BAB III

METODE PENELITIAN

Metode yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan metode pengembangan *system*. Pada pengumpulan data, metode yang digunakan dalam Penelitian yaitu metode yang digunakan untuk mempelajari dan mengumpulkan literatur yang berkaitan dengan penerapan algoritma AES, DES, dan IDEA untuk proses enkripsi dekripsi, serta mempelajari penelitian sebelumnya terkait dengan enkripsi dekripsi. Metode ini bersumber dari jurnal-jurnal ilmiah, makalah, artikel, serta ilmiah lainnya. Tahapan-tahapan penelitian yang peneliti lakukan adalah:

3.1 ALAT YANG DIGUNAKAN

3.1.1 PERANGKAT KERAS (*HARDWARE*)

Perangkat keras yang digunakan pada penelitian ini yaitu 1 perangkat laptop dengan *system* operasi dan spesifikasi hardware sebagai berikut :

Tabel 3. 1 Spesifikasi Perangkat Keras

Sistem Operasi	Windows 10 Home
Processor	AMD Ryzen 5 3500U
RAM	8 GB
SSD	256 GB
HDD	500 GB

3.1.2 PERANGKAT LUNAK (*SOFTWARE*)

Pada penelitian ini menggunakan 2 perangkat virtual yang berjalan pada *virtual environment* VirtualBox, yaitu 1 sebagai *server secure file transfer* dan 1 sebagai *client* dari *secure file transfer*. Spesifikasi perangkat virtual tercantum pada tabel 3.2.

Tabel 3. 2 Spesifikasi Perangkat Virtual

SERVER	Sistem Operasi	Xubuntu-20.04.4
	RAM	1 GB
	Harddisk	10 GB
	Alamat IP	192.168.74.251
CLIENT	Sistem Operasi	Xubuntu-20.04.4
	RAM	1 GB
	Harddisk	10 GB
	Alamat IP	192.168.74.253

3.1.3 *SOFTWARE TOOL* DAN APLIKASI

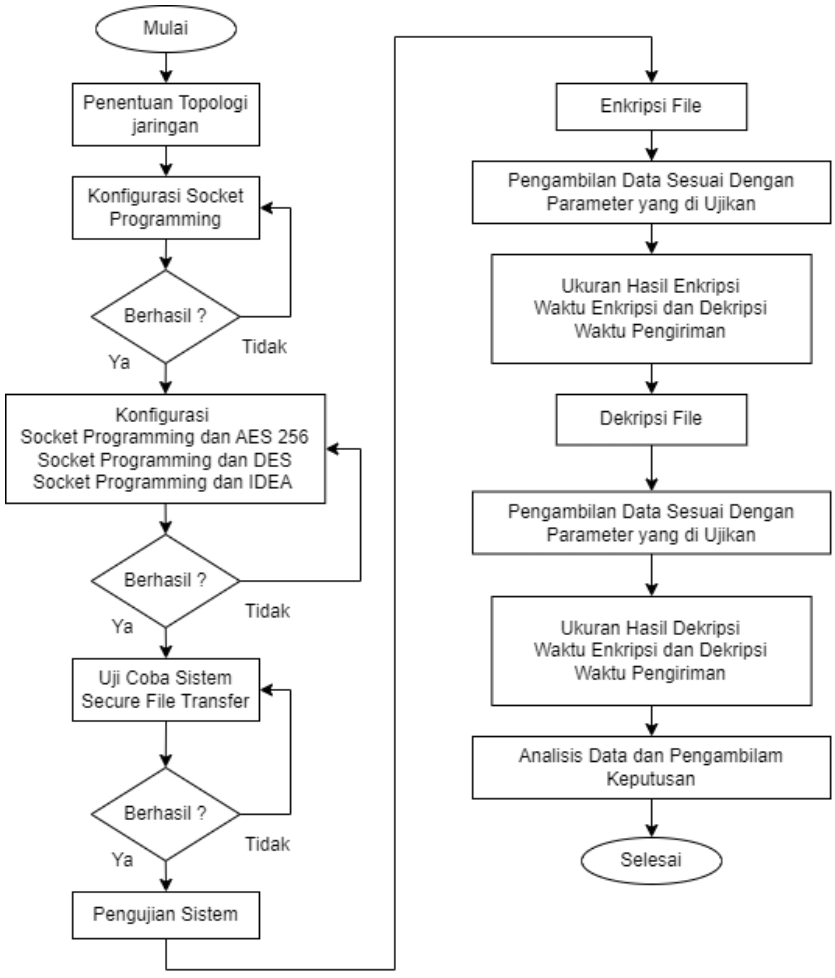
Perangkat lunak sebagai *tool* dan aplikasi yang digunakan pada penelitian ini ditujukan pada Tabel 3.3.

Tabel 3. 3 Software Tool dan Aplikasi

No	Nama Software	Versi	Fungsi
1	VirtualBox	6.1.36	Virtualisasi
2	Mikrotik OS	6.40.1	Router Virtual
3	Wireshark	3.6.7	*Capture* Jaringan

3.2 ALUR PENELITIAN

Penelitian ini dilakukan dalam beberapa tahap seperti diagram alur yang ditujukan pada Gambar 3.1.



Gambar 3. 1 Diagram Alur Penelitian

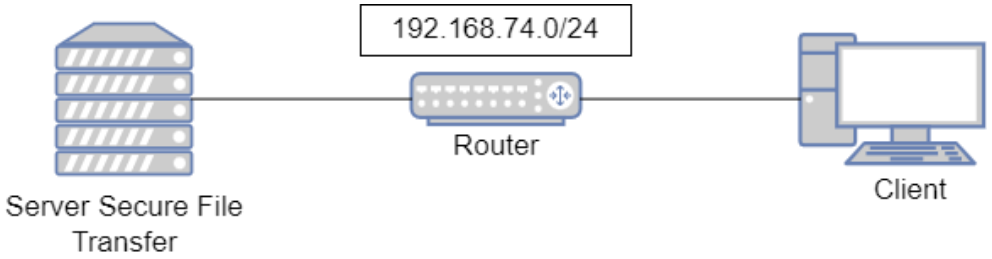
Pada Gambar 3.1 menunjukkan diagram alur perancangan agar penelitian dapat tercapai. Dimulai dari penentuan topologi jaringan sebagai dasar menjalankan

sistem *secure file transfer* menggunakan *socket programming*. Kemudian dilakukan konfigurasi *socket programming* pada mesin vm server dan vm client yang berjalan pada sistem operasi Xubuntu-20.04.4. Apabila konfigurasi berhasil maka akan dilanjutkan dengan konfigurasi *socket programming* dengan algoritma enkripsi yang digunakan (AES-256, DES, dan IDEA). Jika berhasil kemudian dilakukan uji coba terhadap sistem *secure file transfer* yang telah dikonfigurasi. Jika berhasil, kemudian dilakukan pengujian terhadap sistem baik enkripsi maupun dekripsi sesuai dengan parameter yang diujikan. Kemudian dilakukan pengambilan data hasil pengujian berupa parameter ukuran hasil enkripsi maupun dekripsi, waktu enkripsi maupun dekripsi, dan waktu pengiriman.

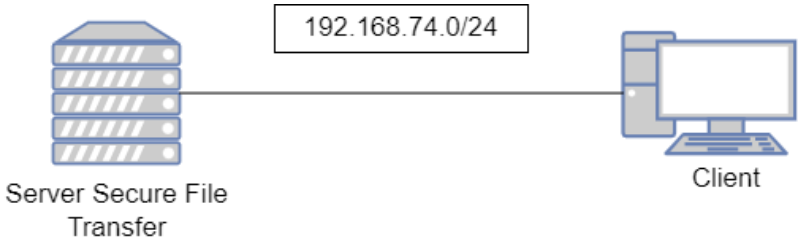
Tahap selanjutnya melakukan analisis terhadap data-data yang telah diperoleh untuk mengetahui performa dari masing-masing algoritma yang diterapkan pada *socket programming*, kemudian dilakukan perbandingan data dari ketiga algortima. Pengambilan kesimpulan dilakukan dengan memperhatikan tujuan dari penelitian agar diperoleh hasil yang sesuai.

3.3 TOPOLOGI JARINGAN

Pada penelitian ini menggunakan 2 topologi jaringan. Topologi pertama ditujukan pada Gambar 3.2 yang terdiri dari 1 *client* yang digunakan untuk sebagai pengirim dan 1 *server* yang digunakan sebagai penerima, kemudian terdapat router yang berfungsi untuk menghubungkan seluruh *virtual machine*. Kemudian Gambar 3.3 menunjukkan topologi kedua, yang terdiri dari 1 *server* dan 1 *client*.



Gambar 3. 2 Topologi Jaringan 1



Gambar 3. 3 Topologi Jaringan 2

3.4 SKENARIO PENGUJIAN

3.4.1 UJI COBA SISTEM *SECURE FILE TRANSFER*

Uji coba dilakukan untuk memastikan sistem dapat berjalan dengan baik dalam melakukan enkripsi, dan dekripsi serta dalam pengiriman *file*. Sebelum membuat sistem *Secure File Transfer*, peneliti terlebih dahulu melakukan beberapa konfigurasi.

3.4.1.1 KONFIGURASI *SOCKET PROGRAMMING*

Konfigurasi *socket programming* dilakukan untuk mempersiapkan Sistem *Secure File Transfer* pada *virtual environment*. Konfigurasi dilakukan pada dua program, yaitu program pengirim (*client*) dan penerima (*server*). Konfigurasi utama adalah menentukan *Host* IP dan *Port* dari dari masing-masing program agar bisa saling terhubung, serta konfigurasi untuk menentukan lokasi *file* yang akan dikirim pada program pengirim (*client*) dan menentukan lokasi *file* yang diterima pada program penerima (*server*).

```
# device's IP address
SERVER_HOST = "0.0.0.0"
SERVER_PORT = 5001
```

(a)

```
# the ip address or hostname of the server, the receiver
host = "192.168.74.251"
# the port, let's use 5001
port = 5001
```

(b)

```
# send data
filename = "file/public.txt"
```

(c)

```
#location receiving file
with open("Receiver/" + filename, "wb") as f:
```

(d)

Gambar 3. 4 Konfigurasi *Socket Programming*
(a) *Host* dan *Port Server*

(b) *Host* dan *Port* pada *Client*

(c) Lokasi *file* pada *Client*

(c) Lokasi *file* diterima pada *Server*

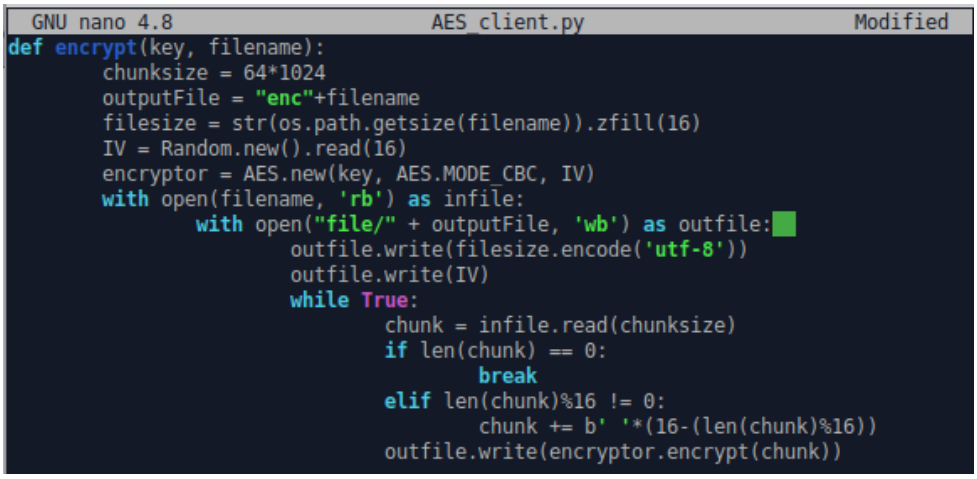
3.4.1.2 KONFIGURASI ALGORITMA AES256

Kofigurasi dilakukan pada dua program, yaitu pengirim (*client*) dan penerima (*server*), dengan menggunakan *library pycryptodome* untuk melakukan pengamanan data. Fungsi-fungsi yang di*import* dari *library* tersebut dapat dilihat pada Gambar 3.4.

```
 GNU nano 4.8                    AES_server.py
from Crypto.Cipher import AES
from Crypto import Random
from Crypto.Hash import SHA256
import socket
import tqdm
import os
from time import process_time
from datetime import datetime
```

Gambar 3. 5 *Import* Fungsi Program Pengirim dan Penerima AES-256

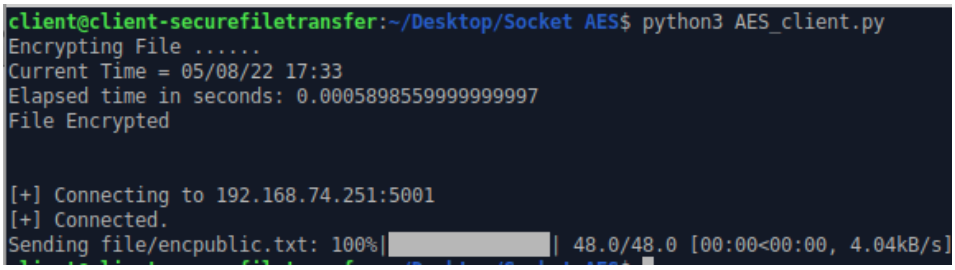
Gambar 3.4 menunjukkan perintah untuk mengimpor *library* pada Program pengirim dan penerima pada algoritma AES-256, untuk mengimpor fungsi dari *library pycryptodome* penggunaan algoritma kriptografi AES, penggunaan *hash256*, serta penggunaan *library* untuk *socket programming* dan *library* pendukung lainnya.



Gambar 3. 6 Fungsi Enkripsi Pada Program Pengirim AES-256

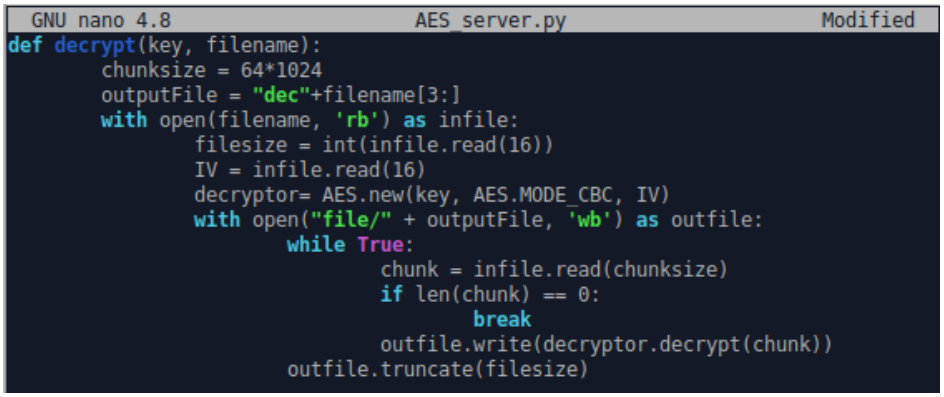
Gambar 3.5 menujukkan fungsi untuk melakukan enkripsi terhadap file yang akan dikirimkan. Pertama program akan memanggil fungsi *encrypt(key, filename)* untuk mengambil *file* yang akan dikirim kemudian akan dilakukan enkripsi terhadap *file* tersebut, setelahkan *file* hasil enkripsi akan mendapat penambahan nama „enc“ pada nama *file*. Kemudian setelah dilakukan enkripsi, *file* dikirimkan ke penerima (*server*).

Program penerima (*client*) pada algoritma AES diberi nama AES_client.py. Untuk menjalankan program tersebut dilakukan dengan menjalankan perintah python3 AES_client.py. Tampilan dari program pengirim (*client*) ketika dijalankan dapat dilihat pada Gambar 3.6.



Gambar 3. 7 Tampilan Pengiriman *File* Pada Program AES client.py

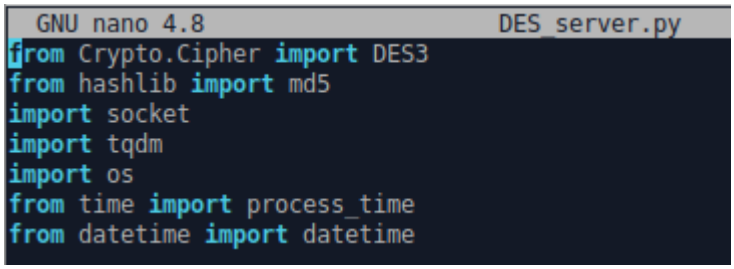
Setelah file diterima oleh *server*, program *server* akan memanggil fungsi *decrypt* (*key*, *filename*) untuk dilakukan dekripsi pada *file* tersebut. Setelah berhasil dilakukan dekripsi, kemudian akan ditambahkan nama „dec“ pada nama *file*. Fungsi program dekripsi ditunjukkan pada Gambar 3.7.



Gambar 3. 8 Fungsi Dekripsi Pada Prorgam Penerima AES

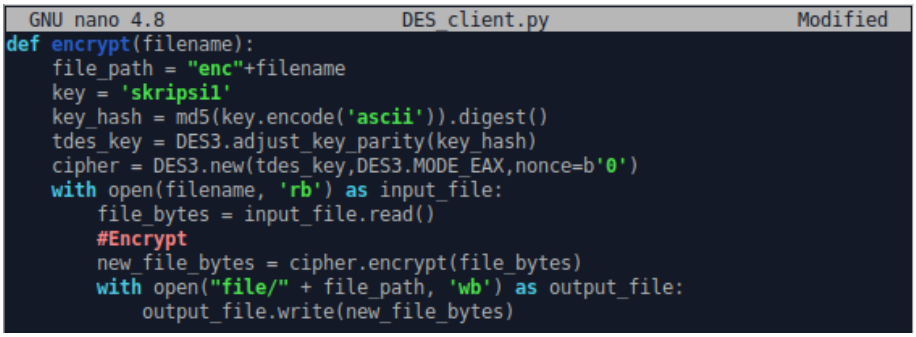
3.4.1.3 KONFIGURASI ALGORITMA DES

Kofigurasi dilakukan pada dua program, yaitu pengirim (*client*) dan penerima (*server*), sama seperti algoritma AES-256, yaitu sama-sama menggunakan *library pycryptodome* untuk melakukan pengamanan data. Fungsi-fungsi yang di*import* dari *library* tersebut dapat dilihat pada Gambar 3.8.



Gambar 3. 9 *Import* Fungsi Pada Program Pengirim dan Penerima DES

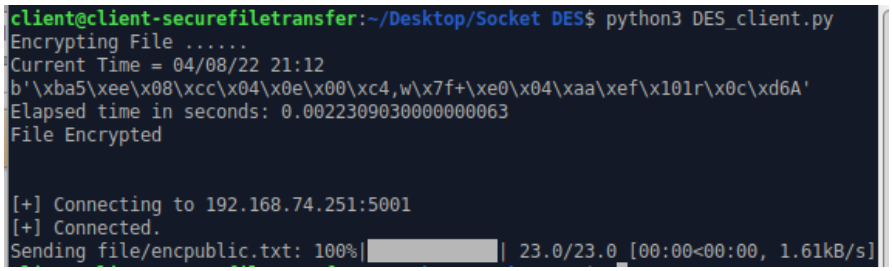
Gambar 3.8 menunjukkan perintah untuk mengimpor *library* pada Program pengirim dan penerima algoritma DES, untuk mengimpor fungsi dari *library pycryptodome* penggunaan algoritma kriptografi DES, penggunaan *Crypto.Chiper*, serta penggunaan *library* untuk *socket programming* dan *library* pendukung lainnya.



Gambar 3. 10 Fungsi Enkripsi pada Program Pengirim DES

Pada Gambar 3.9 menujukkan fungsi untuk melakukan enkripsi terhadap file yang akan dikirimkan. Untuk langkah-langkah yang digunakan hampir sama seperti algoritma AES-256, dimana diawali dengan pemanggilan fungsi *encrypt(filename)* untuk mengambil *file* yang akan dikirim kemudian akan dilakukan enkripsi terhadap *file* tersebut, setelahkan file hasil enkripsi akan mendapat penambahan nama „enc“ pada nama *file*. Kemudian setelah dilakukan enkripsi, file dikirimkan ke penerima (*server*).

Program penerima (*client*) pada algoritma DES diberi nama DES_client.py. Untuk menjalankan program tersebut dilakukan dengan menjalankan perintah python3 DES_client.py. Tampilan dari program pengirim (*client*) ketika dijalankan dapat dilihat pada Gambar 3.10.



Gambar 3. 11 Tampilan Pengiriman *File* pada Program DES_client.py

Setelah file diterima pada sisi *server*, program *server* akan memanggil fungsi *decrypt(filename)* untuk dilakukan proses dekripsi pada file tersebut. Setelah berhasil dilakukan dekripsi, kemudian akan ditambahkan nama „dec“ pada nama file. Fungsi program dekripsi ditunjukkan pada Gambar 3.11.

```
 GNU nano 4.8                    DES_server.py                         Modified
def decrypt(filename):
    file_path = "dec"+filename[3:]
    key = 'skripsi1'
    key_hash = md5(key.encode('ascii')).digest()
    tdes_key = DES3.adjust_key_parity(key_hash)
    cipher = DES3.new(tdes_key,DES3.MODE_EAX,nonce=b'0')
    with open(filename, 'rb') as input_file:
        file_bytes = input_file.read()
        #Encrypt
        new_file_bytes = cipher.decrypt(file_bytes)
        with open("file/" + file_path, 'wb') as output_file:
            output_file.write(new_file_bytes)
```

Gambar 3. 12 Fungsi Dekripsi pada Program Penerima DES

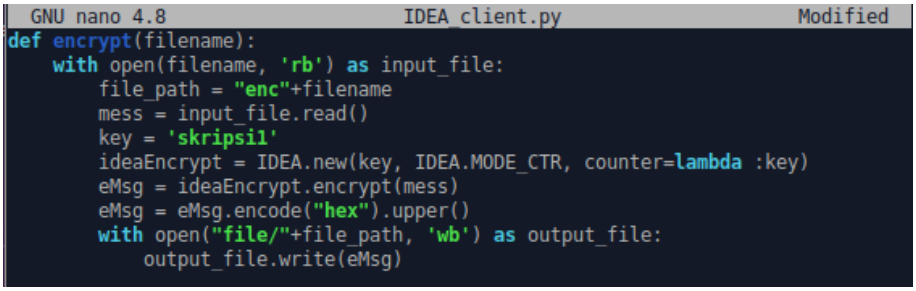
3.4.1.4 KONFIGURASI ALGORITMA IDEA

Kofigurasi dilakukan pada dua program, yaitu pengirim (*client*) dan penerima (*server*), dengan menggunakan *library PyCrypto* dan *PyCryptoplus* untuk melakukan pengamanan data. Fungsi-fungsi yang diimport dari *library* tersebut dapat dilihat pada Gambar 3.12.

```
 GNU nano 4.8                    IDEA_client.py
import socket
import tqdm
import os
import threading
import hashlib
import itertools
import sys
import timeit
from datetime import datetime
from Crypto import Random
from Crypto.PublicKey import RSA
from CryptoPlus.Cipher import IDEA
```

Gambar 3. 13 *Import* Fungsi pada Program Pengirim dan Penerima IDEA

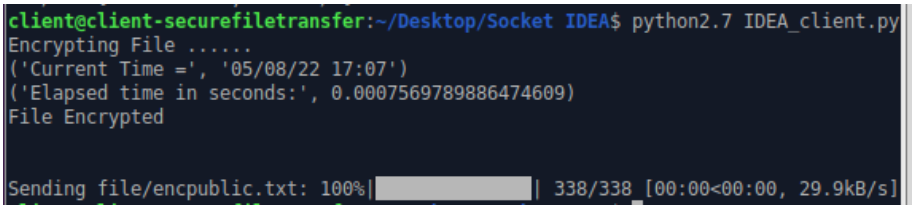
Gambar 3.12 menunjukkan perintah untuk mengimpor *library* pada Program pengirim dan penerima pada algoritma IDEA untuk mengimpor fungsi dari *library PyCrypto* dan *PyCryptoplus*, penggunaan algoritma kriptografi DES, penggunaan *Crypto.Chiper*, serta penggunaan *library* untuk *socket programming* dan *library* pendukung lainnya.



Gambar 3. 14 Fungsi Enkripsi pada Program Pengirim IDEA

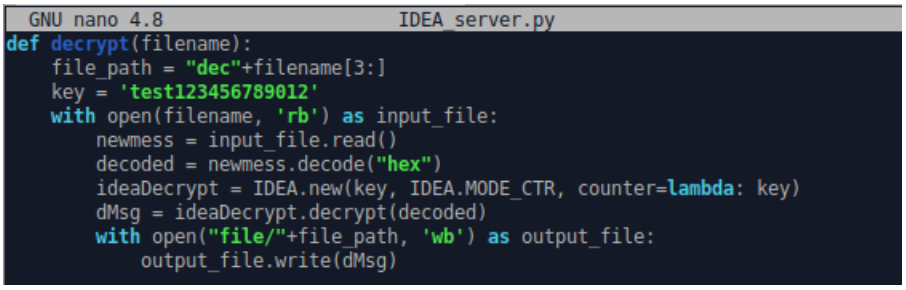
Pada Gambar 3.13 menujukkan fungsi untuk melakukan enkripsi terhadap file yang akan dikirimkan. Untuk langkah-langkah yang digunakan hampir sama seperti algoritma AES-256 dan DES, dimana diawali dengan pemanggilan fungsi *encrypt(filename)* untuk mengambil *file* yang akan dikirim kemudian akan dilakukan enkripsi terhadap *file* tersebut, setelahkan *file* hasil enkripsi akan mendapat penambahan nama „enc“ pada nama *file*. Kemudian setelah dilakukan enkripsi, *file* dikirimkan ke penerima (*server*).

Program penerima (*client*) pada algoritma IDEA diberi nama IDEA_client.py. Untuk menjalankan program tersebut dilakukan dengan menjalankan perintah python2.7 IDEA_client.py. penggunaan python2.7 pada algoritma IDEA disebabkan pada python versi 3 tidak mendukung *library PyCrypto* dan *PyCryptoplus* yang digunakan oleh algoritma IDEA. Tampilan dari program pengirim (*client*) ketika dijalankan dapat dilihat pada Gambar 3.14.



Gambar 3. 15 Tampilan Pengiriman *File* pada Program IDEA_client.py

Setelah *file* diterima pada sisi *server*, program *server* akan memanggil fungsi *decrypt(filename)* untuk dilakukan proses dekripsi pada *file* tersebut. Setelah berhasil dilakukan dekripsi, kemudian akan ditambahkan nama „dec“ pada nama file. Fungsi program dekripsi ditunjukkan pada Gambar 3.14.



Gambar 3. 16 Fungsi Dekripsi pada Program Penerima IDEA

3.4.2 PENGUJIAN *SECURE FILE TRANSFER*

Pengujian dilakukan untuk mengetahui performa dari masing-masing algoritma dalam melakukan enkripsi dan dekripsi pada *secure file transfer* menggunakan *socket programming*. Pengukuran yang diambil didasarkan pada waktu yang diperlukan untuk melakukan dekripsi dan enkripsi *file*, serta pengukuran berdasarkan besar *file* yang dihasilkan.

Dalam pengujian ini penulis menggunakan tiga skenario, pada skenario pertama pengujian menggunakan beberapa tipe *file* dengan berbagai macam ukuran seperti tampak pada Tabel 3.4. kemudian pada skenario kedua penulis menggunakan beberapa tipe *file* dengan ukuran *file* yang hampir sama seperti yang tampak pada Tabel 3.5, dan pada skenario ketiga penulis menggunakan beberapa tipe *file* dengan ukuran yang sama.

Tabel 3. 4 File-file Uji Sistem (Skenario 1)

No	Nama File	Tipe File	Ukuran (*bytes*)
1.	*Public*	*Text Document* (.txt)	234
2.	ArsipFile	WinRAR *archieve* (.rar)	13.388
3.	Laporan	*Microsoft Office Word Document* (.docx)	280.714
4.	PresidenRI	JPG *File* (.jpg)	5.113
5.	VivaLaVida	MP3 *File* (.mp3)	9.685.485
6.	Video1	MP4 *File* (.mp4)	20.373.975
7.	BukuPanduan	PDF *Document* (.pdf)	7.460.661

Tabel 3. 5 File-file Uji Sistem (Skenario 2)

No	Nama File	Tipe File	Ukuran (*bytes*)
1.	*Text*	*Text Document* (.txt)	2.104.850
2.	Arsip	WinRAR *archieve* (.rar)	2.107.610
3.	Doc1	*Microsoft Office Word Document* (.docx)	2.124.908

No	Nama File	Tipe File	Ukuran (*bytes*)
4.	*picture*	JPG *File* (.jpg)	2.107.540
5.	*Music*	MP3 *File* (.mp3)	2.086.171
6.	Video	MP4 *File* (.mp4)	2.085.137
7.	*Document*	PDF *Document* (.pdf)	2.128.513

Tabel 3. 6 File-file Uji Sistem (Skenario 3)

No	Nama File	Tipe File	Ukuran (*bytes*)
1.	Rev	*Text Document* (.txt)	2.104.850
2.	Rev	WinRAR *archieve* (.rar)	2.104.850
3.	Rev	*Microsoft Office Word Document* (.docx)	2.104.850
4.	Rev	JPG *File* (.jpg)	2.104.850
5.	Rev	MP3 *File* (.mp3)	2.104.850
6.	Rev	MP4 *File* (.mp4)	2.104.850
7.	Rev	PDF *Document* (.pdf)	2.104.850