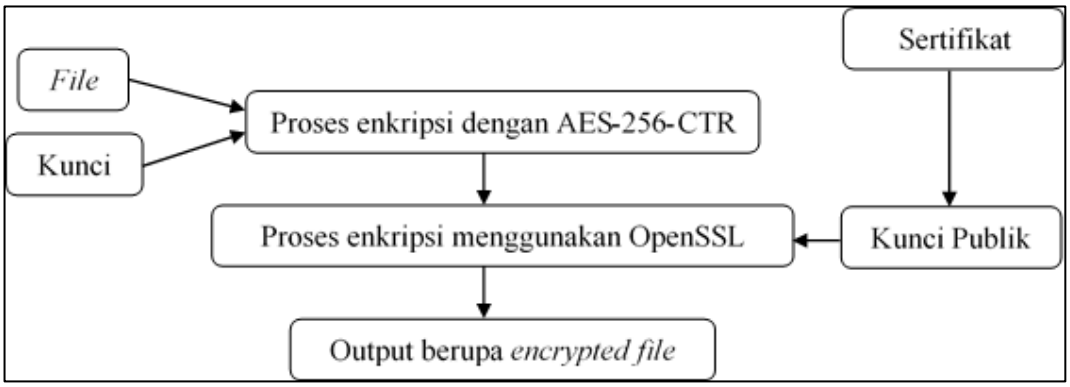
BAB III

METODOLOGI PENELITIAN

3.1. Metode Penelitian

Secara umum metode yang diusulkan pada penelitian ini terdiri dari dua bagian yaitu proses enkripsi dan dekripsi seperti yang diuraikan pada Gambar 3.1 dan Gambar 3.2.



Gambar 3.1 Proses enkripsi

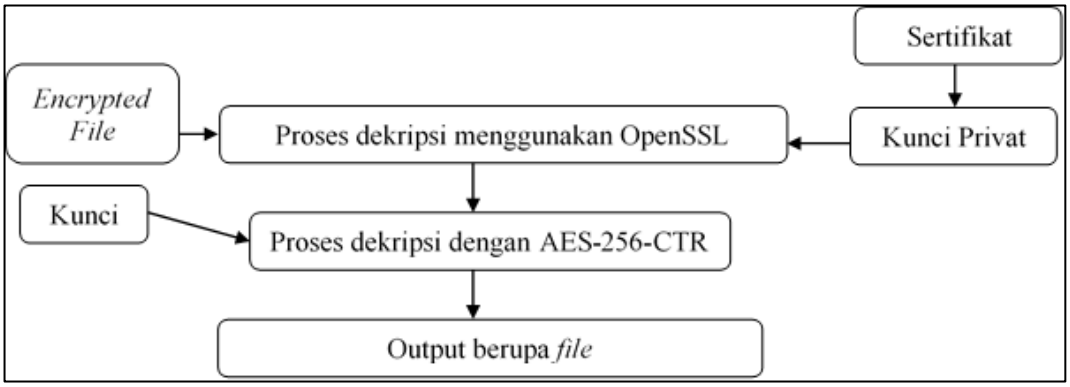
Proses enkripsi terdiri dari 2 bagian dengan tahapan sebagai berikut:

1. Proses Enkripsi dengan algoritma AES-256 mode *CTR*

   File dan kunci rahasia dimasukan ke dalam aplikasi dan dilakukan proses enkripsi dengan algoritma AES-256 mode *CTR* sehingga memberi keluaran berupa *encrypted file tahap 1*.

2. Proses Enkripsi menggunakan OpenSSL

   Proses pertama yaitu memasukan sertifikat dan mengkestraksi kunci publiknya lalu dikombinasikan dengan *encrypted file tahap 1* dan menghasilkan *encrypted file tahap 2*.



Gambar 3.2 Proses dekripsi

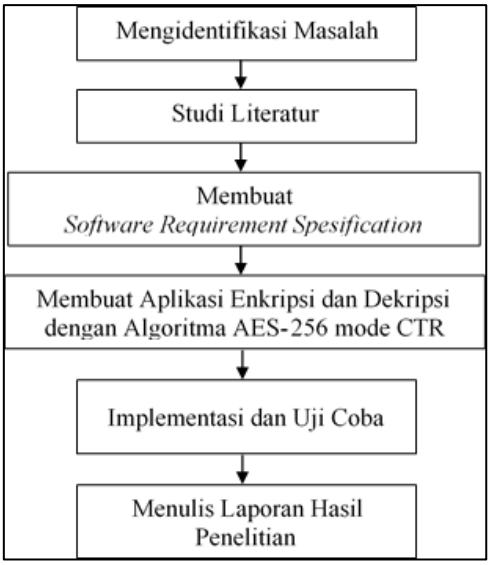
Gambar 3.2 menunjukan bahwa proses dekripsi terdiri dari 2 bagian dengan tahapan sebagai berikut:

1. Proses Dekripsi menggunakan OpenSSL

   Proses awal yaitu memasukan sertifikat dan mengekstraksi kunci privat untuk membuka *encrypted file tahap 2* sehingga data tersebut menjadi *encrypted file tahap 1*.

2. Proses Dekripsi dengan algoritma AES-256 mode *CTR*. Proses ini adalah mengembalikan *encrypted file tahap 1* menjadi *file* semula.

3.2. Tahap Penelitian



Gambar 3.3 Tahap penelitian

Gambar 3.3 menunjukan tahapan penelitian yang akan dilaksanakan, sebagai berikut.

Langkah pertama yaitu tahap mengidentifikasi masalah. Masalah yang akan dicoba dipecahkan pada penelitian ini yaitu bagaimana meningkatkan keamanan data pada proses transmisi dengan mengintegrasikan algoritma AES-256 mode *CTR* dan OpenSSL untuk proses enkripsi dan dekripsi data.

Langkah kedua yaitu tahap studi literatur. Tahap ini dilakukan untuk melihat sejauh mana penelitian sebelumnya dengan mencari jurnal-jurnal ilmiah yang relevan dan terbaru.

Langkah ketiga adalah membuat rancangan *software requirement spesification.* Tahap ini dilakukan pendefinisian kebutuhan-kebutuhan apa saja yang harus ada pada aplikasi enkripsi dan dekripsi serta bagaimana alur kerja dari aplikasi tersebut sehingga bisa menghasilkan keluaran yang sesuai.

Berikut merupakan rancangan *software requirement spesification* yang akan dibuat:

Model *Use Case*

Definisi Aktor

No	Aktor	Deskripsi
1	User	Pengguna adalah orang yang akan melakukan operasi enkripsi maupun dekripsi

Definisi *Use Case*

No	Use case	Keterangan
1	Enkripsi	Merupakan proses enkripsi menggunakan algoritma AES-256 mode CTR dan OpenSSL.
2	Dekripsi	Merupakan proses dekripsi menggunakan algoritma AES-256 mode CTR dan OpenSSL
3	Jenis	Merupakan jenis yang akan diproses, yaitu *hex* atau *binary*

Definisi *Use Case*

Nama *Use Case* : Enkripsi

Skenario :

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan kunci	
2. Memasukan sertifikat	
3. Memasukan *file*	
4. Memilih jenis	
5. Memilih tombol enkripsi	6. Proses enkripsi dilakukan dengan jenis terpilih menggunakan algoritma AES-256 dan OpenSSL
	7. Menampilkan hasil proses enkripsi
Skenario Altrenatif	
Tidak ada	

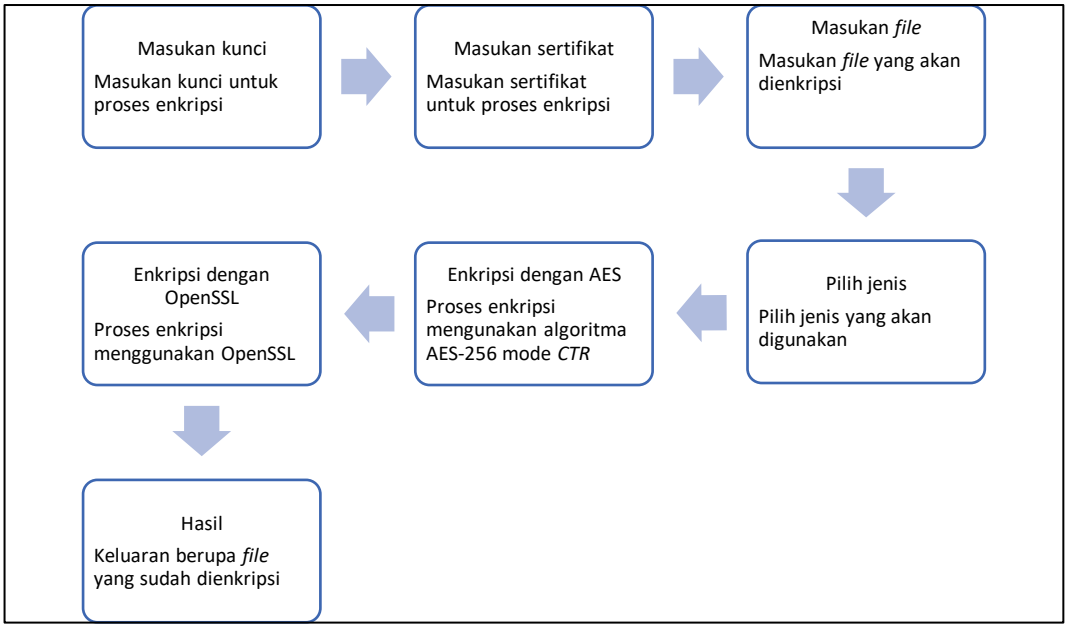
Definisi *Use Case*

Nama *Use Case* : Dekripsi

Skenario :

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan kunci	
2. Memasukan sertifikat	
3. Memasukan *file*	
4. Memilih jenis	
5. Memilih tombol dekripsi	6. Proses dekripsi dilakukan dengan jenis terpilih menggunakan algoritma AES-256 dan OpenSSL
	7. Menampilkan hasil proses dekripsi
Skenario Altrenatif	
1. Tidak ada	

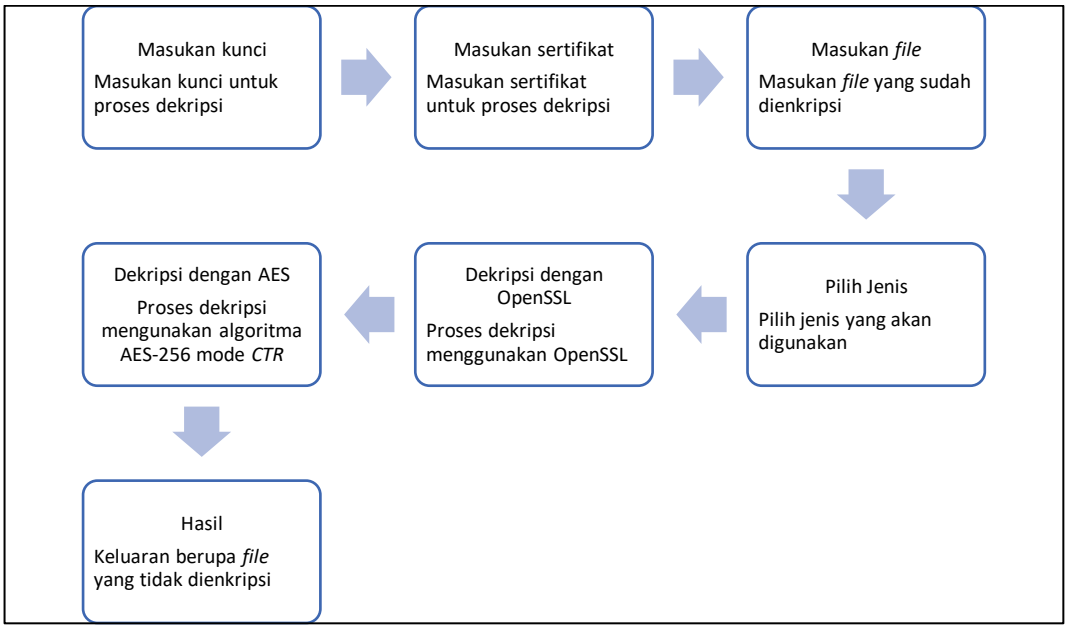
Flowchart proses enkripsi



Proses enkripsi dapat dilakukan dengan menggunakan langkah sebagai berikut:

1. Masukan kunci untuk proses enkripsi. Kunci tersebut berbentuk *string* dengan panjang bebas.
2. Masukan sertifikat. Sertifikat yang dimaksud yaitu berformat pkcs8 yang di dalamnya terdapat kunci privat.
3. Masukan *file* yang akan dilakukan proses enkripsi.
4. Pilih jenis yang akan digunakan, jenis ini ada 2 yaitu *hexa* dan *binary*.
5. Melakukan operasi enkripsi menggunakan algoritma AES-256 mode CTR.
6. Melakukan operasi enkripsi menggunakan OpenSSL.
7. Merupakan hasil *file* yang telah dienkripsi.

Flowchart proses dekripsi

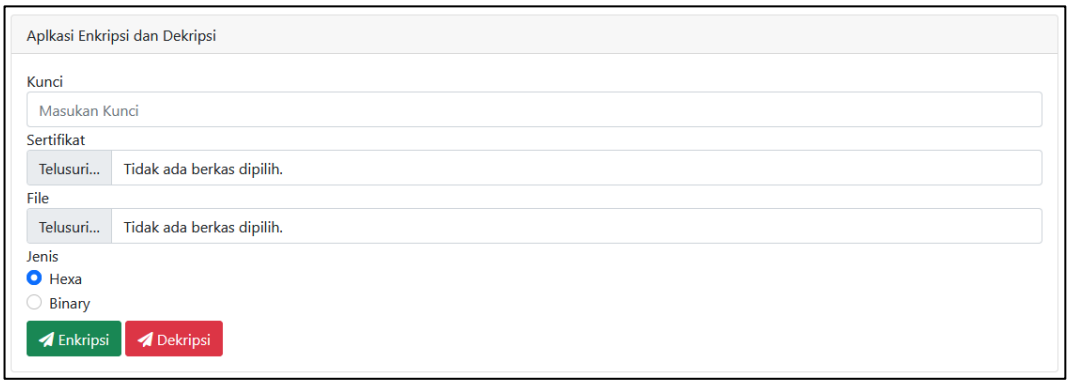


Proses dekripsi dapat dilakukan dengan menggunakan sebagai berikut:

1. Masukan kunci untuk proses enkripsi. Kunci tersebut berbentuk *string* dengan panjang bebas.

2. Masukan sertifikat. Sertifikat yang dimaksud yaitu berformat *pkcs8* yang di dalamnya terdapat kunci privat.
3. Masukan *file* yang akan dilakukan proses dekripsi.
4. Pilih jenis yang akan digunakan, jenis ini ada 2 yaitu *hexa* dan *binary.*
5. Melakukan operasi dekripsi menggunakan OpenSSL.
6. Melakukan operasi dekripsi menggunakan algoritma AES-256 mode CTR.
7. Merupakan hasil *file* yang telah didekripsi.

Langkah keempat yaitu membuat aplikasi enkripsi dan dekripsi sesuai dengan rancangan *software requirement spesification.* Tahap ini adalah realisasi dari langkah ketiga, yaitu mulai pembuatan aplikasi enkripsi dan dekripsi data sesuai dengan *software requirement spesification* yang sudah dirancang.



Gambar 3.4 Tampilan aplikasi

Gambar 3.4 menunjukan tampilan dari aplikasi yang telah dirancang sesuai dengan *software requirement spesification.*

Source Code 3.1 Tampilan aplikasi

```
       <!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta    name="viewport"    content="width=device-width,    initial-
scale=1.0">
    <title>Aplikasi Enkripsi dan Dekripsi Data dengan OpenSSL Algoritma
AES-256</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css"              rel="stylesheet"              integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <link                                              rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css"                       integrity="sha512-
iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8dLJgyAH
IhR35VZc2oM/gI1w=="     crossorigin="anonymous"     referrerpolicy="no-
referrer" />
</head>

<body>
    <div class="container pt-5">
        <div class="card">
            <div     class="card-header">Aplkasi     Enkripsi     dan
Dekripsi</div>
            <div class="card-body">
                <form         action="proses.php"         method="POST"
enctype="multipart/form-data">
                    <div class="form-group">
                        <label>Kunci</label>
                        <input  type="text"  name="kunci"  class="form-
control" placeholder="Masukan Kunci" required>
                    </div>
                    <div class="form-group">
                        <label>Sertifikat</label>
                        <input      type="file"      name="sertifikat"
class="form-control" required>
                    </div>
                    <div class="form-group">
                        <label>File</label>
                        <input type="file" name="berkas" class="form-
control" required>
                    </div>
                    <div class="form-group">
                        <label>Jenis</label>
                        <div class="form-check">
                            <input            class="form-check-input"
type="radio" name="jenis" id="exampleRadios1" value="hex" checked>
```

Source Code 3.2 Lanjutan Source Code 3.1

```
<label class="form-check-label" for="exampleRadios1">
                                Hexa
                            </label>
                        </div>
                        <div class="form-check">
```

```
                            <input            class="form-check-input"
type="radio" name="jenis" id="exampleRadios2" value="bin">
                            <label            class="form-check-label"
for="exampleRadios2">
                                Binary
                            </label>
                        </div>
                    </div>
                    <div class="form-group mt-2">
                                                  <button  name="proses"
value="e" type="submit" class="btn btn-success btn-icon"><i class="fa
fa-paper-plane"></i> Enkripsi</button>
                        <button name="proses" value="d" type="submit"
class="btn  btn-danger  btn-icon"><i  class="fa  fa-paper-plane"></i>
Dekripsi</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
    </div>
</body>
</html>
```

Source code 3.1 merupakan kode untuk tampilan aplikasi yang digunakan.

Source Code 3.3 File proses.php

```
<?php

$kunci = $_POST['kunci'];
$berkas = $_FILES['berkas'];
$sertifikat = $_FILES['sertifikat'];
$jenis = $_POST['jenis'];
$proses = $_POST[‘proses’];
require_once('../enkripsi.php');

$path = './crt/' . uniqid();
move_uploaded_file($sertifikat['tmp_name'], $path);

$encryption = new Encryption($kunci);
$encryption->setCertPath($path);

if ($jenis == 'bin') {
    $encryption->setRaw();
}
```

Source Code 3.4 Lanjutan Source Code 3.3

```
if ($proses == ‘e’) {
 echo $encryption->encrypt(file_get_contents($berkas['tmp_name']));
} else {
 echo $encryption->decrypt(file_get_contents($berkas['tmp_name']));
}
```

Source code 3.3 merupakan kode untuk proses dari halaman awal aplikasi.

Source Code 3.5 File enkripsi.php

```
<?php
/**
 * Created by Ihsan Fawzan
 * 2021/02/02
 */
class Encryption
{
    protected $method = 'aes-256-ctr'; // method yang digunakan
    protected $certPath = null; // lokasi sertifikat
    protected $cert = null; // sertifikat
    protected $raw = false;
    private $password;
    private $iv;

    public function __construct($password)
    {
        if (!$password) {
            throw new Exception("API Key nya bos", 1);
        }

        $this->password   =   hex2bin(hash('sha512',   $password)   .
hash('sha512', $password) . hash('sha512', $password) . hash('sha512',
$password));
        $this->iv                                                    =
openssl_random_pseudo_bytes(openssl_cipher_iv_length($this->method));
    }

    public function setCertPath($path)
    {
        $this->certPath = $path;
        $this->cert = file_get_contents($this->certPath);
    }

    /**
     * Mengubah mode output menjadi binary / raw
     *
     * @return void
     */
    public function setRaw()
    {
        $this->raw = true;
```

Source Code 3.6 Lanjutan Source Code 3.5

```
}

    public function encrypt($plaintext)
    {
        if ($this->certPath == null) {

            if ($this->raw) {
```

```
                return    openssl_encrypt($plaintext,    $this->method,
$this->password, OPENSSL_RAW_DATA, $this->iv) . $this->iv;
            } else {
                return    bin2hex(openssl_encrypt($plaintext,    $this-
>method, $this->password, OPENSSL_RAW_DATA, $this->iv) . $this->iv);
            }
        } else {
            $out  =  openssl_encrypt($plaintext,  $this->method,  $this-
>password, OPENSSL_RAW_DATA, $this->iv);
            return $this->seal($out, $this->iv);
        }
    }

    public function decrypt($encryptedtext)
    {
        if ($this->certPath == null) {

            if ($this->raw) {
                $encryptedtext = $encryptedtext;
            } else {
                if           (ctype_xdigit($encryptedtext)           &&
strlen($encryptedtext) % 2 == 0) {
                    $encryptedtext = hex2bin($encryptedtext);
                } else {
                    return false;
                }
            }

            $iv           =           substr($encryptedtext,           -
openssl_cipher_iv_length($this->method));
            $data = substr($encryptedtext, 0, strlen($encryptedtext) -
openssl_cipher_iv_length($this->method));


            return    openssl_decrypt($data,    $this->method,    $this-
>password, OPENSSL_RAW_DATA, $iv);
        } else {

            $enc = $this->open($encryptedtext);

            if ($enc == false) {
                return false;
            }
```

Source Code 3.7 Lanjutan Source Code 3.5

```
            return   openssl_decrypt($enc[0],   $this->method,   $this-
>password, OPENSSL_RAW_DATA, $enc[1]);
        }

    }

    protected function seal($text, $iv)
```

```
    {
        $pub = openssl_pkey_get_public($this->cert);

        openssl_seal($text, $sealed, $pass, [$pub]);

        if ($this->raw) {
            return $pass[0] . $sealed . $iv;
        } else {
            return bin2hex($pass[0] . $sealed . $iv);
        }
    }

    protected function open($text)
    {
        $private_key = openssl_pkey_get_private($this->cert);

        if ($this->raw) {
            $raw = $text;
        } else {
            $raw = hex2bin($text);
        }

        $pass = substr($raw, 0, 128);
        $iv = substr($raw, -openssl_cipher_iv_length($this->method));
        $data  =  substr($raw,  128,  -openssl_cipher_iv_length($this-
>method));

        openssl_open($data, $out, $pass, $private_key);

        $output = [
            $out,
            $iv
        ];

        return $output;
    }
}
```

Source code 3.5 merupakan kode untuk proses enkripsi maupun dekripsi dengan AES-256 mode CTR dan OpenSSL.

Langkah kelima adalah implementasi dan uji coba. Aplikasi tersebut diuji dengan *dataset*, apakah menghasilkan keluaran yang sesuai atau tidak dan melakukan proses uji coba transmisi data pada protokol *HTTP*. Proses pengujian yang akan dilakukan di antaranya : pengujian waktu enkripsi pada beberapa jenis *file*, pengujian waktu dekripsi pada beberapa jenis *file*, pengujian hasil enkripsi ditransmisikan pada jaringan, pengujian waktu transmisi *file* yang sudah dienkripsi,

pengujian waktu transmisi *file* tanpa enkripsi, pengujian serangan dengan metode *Man-in-the-Middle*, dan mengukur jumlah ketidakteraturan dengan entropi *Shannon*.

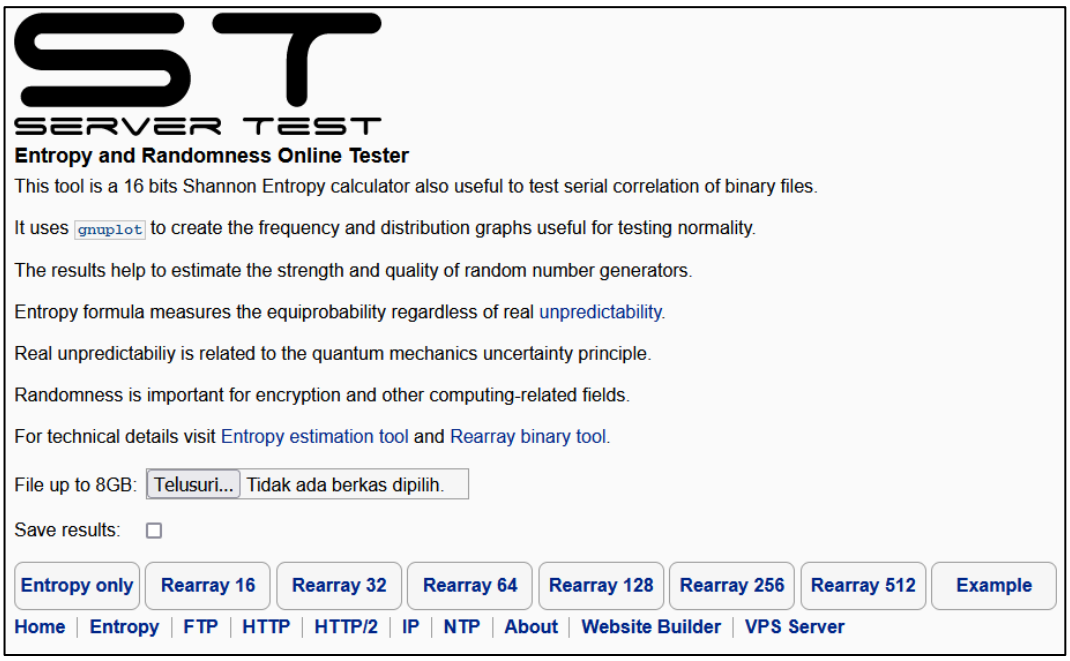
Source Code 3.8 Proses pengujian waktu

```
<?php
$awal = microtime(true);
 // proses enkripsi dan dekripsi
$akhir = microtime(true);
$hasil = $akhir - $awal;
?>
```

Source Code 3.8 merupakan kode yang digunakan untuk melakukan proses pengujian waktu.

Proses pengujian serangan dengan metode *Man-in-the-Middle* menggunakan tools Burp Suite dan dimonitor dengan Wireshark.

Proses pengujian menggunakan entropy *shannon* dilakukan secara *online* menggunakan tools yang dapat diakses dari https://servertest.online/entropy.



Gambar 3.5 Tampilan website tools entropy

Gambar 3.5 merupakan tampilan website dari tools yang digunakan.

Langkah terakhir yaitu membuat laporan hasil penelitian sebagai bahan pertanggung jawaban atas penelitian ini.