Jurnal Tekno Insentif | Vol. 15 | No. 2 | Oktober 2021
DOI : https://doi.org/10.36787/jti.v15i1.497 Halaman 65 - 75

Cryptography: Perancangan *Middleware Web Service Encryptor* menggunakan *Triple Key MD5. Base64,* dan *AES*

MAULYANDA[1], SYAFRIAL FACHRI PANE[2], ROLLY MAULANA AWANGGA[3]

[1,2,3]Politeknik Pos Indonesia, Indonesia
Email : maulyanda@poltekpos.ac.id

Received 27 September 2021 **|** *Revised* 15 Oktober 2021 **|** *Accepted* 25 Oktober 2021

ABSTRAK

Penelitian ini membantu dalam melalukan proses Keamanan data atau informasi untuk menjamin kerahasiaan dan keaslian data atau informasi. Dalam perancangan ini menggunakan Kriptografi sebagai salah satu solusi dalam mengamankan data atau informasi. Metode kriptografi digunakan untuk mempersatukan algoritma Md5, Base64 serta AES (*Advanced Encryption Standard*). Kombinasi dari tiga algoritma menghasilkan *ciphertext*, yang dapat mengamankan data dari proses *tag NFC*. Penelitian ini menggunakan metodologi penelitian yang dapat menyatakan bahwa sistem yang dibangun dapat berfungsi dengan baik dan untuk keamanan nya aman digunakan, dari hasil penerapannya didapatkan hasil persentase keberhasilan 100%. Jadi, penelitian ini mampu menjawab permasalahan yang terjadi pada sistem keamanan data.

Kata kunci: Kriptografi, AES, *Base64*, Md5, *RESTFul*

ABSTRACT

This research helps in carrying out data or information security to ensure the confidentiality and authenticity of data or information. This design uses Cryptography as a solution in securing data or information. Cryptographic methods to unify the Md5, Base64, and AES (Advanced Encryption Standard) algorithms. The combination of the three algorithms produces ciphertext, which can secure data from the NFC tag process. This study uses a research methodology that can state that the system built can function correctly, and for security, it is safe to use because it has a 100% success percentage. So, this researchable to answer the problems that occur in the data security system.

Keywords*: Cryptography, AES, Base64, Md5, RESTFul*

1. PENDAHULUAN

Kriptografi yaitu sebuah metode untuk menjaga keamanan data agar terjamin kerahasiaan data dan meningkatkan aspek keamanan saat penyampaian suatu data atau informasi **(Panjaitan, Susanto, & WM, 2017)**. Kriptografi mempunyai aspek keamanan yaitu melindungi, merahasiakan data atau informasi dari pemalsuan dan pengubahan informasi tersebut **(Hidayat & Afrianto, 2017)**. Kegunaan kriptografi kunci-publik yang penting adalah kunci, oleh karena itu pesan enkripsi menggunakan kunci privat Ketika proses pengiriman data atau informasi dan kunci tersebut ditempatkan pada data yang dikirimkan **(Rohman & Mufti, 2018)**. Para kriptografi, data atau informasi yang dapat dibaca disebut dengan *plaintext*, sedangkan data yang tidak terbaca atau tidak jelas disebut dengan *cyphertext* **(Siswanto, Anif, & Gata, 2018)**. Salah satu algoritma yang dipakai dalam kriptografi adalah AES *(Advanced Encryption Standard)* algoritma enkripsi kunci simetris pada saat ini **(Masoumi & Rezayati, 2014) (Sibarani, Zarlis, & Sembiring, 2017)**. Sedangkan metode yang diterapkan pada algoritma kriptografi penyandi blok AES antara lain ECB, CBC, CFB serta OFB **(Budianto, Amini, & Ariyani, 2017)**. Teknologi untuk keamanan data memiliki tujuan untuk memenuhi kebutuhan proses bisnis dari pengguna atau organisasi suatu perusahaan. **(Pane, Awangga, & Maulyanda, 2018) (Pane, Awangga, & Azhari, 2018)**. Web *service* digunakan untuk menghubungkan *client* dan server untuk dapat melakukan proses pertukaran data atau pesan **(Christanto & Kurniawati, 2016)**. *RESTFul web service* sangat baik dalam proses kinerja server yang stabil ketika digunakan secara bersamaan oleh pengguna **(Carter, Khaire, Novosad, & Chang, 2016)**. Namun keamanan *RESTFul Web service* merupakan poin penting yaitu mencangkup keamanan informasi, serta melindungi kerahasiaan informasi atau data tersebut **(Awangga & Andarsyah, 2016)**. Bahasa pemrograman *python* digunakan dalam penelitian ini karena beberapa alasan yaitu *python* bersifat *open source* **(Permana, Praja, Fatkhuroyan, & Muzayanah, 2018)**, dapat di optimalisasi **(Tibau, Nunes, Bortoluzzi, & Marenzi, 2018) (Moudgalya, 2018)**, dan *Python* adalah bahasa pemrogram multiguna dan sebuah program level tinggi yang dapat ditafsirkan **(López, Pelayo, & Forero, 2016)**. Jenis *microframework* yang digunakan untuk menjalankan kode *python* yaitu *flask* yang merupakan sebuah *framework* dari Bahasa pemrograman *python* dan memanfaatkan *gunicorn* sebagai server untuk menjalankan kode *python* untuk melayani HTTP.

Middleware Web Service merupakan produk yang bekerja untuk memfasilitasi fungsionalitas antar aplikasi. *Middleware Web Service* juga dikenal sebagai manajemen dari layanan web yang dapat menangani hal-hal seperti keamanan atau komunikasi lintas *platform* seperti mengirimkan pesan data atau informasi antar aplikasi yang berbeda. *Middleware Web Service* ini bekerja sebagai arsitektur *client-server* di mana aplikasi layanan *web* adalah *client* dan *middleware* adalah *server,* yaitu menyediakan layanan kepada *client.* Proses keamanan data ini disebut proses Enkripsi yang merupakan bagian penting dalam proses pengamanan data. Aspek penting dari sebuah sistem yaitu *security* (keamanan), namun pemilik sistem masih kurang memperhatikan dari keamanan data dari sistem yang telah dibangun. Karena pentingnya pengamanan data.

Penelitian ini bertujuan menjaga kerahasiaan data dengan mengekripsikan data menggunakan *triple key* Md5, Base64 dan AES pada pemrograman *python* dan menjaga kerahasiaan data E-KTP dari proses *Tag NFC*. Dimana pada penelitian ini mengangkat teknik pengamanan data menggunakan kriptografi yang membuat pengamanan atau kerahasiaan data E-KTP saat proses *tag NFC* di ruangan *informatics research center* (IRC) yang memungkinkan data E-KTP tidak terbaca oleh pengguna yang tidak memiliki akses, dengan *encryption* data atau informasi menjadi lebih sulit untuk diketahui oleh pengguna yang tidak memiliki akses. Dengan

pengamanan maka pemilik E-KTP terhindar dari penyadapan atau pembajakan data penting. Proses enkripsi pada penelitian ini menggunakan kombinasi algoritma *Md5*, *Base64* dan *AES*, untuk memberikan hasil berbentuk *(chipertext).*

Dengan demikian penelitian ini menggunakan *triple key Md5*, *Base64* dan *AES* yang merupakan algoritma untuk *Encoding* dan *Decoding* suatu data. Kombinasi ini diharapkan dapat memperkuat keamanan data atau pesan yang bersifat rahasia, sehingga akan lebih sulit untuk di pecahkan oleh orang-orang yang tidak berhak terhadap data tersebut.

2. METODE

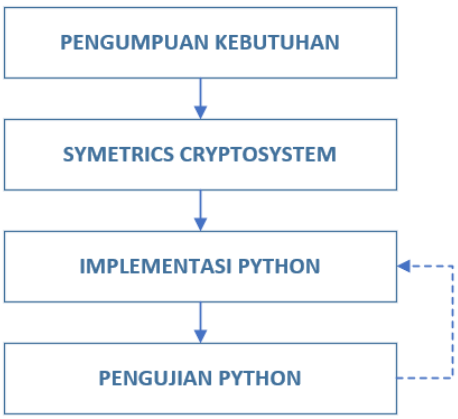
Penelitian ini membahas tentang mengamankan data dengan cara mengenkripsi menggunakan *triple key Md5*, *Base64* dan AES, Adapun penjelasan beserta contohnya dari ketiga algoritma tersebut sebagai berikut:

a. AES: AES *Cipher Blocker Chaining* (CBC) merupakan bentuk lanjutan dari generasi pertama *Enkripsi Cipher Blok* (ECB). Dengan enkripsi CBC, setiap blok *ciphertext* tergantung pada semua blok *plaintext* yang diproses untuk menambah tingkat kerumitan data yang telah dienkripsi. Contoh enkripsi AES CBC dari kata ”Anak Teknik” menjadi ”wkL5tGCFUkVptwZRi/UG8g==”.

b. *Base64*: Algoritma *Base64* digunakan dalam penelitian ini karena merupakan algoritma untuk *encryption* dan *decryption* ke dalam format *ASCII*. Contoh penggunaan dari Algoritma Base64 dalam melakukan enkripsi karakter. Kutipan dari Albert Einstein: ”Imajinasi lebih penting dari pengetahuan” adapun hasil enkripsi ”SW1hamluYXNpIGxlYmloIHBlbnRpbmcgZGFyaSBwZW5nZXRhaHVhbg==”.

c. MD5: *Algoritma* MD5 merupakan sebuah pesan dengan ukuran panjang *message digest* 128 bit, secara *brute force* dibutuhkan percobaan sebanyak 2128 kali untuk menemukan dua buah pesan atau lebih yang sama **(Sumarno & others, 2018)**. Contoh penggunaan Algoritma MD5 dalam melakukan enkripsi karakter nya ”Sayang Kamu” dan hasil enkripsi ”7DBC90A494B88410686214D2A0C877B9”.

Metodologi penelitian merupakan aturan kegiatan, dan prosedur yang digunakan oleh seorang peneliti. Metodologi juga merupakan sebuah analisa teoritis tentang suatu cara. Berikut ini adalah alur dari metodologi penelitian yang dilakukan di dalam penelitian ini yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metodologi Penelitian

2.1 Pengumpulan Kebutuhan

Mengumpulkan kebutuhan sistem *enkriptor* seperti data. Data penelitian ini diperoleh dari pengguna atau pengurus *Informatics Research Center* (IRC), data yang digunakan merupakan data E-KTP. Data tersebut akan di enkripsi untuk menjaga kerahasiaan menggunakan kriptografi.

2.2 *Symetric Cryptosystem*

proses *Symetric Cryptosystem* pada enkripsi yaitu mengganti huruf dari data atau informasi *plaintext* menjadi *ciphertext*. Proses pergantian dilakukan dengan melangkahi huruf setelahnya seperti huruf a digantikan dengan huruf f dan seterusnya, dan proses pengembalian huruf data atau informasi disebut dengan deskripsi.

2.3 Implememtasi *Python*

Proses pada implementasi *python* ini menerjemahkan hasil dari proses enkripsi dan deskripsi ke dalam bahasa pemrograman *python* untuk menjaga kerahasiaan data E-KTP dari proses *tag NFC*.

2.4 Pengujian *Python*

Tahapan ini merupakan tahapan akhir, pada tahap ini akan didapatkan hasil dari identifikasi masalah pada penelitian ini. Pengujian ini menggunakan *BOOM* untuk melihat hasil implementasi *python* apakah ada terdapat masalah atau *error*, jika tidak maka aman untuk digunakan.

3. HASIL DAN PEMBAHASAN

Penelitian ini membahas tentang bagaimana menjaga kerahasiaan data E-KTP dari proses *tag NFC*. dengan cara mengenkripsi menggunakan kombinasi algoritma kriptografi *Triple Key Md5*, *Base64* dan AES, dan diimplantasikan pada bahasa pemrograman *python*.

3.1 Pengumpulan Kebutuhan

Data kebutuhan yang diperoleh dan dikumpulkan langsung dari pengguna laboratorium IRC, seperti Dosen pengurus IRC dan Mahasiswa DIV Teknik Informatika Politeknik Pos Indonesia yang menggunakan laboratorium IRC sebagai tempat sarana penelitiannya. Berikut data yang akan dilakukan nya proses enkripsi dan enkripsi pada penelitian ini yang dapat dilihat pada Tabel 1.

Tabel 1. Data dari Proses *Tag NFC*

Id_tap	Id_pengguna	Tanggal
1	0x40x360x380xB20x210x250x80	19/07/2019 10:02:12
2	0x50x8A0x990x1E0x590x310x0	19/07/2019 10:02:12
3	0x40x8A0x610xBA0x410x2A0x80	21/08/2019 14:15:57
4	0x40x840x7A0x5A0x810x2A0x80	21/08/2019 14:15:57
5	0x40x310x570x5A0x7D0x5B0x80	21/08/2019 14:15:57
6	0x40x600x590xCA0x5B0x2A0x80	21/08/2019 14:20:41
7	0x40x5E0x910xE20x9E0x4F0x80	21/08/2019 14:20:41
8	0x40x840x7A0x5A0x810x2A0x80	21/08/2019 14:24:00
9	0x40x230x240xFA0x800x5B0x80	21/08/2019 16:40:27
10	0x40x360x380xB20x210x250x80	27/08/2019 07:53:28
11	0x40x360x380xB20x210x250x80	27/08/2019 07:53:28

Adapun Dosen pengurus IRC dan Mahasiswa DIV Teknik Informatika Politeknik Pos Indonesia yang menggunakan laboratorium IRC dapat dilihat pada Tabel 2.

Tabel 2. Data Pengurus IRC

id_p	id_pengguna	nama_p
1	0x40x360x380xB20x210x250x80	Muhammad Nur Ikhsan
2	0x40x5E0x910xE20x9E0x4F0x80	Alit Fajar Kurniawan
3	0x50x8A0x990x1E0x590x310x0	Maulyanda
4	0x40x1E0x600x6A0xC60x610x80	Syafrial Fachri Pane
5	0x40x1C0x2E0xA0x420x2A0x80	Rolly Maulana Awangga
6	0x40x310x570x5A0x7D0x5B0x80	Cahya Kurniawan
7	0x40x600x590xCA0x5B0x2A0x80	Aditya Pratama Dharma
8	0x40x230x240xFA0x800x5B0x80	M Raziq Hakim Siregar
9	0x40x840x7A0x5A0x810x2A0x80	R Rifa Fauzi Komara
10	0x40x8A0x610xBA0x410x2A0x80	Faisal Syariffudin

3.2 *Symetric Cryptosystem*

Untuk menjaga kerahasiaan data pada penelitian ini menggunakan *Symetric Cryptosystem* sebagai kunci untuk melakukan proses *encryption* dan *decryption*. Kunci dari proses tersebut harus dirahasiakan untuk menjaga kerahasiaan data atau informasi proses *tag NFC*. Proses Alur Diagram dari *Symetric Cryptosystem* dapat dilihat pada Gambar 2.

Gambar 2. Diagram *Symetric Cryptosystem*

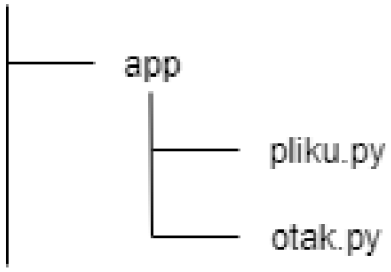
Kunci untuk melakukan proses *encryption* dan *decryption* adalah kunci yang sama (K). Langkah–Langkah detail dari (K) dalam proses enkripsi dan dekripsi dengan penerapan algoritma kriptografi yang dirancang dapat dilihat pada Gambar 3 dan Gambar 4.

Gambar 3. Proses *Encryption*

Gambar 4. Proses *Decryption*

3.3 Implementasi *Python*

Tahap ini merupakan proses dari enkripsi dan deskripsi ke dalam bahasa pemrograman *python* sebagai kerangka kerja dalam pembuatan sistem. Struktur *project* yang dibuat hanya menggunakan 2 *file*, yaitu pliku.py dan otak.py yang dapat dilihat pada Gambar 5.



Gambar 5. Struktur Project

Pada *file* pliku.py terdapat proses dari logika kriptografi, seperti enkripsi dan dekripsi yang mengombinasikan AES CBC, *Base*64 dan *MD5*. Hasil akhir dari proses enkripsi ini berupa *chipertext*, sedangkan proses akhir dari dekripsi ini berupa *plaintext* dari hasil proses data yang berbentuk *chipertext*. Adapun *code* yang terdapat pada *file* pliku.py sebagai berikut:

```
class AESCipher:
    def __init__(self, key):
        self.key = md5(key.encode('utf8')).digest()

    def encrypt(self, data):
        iv = get_random_bytes(AES.block_size)
        self.cipher = AES.new(self.key, AES.MODE_CBC, iv)
        return b64encode(iv + self.cipher.encrypt(pad(data.encode('utf-8'),
            AES.block_size)))

    def decrypt(self, data):
        raw = b64decode(data)
        self.cipher = AES.new(self.key, AES.MODE_CBC, raw[:AES.block_size])
        return unpad(self.cipher.decrypt(raw[AES.block_size:]), AES.block_size)
```

File otak.py adalah sebagai tempat *route* aplikasi dari proses enkripsi dan deskripsi. Pada proses enkripsi menerapkan metode *GET* untuk mendapatkan data dari proses enkripsi dari *file* pliku.py, sebelum mendapatkan data pada proses enkripsi ini akan melakukan proses cek token terlebih dahulu sebagai keamanan lebih dari proses enkripsi ini. Adapun *code* pada proses enkripsi dapat dilihat sebagai berikut:

```
def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.args.get('token')
```

```
        if not token:
            return jsonify({'message' :'Token is missing!'}), 403

        try:
            data = jwt.decode(token, app.config['SECRET_KEY'])
        except:
            return jsonify({'message ' : 'Token is invalid'}), 403

        return f(*args, **kwargs)

    return decorated

@app.route('/enkripsi',methods=['POST', 'GET'])
@token_required
def tes():
    cur = mysql.connection.cursor()

    if request.method=='GET':

     # Read a single record
     result = cur.execute("SELECT * FROM kunciku")
     coba = cur.fetchall()

     mysql.connection.commit()

     return pliku.AESCipher('secret').encrypt(str(coba)).decode('utf-8')

     cur.close()

    else:
      return "error"
```

Hasil dari proses enkripsi seperti yang dapat dilihat di bawah ini merupakan hasil dari kombinasi AES CBC, *Base64* dan *MD5* yang berbentuk *chipertext* (teks tersandi) dari proses enkripsi.

```
yLA0UOqBo7aLBAwKtYgwg5m4pBDzPiXcnnRYBfatN+txeGVdwKt46BJBpYO4YTP/9WI6Jt5r+/KUjdBi
WsAWAv7XImm2Dcdoe0rNi9f47su7Ny6IUreBKVQYU8KAD0LdIsDWtgW+2V+1UM8DdcjHtWlOByiE/j
2zpskroLQZ2mse1p8NRO1JtAB/agH3KuSN96cAjzRBaElD4QEwWEGI9zNWmlqIzAjG6hXrmGfPgkO2O
6u1h2lOsHarRyuVm3sbExYJ3zPqeCLhJOoF6BJ029Er6JBuzwg1S4+ZK5qqEg2CYn20Vnr70JK9hdOrCB
ZPgPgOOROUnzEaWtWy9BEidrqQpfEGbIwy111ZAYCNHvLkT3B3WWO7lIxIBUmkSFmsS6HDqCAC1jx
o9ds12amtgZgLcoS3iVoUuUhgUw7mJfH+OKKQfkzocjct0EgXLM/+TVdtqUFI+RVwPoW2sWsBhMsMv
d/byU/sKLKFqpiUNHL1/XhbIlYwvM7ND7djXy3PigYxhGbNGfo9kxNa4zO2kRygOViBq1yWWWqZT04HT
UMljnyVaxBhGk6LClWbfQyldt0BHdL51Akf6Y/lv2vOBZaHrP80Bc+yHK6eJlPQ3lvZJ1a9FRtPi4hjZEmeJ/i
ilzfr+EUoYdZ3xbGm12c0yOaqqOygoyNJziYYLFpuYVE8rwTd1N0YGKYCzrz4OuHubh2yPKzr3XVHRKko
7oB3UdEYXEwHu0WmUhbj6slGoAklv24NSNw45XQIwrMNtfES/GtIxPGZabn1j2L7J1Rjxoc9e9HdqaGR
0wpmY4xPuV9GE3C9UCZOCZUpHXvC2jcWfYXVxlAuywbBfEgQZgTvvLkA6gsHmey7MC7uzHwQn3FW
0zJi8H4d4oFMHg8Sdx6FmMZ8p1hTjEbYhk8EnD0xhLFY8evWc7joy2NKpLZJ90n0R8lxhlY/G6Ymi1ihg4
5GvKE6J2KtjY7pTIsyTVS43bgkg7X3SNd4iO7K+cU1vL/JK04wCWQyNOCRYa+a29rUXW1ZOWWK19s
eW4DZ2rXEU3GBLCsqTS/05qOn6WwraBBT6938GYLvJv9EZhVQMXW7V2nMNvGsI4B+QuyQgzZoA63
FBXm6gB4w9sF/m0ndjV87H+UD6/Uf41YxvoJCV0ao8ZJ80VYU32343Fge2FeJK4SE2PvBn98KbEoJDo
wLH4jfxGbZVsppVxoiB1DKuiYeW+jdKSj63M0EuRD5qkngKTa3Sgt0nbP6NxrheaokqB6t8irAwcVDicU+
WIHeQqZ5FKOSkvgmC5wEud3m1jTU9SVkDQnpGkLrT2izzxe0oVjYN3gZcjd0TKRD8XDQP+wgTSQsQ
```

```
vCrJIyjpAcRwcQHZhvfbiHQNyRp3zRPyHKs5/ia86ZzQV7cB/4PkiS/XiUiP20Tziw8FgHlkBE0gDReX675U
sHLiuoM4IM5/JiM/27DaAyE3KhDXzNSnkUnYLMGV+8TE1WwkuRTIMOJSmG7LWHvc7bwglIRC3oa41
s4JwX6KU6tv10wC1B0YgilPFUZSXCJ1/uXXKfif+C7e8OrPw0jcRsi75OsaZnVdfZ7dSDJn3gTt61JSnH5s0
zoJS4dENGbSADlzCmsT67UGO6AU74TvtRditWJbrMZtsKhAgHctIWCGOxDDtG/r6Xy41nNxXadByL/Y7
UbV2gkwuoJ5nHaWZwnqxOw40scTYFwHcBphqpKPmbVR+BvIX3amFXePp2dfdhp1y8ZN87VhDlCshR
nNQ8ncWLbUpCuj/4lLZ+huzc3FjJQzEx6JSNogzOewF2Dk753TY/TWb8iMw73ch6RmLf0VT7ZBnFXjGb
11x6Uk/NdkasAjAf9N3VxGHPaABaSO9QYwQlSq+B/ChDyMRY1C2/4ilXxfOO5y2o/u4fNowZnXMaxe8p
22c+ltXCIN6CdwoahFjxXgEfdY6eGcMvsUT00LYq0vqq4EtWUQIAXIa0GHimGvDxYXV7GGFN49wCGm
QCDahueHcR85o9bSw==
```

Sedangkan pada proses dekripsi menggunakan metode *POST* untuk melakukan proses dekripsi dari data enkripsi yang berbentuk *chiptertext* menjadi *plaintext.* Sebelum mendapatkan hasil hampir sama dengan proses enkripsi, pada proses dekripsi akan mengecek token sebagai keamanan lebih sebelum mendapatkan hasil dari proses dekripsi ini. Adapun *code* pada proses dekripsi dapat dilihat sebagai berikut:

```
@app.route('/dekripsi',methods=['POST', 'GET'])
@token_required
def decrypt():
    if request.method=='POST':
        cte = request.form['cte']
        return pliku.AESCipher('secret').decrypt(cte).decode('utf-8')
    else:
      return "error"
```

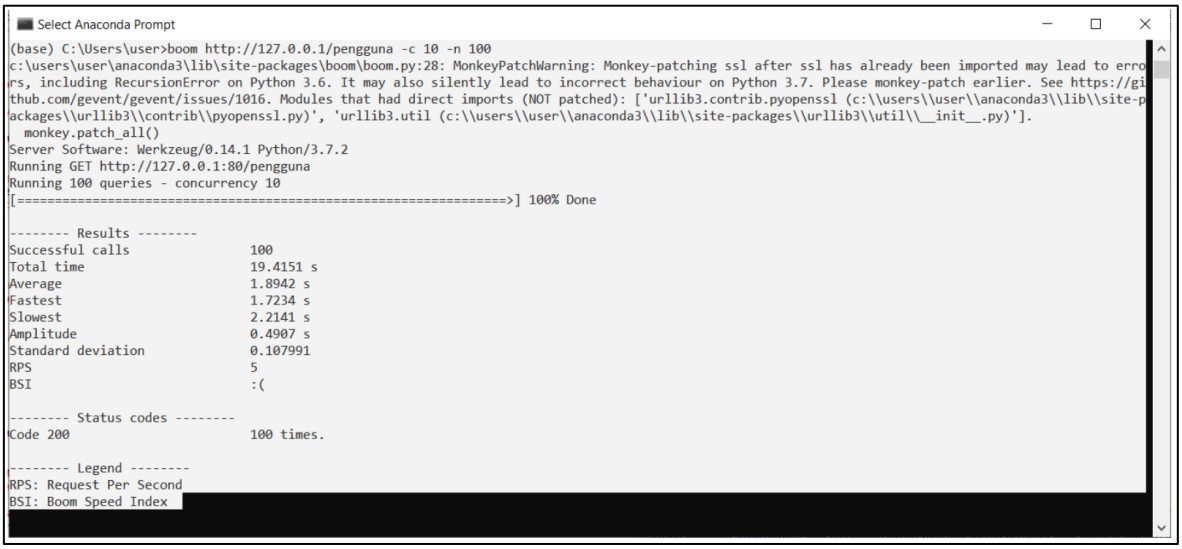
Hasil dari proses deskripsi ini merupakan hasil proses dekripsi dari data berbentuk *chiptertext* yang didapatkan pada proses enkripsi menjadi *plaintext,* sehingga data tersebut dapat dilihat dalam bentuk format XML.

```
({'id_pengguna': '0x40x360x380xB20x210x250x80', 'nama_p': 'Muhammad Nur Ikhsan', 'tanggal': d
atetime.datetime(2019, 7, 19, 10, 2, 12)}, {'id_pengguna': '0x50x8A0x990x1E0x590x310x0 ', 'nama
_p': 'Maulyanda', 'tanggal': datetime.datetime(2019, 7, 19, 10, 2, 12)}, {'id_pengguna': '0x40x8A0x
610xBA0x410x2A0x80', 'nama_p': 'Faisal Syariffudin', 'tanggal': datetime.datetime(2019, 8, 21, 14,
15, 57)}, {'id_pengguna': '0x40x840x7A0x5A0x810x2A0x80', 'nama_p': 'R Rifa Fauzi Komara', 'tang
gal': datetime.datetime(2019, 8, 21, 14, 15, 57)}, {'id_pengguna': '0x40x310x570x5A0x7D0x5B0x8
0', 'nama_p': 'Cahya Kurniawan', 'tanggal': datetime.datetime(2019, 8, 21, 14, 15, 57)}, {'id_pengg
una': '0x40x600x590xCA0x5B0x2A0x80', 'nama_p': 'Aditya Pratama Dharma', 'tanggal': datetime.da
tetime(2019, 8, 21, 14, 20, 41)}, {'id_pengguna': '0x40x5E0x910xE20x9E0x4F0x80', 'nama_p': 'Alit
Fajar Kurniawan', 'tanggal': datetime.datetime(2019, 8, 21, 14, 20, 41)}, {'id_pengguna': '0x40x840
x7A0x5A0x810x2A0x80', 'nama_p': 'R Rifa Fauzi Komara', 'tanggal': datetime.datetime(2019, 8, 21,
14, 24)}, {'id_pengguna': '0x40x230x240xFA0x800x5B0x80', 'nama_p': 'M Raziq Hakim Siregar', 'ta
nggal': datetime.datetime(2019, 8, 21, 16, 40, 27)}, {'id_pengguna': '0x40x360x380xB20x210x250x
80', 'nama_p': 'Muhammad Nur Ikhsan', 'tanggal': datetime.datetime(2019, 8, 27, 7, 53, 28)}, {'id_
pengguna': '0x40x360x380xB20x210x250x80', 'nama_p': 'Muhammad Nur Ikhsan', 'tanggal': dateti
me.datetime(2019, 8, 27, 7, 53, 28)})
```

3.4 Pengujian *Python*

Proses pengujian *python* ini menggunakan *BOOM* yang merupakan paket *tester* dari *python* dimana *BOOM* merupakan pengganti *Apache Bench* yang fungsinya akan melihat kinerja dari sistem enkripsi dan dekripsi yang telah dibuat dengan bahasa pemrograman *python* seperti

keamanan. Dalam pengujian ini menggunakan 100 kueri dengan maksimum 10 pengguna, berikut adalah hasil dari pengujian menggunakan *BOOM* yang dapat dilihat pada Gambar 6.



Gambar 6. Hasil Pengujian Menggunakan *BOOM*

Dari hasil pengujian yang telah dilakukan seperti pada Gambar 6, didapatkan hasil persentase 100% keberhasilan terhadap pengujian *code* program aplikasi, tidak terdapat masalah atau *error* pada proses simulasi sistem kriptografi pada penelitian ini. Sehingga dikategorikan aman untuk digunakan.

4. KESIMPULAN

Setelah melakukan proses enkripsi dan deskripsi dengan menerapkan *Triple Key Md5, Base64* dan AES, untuk menjaga kerahasiaan data proses *tag NFC*. Kesimpulan penelitian sebagai berikut:

1. Penelitian ini berhasil menerapkan algoritma kriptografi menggunakan *triple key Md5, Base64* dan AES untuk enkripsi data E-KTP dari proses *tag NFC* di ruangan *Informatics Research Center* (IRC) yang memungkinkan data E-KTP tersebut aman dan tidak dapat terbaca oleh pengguna yang tidak memiliki akses;
2. Penelitian ini telah berhasil membuat *encrypt* dan *decrypt*, dan di implementasikan pada bahasa pemrograman *python*. *Encrypt* dan *Decrypt* digunakan untuk merahasiakan data E-KTP pada organisasi dengan mengombinasikan *Triple Key* algoritma kriptografi *MD5*, *BASE64* dan AES;
3. Proses pengujian menggunakan metode *BOOM* yang merupakan paket *tester* dari *python*, didapatkan persentase 100% keberhasilan dan tidak terdapat *error* dari sistem algoritma kriptografi yang menggunakan bahasa pemrograman *python*, sehingga proses keamanan nya aman untuk digunakan.

DAFTAR RUJUKAN

Awangga, R. M., & Andarsyah, R. (2016). Pengukuran Performansi Penerapan Asynchronous Daemon Pada Web Service Verifikasi User Di Banana Pi Dengan Metode Benchmarking. *Jurnal Teknik Informatika, 8*, 1--13.

Budianto, W., Amini, S., & Ariyani, P. F. (2017). Aplikasi Pengamanan Dokumen Digital Menggunakan Algoritma Kriptografi Advanced Encryption Standard (Aes-128), Kompresi Huffman Dan Steganografi End Of File (Eof) Berbasis Desktop Pada Cv. Karya Perdana. *Prosiding SNATIF*, 273--280.

Carter, F. W., Khaire, T. S., Novosad, V., & Chang, C. L. (2016). scraps: An open-source Python-based analysis package for analyzing and plotting superconducting resonator data. *IEEE Transactions on Applied Superconductivity, 27*, 1--5.

Christanto, A. T., & Kurniawati, R. (2016). Penerapan Service Oriented Architecture Menggunakan Web Service Pada Aplikasi Perpustakaan Berbasis Android.

Hidayat, A. D., & Afrianto, I. (2017). Sistem Kriptografi Citra Digital Pada Jaringan Intranet Menggunakan Metode Kombinasi Chaos Map dan Teknik Selektif. *Ultimatics: Jurnal Teknik Informatika, 9*, 59--66.

López, A. F., Pelayo, M. C., & Forero, Á. R. (2016). Teaching image processing in engineering using python. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 11*, 129--136.

Masoumi, M., & Rezayati, M. H. (2014). Novel approach to protect advanced encryption standard algorithm implementation against differential electromagnetic and power analysis. *IEEE Transactions on Information Forensics and Security, 10*(2), 256--265.

Moudgalya, K. (2018). Crowdsourced information technology content for education and employment. Dalam *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)* (hal. 39--41). IEEE.

Pane, S. F., Awangga, R. M., & Azhari, B. R. (2018). Qualitative evaluation of RFID implementation on warehouse management system. *Telkomnika, 16*, 1303--1308.

Pane, S. F., Awangga, R. M., & Maulyanda. (2018). Sireuboh: Klasifikasi Data Lokasi Barang Menggunakan Region Of Interest (ROI) dan Algoritma Ransac. *Jurnal Tekno Insentif, 12*, 36--40.

Panjaitan, Y. G., Susanto, A., & WM, I. U. (2017). Enkriptor-Dekriptor Isi E-Mail Berbasis Android Dengan Algoritma Blowfish. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer, 8*, 193--200.

Permana, D. S., Praja, A. S., Fatkhuroyan, F., & Muzayanah, L. F. (2018). Pengolahan dan pemulihan data radar cuaca menggunakan wradlib berbasis python. *Jurnal Meteorologi Dan Geofisika, 17*.

Rohman, F. D., & Mufti, M. (2018). Implementasi Kriptografi Pada Pengiriman Pesan Email Dengan Menggunakan Metode Rc4 Dan Blowfish Berbasis Web Pada Pt. Dascom Jaya Sakti. *SKANIKA, 1*, 788--793.

Sibarani, E. B., Zarlis, M., & Sembiring, R. W. (2017). Analisis Kripto Sistem Algoritma AES Dan Elliptic Curve Cryptography (Ecc) Untuk Keamanan Data. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, 1*, 106--112.

Siswanto, S., Anif, M., & Gata, W. (2018). Penerapan Algoritma Kriptografi TEA Dan Base64 Untuk Mengamankan Email Data Policy Asuransi. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer, 2*(1), 34--41.

Sumarno, S., & others. (2018). Analisis Kinerja Kombinasi Algoritma Message-Digest Algortihm 5 (MD5), Rivest Shamir Adleman (RSA) dan Rivest Cipher 4 (RC4) Pada Keamanan E-Dokumen. *Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOM PRIMA), 2*.

Tibau, M. a., Nunes, B. P., Bortoluzzi, M., & Marenzi, I. (2018). Modeling exploratory search as a knowledge-intensive process. Dalam *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)* (hal. 34--38). IEEE.