Implementasi Sistem Kriptografi Algoritma *AES (256-bit)* Berbasis *Web API* untuk Mengamankan Data Pribadi di CV. Elang Cahaya Sukses Surakarta

Andreas Abi Permana[1], Joni Maulindar[2], Dwi Hartanti[3]

S-1 Teknik Informatika, Universitas Duta Bangsa Surakarta
Jl. Bhayangkara No.55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah

[1]andreas_abi@fikom.udb.ac.id
[2]joni_maulindar@udb.ac.id
[3]dwi_hartanti@udb.ac.id

Abstrak— Peningkatan penggunaan internet dekade terakhir membawa kerahasiaan data menjadi salah satu perhatian utama. Peningkatan tersebut membuat kerahasiaan data menjadi krusial ketika menghadapi serangan siber yang marak terjadi. Kerahasiaan data berarti melindungi data privasi dari pihak yang tidak bertanggung jawab. Kerahasiaan data dapat dicapai dengan menerapkan kriptografi. CV. Elang Cahaya Sukses Surakarta telah menerapkan kriptografi pada aplikasi absensi karyawan, dengan melakukan proses enkripsi data pribadi. Akan tetapi penerapan tersebut kurang menjamin kerahasiaan data ketika terjadi serangan atau akses yang tidak diotorisasi oleh aplikasi, karena kunci enkripsi masih disimpan di dalam kode sumber. Peneliti menggunakan metode studi literatur, observasi dan waterfall untuk menyelesaikan masalah tersebut. Hasil dari penelitian ini yaitu sistem kriptografi algoritma AES (256bit) berbasis web API. Peneliti menyimpulkan bahwa sistem yang dibuat dapat menjadi solusi untuk memperkuat pengamanan data, khususnya dalam penyimpanan data pribadi di CV. Elang Cahaya Sukses.

Kata kunci— kriptografi, algoritma AES (256bit), keamanan sistem, manajemen kunci

Abstract— The increasing use of the internet in the last decade has brought data confidentiality one of the main concerns. This increase makes data confidentiality crucial when facing cyber-attacks that are rife. Data confidentiality means protecting privacy data from irresponsible parties. Data confidentiality can be achieved by applying cryptography. CV. Elang Cahaya Sukses Surakarta has applied cryptography to employee attendance applications by encrypting personal data. However, this application doesn't guarantee data confidentiality in the event of an attack or unauthorized access by the application because the encryption key is still stored in the source code. The researcher used the literature study, observation, and waterfall methods to solve the problem. These research results are the AES (256bit) algorithm cryptography system based on web API. The researcher concludes that the system created can be a solution to strengthen data security, especially in storing personal data in CV. Elang Cahaya Sukses Surakarta.

Keywords— cryptography, AES(256bit) algorithm, security system key management

I. PENDAHULUAN

Peningkatan penggunaan internet pada dekade terakhir, membawa kerahasiaan data menjadi salah satu perhatian utama. Peningkatan tersebut membuat kerahasiaan data menjadi sangat krusial apalagi ketika menghadapi serangan siber yang makin marak terjadi [1]. Kerahasiaan data disini berarti melindungi data privasi dari pihak yang tidak bertanggung jawab. Kerahasiaan data ini dapat dicapai dengan menerapkan ilmu kriptografi melalui proses yang bernama enkripsi dan dekripsi. Tujuan dari penerapan kriptografi ini, yaitu mengamankan data atau dokumen pribadi baik ketika disimpan ke media penyimpanan maupun ketika ditransmisikan ke beberapa jalur komunikasi digital. Kriptografi bekerja dengan menerapkan beberapa algoritma dalam melakukan proses enkripsi dan dekripsi. Setiap algoritma enkripsi memiliki tingkat keamanan, kecepatan, kompleksitas, dan kerumitan tersendiri. Maka dari itu, pemilihan algoritma enkripsi ini merupakan aspek penting dalam kriptografi, Algoritma *AES* menghasilkan hasil terbaik dalam hal waktu enkripsi, throughput dan memiliki tingkat keamanan yang lebih baik [2].

CV. Elang Cahaya Sukses Surakarta *(ECS INFOTECH)* merupakan persekutuan komanditer yang bergerak di bidang konsultasi dan pembuatan sistem berbasis teknologi informasi. Persekutuan komanditer tersebut telah menerapkan perlindungan data pribadi menggunakan ilmu kriptografi pada aplikasi absensi karyawan, yang mana melakukan proses enkripsi data pribadi seperti, nik, tanggal lahir, alamat, dan nomor jaminan sosial dan disimpan di dalam basis data. Akan tetapi penerapan tersebut kurang menjamin kerahasiaan data ketika terjadi serangan atau akses yang tidak diotorisasi oleh aplikasi karena kunci enkripsi masih disimpan di dalam kode sumber. Hal ini menjadi masalah besar ketika pihak tidak bertanggung jawab berhasil masuk ke sistem dan akan dengan sangat mudah untuk melakukan dekripsi data menggunakan kunci yang tersimpan di kode sumber.

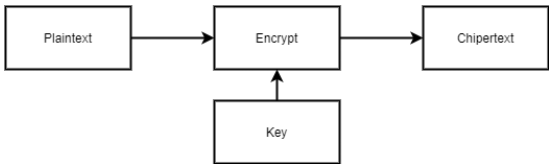
Masalah yang telah dipaparkan di atas dapat diselesaikan dengan penerapan manajemen kunci secara terpisah dari

aplikasi. Manajemen kunci merupakan teknik yang dikembangkan untuk menjaga kerahasiaan kunci. Manajemen kunci ini mencakup beberapa operasi, yaitu pembuatan kunci, distribusi kunci, pemeriksaan kunci, penyimpanan kunci, dan pembuatan cadangan kunci. Oleh karena itu peneliti memutuskan untuk melakukan analisis, perancangan, dan pembuatan aplikasi manajemen kunci tersendiri yang berjalan berbasis *web API* yang diharapkan dapat memberikan perlindungan tambahan keamanan data pribadi yang disimpan pada aplikasi absensi di CV. Elang Cahaya Sukses Surakarta.

II. LANDASAN TEORI

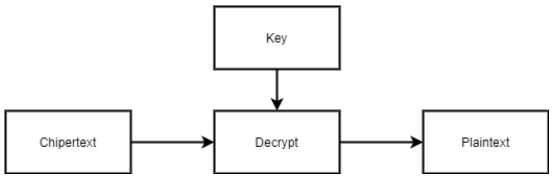
Data pribadi dapat didefinisikan sebagai data individu tertentu yang disimpan, dipelihara, dijaga kebenaran, dan rahasianya [3]. Data tersebut menjadi aset berharga dan memiliki nilai tinggi baik bagi individu personal yang rawan untuk disalahgunakan.

Kriptografi adalah seni dan ilmu menyembunyikan informasi penting dan rahasia agar tidak dilanggar oleh orang yang tidak berwenang [1]. Dalam istilah kriptografi, pesan rahasia ini disebut sebagai *plaintext*, sementara pesan yang sudah dikaburkan atau disembunyikan disebut sebagai *ciphertext*.



Gambar 1. Proses Enkripsi

Proses translasi dari *plaintext* ke *ciphertext* ini disebut proses enkripsi, sebaliknya proses translasi dari *ciphertext* ke *plaintext* disebut proses dekripsi. Pada proses tersebut, diperlukan komponen tambahan yang disebut sebagai kunci yang mana digunakan untuk menyembunyikan serta mengembalikan pesan dalam kriptografi. Proses enkripsi dapat dilihat pada Gambar 1 dan proses dekripsi dapat dilihat pada Gambar 2.



Gambar 2. Proses Dekripsi

Proses ini menggunakan algoritma dengan beberapa parameter tertentu tergantung metode yang digunakan. Kriptografi sendiri terbagi atas 3 macam teknik, yaitu kunci simetris, kunci asimetris, dan fungsi *hash* [2].

Algoritma *Advanced Encryption Standard* (juga disebut sebagai algoritma *Rijndael*) adalah algoritma *cipher* blok simetris yang menggunakan 128, 192, atau 256 *bit* kunci untuk mengubah blok pesan *bit* menjadi *bit ciphertext* [4].

Manajemen Kunci merupakan dasar yang digunakan untuk melakukan manajemen siklus hidup kunci, berupa pembuatan, distribusi, penyimpanan, dan penghancuran dalam sistem kriptografi [5].

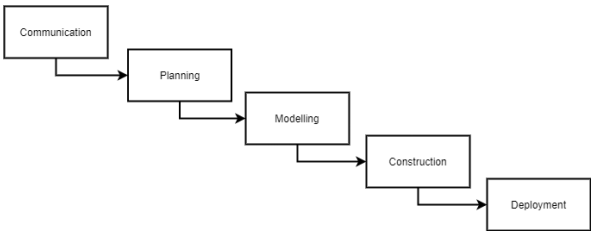
Application Programming Interface (API) adalah suatu layanan berupa aturan-aturan yang memungkinkan aplikasi berkomunikasi dengan aplikasi lain guna saling bertukar informasi satu sama lain. *Application Programming Interface (API)* bertindak sebagai lapisan tengah antara *server* dan aplikasi untuk memproses transfer data. *Application Programming Interface (API)* merupakan salah satu elemen penting dalam rekayasa perangkat lunak yang mempercepat pengembangan suatu perangkat lunak [6].

III. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan adalah studi literature dan pengamatan langsung pada objek penelitian.

Peneliti mencari artikel ilmiah dan buku referensi yang relevan dengan penelitian yang dilakukan. Sumber referensi yang digunakan berkaitangan mengenai implementasi pengamanan data. Peneliti juga melakukan pengamatan langsung dan pengambilan data di CV. Elang Cahaya Sukses Surakarta yang berhubungan dengan proses pengamanan data.

Metode pengembangan yang digunakan pada penelitian ini menggunakan pendekatan terstruktur yaitu waterfall. *Waterfall* merupakan pendekatan yang sistematis dan sekuensial dalam pengembangan perangkat lunak yang dimulai pada tingkatan sistem dan bergerak maju mulai tahap *communication, planning, modelling, construction, deployment* [7]. Tahapan dalam *waterfall* adalah sebagai berikut:



Gambar 3. Proses *Waterfall*

1) Communication

Pada tahap ini, peneliti melakukan inisiasi proyek pembuatan aplikasi dan melakukan proses analisis kebutuhan.

2) Planning

Perencanaan yang dilakukan pada tahap ini yaitu, melakukan perencanaan estimasi waktu dan beban proyek, penjadwalan proses pengembangan aplikasi, dan melakukan penyesuaian terhadap kondisi yang ada di lapangan.

3) Modelling

Pada tahap ini peneliti merancang alur kerja aplikasi, membuat rancangan proses, membuat rancangan basis data, relasi antar tabel, dan melakukan desain antar muka sistem informasi.

4) Construction

Pada tahap ini peneliti melakukan pembuatan aplikasi yang telah didesain, dan melakukan pengujian aplikasi sebelum diserahkan ke pengguna.

5) Deployment

Pada tahap ini peneliti membuat dokumentasi baik penggunaan dan pengembangan aplikasi, menyerahkan aplikasi ke pengguna dan terakhir melakukan pemeliharaan aplikasi secara berkala melalui umpan balik dari pengguna.

IV. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

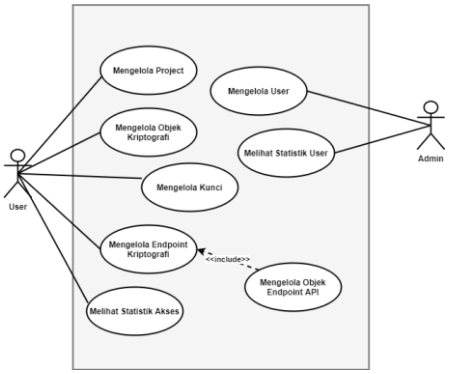
Peneliti melakukan analisis terhadap permasalahan yang dihadapi dan menetapkan kebutuhan perangkat lunak dan bertujuan untuk memahami sistem yang ada, mengidentifikasi masalah dan mencari solusinya.

Hasil analisis kebutuhan sistem yang akan dibangun diperlihatkan pada Tabel 1.

Tabel 1. Analisis Kebutuhan Sistem

No	Hak Akses	Kebutuhan
1	*Admin*	Dapat mengelola pengguna aplikasi
		Dapat melihat statistik pengguna
2	*User*	Dapat mengelola objek kriptografi berdasarkan proyek
		Dapat mengelola objek kriptografi yang akan diamankan
		Dapat mengelola kunci dari setiap objek kriptografi
		Dapat mengelola *endpoint API* untuk proses kriptografi
		Dapat mengelola data objek endpoint *API* pada proyek
		Dapat melihat statistik akses dari *endpoint API*

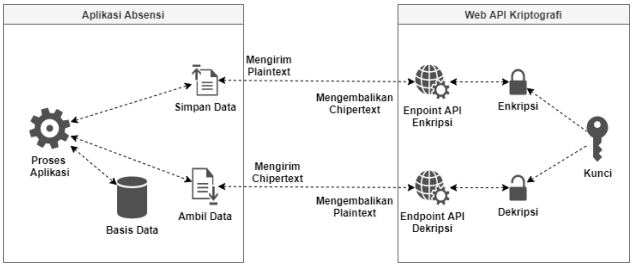
B. *Desain Sistem*



Gambar 4. Use Case Diagram Sistem

Desain sistem menggunakan *Use Case Diagram*. *Use Case Diagram* merupakan diagram dalam *Unified Modeling Language (UML)* yang menentukan fungsionalitas dan fitur perangkat lunak dari sudut pandang pengguna yang terdapat dalam sistem yang akan dikembangkan[7]. *Use Case Diagram* sistem yang dibuat terdapat pada Gambar 4. Diagram tersebut dibuat berdasarkan analisis kebutuhan yang dilakukan sebelumnya.

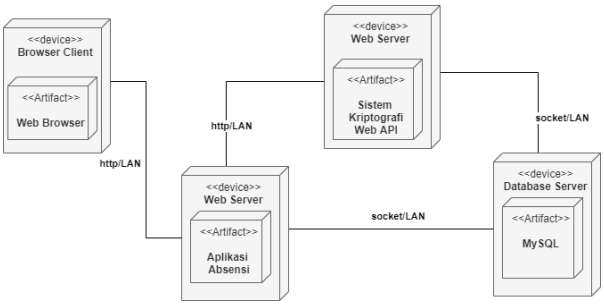
C. Arsitektur Sistem



Gambar 5. Arsitektur Sistem

Sistem keamanan yang dibuat akan memisahkan pengamanan data pada aplikasi lalu diserahkan kepada sistem kriptografi. Sistem ini melakukan proses enkripi, dekripsi, dan penyimpanan kunci. Kunci yang digunakan untuk proses kriptografi, akan dikelola oleh sistem. Sehingga sebelum data disimpan kedalam basis data atau ketika akan diproses oleh aplikasi, data akan di enkripsi atau di dekripsi terlebih dahulu. Gambaran dari arsitektur sistem dapat dilihat pada Gambar 5.

D. *Desain Pemasangan Sistem*



Gambar 6. Deployment Diagram Sistem

Desain pemasangan sistem menggunakan *Deployment Diagram. Deployment Diagram* merupakan diagram dalam *Unified Modeling Language (UML)* yang berfokus pada struktur sistem perangkat lunak dan berguna untuk menunjukkan distribusi fisik sistem perangkat lunak di antara platform perangkat keras dan lingkungan eksekusi [7]. Pada Gambar 6 terdapat 4 *node* yang saling terhubung dalam satu jaringan lokal. Aplikasi absenai berkomunikasi langsung dengan pengguna untuk digunakan, sistem kriptografi untuk mengamankan datanya, dan terakhir basis data untuk menyimpan data.

E. Implementasi Sistem

Implementasi Sistem dibuat menggunakan bahasa pemrograman *Python* dengan kerangka kerja *Django 3*. Basis

data yang digunakan adalah *MySQL*. Sementara data kriptografi yang dikirimkan berbentuk *JSON* dengan media pengiriman melalui *Application Programming Interface*. Pengiriman data permintaan enkripsi maupun dekripsi menggunakan *format* data *JSON* yang telah ditentukan ke *endpoint API* sistem kriptografi.

```
{
    "api-key":"yIFup9n4Pw0b4eieh4RaR9gm1iEVAgVc",
    "objek":"karyawan",
    "aksi":"enkrip",
    "plaintext":"Y29udG9oIGRhdGEga2FyeWF3YW4K"
}
```

Gambar 7. Data JSON enkripsi

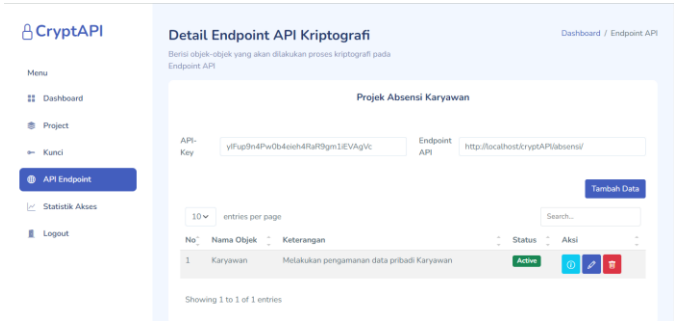
Ketika enkripsi data, aplikasi absensi mengirimkan data *plaintext* dalam bentuk telah di *encode* menjadi *base64*, tipe aksi yang dilakukan yaitu enkrip, objek merupakan penanda aplikasi pada sistem kriptografi, dan terakhir adalah kunci *API* sebagai kunci dan sekaligus penanda ketika pengguna menggunakan layanan sistem kriptografi. Data *JSON* untuk melakukan enkripsi terdapat pada Gambar 7.

```
{
    "api-key":"Y29udG9oIGRhdGEga2FyeWF3YW4K",
    "objek":"karyawan",
    "aksi":"dekrip",
    "ciphertext":"umzPD+RWylRCGpPxshV0oqI6znoY20nymzfxPgsryXo="
}
```

Gambar 8. Data JSON dekripsi

Ketika dekripsi data, format data yang dikirimkan hampir sama. *Ciphertext* dikirimkan dalam bentuk telah di *encode* menjadi *base64*, dan tipe aksi yang dilakukan yaitu dekrip. Data *JSON* untuk melakukan enkripsi terdapat pada Gambar 8.

Kunci *API* dibangkitkan melalui sistem kriptografi. Kunci tersebut bersifat otentik untuk setiap projek dalam sistem dan sebagai otentikasi khusus bagi pengguna ketika menggunakan layanan *API* sistem kriptografi, tanpa kunci sistem tidak akan bisa diakses. Tampilan halaman detail *endpoint API* kriptografi yang berisi kunci *API* dapat dilihat pada Gambar 9.



Gambar 9. Halaman Detail *Endpoint API* Kriptografi

Setiap akses permintaan proses kriptografi, baik enkripsi maupun dekripsi akan tercatat dan tersimpan ke dalam sistem. Data tersebut ditampilkan dalam statistik akses pengguna dan admin dalam bentuk grafik. Tampilan halaman statistik akses pengguna dapat dilihat pada Gambar 10.

Gambar 10. Halaman Statistik Akses Pengguna

Aplikasi absensi menggunakan bahasa pemrograman *PHP* dengan kerangka kerja *CodeIgniter 3*. Peneliti mengimplementasikan sistem kriptografi dengan membuat *library* khusus untuk melakukan proses komunikasi ke layanan *Web API*. Sebelum menyimpan data objek kriptografi ke basis data, *library* tersebut dipanggil untuk melakukan enkripsi data. Contoh penerapannya dapat dilihat pada Gambar 11.

```
$this->load->library('cryptAPI');
$data = array(
    'nik' => $this->cryptAPI->encrypt($this->input->post('nik'),'karyawan'),
    'nama' => $this->cryptAPI->encrypt($this->input->post('nama'),'karyawan'),
    'rfid_tag' => $this->input->post('rfid_tag'),
    'alamat' => $this->encryption->encrypt($this->input->post('alamat'),'karyawan'),
    ...
    );
$insert = $this->m_tambah->master_user($data);
```

Gambar 11. Contoh penerapan enkripsi

Dekripsi data diterapkan setelah data diambil dari basis data, karena data yang diambil masih dalam bentuk *plaintext* maka data harus di dekripsi terlebih dahulu. Aplikasi absensi memanggil *library* untuk melakukan dekripsi, baru setelah itu data dapat diproses ke proses bisnis dalam aplikasi. Contoh penerapannya dapat dilihat pada Gambar 12.

```
public function getMasterUserById($id){
    $datas = $this->m_ambil->getMasterUserById($id);
    $this->load->library('cryptAPI');
    $data = array(
        'nik' => $this->cryptAPI->encrypt($datas['nik'],'karyawan'),
        'nama' => $this->cryptAPI->encrypt($datas['nama'],'karyawan'),
        'rfid_tag' => $datas['rfid_tag'],
        'alamat' => $this->encryption->encrypt($datas['alamat'],'karyawan'),
        ...
    );
    echo json_encode($data);
}
```

Gambar 12. Contoh penerapan dekripsi

F. Uji coba Sistem

Peneliti melakukan pengujian secara langsung setiap fungsi – fungi dari sistem setelah dikonstruksi. Pengujian fungsi utama sistem berupa proses kriptografi dilakukan menggunakan aplikasi *curl*.

```
andreas@DESKTOP-09JFMDJ:~$ echo "contoh data karyawan" | base64
Y29udG9oIGRhdGEga2FyeWF3YW4K
andreas@DESKTOP-09JFMDJ:~$ curl  --request POST  --data '{"api-key":"yIF
up9n4Pw0b4eieh4RaR9gm1iEVAgVc","objek":"karyawan","aksi":"enkrip","plaint
ext":"Y29udG9oIGRhdGEga2FyeWF3YW4K"}' localhost/CryptAPI/Crypt/absensi/
{"status":"Sukses!","ciphertext":"umzPD+RWylRCGpPxshV0oqI6znoY20nymzfxPgs
ryXo="}andreas@DESKTOP-09JFMDJ:~$
andreas@DESKTOP-09JFMDJ:~$
```

Gambar 13. Uji coba enkrip

Pengujian proses enkripsi dilakukan dengan mengirimkan data pengujian untuk menguji jalannya proses enkripsi. Pada Gambar 13, diketahui bahwa proses enkripsi berjalan dengan baik dan menghasilkan keluaran berupa *ciphertext*.

```
andreas@DESKTOP-09JFMDJ:~$ curl  --request POST   --data '{"api-key":"yIFup9
n4Pw0b4eieh4RaR9gm1iEVAgVc","objek":"karyawan","aksi":"dekrip","chipertext":
"umzPD+RWylRCGpPxshV0oqI6znoY20nymzfxPgsryXo="}' localhost/CryptAPI/Crypt/ab
sensi/
{"status":"Sukses!","plaintext":"Y29udG9oIGRhdGEga2FyeWF3YW4K"}andreas@DESKT
OP-09JFMDJ:~$
andreas@DESKTOP-09JFMDJ:~$ echo "Y29udG9oIGRhdGEga2FyeWF3YW4K" | base64 -d
contoh data karyawan
andreas@DESKTOP-09JFMDJ:~$
```

Gambar 14. Uji coba dekrip

Pengujian selanjutnya yaitu, proses dekripsi dilakukan dengan mengirimkan data *ciphertext* sebelumnya untuk menguji jalannya proses dekripsi. Pada Gambar 14, diketahui bahwa proses dekripsi berjalan dengan baik dan menghasilkan keluaran berupa plaintext yang sama ketika sebelum dilakukan enkripsi.

```
mysql> use absensi;
Database changed
mysql> SELECT nik,nama FROM karyawan;
+--------------------------------------------+--------------------------------------------+
| nik                                        | nama                                       |
+--------------------------------------------+--------------------------------------------+
| chI8c+iqsLkJtlizGPDcEkRrbEfAAbSMjI2i3tt0zfQ= | EUUtd8pab47KzF7WuOK6zc9RIq1Qe9/++iwnAvKMGVY= |
| tY1xSZ5VrPX4TuZ7By7mI4CdDw/HG+hhQzZJrHdLfj8= | LAcM99vo5MvtXCvMgXyyPG6q5cCMVIVSucUsQnaiu/k= |
| MQyC+R1ISev9JcmzvnbwdU30+bJdmlBcIdCiJJBwGVQ= | ozskWhGLlukhtEnzQxlZzeP2etX3T3+d96+9hNk/PvL8= |
| mzyyYJS7rY8TN2OKJJ666eyMkRGCpNgMKrCR2JhF5WK= | 7hh3J+vlCRr2Wai49uEkC8TEuEOeGEiB6sy7MDvF0S0= |
| mzSNGLNSPkwpKnsiPmmhXtODR/b0mtCh0n5pAcEyPYY= | 7G2OmuV7dJnooel4by9oN5CS7HJom1s2yegLEWKblrk= |
+--------------------------------------------+--------------------------------------------+
5 rows in set (0.00 sec)
```

Gambar 15. Uji coba penerapan aplikasi absensi

Pengujian terakhir dilakukan dengan menerapkan fungsi pengamanan data menggunakan sistem kriptografi pada aplikasi absensi. Pada Gambar 15, diketahui bahwa proses pengamanan data berjalan dengan baik dan data yang disimpan di basis data berupa data hasil dari proses enkripsi.

V. KESIMPULAN

Kesimpulan yang dapat diperoleh dari Implementasi Sistem Kriptografi Algoritma *AES (256bit)* berbasis *Web API* adalah sebagai berikut:

1) Sistem Kriptografi berbasis *Web API* dapat diterapkan dalam pengamanan data pribadi khusus pada aplikasi absensi di CV. Elang Cahaya Sukses Surakarta.
2) Algoritma *Kriptografi AES (256bit)* dapat diimplementasi pada Sistem Kriptografi berbasis *Web API* yang dibangun dengan bahasa pemrograman *Python* dan kerangka kerja *Django 3*.
3) Sistem yang dibuat dapat menjadi solusi untuk memperkuat pengamanan data, khususnya dalam penyimpanan data pribadi di CV. Elang Cahaya Sukses.

VI. SARAN PENGEMBANGAN

Peneliti menyadai bahwa sistem yang dikembangkan masih jauh dari kata sempurna. Sistem Kriptografi Algoritma AES 265-bit ini masih sangat sederhana sehingga membutuhkan penelitian dan pengembangan lebih lanjut sehingga dapat terbentuknya suatu sistem yang lebih stabil, efisien dan aman. Berikut ini adalah beberapa potensi pengembangan sistem yang dapat dilakukan pada penelitian selanjutnya:

1) Penelitian lanjut pada penerapan sistem kriptografi pengamanan data yang lebih aman dan efisien.
2) Implementasi penambahan opsi algoritma pada pembangkitan kunci.
3) Penambahan fitur untuk migrasi dan pembangkitan ulang kunci.
4) Penyempurnaan desain sistem dan proses.

REFERENSI

[1] M. T. Gençoğlu, “Importance of Cryptography in Information Security,” *IOSR J. Comput. Eng.*, vol. 21, no. 1, hal. 65–68, 2019, doi: 10.9790/0661-2101026568.

[2] M. N. Alenezi, H. Alabdulrazzaq, dan N. Q. Mohammad, “Symmetric encryption algorithms: Review and evaluation study,” *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 2, hal. 256–272, 2020.

[3] I. Thaher, “LEGAL POLITICS: PERSONAL DATA PROTECTION IN PEDULI PROTECT APPLICATIONS IN INDONESIA,” *J. Res. Soc. Sci. Econ. Manag.*, vol. 01, no. 8, hal. 1195–1206, 2022.

[4] D. Yehya dan M. Joudi, “AES Encryption : Study & Evaluation,” *CCEE552 Cryptogr. Netw. Secur.*, no. November, 2020, [Daring]. Tersedia pada: https://www.researchgate.net/publication/346446212_AES_Encryption_Study_Evaluation.

[5] A. Bentajer dan M. Hedabou, “CRYPTOGRAPHIC KEY MANAGEMENT ISSUES IN CLOUD COMPUTING,” *Adv. Eng. Res.*, vol. 34, no. July, hal. 257, 2021, [Daring]. Tersedia pada: https://www.researchgate.net/publication/353523527_A_Two-Station_Radio_Navigation_Method_Using_the_Solution_of_Redundant_Equation_ADVANCES_IN_ENGINEERING_RESEARCH_ADVANCES_IN_ENGINEERING_RESEARCH_VOLUME_34?enrichId=rgreq-75deb1f63700abb75175014668a09035-X.

[6] Z. Halim, R. Canda, R. Sadjirin, dan Z. Sahri, “Machine learning model deployment as API service using Python Web Framework,” *e-Proceedings Inf. Technol. Comput. Sci. Colloq. Ser.*, vol. 1, no. 1, hal. 18–20, 2021, [Daring]. Tersedia pada: https://academic-raub.com/virtcs/index.php/en/article/view/14.

[7] B. R. Maxim dan R. S. Pressman, *Software engineering: A Practioners Appoarch*. 2020.