Analisis Sistem Autentikasi Otp Data Medis Dengan Menggunakan Teknik *Embbeded Symmetric Key*

Analysis Of Otp Authentication System Medical Data Using Embedded Symmetric Key Technique

1st Josua Cong Yapo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
acongsitanggang@student.telkomuniversity.ac.id

2nd Ida Wahidah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
wahidah@telkomuniversity.ac.id

3rd Fardan
Teknik Telekomunikasi
Universitas Telkom
Bandung, Indonesia
fardanfnn@telkomuniversity.ac.id

Abstrak

Perkembangan dunia medis sudah sangat pesat, perangkat medis *wearable* sudah mendapat peranan penting di dalam dunia medis. Berbeda dengan alat yang berkoneksi dengan kabel, *wearable* adalah alat dengan koneksi nirkabel yaiut perangkat yang dapat dikenakan dengan jarak jauh. Pada pembuatan sistem ini, akan di aplikasikan pada sebuah rompi kesehatan, rompi ini akan ditanamkan beberapa sensor seperti, pembacaan detak jantung secara *realtime*, dan lain-lain. Untuk meningkatkan keamanan data, alat ini akan dilengkapi dengan autentikasi *One Time Password*(OTP) dan keamanan kunci simetris menggunakan metode AES-256 dan SHA-256.

Dari hasil pengujian yang telah dilakukan, diketahui bahwa sistem dapat bekerja dengan baik. Kode OTP muncul sesuai dengan data pengguna yang terdaftar, untuk sistem enkripsi juga telah bekerja dengan baik, data-data yang telah di *input* dapat di enkripsi dari data *plaintext* menjadi *ciphertext.*

Kata Kunci : OTP,*Wearable*,Autentikasi AES dan SHA

Abstract

The development of the medical world is very rapid, wearable medical devices have played an important role in the medical world. Unlike a wired device, a wearable is a device with a wireless connection, i.e. a device that can be worn remotely. In making this system to be applied to the health vest, several sensors will be implanted in this vest, such as real-time heart rate readings, and others. To improve data security, this tool will be equipped with One Time Password (OTP) authentication and symmetric key security using AES-256 and SHA-256 methods.

From the results of the tests that have been carried out, it is known that the system can work well. The OTP code that appears is in accordance with the registered user data, for the encryption system has also been running well, the inputted data can be encrypted from plaintext data into ciphertext.

Keywords: OTP, Wearable, AES Authentication and SHA

I. PENDAHULUAN

Pandemi Covid-19 sampai saat ini masih belum juga dapat diredam secara tuntas, berbagai upaya dan tindakan sudah dilakukan pemerintah maupun masyarakat. Protokol kesehatan yang sudah dijalankan untuk menekan peyebaran virus. Pandemi Covid-19 menciptakan suatu kehidupan baru (*new* normal) pada lingkungan masyarakat. *New normal* adalah perubahan perilaku untuk tetap melakukan aktivitas normal dengan menerapkan protokol kesehatan, seperti memakai masker, sering mencuci tangan dengan sabun, hingga menghindari kerumunan orang. Maka dari itu penulis merancang sistem ini, dimana sistem ini akan mendukung aturan pemerintah dimana masyarakat dihimbau agar tetap melakukan aktivitas di dalam rumah, dimana alat ini dapat digunakan bagi masyarakat yang ingin melakukan *checkup* mandiri.

Untuk menambah keamanan sistem, alat akan dilengkapi dengan sistem yang telah dirancang yaitu penambahan autentikasi *One Time Password* (OTP) dan sistem enkripsi dan dekripsi dengan metode AES-256 dan SHA-256. Kode OTP akan bekerja pada saat pengguna akan melakukan *checkup* mandiri, ketika pengguna masuk ke sistem dan berhasil melakukan input data, maka data yang telah di input akan di

enkripsi pada *database* sistem menggunakan AES-256 dan SHA-256. Untuk OTP sendiri adalah kata sandi yang berlaku dalam satu sesi login dan memiliki durasi waktu sandi. *Plaintext* yang akan digunakan untuk OTP terdiri dari kombinasi *username* dan nomor ponsel yang telah terdaftar. *Plaintext* ini akan di *hash* dan di enkripsi menjadi *ciphertext* dengan metode AES-256. Dan dengan skema LCG (*Linear Congruential Generator*) *ciphertext* akan di acak sebanyak 6 kali untuk mendapatkan OTP. Sistem yang dirancang berupa simulasi yang dibuat di dalam satu perangkat.

II. TINJAUAN PUSTAKA

a. *One Time Password*

OTP hanya melindungi sistem terhadap serangan pasif tetapi tidak dapat mengatasi serangan aktif. Sistem ini terdiri dari 2 entitas yaitu generator dan server. Secara umum, generator akan menerima masukan-
berupa *username* dan *challenge* dari server, kemudian menghasilkan OTP. Sedangkan server bertugas mengirimkan *challenge* yang sesuai dengan *user*, memverifikasi OTP yang diterima, dan menyimpan OTP terbaru

b. *Wearable Device*

Wearable device adalah suatu perangkat dengan sensor dan komputasi yang dirancang dan digunakan pada bagian tubuh tertentu untuk memonitor dan menampilkan data kesehatan penggunanya. *Wearable device* dapat berkomunikasi langsung pada perangkat yang sudah tertanam konektivitas nirkabel atau dengan perangkat lain, misalnya *smartphone.* Data dari tubuh pengguna yang dikumpulkan dari perangkat melalui berbagai sensor yang telah terpasang dapat secara langsung diproses dan hasil nya akan langsung ditampilkan ke pengguna.

Gambar 2.1 *Wearable Device*

c. Autentikasi

Autentikasi adalah istilah yang digunakan untuk menggambarkan bagaimana objek dikonfirmasi menjadi objek yang diklaimnya. Objek dapat menjadi manusia atau agen lain yang berpartisipasi dalam sistem. Secara garis besar, autentikasi mencoba mengidentifikasi objek berdasarkan sesuatu yang diketahui, misalnya kata sandi, sesuatu dalam kepemilikan objek. Metode yang terkenal (tetapi bukan satu-satunya) untuk melakukan autentikasi adalah menantang objek untuk memberikan respon berdasarkan rahasia (sandi) dan kemudian mengkonfirmasi bahwa respon terhadap tantangan sudah benar.

Gambar 2.2 *Username* dan *Password*

d. *Advanced Encrypted Standard* (AES)

AES diadopsi dari *National Institute of Standards and Technology* (NIST) sebagai pengganti Standar Enkripsi Data (DES). Ini dilakukan karena kunci yang digunakan pada algoritma DES terlalu pendek (56 bit) sehingga tidak dapat menjamin tingkat keamanan data yang tinggi. Algoritma AES melakukan operasi pada *plaintext* 128bit dan menggunakan kunci identik untuk melakukan enkripsi serta dekripsi. Algoritma AES memproses fakta yang menghalangi bagian 128bit dan melakukan 10,12,14 putaran operasi menggunakan kunci rahasia.

e. *Linear Congruential Generator*(LCG)

Linear Congruential Generator(LCG) adalah metode populer dan paling banyak digunakan untuk menggasilkan angka atau kode acak. LCG ditemukan oleh D.H Lehmer, LCG menggunakan model linier untuk menghasilkan angka acak yang didefenisikan sebagai berikut:

Xn 1=(aXn c) mod m(1) dimana a adalah pengganda, c adalah faktor kenaikan dan m adalah modulus. Parameter a, c dan m harus dipilih dengan hati-hati untuk menghindari pengulangan angka serupa sebelum m. Modulus m harus menjadi bilangan bulat prima yang besar, sedangkan pengganda a harus menjadi bilangan bulat dalam rentang 2,3,…m-1. Panjang siklus LCG tidak akan melebihi modulus m, tetapi dapat dimaksimalkan menggunakan tiga kondisi berikut:

- C relatif prima untuk modulus m;
- Pengganda a-1 adalah kelipatan dari setiap modulus m yang membagi;
- Pengganda a-1 adalah kelipatan dari empat ketika modulus m adalah kelipatan empat juga

f. OTP berdasarkan AES

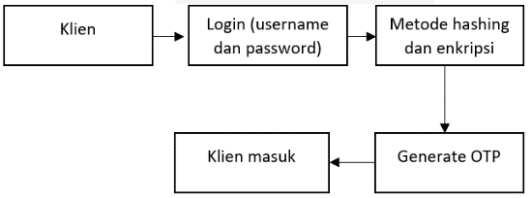
Langkah pertama OTP adalah perhitungan OTP itu sendiri, yang merupakan algoritma untuk menghasilkan kode sandi unuk untuk setiap autentikasi. Berikut adalah proses umum menghasilkan OTP pada sistem.

A. *Plaintext* adalah gabungan dari nama pengguna dan juga nomor ponsel.
B. Nomor ponsel akan di hashing dengan metode SHA-256 sebagai kunci untuk proses enkripsi.
C. *Plaintext* akan di enkripsi dengan metode SHA-256 dengan hasil hashing nomor ponsel sebagai kunci.

Hasil enkripsi ini akan berupa *ciphertext*, hasil ini akan di *randomize* oleh LCG sebanyak 6 kali untuk mendapatkan indeks karakter OTP.

III. METODE

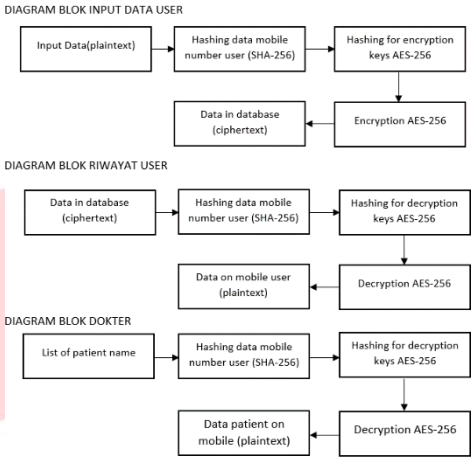
Alat ini akan berupa rompi yang disimulasikan dan berisi sensor seperti pendeteksi detak jantung, suhu badan, dan lain-lain. Sistem akan terhubung dengan dokter secara *realtime*, tujuan sistem ini dibuat agar *user* yang akan menggunakan alat dapat melakukan *checkup* mandiri dirumah tanpa perlu pergi ke pusat kesehatan. Untuk menjaga data pasien, penulis akan meningkatkan sistem keamanan alat dengan cara merancang autentikasi OTP dan melakukan enkripsi data yang telah di *input* kedalam *database*. Alur proses OTP dapat dilihat dalam *flowchart* di bawah.



Gambar 3.1 Desain Sistem OTP

Pada gambar 3.1 autentikasi dilakukan sebelum klien akan menggunakan alat. Dimana sebelumnya klien telah mendaftarkan akun sendiri. Selanjutnya setelah akun terdaftar, klien akan melakukan login dan otomatis akan menerima kode OTP. Kode yang diterima akan dimasukkan oleh klien untuk di verifikasi oleh sistem. Jika kode yang dimasukan ter verfikasi oleh sistem, maka *user* akan dapat masuk dan melakukan input data, jika kode yang dimasukan salah dan tolak oleh sistem, maka *user* harus melakukan input ulang kode OTP.

Pada proses selanjutnya adalah *embbeded key symmetric*, proses ini juga menggunakan *hashing* dan enkripsi SHA-256 dan AES-256. Berikut diagram blok *embbeded key symmetric.*



Gambar 3.2 Diagram Blok *Embbeded Key Symmetric*

Pada gambar 3.2 diagram blok dari input data *user* yaitu data *plaintext* yang akan masuk kedalam *database*, data tersebut akan diubah menjadi *ciphertext* dengan melewati proses *hashing*, data ponsel *user* akan digunakan sebagai kunci untuk enkripsi, lalu pada blok diagram riwayat *user* data dalam database di dekripsi kembali untuk menghasilkan data berupa *plaintext* yang akan ditampilkan pada *mobile application.* Selanjutnya blok diagram dokter dimulai dari list nama *user* yang telah melakukan input data, data ini juga akan di dekripsi kembali untuk menghasilkan data *plaintext* pada sisi dokter.

a. Input

Pada tahap ini menjelaskan perangkat yang digunakan untuk merancang sistem. Perangkat tersebut terdiri dari perangkat lunak dan perangkat keras. Untuk perangkat lunak seperti : *Windows 10 64-bit, Android Studio. React Native, PHP, XAMPP, Visual Studio Code.* Dan untuk perangkat keras seperti : *Asus ROG GL553Vd(intel core i7, Ram: 8Gb), Samsung Galaxy S9, Shield Wi-fi.*

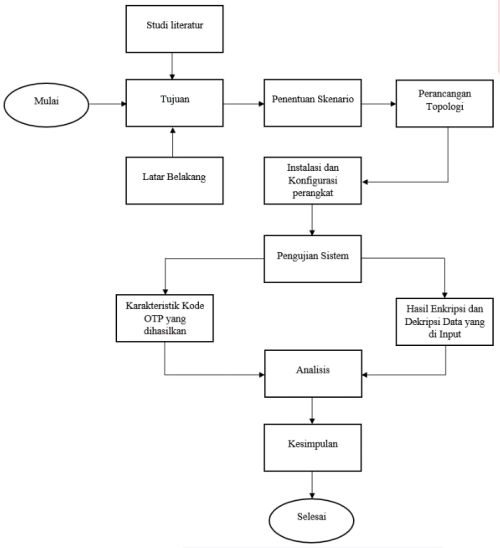
b. Proses

Tahapan proses perancangan sistem keamanan autentikasi OTP dan Kunci Simetris AES-256 dan SHA-256 pada *wearable vest*

c. Flowchart Alur Sistem

Skenario alur penelitian penulis mengacu pada studi literatur, latar belakang dan tujuan

yang menjadi awal untuk perancangan sistem ini. Penulis juga melakukan perancangan topologi jaringan, perancangan skema sistem, dimana tahapan ini akan membangun perancangan sebuah sistem yang dimulai dari analisis kebutuhan, pengumpulan data, desain, hingga pengujian sistem. Selanjutnya akan dilakukan beberapa pengujian yang berupa simulasi untuk mengetahui apakah sistem yang telah dirancang berjalan dengan baik. Penulis juga melakukan pengujian untuk mendapatkan hasil yang akan digunakan untuk menjawab skenario yang telah dirancang dan dapat memberikan kesimpulan tentang sistem yang telah dibuat.



Gambar 3.3 Flowchart Alur Sistem

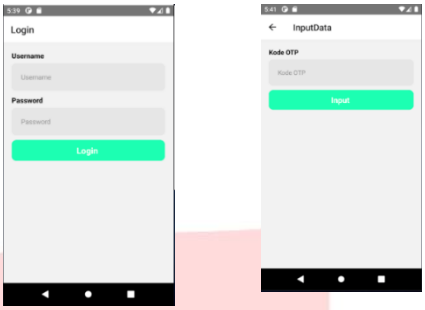
IV. HASIL DAN PEMBAHASAN

a. Tujuan Analisis dan Hasil

Tahapan akhir dari pembuatan sistem ini adalah meningkatkan sistem keamanan alat yang telah ada dalam bentuk simulasi dengan melakukan pengujian pada sistem agar dapat mengetahui apakah sistem yang telah dirancang dapat bekerja dan berfungsi dengan baik atau tidak. Pengujian dan analisis terhadap sistem bertujuan juga untuk mengetahui cara kerja dari komponen secara keseluruhan dari setiap komponen *input*, proses, dan komponen *output*. Sehingga dari hasil pengujian dan pengukuran dapat dilakukan evaluasi dan perbaikan terhadap sistem agar memperoleh hasil yang maksimal.

b. Hasil

Sistem ini berupa simulasi *mobile appliaction* kode OTP yang diterima akan ditampilkan pada *mobile* sistem yang telah rancang. Pada sisi *mobile* terdapat beberapa fitur seperti, fitur *login, input* kode OTP yang diterima. Berikut *display mobile application* yang telah dibuat.



Gambar 4.1 Tampilan *mobile* yang dirancang

c. Hasil Pengujian Sistem

Pada penelitian ini dihasilkan kunci yang didapatkan dari proses *hashing* nomor ponsel pengguna, proses ini menggunakan metode SHA-256. Hasil dari proses ini akan digunakan sebagai kunci untuk enkripsi dengan AES-256. Berikut hasil *hashing* nomor ponsel

d. Hasil Pengujian Enkripsi

Setelah kunci didapatkan dengan metode SHA-256, selanjutnya adalah hasil dari proses enkripsi dengan metode AES-256. *Username* dan nomor ponsel akan menjadi *plaintext*, dan hasil *hashing* sebelumnya akan menjadi kunci. Berikut hasil enkripsi dengan metode AES-256.

	Username	Nomor Ponsel	OTP
1	*User*1	082212345678	qwqPqf
2	*User*2	085320209011	AIEw8v
3	*User*3	089678129971	3RBppt
4	*User*4	088977631183	0Tim4t
5	*User*5	081544340323	5q+fDN

e. Hasil *One Time Password* (OTP)

Berikut hasil OTP yang ditampilkan pada *mobile application.* mendapatkan kode OTP yang berbeda. Kode ini didapatkan setelah hasil *ciphertext* di acak dengan metode *Linear Congruential Generator* sebanyak 6 kali.

f. *Response Time pada Localhost*

Pada sistem ini dilakukan pengujian *response time*, tujuan dari pengujian ini adalah untuk mengetahui seberapa lama waktu yang diperlukan server untuk mengirimkan kode OTP kepada pengguna. Percobaan ini dilakukan 10

kali pada *localhost* dan didapat kan hasil seperti pada tabel.

Tabel 4.4 Hasil percobaan *response time* pada *localhost*

No.	Percobaan Ke-	*Response Time* (ms)
1	1	220
2	2	317
3	3	157
4	4	171
5	5	159
6	6	161
7	7	172
8	8	149
9	9	166
10	10	280

Pada Tabel 4.4 *response time* terlama berada pada percobaan ke-2 yaitu 317ms dan waktu tercepat berada pada percobaan ke-8 yaitu 149ms, untuk rata-rata waktu respon pada *localhost* yaitu 195,2ms.

g. Response Time pada Hosting

Pada sistem ini juga dilakukan pengujian waktu respon pada *hosting.* Pengujian dilakukan sebanyak 10 kali dan didapatkan hasil seperti pada tabel.

Tabel 4.5 Hasil percobaan *response time* pada *hosting*

No.	Percobaan Ke-	*Response Time* (ms)
1	1	773
2	2	929
3	3	731
4	4	798
5	5	2.140
6	6	3.167
7	7	2.918
8	8	953
9	9	1.932
10	10	2.134

Waktu terlama pada hosting berada pada percobaan ke-6 yaitu 3.167ms, dan waktu respon tercepat berada pada percobaan ke-3 yaitu 731ms, dan rata-rata untuk *response time* pada *hosting* yaitu 1.647,5ms. Hasil pada hosting sedikit lebih lama dengan *localhost* karena pengujian pada *hosting* tidak dilakukan dalam satu perangkat yang sama.

h. *Embbeded Key Symmetric*

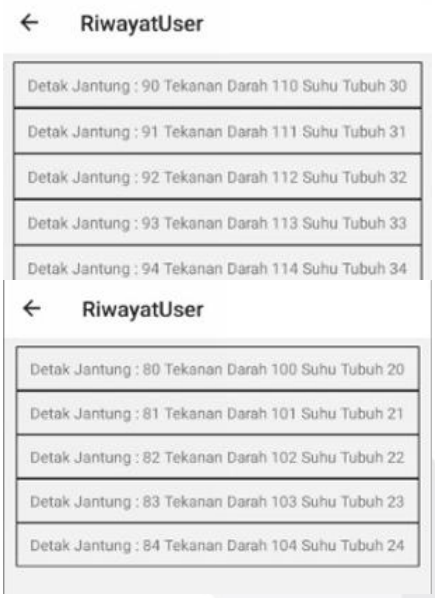
Pada sistem ini, pengujian ini dilakukan untuk melihat apakah data yang telah di *input* oleh klien dapat di enkripsi menjadi *ciphertext* dan tetap menjadi *plaintext* dalam *mobile application* pada sisi dokter maupun klien. Hasil enkripsi dapat dilihat pada tabel.

Tabel 4.6 Hasil Enkripsi Data yang di Input

User 1	Detak Jantung (bpm)	*Ciphertext*
1	90	cmPKaECS/RRZF+xquYRaKw==
2	91	IXarHs3EZ7mWAqLL8SEczQ==
3	92	OjQoTwsDwbNTLev21tx9lQ==
4	93	kMg6Aa+Zhbj8bEKnnYITeg==
5	94	+xMCToQLvbmRyZ/jX5cFA==
User 1	Tekanan Darah (mmHg)	*Ciphertext*
1	110	8BxZ06isNwvJiMJ61TBgsA==
2	111	ieYst8ugUlWSVZgnfA6h0w==
3	112	Pv7Q2Ous1rAe9X8PWv2hvA==
4	113	HELxHj679Vy+44wokxNsPg==
5	114	t1afErTkF/GBSEEQ2iPZOA==
User 1	Detak Jantung (bpm)	*Ciphertext*
1	30	NWPwJH/auSYr01J5qgZR6w==
2	31	fVC5Hac8mFRHXpqKzVKGIA==
3	32	RYblv0pfB0O51hRL3XzTZg==
4	33	BV4OrcrQx+mRm3PDkToCzA==
5	34	vZSA7gPlUUSFCcW91jtT4w==
User 2	Detak Jantung (bpm)	*Ciphertext*
1	80	/zgcThPv6gVBQEGD5EoV8g==
2	81	KBDcONehmSUQo6CDjwhL4A==
3	82	/aDM9g1KLPbmFHs7KXnPrw==
4	83	nun4InjeCoA1QKd5VOTG6w==
5	84	wo+2mL9qFrhd0bbXodXUNQ==
User 2	Tekanan Darah (mmHg)	*Ciphertext*
1	100	LfNG8O86Xn2LKUbj4jc7BQ==
2	101	2LaDjDvo+PXvZtZSRK0z8g==

3	102	rBfBiKJXAU6QsE9xMgDAPw==
4	103	G8Lt/NuSti1q8E+ST1J5ow==
5	104	TlNma//LJo+N431HMUwXA==
User 2	Detak Jantung (bpm)	*Ciphertext*
1	20	4eNhydZlVq47bLppLrYzOw==
2	21	uNIhH/2WRJjSZcUO0+wHRQ==
3	22	aorNaqIJINCC2oTWtQUS5g==
4	23	mEMHNEsTPDYTASebvprfkg==
5	24	QxTjjOZvlHxmYd4iNJgD6Q==

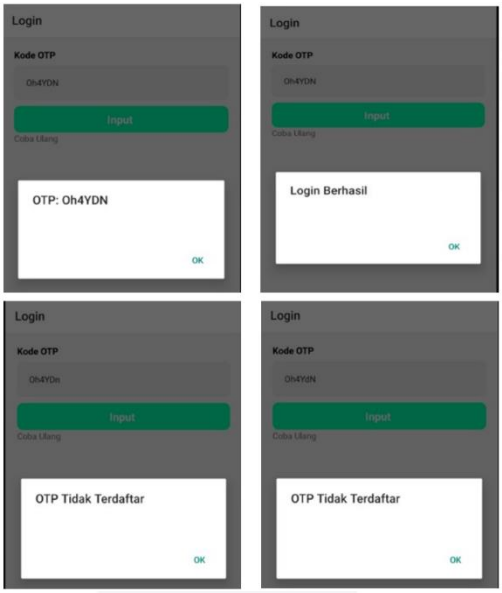
Pada tabel 4.6 dapat dilihat, data detak jantung, tekanan darah, dan juga suhu tubuh yang telah di input merupakan data *plaintext*, setelah melakukan proses *hashing* dan enkripsi maka didapatkan data pada server menjadi *ciphertext.* Untuk data pada sisi *mobile application* akan tetap berupa *plaintext.* Berikut lampiran data pada sisi *mobile application.*



Gambar 4.1 Data pada *Mobile Application*

i. Skenario Keamanan Sistem

Pada pengujian ini, dilakukan percobaan login dengan menggunakan OTP yang telah di dapatkan. Percobaan pertama dilakukan dengan memasukkan kode yang sesuai dengan OTP yang diterima. Lalu percobaan berikut nya dilakukan dengan merubah salah satu *font* dari kode OTP yang diterima, dan hasil yang didapatkan ditunjjukan pada gambar.



Hasil yang didapatkan adalah *user* tidak dapat *login* jika kode OTP yang diterima dirubah atau *user* tidak melakukan *input* sesuai dengan OTP yang diterima.

j. Kontrol Parameter

Pada percobaan ini dilakukan dalam bentuk simulasi, skenario yang dipilih adalah *response time* sistem dalam mengirim kode OTP jika sistem sedang diakses dengan jumlah *user* yang besar. Penulis menggunakan *tool* Loadium.io untuk mendapatkan hasil *response time.* Pengujian dilakukan sebanyak 2 kali. Percobaan pertama dengan jumlah *user* 150 dan percobaan kedua dengan jumlah *user* 250. Berikut hasil *response time* sistem dapat dilihat pada gambar.

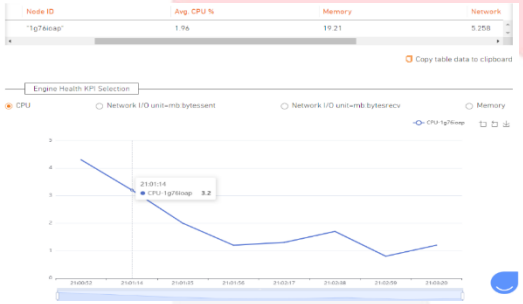
Gambar 4.3 *Response Time* sistem ketika diakses 150 *user*

Pada gambar diatas *response time* dan *latency* berada pada grafik yang sama. Pada

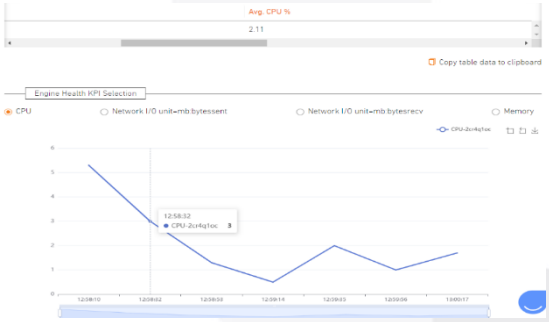
gambar 4.3 menunjukkan hasil untuk total 150 *user* yang melakukan *login* secara bersamaan. Untuk rata-rata *response time* pada jumlah *user* 150 berada pada 1.013,19ms, dan untuk jumlah *user* 250 berada pada 1.625ms. pengujian ini dilakukan untuk melihat seberapa besar pengaruh jumlah *user* ketika melakukan *login* bersamaan dengan *response time.* Dari hasil yang didapatkan dapat disimpulkan *response time* OTP relatif cepat dan stabil.

k. CPU *Usage*

Pada bagian ini, pengujian dilakukan seberapa besar CPU dari perangkat yang digunakan ketika menjalankan sistem yang sudah dirancang. Hasil dari pengujian dapat dilihat pada gambar berikut:



Gambar 4.5 CPU *Usage* 150 *User*



Gambar 4.6 CPU *Usage 250 User*

Pada skenario ini, hanya akan melihat hasil dari CPU yang terpakai ketika menjalakan sistem yang telah dirancang. Percobaan ini dilakukan dengan menggunakan *tool* Loadium.io, dimana hasil CPU ini didapatkan ketika sistem sedang diakses dengan jumlah *user* yang telah ditentukan. Pada gambar 4.5 jumlah *user* sebanyak 150 dan total rata-rata cpu yang terpakai adalah 1,96%. Pada gambar 4.6 jumlah *user* sebanyak 250 dan total rata-rata cpu yang terpakai adalah 2,11%. Untuk hasil rata-rapa cpu yang terpakai tergolong rendah

4.2.10 Avalanche Effect

Avalanche effect merupakan salah satu fungsi yang diinginkan pada algoritma enkripsi. *Avalanche effect* merupakan kemampuan dimana jika sedikti perubahan pada *plaintext* atau kunci akan menghasilkan perubahan yang cukup besar pada *ciphertext.* Efek ini memastikan bahwa penyerang tidak bisa dengan mudah menebak *plaintext* melalui analisis statistika. Algoritma yang tidak memenuhi metode ini akan mudah ditebak menggunakan analisis statistika.

Pada percobaan ini enkripsi menggunakan data awal yang telah dimodifikasi beberapa data tetapi masih menggunakan kunci yang sama. Kemudian hasil enkripsi dirubah menjadi bit dan *ciphertext* awal akan dibandingkan dengan *ciphertext* setelah data dimodifikasi untuk menghitung perbedaan jumlah bit. Hasil pengujian dapat dilihat pada tabel.

Tabel 4.7 Hasil *Avalanche Effect*

Plaintext A	Ciphertext	Plaintext B	Ciphertex
User1082123456678	gpFMRZllDlh8/fQYOqL4jZthoNZOJyPXqAwDe9jLjKs=	User108221234{9386}	709Jt71Bc4vz8WrVFk+9qVgHi5C6wN6Kg=
User2085320209011	E8b2dD/A/x0wvaYId1G043Elpv3JjGhNoY3QCEWPN58=	User208532020{4732}	qXTzk6RpXWwHKbr0mcYPkM6aPVkihxIF
User308967812997 1	c32by/V6TsLChMKpwMNpRCzBtuGM/hXISmd5trezyrE=	User308967812{1235}	rOns9PL0pVoymsieZIxAqdnVLw/BPkfWM
User4088977631183	sSRMpju6oKbieya4GljYoTmntGeV3qep8oToz0UCsSY=	User408897763{0372}	eVZFGnnlxxj15yUW9ENmPv9cH5vg4QDCr
User5081544340323	Dqjvp/fKyUMtFzNdCjayO9J5pDkf+4GdYyQbPGfuXBc=	User508154434{2391}	frTv0lRUElGDL7AP62ulZtFeyr3brgfLE=

Pada tabel 4.7 dapat dilihat empat indeks karakter yang diganti menghasilkan rata-rata perbedaan bit pada setiap kunci sebesar 144,8 bit dan untuk *avalanche effect* mendapatkan rata-rata sebesar 40,0468%

V. KESIMPULAN

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem yang telah dibuat sudah dapat meningkatkan sisi keamanan alat rompi kesehatan
2. Pada skenario OTP, telah berfungsi untuk melakukan verifikasi koneksi dengan alat rompi

3. OTP tidak akan aktif jika ada *font* yang berbeda
4. Kode OTP yang diterima untuk melakukan verifikasi dapat diterima dalam waktu relatif singkat
5. Data yang di input oleh *user* dalam *plaintext* dapat ter enkripsi menjadi *ciphertext*
6. Sistem yang dirancang relatif stabil karena *response time* yang dihasilkan ketika sistem diakses dalam jumlah besar relatif rendah
7. CPU yang terpakai ketika sistem dijalankan relatif rendah

REFERENSI

1. Kementrian Kesehatan RI. (2020). Pedoman Pencegahan dan Pengendalian Coronavirus Disease. Direktorat Jenderal Pencegahan dan Pengendalian Penyakit, 1-136
2. Buana, D. R. (2020). Analisis Perilaku Masyarakat Indonesia dalam Menghadapi Pandemi Virus Corona (Covid-19) dan Kiat Menjaga Kesejahteraan Jiwa. National Research Tomsk State University, Universitas Mercu Buana
3. T. Yaqoob, H. Abbas and M. Atiquzzaman, "Security Vulnerabilities, Attacks, Countermeasures, and Regulations of Networked Medical Devices—A Review," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3723-3768, Fourthquarter 2019, doi: 10.1109/COMST.2019.2914094.
4. Istihadi Ali. 2004. *Penerapan Sistem Autentikasi One Time Password Untuk ProsesLogin*. http://www.cert.or.id/~budi/courses/ec7010/dikmenjur/ali- report.doc
5. H. Zodpe and A. Sapkal, "An Efficient AES Implementation using FPGA with Enhanced Security Features," Journal of King Saud University - Engineering Sciences, 2018.
6. Imamah. (2018). One Time *Password* (OTP) Based on Advanced Encrypted Standard (AES) and Linear Congruential Generator(LCG). 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS). doi:10.1109/eeccis.2018.8692931
7. W. J. Long and W. Lin, "An authentication protocol for *wearable* medical devices," 2017 13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT), Stony Brook, NY, 2017, pp. 1- 5, doi: 10.1109/CEWIT.2017.8263140.
8. K I Santoso, "Dua Faktor Pengamanan Login Web Menggunakan Otentikasi One Time *Password* Dengan Hash SHA", Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013),pp. 204-210, 2013
9. Murti Setia D.2009.Autentikasi *User* pada Aplikasi berbasis web
10. Daor, Joa & Daemen, Joan & Rijmen, Vincent (1999). AES proposal : Rijndael
11. I. Imamah, A. Djunaidy and M. Husni, "Penerapan AES untuk Otentikasi Akses Cloud Computing," Jurnal Ilmiah Simantec, vol. 4, pp. 27-34, 2014
12. A. E. T. Tchendjeu, . R. Tchitnga and H. B. Fotsin, "FPGA Implementation of Linear Congruential Generator Based on Block Reduction Technique," Journal of Circuits, Systems and Computers, vol. 27, no. 10, 2018
13. Y. Huang, Z. Huang, H. Zhao and X. Lai, "A new One-time *Password* Method," in IERI Procedia, 2013
14. Jagannathan, S., & Sorini, A. (2016). Self-authentication in medical device *software*: An approach to include cybersecurity in legacy medical devices. 2016 IEEE Symposium on Product Compliance Engineering
15. Long, W. J., & Lin, W. (2017). *An authentication protocol for wearable medical devices. 2017 13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWI).*
16. Ari Dwi Ananto, 2014, Desain dan Implementasi Aplikasi SMS (Short Message Service) pada Android Menggunakan Algoritma AES, Universitas Telkom, Bandung.