CyberSecurity dan Forensik Digital e-ISSN: 2615-8442
Vol. 8, No. 1, Mei 2025, hlm. 1-9

Analisis dan Implementasi Algoritma Bcrypt dengan Affine Cipher untuk Pengamanan Password pada Aplikasi Web

Nugroho Dwi Aji[1], Tomi Tri Sujaka[2], Husain[3], Ondi Asroni[4], Kurniadin Abd. Latif[5]

[1,2,3,4,5]Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Bumigora

Email: [1]2101020042@universitasbumigora.ac.id, [2]tomi_tri@universitasbumigora.ac.id, [3]husain@universitasbumigora.ac.id, [4]ondi@universitasbumigora.ac.id , [5]kurniadin@universitasbumigora.ac.id

Abstrak

Teknologi informasi saat ini semakin berkembang dengan berkembangnya teknologi tersebut membuat segala kalangan dapat mengakses aplikasi web, hal tersebut menjadikan keamanan pada aplikasi web harus di perhatikan. Salah satu teknik yang dapat digunakan sebagai pengamanan pada aplikasi web yaitu menerapkan algoritma kriptografi pada *password*. Penelitian ini bertujuan untuk menganalisis dan mengimplementasi algoritma kriptografi *bcrypt* dengan *affine cipher* yang akan di modifikasi dengan menggunakan algoritma *ROT13* untuk pengamanan *password* pada aplikasi web. Penelitian dilakukan dengan menggunakan metode pengembangan sistem yaitu *Secure Sistem Development Life Cycle* (SSDLC) yang meliputi *Reqirements analysis*, Desain, *Development*, dan *Testing* menggunakan *bruteforce* dan *sniffing*. Hasil penelitian ini menunjukkan bahwa analisis dan implementasi algoritma *bcrypt* dengan *affine cipher* untuk pengamanan *password* pada aplikasi web sesuai rancangan. Uji coba menunjukkan bahwa dengan menerapkan algoritma *bcrypt* dan *affine cipher* yang telah dimodifikasi menggunakan algoritma *ROT13* akan memperlambat dari serangan *bruteforce*. Sedangkan implementasi pada pengiriman kredensial dari *frontend* menuju *backend* menunjukkan kredensial tidak dapat dilakukan serangan *sniffing* menggunakan *wireshark*. Kesimpulan dari penelitian ini bahwa menerapkan algoritma kriptografi *bcrypt* dan *affine cipher* yang telah dimodifikasi menggunakan algoritma *ROT13* akan menambahkan lapisan keamanan pada sistem yang dapat menghambat serangan *bruteforce* dan menghentikan serangan *sniffing*. Penelitian ini juga menyarankan penggunaan kriptografi *bcrypt* dan *affine cipher* yang telah dimodifikasi menggunakan algoritma *ROT13* sebagai lapisan pengamanan pada sistem.

Kata kunci: *Affine cipher*, *Bcrypt*, Kriptografi, *Password*, Pengamanan, *ROT13*.

Analysis and Implementation Of Bcrypt Algorithm with Affine Cipher for Password Security in Web Applications

Abstract

Information technology is currently growing with the development of this technology making all circles able to access web applications, this makes security in web applications must be considered. One technique that can be used as security in web applications is applying cryptographic algorithms to passwords. This research aims to analyze and implement the bcrypt cryptographic algorithm with affine cipher which will be modified using the ROT13 algorithm for password security in web applications. The research was conducted using the system development method, namely Secure System Development Life Cycle (SSDLC) which includes Reqirements analysis, Design, Development, and Testing using brute-force and sniffing. The results of this research show that the analysis and implementation of the bcrypt algorithm with affine cipher for password security in web applications are as designed. The test shows that by applying bcrypt and affine cipher algorithms that have been modified using the ROT13 algorithm will slow down brute-force attacks. While the implementation on sending credentials from the front-end to the back-end shows that credentials cannot be sniffed using wireshark. The conclusion of this research is that applying bcrypt and affine cipher cryptographic algorithms that have been modified using the ROT13 algorithm will add a layer of security to the system that can inhibit brute-force attacks and stop sniffing attacks. This research also suggests the use of bcrypt and affine cipher cryptography that has been modified using the ROT13 algorithm as a security layer on the system.

Keywords*: Affine cipher, Bcrypt, Cryptography, Password, Security, ROT13.*

1. PENDAHULUAN

Teknologi informasi saat ini semakin berkembang, dengan berkembangnya teknologi informasi tersebut semua kalangan dapat mengakses aplikasi web. Hal ini menjadikan keamanan data pada aplikasi web menjadi sangat penting. Bahkan semakin berkembangnya teknologi akan membuat serangan pembobolan data oleh *cracker* akan semakin beragam (Laksana & Mulyani, 2024). Untuk mencegah hal tersebut terjadi, hal yang paling sederhana yang dapat kita lakukan adalah menggunakan *password* atau kata sandi yang kuat untuk mengamankan data penting yang ada di dalam aplikasi web yang kita buka dari tangan yang tidak bertanggung jawab (Gunawan, 2021). Walau begitu pengamanan ini tidak sepenuhnya menjamin keamanan data yang ada pada aplikasi web dikarenakan sudah banyak teknik untuk menembus keamanan ini.

Beberapa penelitian sebelumnya yang terkait dengan penelitian yang penulis lakukan. Penelitian sebelumnya telah mengeksplorasi berbagai pendekatan kriptografi untuk meningkatkan keamanan *password*. (Batubara et al., 2021) menganalisis performa Bcrypt dalam menghadapi serangan *brute-force*, tetapi hanya berbasis studi literatur tanpa implementasi praktis. Penelitian serupa oleh (Febrian et al., 2023) mengimplementasikan Bcrypt pada generator hashing berbasis web menggunakan Laravel, namun masih terbatas pada satu lapisan enkripsi tanpa kombinasi algoritma lain. Sementara itu, (Nur, 2023) memperkenalkan penggunaan Bcrypt dengan OTP, tetapi fokusnya pada autentikasi dua faktor, bukan penguatan enkripsi *password*. Beberapa peneliti juga mencoba mengombinasikan Bcrypt dengan algoritma lain, namun dengan tujuan yang berbeda. (Divva et al., 2022) menggabungkan Bcrypt dan AES, tetapi penerapannya hanya untuk pengamanan file, bukan proteksi *password* di aplikasi web. Di sisi lain, (Kurniasih et al., 2023) memodifikasi Affine Cipher dengan teknik steganografi (Least Significant Bit-2) untuk penyembunyian pesan dalam gambar, tetapi tidak mengintegrasikannya dengan teknik hashing seperti Bcrypt.

Meskipun berbagai penelitian telah dilakukan, masih terdapat beberapa celah. Pertama, belum ada studi yang menggabungkan *Bcrypt* dengan *Affine Cipher* yang dimodifikasi menggunakan *ROT13* untuk pengamanan *password* di aplikasi web. Kedua, sebagian besar penelitian terdahulu hanya menguji resistensi terhadap satu jenis serangan (seperti *brute-force*), tanpa mempertimbangkan ancaman lain seperti sniffing. Ketiga, metodologi yang digunakan seringkali terbatas pada literature review atau implementasi dasar tanpa pendekatan *Secure* SDLC (*Software Development Life Cycle*) yang komprehensif. Penelitian ini bertujuan untuk mengisi gap tersebut dengan mengusulkan solusi hybrid *cryptography* yang mengombinasikan *Bcrypt* (untuk hashing) dan *Affine Cipher-ROT13* (untuk enkripsi). Pendekatan ini tidak hanya memperkuat proteksi terhadap serangan *brute-force*, tetapi juga memastikan kerahasiaan data selama transmisi dengan mengacak kredensial sehingga tidak terbaca saat terjadi sniffing. Selain itu, penelitian ini menggunakan metodologi *Secure* SDLC untuk memastikan keamanan diterapkan sejak tahap desain hingga implementasi. Hasil pengujian menunjukkan bahwa kombinasi algoritma ini secara signifikan meningkatkan keamanan sistem dibandingkan penggunaan algoritma tunggal seperti yang dilakukan dalam penelitian-penelitian sebelumnya. Dengan demikian, penelitian ini memberikan kontribusi baru dalam pengembangan sistem keamanan password yang lebih robust dan tahan terhadap *multi-vektor* serangan.

Berdasarkan pemaparan di atas penulis tertarik melakukan penelitian analisis dan implementasi *bcrypt* dan *affine cipher* untuk pengamanan password pada aplikasi web. Dengan adanya kelemahan dari algoritma *affine cipher* tersebut, penulis ingin menggabungkan enkripsi sederhana affine cipher yang telah dimodifikasi menggunakan ROT13 dengan algoritma hashing *bcrypt*. Hasil dari penelitian ini merupakan contoh dari bagaimana *bcrypt* dan *affine cipher* itu dapat menjadi pengamanan hybrid guna meningkatkan keamanan dari sebuah *password* dalam aplikasi web.

Berdasarkan latar belakang yang telah dipaparkan, penulis merumuskan rumusan masalah yaitu Bagaimana efektivitas kombinasi algoritma bcrypt dan Affine Cipher yang dimodifikasi dengan *ROT13* dalam meningkatkan keamanan password terhadap serangan *brute-force* dan sniffing, serta bagaimana implementasinya dalam aplikasi web? Dengan tujuan dari penelitian ini adalah untuk menggabungkan dua jenis algoritma kriptografi yakni algoritma *encrypt affine cipher* yang telah dimodifikasi menggunakan algoritma *ROT13* dan algoritma hashing *bcrypt*. Sehingga gabungan kedua algoritma tersebut di harapkan menambah lapisan keamanan dari password pada aplikasi web.

Pengujian *brute-force* dilakukan dengan menggunakan skrip Python yang mencoba kombinasi email dan password dari daftar kata (*wordlist*) berisi 20 entri. Waktu yang dibutuhkan untuk menemukan password yang benar diukur dalam detik. Pengujian *sniffing* dilakukan dengan Wireshark untuk memantau paket data yang dikirim selama proses login. Keberhasilan pengujian *brute-force* diukur berdasarkan peningkatan waktu yang dibutuhkan untuk menemukan password yang benar. Sedangkan keberhasilan pengujian *sniffing* diukur berdasarkan ketidakmampuan alat untuk menangkap *password* dalam bentuk plaintext.

2. KAJIAN PUSTAKA

Penelitian yang berjudul "Optimasi Keamanan Web Server Terhadap Serangan *Brute-Force* Menggunakan *Penetration Testing*" menyebutkan Badan Siber dan Sandi Negara mengeluarkan data pada tahun 2020 yang mengatakan bahwa serangan web atau cyber attack mendeteksi jumlah serangan naik 4 kali lipat jumlah serangan dari tahun sebelumnya yaitu 2019 dengan jumlah serangan hanya 98 juta (Fachri, 2023). Sedangkan berdasarkan laporan Microsoft Entra, platform manajemen identitas digital, terjadi rata-rata 600 juta serangan siber berbasis identitas per hari antara Juli 2023 hingga Juni 2024. Data menunjukkan bahwa hampir seluruh serangan (99%) secara spesifik menargetkan kredensial kata sandi pengguna. Selama periode satu tahun tersebut, sistem keamanan Microsoft berhasil mencegah sekitar 7.000 upaya peretasan kata sandi setiap detiknya - jumlah yang jika dikalkulasi setara dengan lebih dari 220 triliun serangan dalam setahun. Serangan – serangan tersebut akan menjadi ancaman yang serius bagi kerahasiaan dan keamanan data pengguna (Fa'izi, 2024).

Serangan yang paling sering digunakan dalam keamanan password adalah *brute-force* dengan *dictionary attack* dan *sniffing attack*. Teknik *brute-force* merupakan metode peretasan dengan menggunakan trial and error untuk mendapatkan akses kedalam akun, kredensial *login*, ataupun kunci enkripsi. Dalam serangan *brute-force* peretas akan bekerja dengan semua kemungkinan kombinasi dengan harapan dapat menemukan kombinasi yang benar (Revenkov et al., 2021) dan teknik *dictionary attack* merupakan serangan pencarian kata sandi dengan mencoba semua kata sandi yang ada dalam kamus atau daftar kata yang sudah ada. Kamus ini bisa berupa kata-kata umum, kombinasi angka, huruf, dan simbol, serta kata-kata yang mungkin terkait dengan pengguna atau organisasi tertentu. Serangan ini bertujuan untuk menemukan kata sandi yang tepat atau mendekati kata sandi yang digunakan oleh target (Sunandar Informatika et al., 2024). Sedangkan teknik *sniffing attack* merupakan jenis serangan siber di mana penyerang memanfaatkan perangkat lunak atau perangkat keras untuk menyadap lalu lintas data di jaringan(Luthfansa & Rosiani, 2021). Tujuannya adalah mencuri informasi sensitif seperti kata sandi, data login, atau informasi kartu kredit. Penyerang biasanya menggunakan alat seperti Wireshark atau tcpdump untuk menangkap dan menganalisis paket data yang dikirim melalui jaringan, terutama pada jaringan yang tidak terenkripsi(Naufal Ayman & Nurhadiyanto, 2024). Untuk mengurangi ancaman dari serangan tersebut ada beberapa teknik pengamanan *password*, salah satunya adalah teknik kriptografi.

Teknik kriptografi seperti enkripsi simetris dan asimetris, hashing, dan tanda tangan digital sangat penting untuk melindungi data dari ancaman ini. Implementasi kriptografi memerlukan penilaian risiko yang detail, pemilihan algoritma yang tepat, manajemen kunci yang aman, serta pengujian dan pemeliharaan rutin (Azhari et al., 2022). Dengan strategi perlindungan yang efektif dan pembaruan keamanan terus-menerus, individu dan organisasi dapat mengurangi risiko serangan dan melindungi data mereka dari berbagai ancaman (Ramalinda & Rachmat Raharja, 2024). Dalam penelitian ini digunakan *affine cipher* yang telah dimodifikasi dengan menggunakan algoritma ROT13 digabungkan dengan algoritma *bcrypt* untuk mengamankan data password pengguna.

Affine cipher adalah salah satu teknik kriptografi enkripsi yang paling sederhana yang di mana plainteks akan dikalikan dengan sebuah nilai dan ditambahkan dengan sebuah pergeseran. Algoritma ini akan digunakan untuk proses enkripsi dan dekripsi (Kurniasih et al., 2023). *Affine Cipher* memiliki kelemahan yang cukup besar, yaitu mudah ditembus dengan metode analisis frekuensi. Metode ini mengandalkan fakta bahwa dalam bahasa Inggris, huruf E adalah huruf yang paling sering muncul, sehingga jika kita menganalisis frekuensi kemunculan huruf dalam sebuah pesan yang terenkripsi dengan *affine cipher*, maka kita dapat menebak geseran yang digunakan dan membuka sandi dengan mudah(Siregar, 2023).

ROT13 merupakan algoritma enkripsi varian sederhana dari sandi Caesar, yang setiap huruf dalam teks akan dirubah dengan huruf yang terletak 13 posisi lebih jauh dalam alfabet. Contohnya, huruf A akan menjadi N, B menjadi O, dan seterusnya hingga Z yang akan menjadi M. Proses tersebut bersifat simetris sehingga untuk mendekripsi teks yang di enkripsi dengan *ROT13* akan menggunakan algoritma yang sama kembali (Abi Assyarif et al., 2023)

Bcrypt adalah fungsi hash yang dibuat oleh *Blowfish* dan *Crypt* yang merupakan fungsi hash utama pada pemrograman password di UNIX. *Blowfish* melakukan penggabungan dengan fungsi hash Crypt agar pemecahan password memerlukan waktu yang lama(Indrawan & Sa'uda, 2024). Pada *bcrypt* memiliki tahap inisialisasi kunci bernama "*eksblowfish*", yang memiliki makna expensive key schedule *blowfish*. *Bcrypt* adalah hashing kata sandi dengan jumlah ilustrasi yang lebih banyak untuk membuatnya lebih kuat melawan serangan pencarian *bruteforce* serta meningkatkan daya komputasi dengan menggabungkan garam dari melindungi dari *rainbow table attack* (Batubara et al., 2021).

3. METODOLOGI

Dalam penelitian ini, penulis menggunakan metode penelitian studi literatur dan metode pengembangan sistem *Secure software development life cycle* (SSDLC). Langkah SSDLC yang digunakan mencakup *requirement analysis*, desain, *development* dan *testing*.

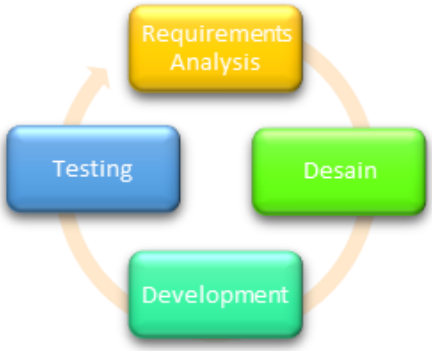
Penelitian ini memiliki beberapa keterbatasan, seperti pengujian yang hanya dilakukan pada lingkungan lokal dan belum mencakup serangan lain seperti SQL *injection* atau XSS. Selain itu, ukuran *wordlist* yang digunakan dalam pengujian *brute-force* masih terbatas.

3.1. Studi Literatur

Pada tahap studi literatur penulis melakukan analisis untuk mendukung penelitian ini mengenai algoritma *bcrypt*, *affine cipher* serta ROT13 untuk pengamanan *password* pada aplikasi web dengan melakukan pengumpulan data pada berbagai sumber terpercaya.

3.2. Metode Pengembangan Sistem

Pengembangan sistem yang digunakan adalah *Secure software development life cycle* (SSDL) dengan langkah-langkah yang digunakan sebagai berikut:



Gambar 1. Alur SSDLC yang digunakan

Pada penelitian ini dibuat sistem pengamanan *password* menggunakan dua algoritma kriptografi yakni algoritma encripsi *affine cipher* yang telah dimodifikasi menggunakan ROT13 dan algoritma hashing *bcrypt*. Secara garis besar tahapan penelitian menggunakan langkah metode *Secure Software Development Life Cycle* (SSDLC) dapat dijelaskan sebagai berikut:

3.2.1. Requirement Analysis

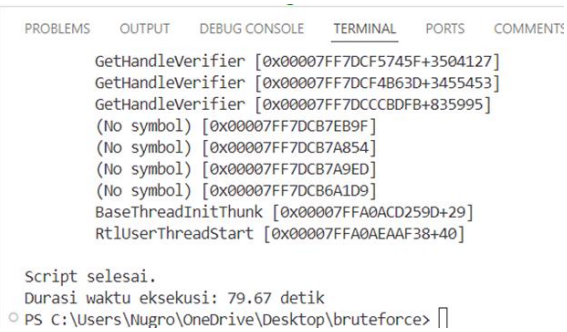
Pada tahap ini penulis melakukan analisis kerentanan dan kebutuhan untuk membuat contoh sistem keamanan menggunakan gabungan algoritma kriptografi *bcrypt* dan *affine cipher* yang dimodifikasi menggunakan ROT13 dilakukan untuk mengetahui apa saja yang dibutuhkan pada sistem.

1. *Risk Assesment*

Untuk menilai kerentanan pada penelitian ini digunakan metode *penetration bruteforce* testing dan *sniffing* testing berikut adalah *risk assessment* terhadap web tanpa menggunakan gabungan algoritma kriptografi *bcrypt* dan *affine cipher* yang dimodifikasi menggunakan ROT13.

2. *Bruteforce*

Langkah pertama yang dilakukan *cracker* untuk melakukan serangan *bruteforce* adalah menganalisis dari target. Analisis dilakukan untuk melihat struktur dari halaman web yang akan diserang dengan menekan tombol f12 dan memilih menu elemen untuk melihat struktur web yang dipakai. Selanjutnya yang dibutuhkan ialah hal yang terjadi jika mengirimkan kredensial yang salah sebagai parameter untuk *script bruteforce* memulai percobaan selanjutnya. Setelah semua parameter yang dibutuhkan lengkap maka *script* dapat di buat dengan menggunakan *python*. Untuk *dectionary* yang digunakan saat melakukan uji coba serangan *bruteforce* menggunakan file *excel* dengan beberapa email dan *password* akan digunakan sebagai word *list* untuk mencoba melakukan *bruteforce attack*. Jika berhasil akan langsung masuk ke dalam sistem dan pada log *script* akan memunculkan waktu berjalanya *script* seperti gambar

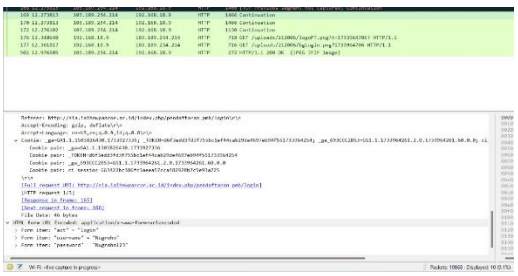


Gambar 2. Hasil *bruteforce* test tanpa kriptografi

Terlihat pada terminal yang terdapat pada gambar 3.5 untuk menemukan email dan *password* yang benar dengan target web yang tidak mengimplementasikan teknik kriptografi dalam hal penelitian ini yakni *bcrypt* dan *affine cipher* yang telah di modifikasi dengan algoritma sederhana ROT13 membutuhkan waktu 79.67 detik.

3. *Sniffing*

Hal pertama yang dilakukan oleh seorang *cracker* untuk melakukan *sniffing* adalah menjalankan *tools sniffing* (wireshark). Dengan menggunakan *username* Nugroho Dwi Aji dan menggunakan *password* Nugroho123. Setelah mengisi *username* dan *password* tekan tombol *login* untuk mengeksekusi proses *login* dengan tetap menjalankan *tools wireshark*.



Gambar 3. Hasil sniffing tanpa kriptografi

Pada gambar 3 dapat dilihat wireshark dapat menangkap *plainteks* yang dimasukkan oleh

pengguna untuk melakukan proses *login* dikarenakan pada saat proses *login* sistem tidak dilengkapi dengan pengamanan data agar terhindar dari kejahatan *sniffing*.

4. Kebutuhan perangkat lunak

Kebutuhan perangkat lunak yang digunakan dalam penelitian ini untuk membangun sistem adalah:

- Teks editor (Visual Studio Code)
- Sistem Operasi (Windows 11)
- NodeJS
- Bahasa pemrograman (Java Script)
- Framework (ExpressJS dan React + Vite)
- Database (MySql)
- Server database (Laragon)
- Bahasa pemrograman penetration testing (Python)
- Bcrypt

5. Kebutuhan perangkat keras

Kebutuhan perangkat keras yang digunakan pada penelitian ini untuk membangun sistem adalah sebuah laptop yang memiliki spesifikasi berikut:

- Processor : AMD Ryzen 7
- RAM : 8 GB
- Penyimpanan : 2 x SSD NVME 512 GB

6. Kebutuhan sumber daya manusia

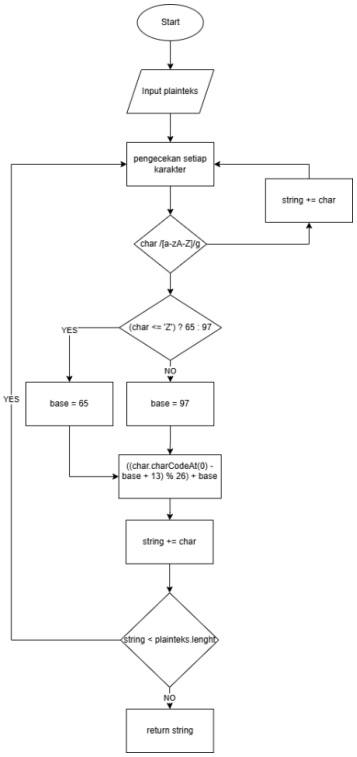
Perangkat lunak yang telah disebutkan di atas memerlukan sumber daya manusia yang mampu mengoperasikan sistem tersebut dengan baik dan benar.

3.2.2. Desain Penelitian

Tahap ini dilakukan *thread* modeling dan desain *review* sehingga dapat dihasilkan model proses sistem saat registrasi dan *login* sehingga dapat dilihat secara keseluruhan sebagai berikut:

1. Algoritma ROT13

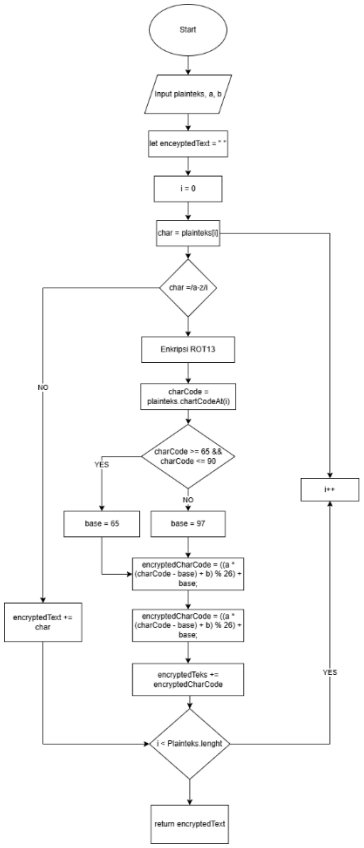
Algoritma ROT13 akan diimplementasikan pada algoritma enkripsi *affne cipher* sebagai pengaman tambahan. *Plainteks* yang *diinputkan* oleh *user* akan dikalikan 13 sehingga nilai *string* yang akan dikembalikan akan berubah dari nilai awal. Pada algoritma tersebut hanya mengalikan karakter *char*.



Gambar 4. Proses algoritma ROT13

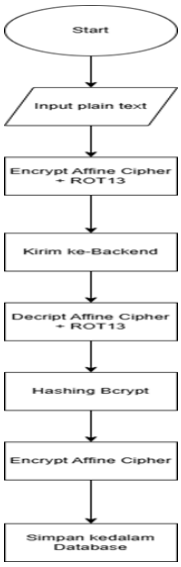
2. Algoritma Affine Cipher + ROT13

Alur dari proses algoritma *affine cipher* yang dimulai dari saat memasukkan dari *plaintext* dan kunci yang akan dipakai untuk mengenkripsi dari *plaintext* tersebut.



Gambar 5 Proses algoritma affine cipher + ROT13

3. Proses Pengamanan Saat Registrasi

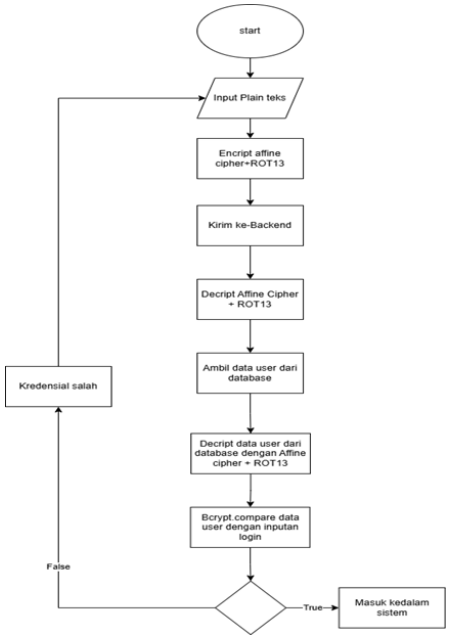


Gambar 6. Proses Pengamanan Saat Registrasi

Flowchart pada gambar 6 tersebut memperlihatkan alur registrasi ataupun enkripsi sistem secara keseluruhan. Dimulai dari penginputan *plainteks* dan akan di enkripsi terlebih dahulu menggunakan algoritma *Affine cipher* yang telah di modifikasi menggunakan ROT13 sebelum dikirimkan menuju *backend*. Setelah itu pada bagian *backend* akan mendekrip kredensial yang telah dikirimkan oleh *user* dan selanjutnya akan di-hashing menggunakan algoritma *bcrypt* dan di-encripsi menggunakan algoritma *Affine Cipher* yang telah dimodifikasi menggunakan ROT13 yang selanjutnya baru akan disimpan kedalam database.

4. Proses pengamanan saat login

Flowchart pada gamabr 7 tersebut memperlihatkan bagaimana alur login maupun decript pada sistem secara keseluruhan. Dimulai dari penginputan plainteks dan akan di-ekripsi terlebih dahulu menggunakan algoritma Affine cipher yang telah di modifikasi menggunakan ROT13 sebelum dikirimkan menuju back-end. Setelah itu pada backend akan diencript terlebih dahulu dan akan di bandingkan menggunakan fungsi yang ada pada bcrypt dengan kredensial yang terdapat didalam database yang telah di decript terlebih dahulu. Apabila menghasilkan false akan mengembalikan nilai kredensial salah dan jika bernilai true akan dapat memasuki sistem.



Gambar 7. Proses pengamanan saat login

3.2.3. *Development*

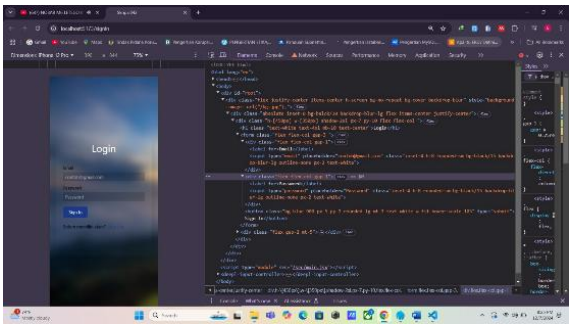
Pada tahap ini penulis melakukan pengkodean sesuai dengan desain dan *thread modeling* yang telah di rancang pada tahap sebelumnya, dengan mengimplementasikan algoritma *bcrypt* pada saat penyimpanan *password* kedalam database dan menggunakan fungsi *compare* saat ingin melakukan login. Untuk *affine cipher* yang telah dimodifikasi menggunakan ROT13, *plainteks* akan enkrip menggunakan algoritma ROT13 setelah itu baru akan di enkripsi menggunakan algoritma *affine cipher*.

4. PEMBAHASAN

4.1. Testing

4.1.1. Bruteforce Testing

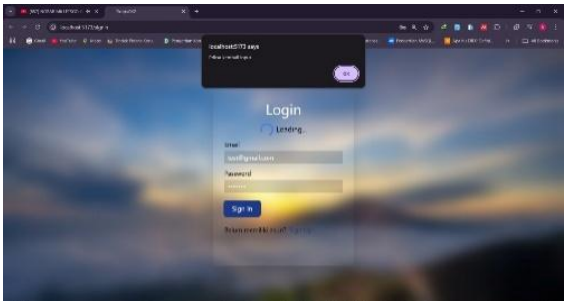
Langkah pertama yang dilakukan cracker untuk melakukan serangan *brute-force* adalah menganalisis dari target. Analisis dilakukan untuk melihat struktur dari halaman web yang akan diserang dengan menekan tombol f12 dan memilih menu elemen untuk melihat struktur web yang dipakai.



Gambar 8. Identifikasi atribut target

Terlihat Web memiliki dua input field untuk email dan *password* dengan email menggunakan placeholder Contoh@gmail.com dan untuk password

menggunakan *placeholder Password* dua parameter tersebut dapat dijadikan parameter untuk serangan *brute-force* serta memiliki tag button yang memiliki teks *Sign In* sebagai tombol untuk mengirim kredensial. Selanjutnya yang dibutuhkan ialah hal yang terjadi jika mengirimkan kredensial yang salah sebagai para meter untuk script *brute-force* memulai percobaan selanjutnya



Gambar 9. Identifikasi respon target

Terlihat saat kredensial salah muncul sebuah alert yang bertuliskan “Priksa kembali input” alert tersebut akan dijadikan parameter yang mengindikasikan bahwa kredensial salah. Setelah semua parameter yang dibutuhkan lengkap maka script dapat di buat dengan menggunakan python.

Script yang digunakan pada percobaan dengan kriptografi sama persis dengan *script* yang digunakan untuk percobaan pada sistem yang tanpa menggunakan kriptografi. File excel yang digunakan pun sama persis dengan yang digunakan untuk menguji coba sistem yang telah diterapkan kriptografi dalam hal ini adalah bcrypt dan affine cipher yang telah di modifikasi menggunakan algoritma sederhana *ROT13*.

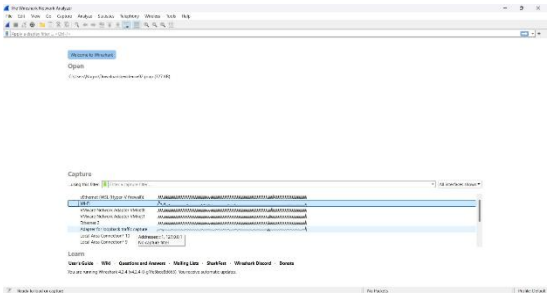
Adapun hasil dari penetration testing pada sistem yang telah dibangun dapat di lihat sebagai berikut

Gambar 10. Hasil uji coba *bruteforce*

Terlihat pada terminal yang terdapat pada gambar untuk menemukan email dan *password* yang benar dengan target web yang mengimplementasikan teknik kriptografi dalam hal penelitian ini yakni bcrypt dan affine cipher yang telah di modifikasi dengan algoritma sederhana *ROT13* membutuhkan waktu 100.57 detik.

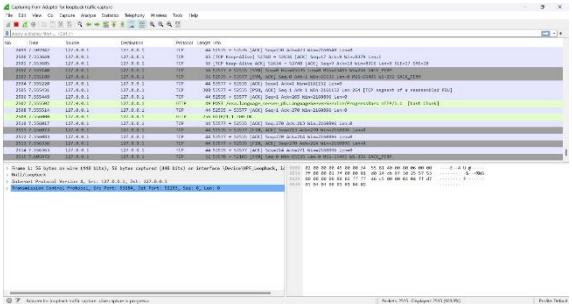
4.1.2. Sniffing Tessting

Langkah pertama yang dilakukan untuk melakukan penetration testing dengan teknik *sniffing* adalah membuka tools *sniffing*, pada penelitian kali ini penulis menggunakan wireshark sebagai *tools* untuk melakukan *sniffing*.



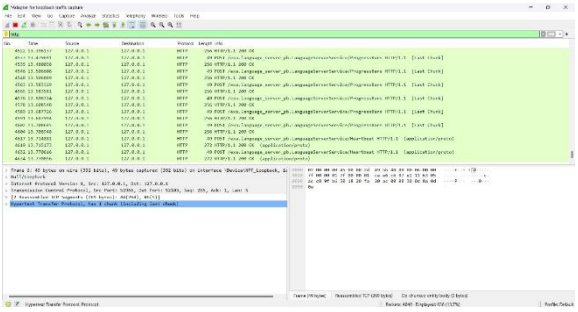
Gambar 11. Tampilan wireshark

Setelah wireshark berjalan terlihat ada beberapa jaringan yang aktif. Setelah wireshark terbuka pilih jaringan yang digunakan untuk membuka web, pada penelitian ini web yang di bangun berjalan pada localhost yang memiliki ip 127.0.0.1.



Gambar 12. Lalulintas jaringan

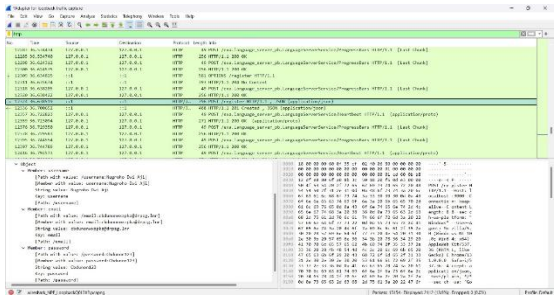
Setelah berada pada halaman jalur jaringan yang dimiliki dari koneksi yang akan di *sniffing*, selanjutnya melakukan filter protocol yang akan di sniffing. Dalam penelitian kali ini protocol yang akan disniffing adalah HTTP, masukkan kata kunci http pada kolom filter.



Gambar 13. Filterisasi protocol

Terlihat pada gambar yang tampil hanyalah yang menggunakan protocol HTTP. Setelah menjalankan wireshark dan melakukan filterisasi, lakukan proses *registrasi* dan proses *login* untuk melihat apakah kredensial yang akan dikirimkan menuju *back-end* dapat terlihat oleh *sniffer* atau tidak. Setelah melakukan login hentikan proses scan dengan menekan tombol kotak merah pada pojok kanan atas pada halaman wireshark dan cari yang memiliki

protocol http/json untuk melihat kredensial yang dikirimkan seperti terlihat.



Gambar 14. Hasil *sniffing attack*

Terlihat pada gambar Kolom info memperlihatkan bahwa kredensial dikirimkan menuju *back-end* menggunakan metode *post* dan menggunakan *protocol http* dengan *endpoint* /register dalam bentuk json. Untuk kredensial yang dibawa dapat dilihat pada kolom bawah halaman pojok samping kiri terlihat email dan *password* tidak dapat dilihat *plainteks* apa yang di-*inputkan* oleh pengguna melainkan *cracker* hanya dapat melihat *cipherteks* yang didapatkan dari *plainteks* yang telah di enkripsi.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, penerapan algoritma kriptografi hashing *bcrypt* dan enkripsi *affine cipher* yang dimodifikasi dengan ROT13 terbukti meningkatkan keamanan sistem. Uji coba *brute-force* menunjukkan bahwa kombinasi email dan *password* membutuhkan waktu 100,57 detik untuk dilacak, lebih lama dibandingkan sistem tanpa kriptografi (79,67 detik). Dalam skenario nyata, penggunaan daftar kata yang lebih besar akan semakin memperlambat proses peretasan, membuktikan bahwa lapisan kriptografi ini efektif mengurangi risiko serangan *brute-force*.

Pada uji coba *sniffing*, kredensial yang dienkripsi dengan *affine cipher-ROT13* hanya menampilkan *ciphertext*, sehingga tidak terbaca oleh penyerang. Sebaliknya, pengujian pada situs tanpa enkripsi memperlihatkan kredensial dalam bentuk teks jelas. Hasil ini membuktikan bahwa modifikasi algoritma enkripsi tersebut mampu melindungi data sensitif selama transmisi.

Dengan demikian, integrasi *bcrypt* dan *affine* cipher-ROT13 memberikan perlindungan ganda: memperlambat serangan *brute-force* dan mencegah kebocoran informasi melalui *sniffing*.

Kombinasi algoritma *bcrypt*, *Affine Cipher*, dan ROT13 dalam penelitian ini menawarkan beberapa keunggulan signifikan. Pertama, sistem ini memberikan perlindungan ganda melalui enkripsi berlapis. *Password* yang *diinput* pengguna diacak terlebih dahulu dengan ROT13 dan *Affine Cipher* sebelum di-hashing menggunakan *bcrypt*, sehingga serangan *brute-force* memerlukan waktu lebih lama untuk memecahkan kedua lapisan enkripsi tersebut. Selain itu, enkripsi pada tahap transmisi data terbukti efektif mencegah serangan *sniffing*, seperti yang ditunjukkan dalam pengujian menggunakan *Wireshark*, di mana kredensial hanya muncul sebagai *ciphertext*. Kedua, kombinasi ini tetap mempertahankan kompatibilitas dengan standar keamanan modern. *Bcrypt*, dengan penggunaan *salt* dan *cost factor*, telah dikenal *resisten* terhadap serangan *brute-force* dan *rainbow table*, sementara modifikasi ROT13 pada *Affine Cipher* membantu mengurangi kerentanan algoritma tersebut terhadap analisis frekuensi. Ketiga, implementasinya relatif sederhana dan tidak memerlukan sumber daya komputasi yang berat, sehingga cocok untuk aplikasi skala kecil hingga menengah.

Namun, pendekatan ini juga memiliki beberapa kelemahan yang perlu dipertimbangkan. Dari segi performa, proses enkripsi berlapis dapat menimbulkan *overhead* komputasi, terutama pada aplikasi dengan lalu lintas tinggi. Pengujian yang dilakukan hanya menggunakan *wordlist* kecil (20 entri), sehingga dampaknya terhadap sistem dengan beban besar belum terukur. Dari aspek keamanan, meskipun ROT13 menambah kompleksitas, *Affine Cipher* tetap rentan terhadap analisis frekuensi jika penyerang memiliki sampel *ciphertext* yang cukup. Selain itu, implementasi yang tidak hati-hati berpotensi membuka celah untuk serangan side-*channel*, seperti *timing attack*. Kelemahan lain terletak pada manajemen kunci; *Affine Cipher* memerlukan kunci enkripsi yang aman, dan jika kunci ini bocor, seluruh lapisan enkripsi menjadi tidak efektif.

Untuk penelitian selanjutnya, disarankan melakukan pengujian skala besar dengan *wordlist* yang lebih kompleks dan *mensimulasikan* serangan *distributed brute-force*. Selain itu, perlu dipertimbangkan penggunaan algoritma enkripsi modern seperti AES untuk menggantikan Affine Cipher guna meningkatkan keamanan transmisi data. Analisis keamanan formal dengan *tools* seperti *ProVerif* atau *CryptoVerif* juga dapat membantu memverifikasi ketahanan sistem secara matematis. Dengan demikian, meskipun kombinasi algoritma ini memberikan peningkatan keamanan, pemahaman mendalam tentang keterbatasannya akan membantu pengembangan solusi yang lebih *robust* di masa depan.

DAFTAR PUSTAKA

Abi Assyarif, R., Eka Yuliastuti, G., & Nurina Prabiantissa, C. (2023). *Implementasi Kombinasi Algoritma Rot13 dan Vernam Cipher untuk Keamanan Data Teks*.

Azhari, M., Perwitosari, J., & Ali, F. (2022). Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES). *Jurnal Pendidikan Sains Dan Komputer*, *2*(1), 2809–476. https://doi.org/10.47709/jpsk.v2i1.1390

Batubara, T. P., Efendi, S., & Nababan, E. B. (2021). Analysis Performance BCRYPT Algorithm to Improve Password Security from *Brute Force*. *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1811/1/012129

Divva, G., Zulma, M., Seta, H. B., & Yuniati, T. (2022). Implementasi Algoritma AES Dan Bcrypt untuk Pengamanan File Dokumen. *Informatik: Jurnal Ilmu Komputer*, *18*(2), 163–176.

Fachri, F. (2023). Optimasi Keamanan Web Server Terhadap Serangan *Brute-Force* Menggunakan Penetration Testing. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *10*(1), 51–58. https://doi.org/10.25126/jtiik.2023105872

Fa'izi, M. B. N. (2024, October). *600-juta-serangan-siber-harian*. Cyberhub.Id.

Febrian, D., Christian, Y., Sutisna, A. W., Budiarto, G., Syukur, M., Ramadhan, X. H., Kuntoro, A. Y., & Fahlapi, R. (2023). Implementation Of Bcrypt Algorithm On Website-Based Hashing Generator Using Laravel Framework. *Journal of Information System, Informatics and Computing*, *7*(2), 199. https://doi.org/10.52362/jisicom.v7i2.1130

Gunawan, H. (2021). PENGUKURAN KESADARAN KEAMANAN INFORMASI DAN PRIVASI DALAM SOSIAL MEDIA. *Jurnal Muara Sains, Teknologi, Kedokteran Dan Ilmu Kesehatan*, *5*(1), 1. https://doi.org/10.24912/jmstkik.v5i1.3456

Indrawan, D. H., & Sa'uda, S. (2024). Perangkat Lunak Pendataan Kehadiran Siswa Berbasis Smart QR Card Menggunakan Algoritma Bcrypt. *SATIN - Sains Dan Teknologi Informasi*, *10*(1), 211–1120. https://doi.org/10.33372/stn.v9i2.1000

Kurniasih, F., Marwati, R., Ririn, D., Program, S., Matematika, S., Matematika, P., Ilmu, D., & Alam, P. (2023). Penggabungan Affine Cipher dan Least Significant Bit-2 untuk Penyisipan Pesan Rahasia pada Gambar. *Jurnal EurekaMatika*, *11*(2), 79–88. https://ejournal.upi.edu/index.php/JEM

Laksana, T. G., & Mulyani, S. (2024). PENGETAHUAN DASAR IDENTIFIKASI DINI DETEKSI SERANGAN KEJAHATAN SIBER UNTUK MENCEGAH PEMBOBOLAN DATA PERUSAHAAN. *Jurnal Ilmiah Multidisiplin (JUKIM)*, *3*(1), 109–122.

Luthfansa, Z. M., & Rosiani, U. D. (2021). Pemanfaatan Wireshark untuk Sniffing Komunikasi Data Berprotokol HTTP pada Jaringan Internet. *Journal Information Engineering and Educational Technology*, *5*(1).

Naufal Ayman, D., & Nurhadiyanto, L. (2024). Analisis Kejahatan Siber Sniffing Pada Media Sosial Whatsapp (Studi Kasus Kurir Paket Bodong). *IKRA-ITH HUMANIORA: Jurnal Sosial Dan Humaniora*, *8*(2), 373–384. https://doi.org/10.37817/ikraith-humaniora.v8i2

Nur, J. (2023). Implementation Of Bcrypt Algorithm For Sipapeda Website Security At Bappeda Office, Buton Regency With One Time Password Method. In *JEAT: Journal of Electrical and Automation Technology* (Vol. 2, Issue 2).

Ramalinda, D., & Rachmat Raharja, A. (2024). Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi. *Journal of International Multidisciplinary Research*, *2*(6), 665–671. https://journal.banjaresepacific.com/index.php/jimr

Revenkov, P. V, Berdyugin, A. A., & Makeev, P. V. (2021). *Research on Brute Force and Black Box Attacks on ATMs*. https://www.cbr.ru/Content/Document/File/83253/onrib_2021.pdf

Siregar, S. (2023). Implementasi Mode Operasi Cipher Block Chaining (CBC) Untuk Mengoptimalkan Algoritma Affine Cipher Dalam Pengamanan Data. *Bulletin of Information System Research (BIOS)*, *1*(3), 99–109. https://journal.grahamitra.id/index.php/bios

Sunandar Informatika, A., Singaperbangsa Karawang Jl HSRonggo Waluyo, U., Timur, T., & Barat, J. (2024). IMPLEMENTASI PENETRATION TESTING DAN WORDLIST GENERATOR DALAM PENGUJIAN KEAMANAN JARINGAN MENGGUNAKAN METODE DICTIONARY ATTACK. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 3).