Jurnal Ilmu Komputer Aceh

E-ISSN: 2986-7797
Edisi : Volume 2 Nomor 3 | Oktober 2025

Available online at https://jurnal.fikompublisher.com/ilka/article/view/22

Optimasi Algoritma SHA-256 dan Metode *Salt* Untuk Pengamanan Akun Calon Santri Baru Pesantren Almuslim

Syauqa Mardhatillah[1], Sriwinar[2], Dedy Armiady[3]
[1,2,3] Universitas Almuslim

ARTICLE INFO

Article history:

Received : 08 Oktober 2025
Revised : 18 November 2025
Accepted : 19 November 2025

Keywords:
SHA-256, Salt, OpenSSL 3

Kata kunci:
SHA-256, Salt, OpenSSL 3

Corresponding Author:
Syauqa Mardhatillah

ABSTRACT

[Optimization of the SHA-256 Algorithm and Salt Method for Securing New Student Accounts at Almuslim Islamic Boarding School] Data security is a crucial aspect of applications that implement a login system. This is due to the various techniques used by unauthorized parties to gain access to account information. Therefore, protecting passwords must be a top priority, including in the management of prospective student accounts at Almuslim Islamic Boarding School. This study aims to design and implement a login system for online student registration by securing user accounts using the SHA-256 hashing algorithm combined with the salt method. The security process is carried out by hashing the password after adding a unique salt, ensuring that the stored data is not in plain text. The results of testing show that the system is able to securely store account data and successfully verify users during login through hash and salt matching.

ABSTRAK

Keamanan data menjadi aspek yang sangat penting dalam aplikasi yang menerapkan sistem *login*. Karena terdapat berbagai upaya dari pihak tidak bertanggung jawab untuk memperoleh informasi akun, oleh karena itu perlindungan terhadap *password* harus menjadi prioritas, termasuk dalam pengelolaan akun calon santri baru pesantren Almuslim. Tujuan penelitian ini adalah untuk merancang dan mengimplementasikan sistem *login* untuk pendaftaran *online* santri baru dengan pengamanan akun menggunakan algoritma *hash* SHA-256 yang dikombinasikan dengan metode *salt*. Proses pengamanan dilakukan dengan cara meng-*hash password* yang telah ditambahkan *salt* unik, sehingga data yang tersimpan tidak berupa teks asli. Hasil pengujian menunjukkan bahwa sistem dapat menyimpan data akun dengan aman, serta berhasil memverifikasi pengguna saat *login* melalui pencocokan *hash* dan *salt*.

PENDAHULUAN

Dengan berkembangnya teknologi informasi, sistem informasi kini memegang peranan penting dalam meningkatkan efisiensi dan kinerja organisasi, termasuk lembaga pendidikan [1], [2]. Sistem informasi berbasis web mempermudah penyimpanan serta pengolahan data secara cepat, tepat, dan relevan [3], [4], [5]. Namun, kemudahan akses melalui internet juga membawa risiko terhadap keamanan data [6]. Pihak-pihak tidak bertanggung jawab dapat memanfaatkan berbagai teknik, seperti sniffing, untuk mencuri data login menggunakan aplikasi tertentu [7].

Oleh karena itu, perlindungan data—khususnya pada sistem yang menerapkan autentikasi login—menjadi sangat krusial [8]. Hal ini juga berlaku dalam pengelolaan akun pendaftaran calon santri baru secara daring di Pesantren Almuslim.

Pesantren Almuslim merupakan salah satu lembaga pendidikan Islam yang berlokasi di Matangglumpangdua, Bireuen, Aceh. Seiring dengan perkembangan zaman, pesantren ini telah mengadopsi sistem pendaftaran santri baru secara *online* guna meningkatkan efisiensi pelayanan [9]. Kemajuan ini mencerminkan transformasi pesantren menjadi institusi modern yang memanfaatkan teknologi dalam proses administrasinya [10], [11]. Namun demikian, penggunaan sistem *online* juga membawa risiko kebocoran data, terutama pada akun calon santri yang menyimpan informasi penting [12]. Penyimpanan data secara daring tanpa sistem pengamanan yang memadai dapat membuka celah bagi peretasan dan penyalahgunaan data pribadi.

Untuk mengatasi tantangan tersebut, diperlukan penerapan metode kriptografi dalam sistem informasi guna melindungi privasi data. Kriptografi adalah teknik untuk mengubah data asli menjadi bentuk terenkripsi yang tidak dapat dibaca langsung, dikenal sebagai ciphertext [13]. Salah satu algoritma yang umum digunakan dalam keamanan data adalah Secure Hash Algorithm 256-bit (SHA-256) [14]. Algoritma ini mengubah data menjadi hash 256-bit dalam format heksadesimal yang bersifat unik, tidak dapat dikembalikan ke bentuk semula, serta memiliki efek avalanche dan ketahanan terhadap tabrakan data (collision) [15]. SHA-256 sering dimanfaatkan dalam sistem login untuk menjaga kerahasiaan serta keutuhan data pengguna .

Untuk memperkuat sistem keamanan, algoritma SHA-256 dapat dipadukan dengan metode salt [16]. Salt merupakan nilai acak yang disisipkan ke dalam input sebelum dilakukan proses hashing, sehingga jika dua pengguna memiliki kata sandi yang identik, hasil hash-nya tetap berbeda [17]. Penerapan salt dapat meningkatkan perlindungan terhadap password dan mengurangi risiko serangan seperti rainbow table maupun brute force [18], [19], [20].

METODE

A. SHA-256

Secure Hashing Algorithm (SHA-256 bit) adalah salah satu algoritma kriptografi modern yang paling sering digunakan untuk menjaga keamanan data, khususnya dalam perlindungan password. Algoritma ini dibuat oleh National Security Agency (NSA) dan resmi dipublikasikan oleh National Institute of Standards and Technology (NIST) pada tahun 2001 [21].

SHA-256 menghasilkan nilai hash sepanjang 256-bit atau 64 karakter dalam format heksadesimal (Secure Hash Standard, 2015). Sebagai bagian dari keluarga SHA-2 yang menggantikan SHA-1, SHA-256 menjadi algoritma yang paling sering digunakan di antara variannya seperti SHA-224, SHA-384, dan SHA-512/256. SHA-2 secara luas diandalkan dalam sistem keamanan digital, termasuk dalam implementasi sertifikat TLS/SSL di berbagai belahan dunia [22], [23].

SHA-256 dikenal sangat kuat terhadap berbagai bentuk serangan kriptografi, salah satunya adalah *collision attack*, karena algoritma ini dirancang agar hampir mustahil menghasilkan dua pesan berbeda dengan *hash* yang sama. Setiap perubahan kecil pada pesan akan menghasilkan perubahan drastis pada nilai *hash*, yang disebut efek *avalanche*. Selain itu, algoritma ini mampu menjaga panjang *hash* tetap meskipun ukuran pesan yang dimasukkan bervariasi, menjadikannya sangat andal dalam menjaga integritas dan keamanan data [24].

B. Salt

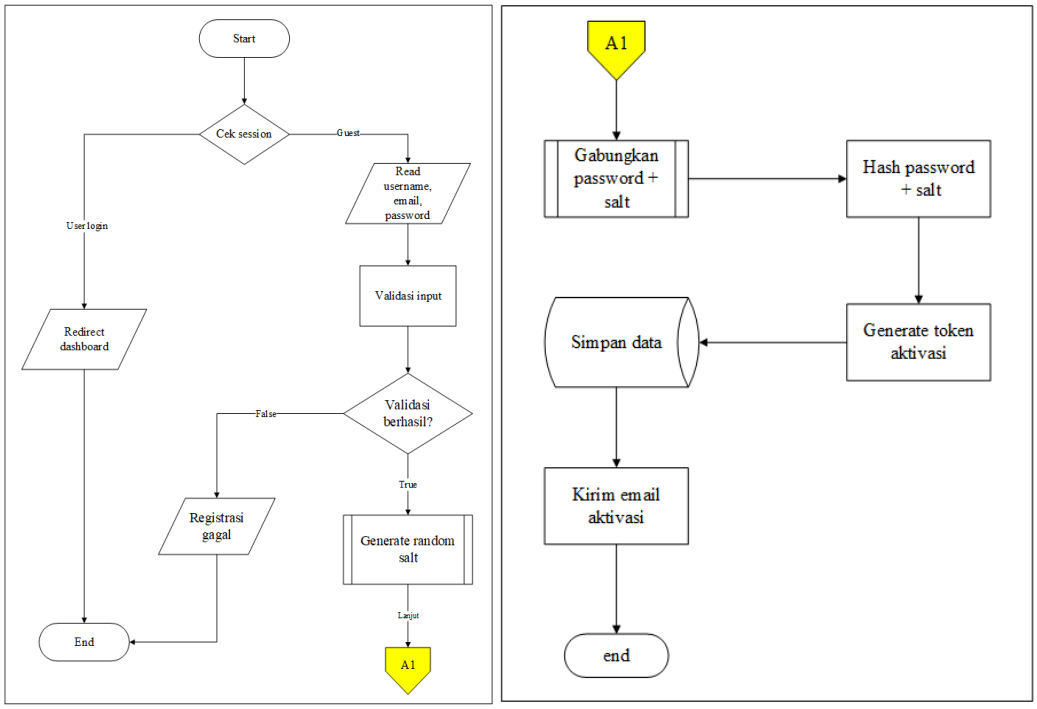
Salt merupakan rangkaian karakter acak yang disisipkan ke dalam password sebelum dilakukan proses hashing .

Tujuannya adalah membuat hasil *hash* lebih unik dan sulit ditebak, terutama jika dua pengguna memiliki *password* yang sama. Dalam penelitian ini, *salt* yang digunakan bersifat dinamis, sehingga meskipun *password* sama, hasil *hashing* tetap berbeda. *Salting* tidak memengaruhi pengalaman pengguna, namun secara signifikan menambah kompleksitas *hash* dan melindungi dari serangan seperti *table hash*, *dictionary attack*, dan *brute force[25].*

Penambahan *salt* ke dalam proses enkripsi telah banyak diterapkan oleh berbagai organisasi dan situs web untuk memperkuat keamanan *password* [26]. Meskipun terdapat banyak jenis algoritma enkripsi seperti *Blowfish*, RSA, dan SHA-256, penggunaan algoritma saja tidak cukup aman terhadap serangan *brute force*. Oleh karena itu, penelitian ini menggabungkan metode SHA-256 dengan teknik *salting* untuk mengenkripsi *password* pengguna, dengan memperhatikan penempatan *salt* yang tepat agar menghasilkan *hash* yang kuat dan aman [27].

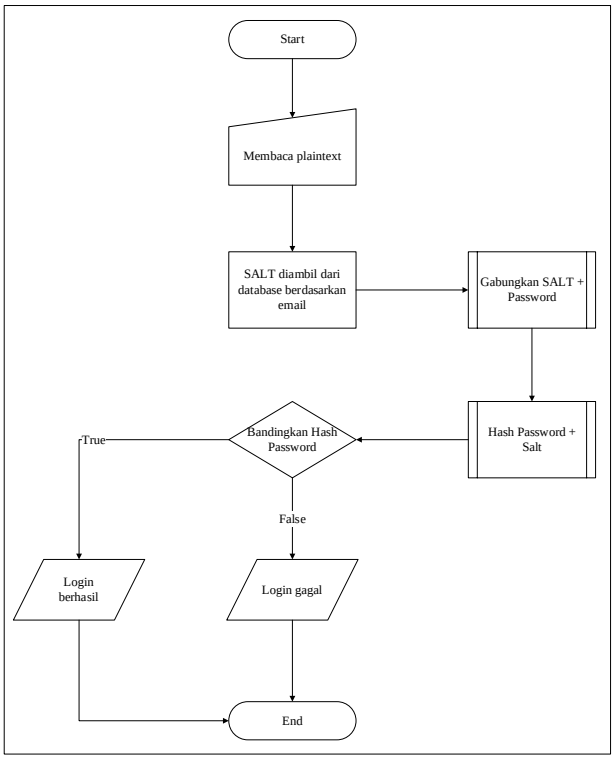
C. Perancangan Sistem

Salah satu bagian penting yang dibahas adalah proses transformasi kata menjadi *chipertext*, yaitu perubahan dari teks yang dapat dibaca menjadi teks yang terenkripsi menggunakan algoritma SHA-256. Selanjutnya, untuk memperkuat keamanan, diterapkan teknik *salt*, penambahan elemen acak sebanyak 64 karakter ke dalam *input* sebelum proses *hashing* dilakukan. Teknik ini bertujuan untuk menghasilkan nilai *hash* yang unik, meskipun *input* yang digunakan sama. *Salt* tersebut dibangkitkan secara otomatis menggunakan *library OpenSSL* versi 3 yang tersedia dalam PHP 8. Proses keseluruhan ini digambarkan secara rinci dalam *flowchart* yang menyertai bagian ini.



Gambar 1 Flawchat registrasi akun dan proses hasting

Diagram *flowchart* di atas menggambarkan sistem proses keseluruhan transformasi kata menjadi *chippertext*, perubahan dari teks yang dapat dibaca menjadi teks yang terenkripsi menggunakan algoritma SHA-256.



Gambar 2 Flowchart validasi chipertext

Diagram *flowchart* di atas menggambarkan sistem membaca *plaintext,* yaitu ketika pengguna atau calon santri memasukkan email dan *password* pada halaman *login*, sistem kemudian mengambil *salt* berdasarkan email tersebut dari *database*. Kemudian *password* yang dimasukkan oleh calon santri digabungkan dengan *salt*, lalu dilakukan proses *hashing, dan* hasil *hashing* kemudian dibandingkan dengan nilai *hash password* di *database*. Jika hasilnya cocok, maka *login* dinyatakan berhasil. Sebaliknya, jika hasilnya tidak cocok, maka *login* dinyatakan gagal.

HASIL DAN PEMBAHASAN

A. Penyimpanan Data

Gambar berikut ini menunjukkan hasil penyimpanan data *password* yang telah dienkripsi menggunakan algoritma SHA-256 dengan penerapan metode *salt*. Kolom *password* menyimpan hasil *hashing* dari *password* asli yang telah digabungkan dengan nilai *salt*, sedangkan kolom salt menyimpan nilai *acak* yang dihasilkan secara unik untuk setiap akun menggunakan fungsi *openssl_random_pseudo_bytes*() pada PHP 8.

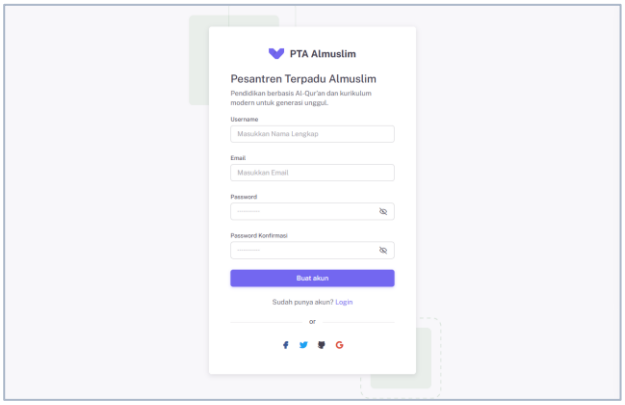
Dari tabel terlihat bahwa meskipun dua pengguna mungkin menggunakan *password* yang sama, nilai *hash* yang dihasilkan tetap berbeda karena adanya penambahan *salt* yang unik. Hal ini bertujuan untuk mencegah serangan seperti *rainbow table attack*, di mana pelaku bisa menebak *password* berdasarkan nilai *hash* yang umum. Dengan teknik ini, sistem keamanan menjadi lebih kuat dan tidak rentan terhadap pencurian *data password*, karena nilai *hash* tidak dapat dibalik ke bentuk aslinya, bahkan jika nilai *password* memiliki pola yang sama.

password	salt
0bb4416b660ce8693010fba1fec801e10608dd111374f1b7da...	1208ee120e2bc0ed5892d198ba172f40094b1c4f21f5d20e79...
a1aa8c2f72dac3d1e94bb7c77c9c7354dbd9ec0297013c0427...	802f7ac7e218c11e043173b833ae61737e15c00067c9c4bd9f...
d4ef1f2bfdf78ff79f1a427ee0fece9575ecf4b81035dbc32e...	2724fcc9ec6675aa7062e9b6440ab6fd7472fa6603eca56aa2...
ceed79ac500b9d58f55e3e504a5366f7ce2f65abbf5315719e...	18c0b5647967269ba012a60e92dc58048e96ed91c917e81276...
d37e18993fef0cf8dcb000fadd66a5f4491b6080aacd51c1b5...	6e40c7144d14d9cbcd96764edd856314c6263e3fee8c8e8105...

Gambar 3 NIlai hash dan salt

B. Implementasi Halaman Registrasi akun

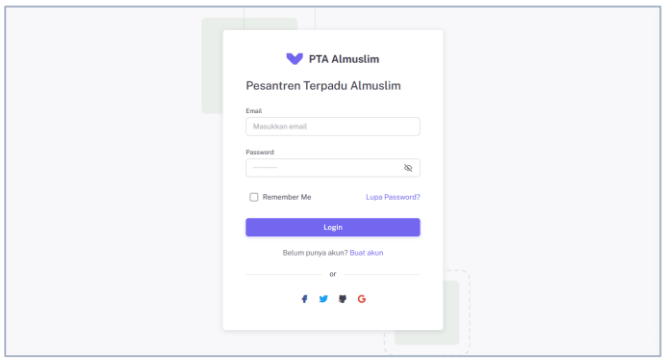
Tahap implementasi adalah proses di mana aplikasi yang telah dirancang dan dikembangkan mulai diuji untuk menilai kelayakannya sebelum digunakan secara penuh sesuai dengan fungsinya. Tahap ini bertujuan untuk memastikan bahwa aplikasi dapat berfungsi dengan optimal, sesuai dengan kebutuhan pengguna, serta mampu mewujudkan sistem yang telah dirancang sebelumnya.



Gambar 4 Antarmuka halamn registrasi akun

C. Implementasi Halaman *Login*

Tampilan halaman *login* sistem pendaftaran santri baru Pesantren Terpadu Almuslim yang terdapat beberapa elemen. Pengguna diminta memasukkan email dan kata sandi untuk mengakses sistem. Tersedia fitur tampilkan kata sandi, tombol *login*, serta tautan untuk pemulihan akun dan pendaftaran akun baru. Tampilan ini dirancang sesederhana mungkin dan mudah digunakan.



Gambar 5 Antarmuka halaman login

KESIMPULAN

Merujuk pada hasil analisis, perancangan, dan implementasi yang telah dipaparkan sebelumnya, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem berhasil menciptakan random salt dengan panjang tetap dan dinamis yang berfungsi sebagai teks tambahan pada password, meningkatkan lapisan keamanan awal.
2. Password yang telah ditambahkan salt berhasil di hashing menggunakan algoritma SHA-256, dengan nilai hash tersimpan secara aman di dalam basis data.

Sistem berhasil meningkatkan keamanan akun pengguna dengan menjadikan tabel precomputed serangan rainbow table tidak efektif, karena tidak ada nilai hash yang dapat dicocokkan dengan tabel tersebut.

REFERENCES

[1] O. Dakhi, M. Masril, R. Novalinda, J. Jufrinaldi, and A. Ambiyar, "Analisis Sistem Kriptografi dalam Mengamankan Data Pesan Dengan Metode One Time Pad Cipher," *INVOTEK: Jurnal Inovasi Vokasional dan Teknologi*, vol. 20, no. 1, 2020, doi: 10.24036/invotek.v20i1.647.

[2] R. Rivaldi and S. Subandi, "IMPLEMENTASI KEAMANAN DATA ARSITEKTUR MENGGUNAKAN ALGORITMA KRIPTOGRAFI DENGAN METODE RIVEST CODE (4 RC4)PADA PT.NAVIRI INDAH CEMERLANG," *SKANIKA*, vol. 4, no. 2, 2021, doi: 10.36080/skanika.v4i2.2249.

[3] F. D. Putra, J. Riyanto, and A. F. Zulfikar, "Rancang Bangun Sistem Informasi Manajemen Aset pada Universitas Pamulang Berbasis WEB," *Journal of Engineering, Technology, and Applied Science*, vol. 2, no. 1, 2020, doi: 10.36079/lamintang.jetas-0201.93.

[4] F. Fitriani and I. Muslem R, "E-Absensi Mahasiswa Fakultas Ilmu Komputer Universitas Almuslim Berbasis Web," *JURNAL TIKA*, vol. 5, no. 3, 2021, doi: 10.51179/tika.v5i3.141.

[5] S. Winar, E. Rizki Putra, and I. Muslem R., "Sistem Informasi Kalkulasi Zakat Pada Kantor Baitul Mal Kabupaten Bireuen Berbasis Android," *Jurnal TIKA*, vol. 7, no. 3, 2022, doi: 10.51179/tika.v7i3.1584.

[6] A. R. Gunawan, N. P. Sastra, and D. M. Wiharta, "Penerapan Keamanan Jaringan Menggunakan Sistem Snort dan Honeypot Sebagai Pendeteksi dan Pencegah Malware," *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 1, 2021, doi: 10.24843/mite.2021.v20i01.p09.

[7] D. W. Wesson, "Sniffing behavior communicates social hierarchy," *Current Biology*, vol. 23, no. 7, 2013, doi: 10.1016/j.cub.2013.02.012.

[8] R. Dewi, T. M. Johan, and I. Muslem R., "Aplikasi Kriptografi Dalam Mengamankan Pesan Teks Dengan Metode Algoritma Rc4 Berbasis Android," *JURNAL TIKA*, vol. 6, no. 01, 2021, doi: 10.51179/tika.v6i01.416.

[9] A. Wijaya and T. Sutabri, "ANALISIS INFORMATION TECHNOLOGY SERVICE MANAGEMENT (ITSM) APLIKASI ABSENSI ONLINE PADA BADAN KEPEGAWAIAN DAN PENGEMBANGAN SUMBER DAYA MANUSIA OGAN ILIR," *Blantika : Multidisciplinary Journal*, vol. 2, no. 1, 2023, doi: 10.57096/blantika.v2i1.7.

[10] U. Aedi and Asep Amaludin, "Modernisasi Sistem Manajemen Pesantren Dengan SIAP(Sistem Informasi Administrasi Pesantren) Pada Pondok Pesantren Sirojuth Tholibin," *Journal of Islamic Management*, vol. 2, no. 2, 2022, doi: 10.15642/jim.v2i2.830.

[11] A. B. Setiawan and J. Sulaksono, "SISTEM INFORMASI MANAJEMEN SANTRI DI PONDOK PESANTREN AL ISHLAH KOTA KEDIRI," *Network Engineering Research Operation*, vol. 4, no. 2, 2019, doi: 10.21107/nero.v4i2.122.

[12] R. Dewi, T. M. Johan, and I. Muslem R., "Aplikasi Kriptografi Dalam Mengamankan Pesan Teks Dengan Metode Algoritma Rc4 Berbasis Android," *JURNAL TIKA*, vol. 6, no. 01, 2021, doi: 10.51179/tika.v6i01.416.

[13] A. S. Assaidi and A. Amborowati, "Perancangan Aplikasi Diary Menggunakan Algoritma Kriptografi Rc6 Berbasis Android," *Seminar Nasional Teknologi Informasi dan Multimedia*, vol. 4, no. 7, 2016.

[14] S. Suhaili and N. Julai, "FPGA-based Implementation of SHA-256 with Improvement of Throughput using Unfolding Transformation," *Pertanika J Sci Technol*, vol. 30, no. 1, 2022, doi: 10.47836/PJST.30.1.32.

[15] T. H. Tran, H. L. Pham, and Y. Nakashima, "A High-Performance Multimem SHA-256 Accelerator for Society 5.0," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3063485.

[16] Sutriman and B. Sugiantoro, "Analysis of password and salt combination scheme to improve hash algorithm security," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, 2019, doi: 10.14569/IJACSA.2019.0101158.

[17] S. E. S. Castelo, R. J. L. Apostol, D. M. A. Cortez, R. M. Dioses, M. C. R. Blanco, and V. A. Agustin, "Modification of SHA-512 using Bcrypt and salt for secure email hashing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, 2024, doi: 10.11591/ijeecs.v33.i1.pp398-404.

[18] Z. Li, D. Wang, and E. Morais, "Quantum-Safe Round-Optimal Password Authentication for Mobile Devices," *IEEE Trans Dependable Secure Comput*, vol. 19, no. 3, 2022, doi: 10.1109/TDSC.2020.3040776.

[19] H. Liu, Y. Xu, and C. Ma, "Chaos-based image hybrid encryption algorithm using key stretching and hash feedback," *Optik (Stuttg)*, vol. 216, 2020, doi: 10.1016/j.ijleo.2020.164925.

[20] P. Gauravaram, "Security analysis of salt||password hashes," in *Proceedings - 2012 International Conference on Advanced Computer Science Applications and Technologies, ACSAT 2012*, 2012. doi: 10.1109/ACSAT.2012.49.

[21] J. Wang, G. Liu, Y. Chen, and S. Wang, "Construction and Analysis of SHA-256 Compression Function Based on Chaos S-Box," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3071501.

[22] O. A. Manankova, M. Z. Yakubova, and A. S. Baikenov, "Cryptanalysis the SHA-256 Hash Function Using Rainbow Tables," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 4, 2022, doi: 10.52549/ijeei.v10i4.4247.

[23] K. Nakamura, K. Hori, and S. Hirose, "Algebraic fault analysis of sha-256 compression function and its application," *Information (Switzerland)*, vol. 12, no. 10, 2021, doi: 10.3390/info12100433.

[24] F. N. Khasanah, "Application of Hash Sha-256 Algorithm in Website-Based Sales Software Engineering," *Journal of Applied Data Sciences*, vol. 3, no. 1, 2022, doi: 10.47738/jads.v3i1.50.

[25] S. Boonkrong and C. Somboonpattanakit, "Dynamic salt generation and placement for secure password storing," *IAENG Int J Comput Sci*, vol. 43, no. 1, 2016.

[26] M. Risqi Firdaus -13520043, "Analisis Penggunaan Algoritma Bcrypt dengan Garam (Salt) untuk Pengamanan Password dari Peretasan."

[27] A. Nugroho and T. Mantoro, "Salt Hash Password Using MD5 Combination for Dictionary Attack Protection," in *Proceedings - 2023 6th International Conference on Computer and Informatics Engineering: AI Trust, Risk and Security Management (AI Trism), IC2IE 2023*, 2023. doi: 10.1109/IC2IE60547.2023.10331606.