Jurnal Informatika: Jurnal pengembangan IT

Vol. 11, No. 2, 2026
ISSN : 2477-5126, DOI:10.30591/jpit.v11i2.10280

Perbandingan Bcrypt, Argon2, dan PBKDF2 pada Keamanan SIMPEG Berbasis Web

Muchamad Gilang Dwi Saputra[1], Denar Regata Akbi[2]
1,2 Program Studi Informatika, Universitas Muhammadiyah Malang, Malang, 65144, Indonesia
[1]gilds140703@webmail.umm.ac.id, [2]dnarregata@umm.ac.id

Info Artikel

Riwayat Artikel:

Received 2026-02-06
Revised 2026-05-07
Accepted 2026-05-15

Abstract *– Password security is an important aspect of the web-based Employee Management Information System (SIMPEG) because this system contains user data and sensitive employee information. This study aims to compare the performance and security resilience of the Bcrypt, Argon2, and PBKDF2 algorithms using the SIMPEG authentication module. The method used is applied experimentation where the three algorithms are implemented in the registration and login processes, using a dataset of 200 synthetic passwords divided into four levels of complexity. Performance testing is based on hashing time, verification time, hash length, and resource usage. Security testing has been conducted with a dictionary attack simulation using Hashcat, NVIDIA RTX 3060 6 GB GPU, and a 5000-password wordlist. The test results show that the lowest hashing and verification times are for PBKDF2, which are 226.007 ms and 228.536 ms, followed by Bcrypt with 317.610 ms and 320.693 ms. Argon2 has the highest processing times with 1403.172 ms for hashing and 1198.050 ms for verification. However, Argon2 has the best resistance to dictionary attacks with a cracking time of 6 hours and 35 minutes and a hash rate of 19 H/s, better than Bcrypt and PBKDF2. Thus, Argon2 is recommended for SIMPEG with a higher priority on password security, while Bcrypt can be a more balanced alternative between security and performance.*

Keywords: *Argon2; Bcrypt; Dictionary Attack; PBKDF2; SIMPEG.*

Corresponding Author:
Muchamad Gilang Dwi Saputra
Email:
gilds140703@webmail.umm.ac.id

Abstrak *– Keamanan password adalah aspek penting dari Sistem Informasi Manajemen Kepegawaian (SIMPEG) berbasis web karena sistem ini berisi data pengguna dan informasi sensitif pegawai. Penelitian ini bertujuan untuk membandingkan performa dan ketahanan keamanan algoritma Bcrypt, Argon2, dan PBKDF2 menggunakan modul autentikasi SIMPEG. Metode yang digunakan adalah eksperimen terapan di mana ketiga algoritma diterapkan dalam proses registrasi dan login, menggunakan dataset 200 password sintetis yang dibagi menjadi empat tingkat kompleksitas. Pengujian performa didasarkan pada waktu hashing, waktu verifikasi, panjang hash, dan penggunaan sumber daya. Uji keamanan telah dilakukan dengan simulasi dictionary attack menggunakan Hashcat, GPU NVIDIA RTX 3060 6 GB, dan wordlist 5000 password. Hasil pengujian menunjukkan bahwa waktu hashing dan verifikasi terendah adalah PBKDF2, yaitu 226,007 ms dan 228,536 ms, diikuti oleh Bcrypt dengan 317,610 ms dan 320,693 ms. Argon2 memiliki waktu pemrosesan tertinggi dalam hashing 1403,172 ms dan dalam verifikasi 1198,050 ms. Namun, Argon2 memiliki ketahanan terbaik terhadap dictionary attack dengan waktu cracking 6 jam 35 menit dan hash rate 19 H/s, lebih baik daripada Bcrypt dan PBKDF2. Dengan demikian, Argon2 direkomendasikan untuk SIMPEG dengan prioritas keamanan password yang lebih tinggi, sedangkan Bcrypt dapat menjadi alternatif yang lebih seimbang antara keamanan dan performa.*

Kata Kunci*: Argon2, Bcrypt, Dictionary Attack, PBKDF2, SIMPEG.*

I. PENDAHULUAN

Aplikasi web telah menjadi platform yang sering digunakan untuk mengelola layanan digital karena dapat menyediakan akses data yang cepat, terintegrasi, dan fleksibel. Aplikasi web digunakan di berbagai instansi untuk mengelola informasi penting, seperti data administrasi, data pengguna, layanan instansi, autentikasi, dan layanan digital lainnya. Namun, penggunaan aplikasi web yang luas juga meningkatkan risiko keamanan karena sistem berbasis web sering menjadi target serangan siber. Penelitian sebelumnya yang diterbitkan dalam jurnal Informatika: Jurnal Pengembangan IT (JPIT) menunjukkan bahwa aplikasi web berpotensi rentan jika tidak dilengkapi dengan mekanisme keamanan yang memadai, sehingga pengujian keamanan adalah hal penting dalam pengembangan sistem [1]. Oleh karena itu, keamanan aplikasi web perlu dipertimbangkan secara menyeluruh, terutama pada bagian autentikasi pengguna.

Sistem Informasi Manajemen Kepegawaian atau SIMPEG adalah salah satu sistem informasi berbasis web yang memerlukan perlindungan keamanan tinggi. SIMPEG digunakan untuk mengelola data karyawan, informasi administratif, riwayat pekerjaan, dan akun pengguna yang terkait dengan proses autentikasi sistem. Data yang

disimpan dalam SIMPEG adalah data sensitif karena berkaitan dengan informasi identitas dan pekerjaan. Sistem autentikasi yang tidak dirancang dengan baik membuat sistem rentan terhadap penyalahgunaan akses, pencurian data atau kebocoran data. Masalah serupa terjadi pada sistem informasi institusi lainnya, misalnya, Sistem Informasi Akademik yang memerlukan perlindungan password melalui mekanisme hashing yang lebih aman [2].

Dalam sistem informasi, password adalah elemen dasar dari proses autentikasi pengguna. Password mudah untuk diterapkan, tetapi juga merupakan kerentanan yang sering dieksploitasi oleh penyerang, jika password disimpan dalam *plaintext* atau menggunakan metode hashing yang lemah, keamanannya mungkin akan terganggu. Dalam aplikasi web, hashing password adalah langkah kritis untuk melindungi kredensial pengguna jika database diakses oleh pihak yang tidak berwenang [3]. Dengan demikian, pemilihan pendekatan hashing password harus mempertimbangkan seberapa baik algoritma tersebut mampu melindungi password dari upaya pemulihan atau pencocokan secara tidak sah.

Ancaman terhadap keamanan password semakin meningkat seiring dengan berkembangnya teknik cracking password. Serangan seperti *brute force* dan *dictionary attack* dapat digunakan untuk mencoba berbagai kombinasi password hingga ditemukan nilai yang sesuai dengan hash password yang tersimpan di database. Perkembangan kemampuan komputasi modern, termasuk penggunaan GPU dan optimasi performa dalam proses *cracking*, memungkinkan serangan dilakukan lebih cepat dibandingkan metode konvensional [4]. Kondisi ini menunjukkan bahwa evaluasi algoritma hashing password tidak hanya perlu mempertimbangkan ketahanan terhadap serangan berbasis komputasi modern, tetapi juga aspek performa, penggunaan memori, dan karakteristik hash yang dihasilkan.

Algoritma yang umum digunakan untuk penyimpanan password adalah Bcrypt, Argon2, dan PBKDF2. Bcrypt adalah algoritma hashing adaptif yang menggunakan *cost factor* untuk meningkatkan beban komputasi, sehingga membuat serangan *brute force* lebih sulit. Argon2 adalah algoritma yang memiliki karakteristik *memory-hard*, sehingga memerlukan lebih banyak sumber daya memori, dan dengan demikian membuatnya lebih tahan terhadap serangan GPU dan perangkat keras khusus. PBKDF2 adalah fungsi derivasi kunci yang cukup populer karena ketersediannya yang luas dan kemungkinan untuk meningkatkan jumlah iterasi guna meningkatkan keamanan password. Perbedaan karakteristik menunjukkan bahwa setiap algoritma memiliki kelebihan dan keterbatasan yang perlu dievaluasi sesuai dengan konteks implementasinya [5].

Penelitian sebelumnya telah menunjukkan bahwa Bcrypt dapat diterapkan dalam sistem berbasis web untuk memperkuat dan meningkatkan keamanan data pengguna. Implementasi AES dan Bcrypt pada situs web jahitku menunjukkan bahwa Bcrypt dapat digunakan untuk mengamankan password pengguna dalam sistem database [6]. Selain itu, penelitian lain menyatakan bahwa Argon2 dapat diterapkan pada sistem aplikasi web karena memiliki karakteristik yang sesuai untuk kebutuhan hashing password modern [7]. Bcrypt dikenal dapat meningkatkan keamanan password dalam hal serangan *brute force* dengan meningkatkan beban komputasi [8]. Di sisi lain, PBKDF2 dapat digunakan dalam mekanisme perlindungan password karena menggunakan jumlah iterasi dan salt untuk membuatnya rumit dan sulit untuk proses *cracking* [9].

Dalam beberapa penelitian terdahulu telah membahas keamanan aplikasi web, implementasi algoritma hashing password, dan pengujian terhadap serangan berbasis password. Tetapi setiap penelitian memiliki fokus dan lingkup yang berbeda. Perbandingan penelitian terdahulu menjadi dasar penelitian ini ditunjukkan pada Tabel 1.

TABEL 1
PERBANDINGAN PENELITIAN TERDAHULU

Judul Penelitian	Fokus Penelitian	Algoritma/Metode	Keterbatasan Penelitian Terdahulu	Relevansi dengan Penelitian Ini
Pengujian Keamanan Fitur Upload File pada Sistem Aplikasi Web [1]	Keamanan aplikasi web	Pengujian kerentanan fitur upload file	Tidak membahas keamanan password dan algoritma hashing	Menjadi dasar bahwa aplikasi web perlu diuji dari sisi keamanan
Penerapan Algoritma Bcrypt untuk Pengamanan Password pada Sistem Informasi Akademik (SIAK) [2]	Pengamanan password pada sistem informasi akademik	Bcrypt	Hanya berfokus pada Bcrypt dan belum membandingkan dengan Argon2 serta PBKDF2	Mendukung pentingnya hashing password pada sistem informasi institusional
Password Hashing Methods and Algorithms on the .NET Platform [5]	Perbandingan metode password hashing	PBKDF2, Bcrypt, Scrypt, Argon2	Konteks pengujian pada platform .NET, bukan SIMPEG berbasis web	Menjadi acuan utama perbandingan Bcrypt, Argon2, dan PBKDF2

Implementasi Algoritma AES dan Bcrypt untuk Pengamanan Data Pengguna pada Aplikasi web Jahitku [6]	Pengamanan data pengguna pada aplikasi web	AES dan Bcrypt	Tidak membandingkan Bcrypt dengan Argon2 dan PBKDF2	Mendukung penerapan Bcrypt pada sistem berbasis web
Simulasi Hashing Password Menggunakan Argon2 dan Scrypt serta Pengembangan Fitur Logging Jaringan Real-Time Berbasis Aplikasi web [7]	Simulasi hashing password pada sistem aplikasi web	Argon2 dan Scrypt	Tidak membahas Bcrypt dan PBKDF2 secara langsung	Mendukung penggunaan Argon2 pada sistem berbasis web
Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force [8]	Analisis performa Bcrypt terhadap brute force	Bcrypt	Tidak membandingkan dengan Argon2 dan PBKDF2	Mendukung analisis Bcrypt terhadap serangan brute force
Analysis Attackers' Methods with Hashing Secure Password Using CSPRNG and PBKDF2 [9]	Analisis metode serangan terhadap password hashing	CSPRNG dan PBKDF2	Fokus pada PBKDF2 dan belum diterapkan pada SIMPEG	Mendukung penggunaan PBKDF2 dalam pengamanan password

Tabel 1 menunjukkan bahwa berbagai aspek keamanan pada aplikasi web, implementasi hashing password, dan serangan berbasis password telah diteliti dalam penelitian sebelumnya. Namun, sebagian besar penelitian masih berfokus pada algoritma tertentu, misalnya Bcrypt, Argon2, atau PBKDF2, dan dalam konteks sistem yang berbeda. Selain itu, tidak ada penelitian yang secara langsung membandingkan ketiga algoritma dalam konteks sistem SIMPEG berbasis web, dengan mempertimbangkan performa hashing, pemanfaatan sumber daya, dan ketahanan terhadap *dictionary attack* berbasis GPU.

Kesenjangan penelitian ini terletak pada kurangnya evaluasi empiris yang secara khusus membandingkan Bcrypt, Argon2, dan PBKDF2 dalam konteks SIMPEG berbasis web. Penelitian terdahulu telah memberikan kontribusi pada keamanan password, tetapi tidak mengintegrasikan secara komprehensif uji performa, karakteristik implementasi, dan simulasi serangan dalam satu lingkungan sistem yang sama. Selain itu, evaluasi terhadap ketahanan algoritma hashing pada sistem informasi kepegawaian masih terbatas, meskipun sistem seperti SIMPEG menyimpan data sensitif dan memerlukan mekanisme autentikasi yang kuat. Oleh karena itu, diperlukan penelitian yang dapat memberikan gambaran empiris tentang algoritma hashing yang paling cocok dan sesuai untuk diterapkan di SIMPEG.

Kebaruan penelitian ini terletak pada implementasi langsung tiga algoritma hashing password, yaitu Bcrypt, Argon2, dan PBKDF2, dalam modul autentikasi SIMPEG berbasis web. Penelitian ini tidak hanya membandingkan waktu hashing dan verifikasi, tetapi juga mengevaluasi penggunaan sumber daya serta ketahanan terhadap dictionary attack berbasis GPU. Pengujian terhadap Argon2 menjadi krusial karena algoritma ini memiliki karakteristik memory-hard yang dirancang untuk meningkatkan ketahanan terhadap serangan modern dan telah mulai dievaluasi dalam implementasi perangkat lunak nyata [10]. Melalui pendekatan tersebut, penelitian ini memberikan kontribusi empiris dalam mengevaluasi keseimbangan antara performa dan keamanan pada sistem informasi kepegawaian.

Penelitian ini bertujuan untuk membandingkan performa dan tingkat keamanan algoritma Bcrypt, Argon2, dan PBKDF2 pada SIMPEG berbasis web. Evaluasi dilakukan berdasarkan waktu hashing, waktu verifikasi, panjang hash, penggunaan sumber daya, dan ketahanan terhadap dictionary attack berbasis GPU. Hasil penelitian ini diharapkan dapat memberikan rekomendasi dan kontribusi teknis kepada pengembang sistem dan instansi pemerintah untuk pemilihan algoritma hashing password yang sesuai untuk sistem informasi yang mengelola data sensitif. Selain itu, diharapkan penelitian ini dapat digunakan dan menjadi acuan sebagai referensi dalam memperkuat keamanan autentikasi pada sistem informasi berbasis web.

II. METODE

Penelitian ini menerapkan metode eksperimen terapan yang bertujuan untuk mengevaluasi performa dan ketahanan keamanan algoritma hashing password dalam skenario autentikasi SIMPEG berbasis web. Metode eksperimen dipilih karena penelitian ini tidak hanya membahas algoritma secara teoritis, tetapi juga mengimplementasikan Bcrypt, Argon2, dan PBKDF2 secara langsung pada modul registrasi dan login. Ketiga

algoritma tersebut dipilih karena merupakan metode password hashing modern yang memiliki karakteristik berbeda dalam hal performa, penggunaan sumber daya, dan ketahanan terhadap serangan berbasis perangkat keras modern [5].

Penelitian ini dilakukan melalui beberapa tahap, yaitu perancangan pengujian, penentuan lingkungan pengujian, penyusunan dataset password, konfigurasi parameter algoritma hashing, pengujian performa sistem, pengujian keamanan menggunakan dictionary attack berbasis GPU, validasi dan pengolahan data, serta analisis hasil pengujian. Seluruh tahap dirancang agar proses pengujian dapat dilaksanakan secara terukur, konsisten, dan dapat direplikasi.

A. Desain Penelitian

Desain penelitian ini menggunakan pendekatan komparatif kuantitatif dengan membandingkan tiga algoritma hashing password, yaitu Bcrypt, Argon2, dan PBKDF2. Perbandingan dilakukan pada lingkungan sistem yang sama, sehingga perbedaan hasil yang diperoleh dapat dikaitkan dengan karakteristik masing-masing algoritma dan bukan dengan perbedaan konfigurasi perangkat atau sistem.

Objek penelitian adalah modul autentikasi dalam SIMPEG berbasis web yang telah dimodifikasi menjadi tiga versi implementasi. Setiap versi menggunakan algoritma hashing yang berbeda dalam proses registrasi dan login pengguna. Variasi implementasi algoritma ditunjukkan pada Tabel 2.

TABEL 2
VARIASI IMPLEMENTASI ALGORITMA HASHING PADA SIMPEG

Versi Sistem	Algoritma Hashing
SIMPEG-1	Bcrypt
SIMPEG-2	Argon2
SIMPEG-3	PBKDF2

Bcrypt, Argon2, dan PBKDF2 dipilih karena pendekatan teknis mereka yang berbeda dalam keamanan password. Bcrypt menggunakan mekanisme *cost factor* untuk meningkatkan beban komputasi, Argon2 memanfaatkan pendekatan *memory-hard* untuk meningkatkan kebutuhan memori, sementara PBKDF2 menerapkan iterasi berulang dan salt untuk memperkuat proses derivasi kunci [5].

B. Lingkungan Pengujian

Pengujian dilakukan di lingkungan pengembangan lokal untuk memberikan kontrol yang konsisten atas kondisi eksperimen. Semua algoritma diuji pada perangkat keras dan perangkat lunak yang sama. Lingkungan yang sama digunakan untuk memastikan perbandingan performa yang valid di antara algoritma. Spesifikasi lingkungan pengujian dapat dilihat pada Tabel 3.

TABEL 3
LINGKUNGAN PENGUJIAN

Komponen	Spesifikasi
CPU	Intel Core i7-12650H
GPU	NVIDIA RTX 3060 6 GB
RAM	16 GB DDR5 4800 MHz
Sistem Operasi	Windows 11 Home Single Language 64-bit
Platform Pengembangan	Laragon
Bahasa Pemrograman	PHP
Database	MySQL
Tool Pengujian Serangan	Hashcat v7.1.2
Mode Serangan	Dictionary Attack
Wordlist	5000 password sintetis

Penggunaan GPU dalam pengujian keamanan dilakukan karena serangan password cracking dapat memanfaatkan pemrosesan paralel untuk mempercepat pencocokan hash. Argon2 juga relevan diuji di lingkungan GPU, karena algoritma ini dirancang dengan parameter memori dan paralelisme yang mempengaruhi efektivitas serangan berbasis perangkat keras [11].

C. Dataset Password

Dataset password yang digunakan dalam penelitian ini terdiri dari 200 password plaintext sintetis dengan panjang minimal 8 karakter. Perlu dicatat bahwa dataset ini tidak diambil dari password pengguna nyata tetapi dihasilkan dengan cara yang terkontrol untuk keperluan eksperimental. Dataset sintetis digunakan untuk menjaga aspek etika penelitian dan menghindari penggunaan kredensial pengguna nyata.

Dataset ini dikategorikan ke dalam empat kelas kompleksitas, yaitu password lemah, sedang, kuat, dan sangat kuat. Setiap kategori terdiri dari 50 password. Pembagian ini dilakukan agar pengujian dapat mewakili variasi pola password yang mungkin digunakan oleh pengguna sistem. Karakteristik dataset ditunjukkan dalam Tabel 4.

TABEL 4

KARAKTERISTIK DATASET PASSWORD

Kategori Password	Jumlah Data	Karakteristik
Lemah	50	Huruf kecil atau angka sederhana dengan pola umum
Sedang	50	Kombinasi huruf kecil dan angka
Kuat	50	Kombinasi huruf besar, huruf kecil, dan angka
Sangat Kuat	50	Kombinasi huruf besar, huruf kecil, angka, dan simbol
Total	200	Variasi kompleksitas password

Untuk meningkatkan validitas hasil pengujian, ukuran dataset ditingkatkan dari 15 password menjadi 200 password. Dataset yang lebih besar dan lebih bervariasi dapat memberikan gambaran yang lebih representatif terhadap performa hashing dan ketahanan password dalam skenario pengujian. Metode pengujian variasi karakter password juga digunakan dalam penelitian sebelumnya untuk menguji pengaruh karakteristik input terhadap keamanan dan performa hashing [12].

D. Parameter Algoritma Hashing

Setiap algoritma dikonfigurasi menggunakan parameter yang telah ditetapkan sebelum proses pengujian dimulai. Penetapan parameter dilakukan agar pengujian dapat berjalan konsisten pada seluruh dataset. Parameter algoritma hashing yang digunakan dalam studi ini ditunjukkan dalam Tabel 5.

TABEL 5

PARAMETER ALGORITMA HASHING

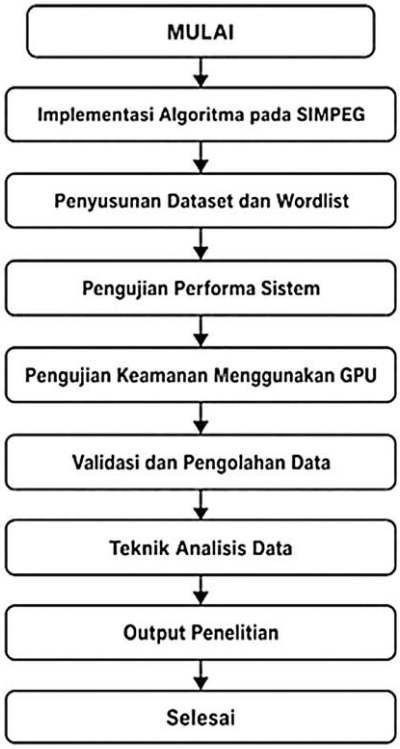
Algoritma	Salt	Parameter	Keterangan
Bcrypt	Otomatis	cost = 12	Meningkatkan beban komputasi hashing
Argon2	Otomatis	memory = 262144 KB, time cost = 4, parallelism = 1	Memory-hard dan lebih tahan terhadap serangan berbasis GPU
PBKDF2	16 byte random salt	iteration = 260000	Kompatibel luas dan berbasis iterasi

Parameter Bcrypt menggunakan cost = 12, karena nilai cost mempengaruhi keseimbangan antara keamanan dan performa. Semakin tinggi nilai cost, semakin besar beban komputasi yang diperlukan untuk menghasilkan hash, membuat proses brute force menjadi lebih sulit, sementara waktu pemrosesan juga meningkat [12]. Oleh karena itu, cost = 12 dipilih sebagai konfigurasi yang masih memungkinkan diterapkan pada sistem autentikasi berbasis web tanpa menghasilkan latensi yang terlalu tinggi.

Argon2 dikonfigurasi dengan memori = 262144 KB, time cost = 4 dan paralelisme = 1. Parameter ini dipilih karena Argon2 memiliki karakteristik memory-hard yang meningkatkan kebutuhan memori dalam proses hashing, sehingga serangan berbasis GPU menjadi lebih sulit dilakukan secara efisien [11]. Sementara itu, PBKDF2 menggunakan 260000 iterasi dan 16-byte random salt untuk meningkatkan biaya komputasi dalam proses penebakan password. PBKDF2 masih relevan untuk digunakan karena memiliki kompatibilitas yang luas dan mendukung konfigurasi iterasi tinggi sebagai mekanisme penguatan password [13].

E. Prosedur Penelitian

Prosedur penelitian dirancang secara sistematis sehingga setiap tahap pengujian dapat dilakukan secara terukur dan dapat direplikasi. Tahapan penelitian dimulai dari implementasi algoritma pada SIMPEG, penyusunan dataset dan wordlist, pengujian performa sistem, pengujian keamanan menggunakan GPU, validasi dan pemrosesan data, analisis data, dan penyusunan output penelitian. Alur prosedur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Prosedur Penelitian

Tahapan penelitian dijelaskan sebagai berikut.

1) *Implementasi Algoritma pada SIMPEG*: Pada tahap ini, algoritma Bcrypt, Argon2, dan PBKDF2 diintegrasikan ke dalam modul registrasi dan login SIMPEG. Implementasi Bcrypt dan Argon2 menggunakan fungsi hashing password yang tersedia di PHP, PBKDF2 menggunakan fungsi hash_pbkdf2(). Semua algoritma dijalankan pada dataset password yang sama, sehingga perbandingan menjadi lebih objektif. Implementasi Bcrypt pada sistem berbasis web telah banyak digunakan karena mendukung mekanisme salt dan cost yang dapat meningkatkan keamanan penyimpanan password [14], [15].

2) *Penyusunan Dataset dan Wordlist*: Pada tahap ini, dataset password sebanyak 200 data disusun berdasarkan empat kategori kompleksitas. Selain dataset password, penelitian ini juga menggunakan wordlist sebanyak 5000 password sintetis untuk pengujian dictionary attack. Semua password plaintext dalam dataset pengujian ada dalam wordlist, tetapi posisinya diacak sehingga proses pengujian tidak berjalan sesuai urutan dataset. Wordlist dihasilkan secara sintetis untuk menjaga pengujian tetap terkendali dan tidak menggunakan password asli pengguna.

3) *Pengujian Performa Sistem*: Pengujian performa dilakukan untuk mengukur pengaruh masing-masing algoritma pada proses autentikasi sistem. Dua proses utama diuji, yaitu proses hashing saat registrasi dan proses verifikasi saat login. Semua password dalam dataset diproses menggunakan Bcrypt, Argon2, dan PBKDF2. Metrik yang dicatat adalah waktu untuk hashing, waktu untuk verifikasi, panjang hash, penggunaan CPU, penggunaan RAM. Evaluasi performa diperlukan, karena pemilihan algoritma hashing ditentukan tidak hanya oleh tingkat keamanan tetapi juga oleh kemampuan sistem untuk memberikan waktu respons yang dapat diterima oleh pengguna [16].

4) *Pengujian Keamanan Menggunakan GPU*: Pengujian keamanan dilakukan dengan simulasi dictionary attack menggunakan Hashcat v7.1.2 dengan GPU NVIDIA RTX 3060 6 GB. Pengujian ini bertujuan untuk mengevaluasi seberapa tahan hash yang dihasilkan oleh setiap algoritma terhadap serangan wordlist. Wordlist yang digunakan adalah 5000 password dan telah memuat seluruh plaintext password dari dataset pengujian dalam posisi acak. Parameter evaluasi yang dicatat adalah durasi cracking, jumlah hash yang berhasil dipulihkan, hash rate, dan tingkat keberhasilan cracking. Relevansi penggunaan GPU dalam

pengujian ini adalah bahwa komputasi paralel dapat mempercepat proses password cracking, dengan algoritma yang memiliki karakteristik memory-hard seperti Argon2 yang dirancang untuk meningkatkan biaya serangan tersebut [11].

5) *Validasi dan Pengolahan Data*: Validasi dilakukan dengan memeriksa konsistensi hasil pengujian dari setiap algoritma. Data yang dicatat mencakup hasil hashing, hasil verifikasi, penggunaan sumber daya, dan hasil dictionary attack. Jika ada nilai yang tidak wajar akibat gangguan proses sistem, data akan dianalisis ulang untuk memastikan tidak mempengaruhi hasil akhir. Selanjutnya, data diproses dengan menghitung nilai rata-rata dari setiap metrik yang diuji.

F. Teknik Analisis Data

Analisis data dilakukan menggunakan pendekatan kuantitatif komparatif. Data hasil pengujian dianalisis berdasarkan dua aspek utama, yaitu performa sistem dan ketahanan keamanan. Analisis aspek performa menggunakan waktu hashing, waktu verifikasi, panjang hash, penggunaan CPU, dan penggunaan RAM. Sementara itu, aspek keamanan dianalisis dengan durasi cracking, hash rate, jumlah hash yang berhasil diambil, dan tingkat keberhasilan pemecahan. Hasil dari setiap algoritma kemudian dibandingkan untuk menentukan algoritma hashing yang memiliki keseimbangan terbaik antara performa dan keamanan dalam SIMPEG berbasis web. Untuk memperjelas hubungan antara teknik analisis, variabel yang dianalisis, dan tujuan analisis, teknik analisis data yang digunakan dalam penelitian ini ditunjukkan pada Tabel 6.

TABEL 6

TEKNIK ANALISIS DATA

Teknik Analisis Data	Variabel yang Dianalisis	Tujuan Analisis
Perbandingan numerik	Waktu hashing, waktu verifikasi, Panjang hash	Menilai efisiensi algoritma
Analisis penggunaan sumber daya	CPU dan RAM	Menilai beban komputasi sistem
Analisis ketahanan keamanan	Waktu cracking, hash rate, jumlah hash yang berhasil dipulihkan	Menilai ketahanan terhadap dictionary attack
Interpretasi komparatif	Performa dan keamanan setiap algoritma	Menentukan algoritma paling sesuai untuk SIMPEG

Berdasarkan teknik analisis yang diterapkan, hasil uji dari setiap algoritma dibandingkan secara sistematis untuk mendapatkan gambaran perbedaan dalam performa dan ketahanan keamanan. Algoritma dengan waktu hashing dan verifikasi yang dapat diterima, penggunaan sumber daya yang wajar, dan ketahanan terhadap cracking yang lebih tinggi lebih cocok untuk diterapkan dalam SIMPEG berbasis web. Hasil analisis kemudian digunakan sebagai dasar untuk menyiapkan rekomendasi algoritma hashing password yang paling sesuai untuk memenuhi kebutuhan sistem informasi kepegawaian.

G. Output Penelitian

Hasil penelitian ini adalah perbandingan performa dan keamanan algoritma Bcrypt, Argon2, dan PBKDF2 pada SIMPEG berbasis web. Output penelitian meliputi tabel performa hashing dan verifikasi, tabel penggunaan sumber daya, grafik perbandingan waktu hashing, grafik hash rate, tabel hasil dictionary attack, serta rekomendasi algoritma hashing yang sesuai untuk SIMPEG.

III. HASIL DAN PEMBAHASAN

A. Hasil Implementasi Algoritma Hashing pada SIMPEG

Hasil implementasi menunjukkan bahwa algoritma Bcrypt, Argon2, dan PBKDF2 berhasil diterapkan pada modul autentikasi SIMPEG berbasis web, terutama dalam proses registrasi dan login. Sistem mengubah password plaintext sintetis menjadi nilai hash sesuai dengan algoritma yang digunakan dan memverifikasi password yang dimasukkan pengguna serta hash yang disimpan di database sepanjang proses login. Semua data uji sebanyak 200 password berhasil diproses dengan status sukses pada ketiga algoritma. Ringkasan hasil implementasi algoritma hashing ditunjukkan pada Tabel 7.

TABEL 7
HASIL IMPLEMENTASI HASHING PADA SISTEM

Algoritma	Proses	Jumlah Data	Status	Panjang Hash	Salt	Parameter	Keterangan
Bcrypt	Register dan Login	200	Sukses	60 karakter	Otomatis	cost = 12	Hash tetap 60 karakter dan mendukung cost factor
Argon2	Register dan Login	200	Sukses	98 karakter	otomatis	memory = 262144 KB, time cost = 4, parallelism = 1	Memory-hard dan lebih tahan terhadap serangan berbasis GPU
PBKDF2	Register dan Login	200	Sukses	83 karakter	16 byte random salt	Iteration = 260000	Kompatibel luas dan berbasis iterasi

Tabel 7 menunjukkan bahwa ketiga algoritma menghasilkan format hash yang berbeda. Bcrypt menghasilkan hash dengan panjang tetap 60 karakter, Argon2 menghasilkan hash sepanjang 98 karakter, dan PBKDF2 menghasilkan hash sepanjang 83 karakter. Perbedaannya disebabkan oleh format penyimpanan parameter yang berbeda dari masing-masing algoritma. Bcrypt menyediakan informasi tentang cost factor, Argon2 menyertakan parameter memory, time cost, dan parallelism, serta PBKDF2 menyediakan informasi tentang salt dan jumlah iterasi. Hasilnya menunjukkan bahwa implementasi ketiga algoritma tersebut berjalan sesuai dengan karakteristik teknis masing-masing algoritma.

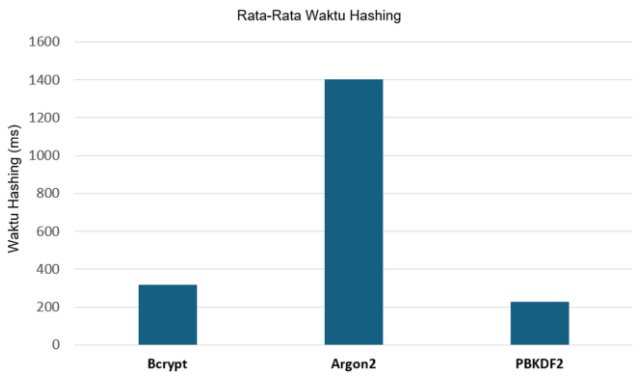
B. Hasil Pengujian Waktu Hashing

Pengujian waktu hashing adalah untuk mengetahui berapa lama setiap algoritma memerlukan waktu untuk menghasilkan hash password saat pengguna registrasi. Pengujian dilakukan pada 200 password sintetis dengan empat tingkat kompleksitas, yaitu lemah, sedang, kuat, dan sangat kuat. Hasil pengujian waktu hashing ditunjukkan pada Tabel 8.

TABEL 8
HASIL PENGUJIAN WAKTU HASHING

Algoritma	Jumlah Data	Rata-Rata (ms)	Minimum (ms)	Maksimum (ms)
Bcrypt	200	317,610	246,624	506,895
Argon2	200	1403,172	911,567	1776,586
PBKDF2	200	226,007	217,848	617,706

Berdasarkan Tabel 8, PBKDF2 memiliki rata-rata waktu hashing terendah, yaitu 226,007 ms, diikuti oleh Bcrypt dengan rata-rata waktu hashing 317,610 ms dan Argon2 memiliki rata-rata waktu hashing tertinggi yaitu 1403,172 ms. Perbedaan ini menunjukkan bahwa setiap algoritma memiliki beban komputasi yang berbeda. PBKDF2 lebih ringan dalam hal waktu hashing, sementara Argon2 memerlukan waktu lebih lama karena memiliki karakteristik memory-hard yang dirancang untuk meningkatkan kebutuhan sumber daya selama proses hashing [5], [12]. Untuk memperjelas perbandingan waktu hashing antar algoritma, hasil pengujian divisualisasikan dengan grafik pada Gambar 2.



Gambar 2. Grafik Perbandingan Waktu Hashing

Gambar 2 menunjukkan bahwa Argon2 memiliki waktu hashing yang jauh lebih tinggi dibandingkan Bcrypt dan PBKDF2. Meskipun demikian, waktu hashing yang lebih tinggi pada Argon2 tidak selalu menjadi kelemahan. Dalam konteks password hashing, peningkatan waktu pemrosesan dapat menjadi bagian dari mekanisme pertahanan karena akan membuat proses penebakan password menjadi lebih mahal secara komputasi. Di sisi lain, PBKDF2 memiliki performa yang paling ringan, tetapi tingkat keamanannya sangat bergantung pada jumlah iterasi dan penggunaan salt yang kuat [13], [17].

C. Hasil Pengujian Waktu Verifikasi Login

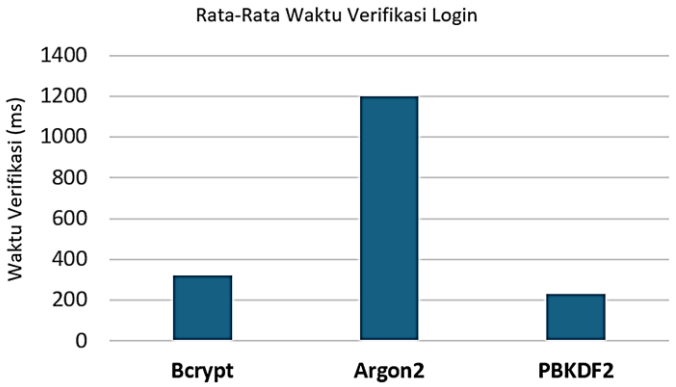
Selain proses hashing selama registrasi, studi ini juga menguji waktu untuk memverifikasi password saat login. Waktu verifikasi menjadi penting karena proses ini langsung terkait dengan pengalaman pengguna dalam menggunakan SIMPEG. Hasil pengujian waktu verifikasi login ditunjukkan pada Tabel 9.

TABEL 9

HASIL PENGUJIAN WAKTU VERIFIKASI LOGIN

Algoritma	Jumlah Data	Rata-Rata (ms)	Minimum (ms)	Maksimum (ms)
Bcrypt	200	320,693	246,455	549,681
Argon2	200	1198,050	590,547	1829,009
PBKDF2	200	228,536	216,064	601,313

Tabel 9 menunjukkan bahwa PBKDF2 memiliki rata-rata waktu verifikasi terendah, yaitu 228,536 ms. Bcrypt berada pada posisi menengah dengan rata-rata 320,693 ms, sedangkan Argon2 memiliki rata-rata waktu verifikasi tertinggi sebesar 1198,050 ms. Hasilnya menunjukkan bahwa Argon2 memiliki beban verifikasi yang lebih tinggi dibandingkan dengan dua algoritma lainnya. Namun, dalam kasus sistem yang mengutamakan keamanan password, peningkatan waktu verifikasi masih dapat ditoleransi selama tidak menghasilkan latensi yang terlalu besar bagi pengguna. Perbandingan waktu verifikasi login pada ketiga algoritma ditunjukkan dengan grafik pada Gambar 3.



Gambar 3. Grafik Perbandingan Waktu Verifikasi Login

Gambar 3 menunjukkan pola yang sama dengan hasil waktu hashing. Argon2 adalah algoritma dengan waktu verifikasi tertinggi, sedangkan PBKDF2 adalah algoritma dengan waktu verifikasi terendah. Hal ini menunjukkan bahwa Argon2 memiliki beban pemrosesan yang lebih besar, tetapi hal tersebut berkaitan dengan desain keamanan yang lebih kuat terhadap serangan berbasis perangkat keras modern [10], [11].

D. Hasil Penggunaan Sumber Daya Sistem

Pengujian penggunaan sumber daya dilakukan untuk mengetahui dampak masing-masing algoritma terhadap CPU dan RAM pada proses registrasi dan login. Hasil pengujian penggunaan sumber daya ditunjukkan pada Tabel 10.

TABEL 10
HASIL PENGGUNAAN SUMBER DAYA SISTEM

Algoritma	Rata-Rata CPU Register (%)	Rata-Rata CPU Login (%)	Rata-Rata RAM Register (MB)	Rata-Rata RAM Login (MB)
Bcrypt	6,706	6,854	2,661	2,022
Argon2	7,414	7,915	2,346	1,965
PBKDF2	6,339	6,801	2,289	1,917

Berdasarkan Tabel 10, penggunaan CPU tertinggi adalah Argon2 baik dalam proses registrasi maupun login. Argon2 menggunakan CPU rata-rata 7,414% selama proses registrasi dan 7,915% selama proses login. Bcrypt dan PBKDF2 memiliki penggunaan yang lebih rendah terhadap CPU. Di sisi RAM, seluruh algoritma masih dalam konsumsi yang relatif rendah di lingkungan pengujian lokal. Namun, ketika sistem diterapkan di server produksi di mana jumlah pengguna bersamaan yang lebih besar, perbedaan dalam penggunaan sumber daya masih harus dipertimbangkan. Hasil ini menunjukkan bahwa Argon2 secara beban komputasi lebih tinggi dibandingkan Bcrypt dan PBKDF2. Kondisi tersebut sesuai dengan karakteristik Argon2 sebagai algoritma yang memerlukan banyak memori. Penggunaan Argon2 pada sistem nyata harus disesuaikan dengan kapasitas server, jumlah pengguna aktif, dan kebutuhan keamanan sistem. Bcrypt mungkin menjadi alternatif yang lebih seimbang ketika sistem memerlukan performa yang stabil, sedangkan PBKDF2 dapat digunakan untuk sistem yang memerlukan kompatibilitas lintas platform dengan konfigurasi iterasi tinggi [11], [13], [16].

E. Perbandingan Performa Bcrypt, Argon2, dan PBKDF2

Untuk memperoleh gambaran menyeluruh, hasil pengujian hashing, verifikasi, CPU, dan RAM diringkas dalam Tabel 11.

TABEL 11
PERBANDINGAN PERFORMA BCRYPT, ARGON2, DAN PBKDF2

Algoritma	Rata-Rata Hashing (ms)	Rata-Rata Verifikasi (ms)	Rata-Rata CPU (%)	Rata-Rata RAM (MB)	Keterangan
Bcrypt	317,610	320,693	6,780	2,341	Performa stabil dan masih efisien, cocok sebagai alternatif yang seimbang
Argon2	1403,172	1198,050	7,665	2,155	Paling berat dari sisi waktu proses, tetapi lebih kuat karena karakteristik *memory-hard*
PBKDF2	226,007	228,536	6,570	2,103	Paling ringan dari sisi waktu, tetapi keamanannya sangat bergantung pada jumlah iterasi dan salt

Berdasarkan Tabel 11, PBKDF2 memiliki performa paling ringan dalam hal waktu hashing dan verifikasi. Bcrypt berada di posisi tengah dengan performa yang stabil, sementara Argon2 memiliki waktu pemrosesan tertinggi. Namun, pada konteks password hashing, algoritma tercepat tidak selalu menjadi pilihan paling aman. Dalam skenario cracking, jika parameter yang digunakan tidak cukup kuat, algoritma yang terlalu ringan dapat lebih mudah diproses. Oleh karena itu, evaluasi performa perlu dibaca bersama dengan hasil pengujian keamanan menggunakan Hashcat

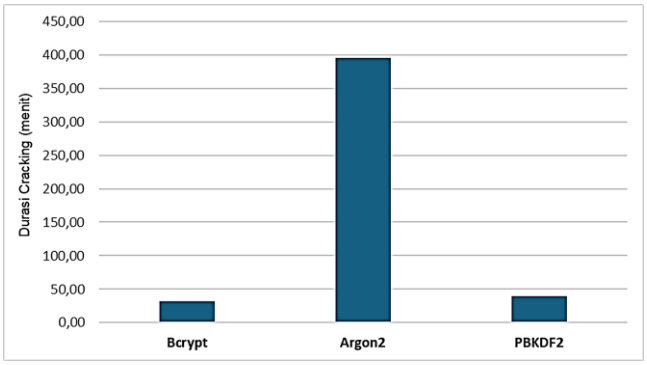
F. Hasil Simulasi Dictionary Attack Menggunakan Hashcat

Pengujian keamanan dilakukan melalui simulasi dictionary attack menggunakan Hashcat dengan dukungan GPU NVIDIA RTX 3060 6 GB. Pengujian dilakukan pada 200 hash untuk setiap algoritma menggunakan wordlist sintetis sebanyak 5000 password. Wordlist terdiri dari semua password plaintext dalam dataset pengujian dalam urutan acak. Tujuan dari pengujian ini adalah untuk memperkirakan ketahanan relatif masing-masing algoritma terhadap peretasan berbasis GPU menggunakan indikator durasi cracking, hash rate, dan jumlah hash yang berhasil dipulihkan. Penggunaan salt dan mekanisme penguatan password masih merupakan aspek penting dalam penyimpanan password karena dapat meningkatkan kesulitan proses penebakan password [17]. Ringkasan hasil simulasi dictionary attack ditunjukkan pada Tabel 12.

TABEL 12
HASIL SIMULASI DICTIONARY ATTACK MENGGUNAKAN HASHCAT

Algoritma	Mode Hashcat	Jumlah Hash Diuji	Wordlist	Hash Berhasil Dipulihkan	Tingkat Keberhasilan	Hash Rate	Durasi Cracking
Bcrypt	3200	200	5000	200	100%	317 H/s	31 menit 36 detik
Argon2	34000	200	5000	200	100%	19 H/s	6 jam 35 menit
PBKDF2	10900	200	5000	200	100%	465 H/s	39 menit 42 detik

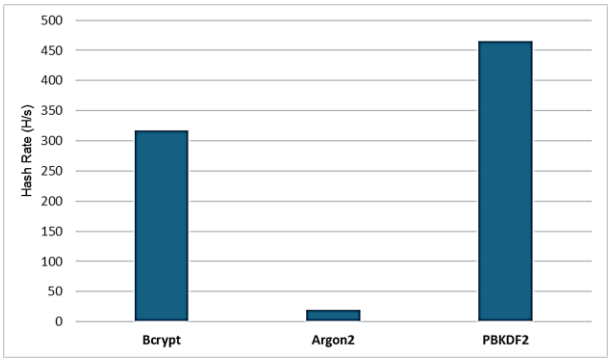
Berdasarkan Tabel 12, ketiga algoritma memiliki tingkat keberhasilan cracking sebesar 100% karena seluruh hash dari setiap algoritma berhasil dipulihkan. Hasil ini terjadi karena wordlist yang digunakan sudah mengandung semua password plaintext dari dataset pengujian. Dengan demikian, subjek analisis keamanan di sini bukanlah keberhasilan atau kegagalan hash untuk dipulihkan, melainkan ukuran biaya komputasi yang diperlukan untuk melakukan proses cracking. Dalam hal ini, durasi cracking dan hash rate mewakili biaya komputasi. Untuk memperjelas perbedaan durasi cracking pada masing-masing algoritma, hasil pengujian divisualisasikan dengan grafik pada Gambar 4.



Gambar 4. Grafik Perbandingan Durasi Cracking

Gambar 4 menunjukkan bahwa Argon2 membutuhkan durasi cracking paling lama, yaitu 6 jam 35 menit. Sementara itu, Bcrypt membutuhkan waktu 31 menit 36 detik dan PBKDF2 membutuhkan waktu 39 menit 42 detik. Durasi cracking yang jauh lebih lama dari Argon2 menunjukkan bahwa algoritma ini menggunakan lebih banyak biaya komputasi ketika dihitung oleh Hashcat dalam skenario dictionary attack berbasis GPU. Oleh karena itu, dalam hal waktu yang dibutuhkan untuk proses cracking, Argon2 adalah yang paling tahan dari ketiga algoritma yang diuji.

Selain waktu cracking, ukuran lain yang penting adalah hash rate. Hash rate mewakili kecepatan Hashcat dalam memproses hash untuk algoritma tertentu. Perbandingan hash rate pada ketiga algoritma ditunjukkan pada Gambar 5.



Gambar 5. Grafik Perbandingan Hash Rate

Berdasarkan Gambar 5, PBKDF2 memiliki hash rate tertinggi, yaitu 465 H/s, dan Bcrypt 317 H/s, sedangkan Argon2 memiliki hash rate terendah, yaitu 19 H/s. Hash rate yang lebih rendah menunjukkan bahwa proses pencocokan hash lebih lambat, sehingga serangan dictionary attack menjadi kurang efisien. Nilai hash rate yang rendah pada Argon2 menandakan bahwa algoritma ini lebih sulit diproses dalam skenario cracking berbasis GPU

dibandingkan dengan Bcrypt dan PBKDF2. Temuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa serangan brute force dan dictionary attack dapat menjadi ancaman serius terhadap password hash, terutama ketika penyerang memanfaatkan pemrosesan dan perangkat komputasi modern saat ini [18]. Kekuatan sebuah password juga harus diuji dengan mempertimbangkan hubungan antara proses hashing dan cracking, karena setiap algoritma memiliki biaya komputasi yang berbeda [19].

G. Analisis Keunggulan Argon2

Keunggulan Argon2 dalam penelitian ini adalah karena memiliki karakteristik memory-hard. Berbeda dengan PBKDF2 yang bergantung pada iterasi dan Bcrypt dengan cost factor, Argon2 membuat proses hashing lebih memakan banyak memori. Ini membuat komputasi paralel kurang efisien dalam skenario serangan berbasis GPU, karena penyerang tidak hanya membutuhkan kemampuan komputasi yang tinggi tetapi juga alokasi memori yang lebih besar. Ini menjelaskan mengapa Argon2 menghasilkan hash rate terendah dan durasi pemecahan terlama. Hasil ini sejalan dengan penelitian sebelumnya yang mengidentifikasi Argon2 sebagai solusi keamanan password yang kuat [20].

Bcrypt masih menunjukkan performa yang seimbang karena mekanisme cost factor untuk meningkatkan beban komputasi. Dalam penelitian ini, Bcrypt memiliki waktu hashing dan verifikasi yang lebih rendah dibandingkan Argon2, namun masih menunjukkan ketahanan yang lebih baik daripada PBKDF2 dalam hash rate. Penggunaan Bcrypt dalam aplikasi pengguna dan sistem pegawai juga telah digunakan untuk meningkatkan keamanan penyimpanan password [21], [22]. Selain itu, risiko kebocoran kredensial tetap menjadi ancaman signifikan bagi sistem autentikasi, dan keamanan password tidak hanya bergantung pada satu mekanisme teknis saja [23]. Bcrypt lebih cocok untuk penyimpanan password daripada algoritma hash lama seperti MD5 dan SHA-1, karena dapat meningkatkan beban komputasi selama proses cracking password [24]. Sementara itu, penerapan Argon2 dalam sistem keamanan data berbasis web memperkuat relevansi algoritma ini untuk perlindungan data sensitif [25].

H. Perbandingan Hasil Penelitian dengan Studi Terdahulu

Hasil penelitian ini dibandingkan dengan penelitian terdahulu untuk menentukan kesesuaian hasil penelitian ini dengan penelitian sebelumnya yang membahas peretasan password, hashing password, Bcrypt, Argon2, PBKDF2, dan keamanan kredensial. Perbandingan ini diperlukan untuk menunjukkan bahwa hasil pengujian SIMPEG berbasis web didasarkan pada bukti empiris dan relevan dengan penelitian yang ada. Ringkasan perbandingan hasil penelitian ini dengan studi terdahulu ditunjukkan pada Tabel 13.

TABEL 13
PERBANDINGAN HASIL PENELITIAN DENGAN STUDI TERDAHULU

Judul Penelitian	Fokus Penelitian	Hasil	Kesesuaian dengan penelitian ini
an Effective Mechanism for Securing and Managing Password Using Aes-256 Encryption & Pbkdf2 [17]	Perlindungan password menggunakan salt dan mekanisme penguatan keamanan	Pengamanan password membutuhkan mekanisme salt dan penguatan proses penyimpanan	Sejalan, PBKDF2 pada penelitian ini digunakan dengan salt dan iterasi tinggi
Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming [18]	Brute force dan dictionary attack	Serangan password dapat dipercepat melalui pemrosesan paralel	Sejalan, penelitian ini menggunakan Hashcat dan GPU untuk simulasi dictionary attack
PassCrack: Cracking, Hashing, and Strength Testing for a Secure Digital Future [19]	Cracking, hashing, dan pengujian kekuatan password	Pengujian kekuatan password perlu mempertimbangkan proses hashing dan cracking	Sejalan, penelitian ini mengukur performa hashing dan ketahanan cracking
Securing Passwords: An Approach Inculcating Argon2 and Three-Fish Algorithm [20]	Pengamanan password menggunakan Argon2	Argon2 dapat digunakan untuk memperkuat keamanan password	Sejalan, Argon2 menunjukkan ketahanan tertinggi pada pengujian Hashcat
Implementasi Bcrypt dengan SHA-256 pada Password Pengguna Aplikasi Golek Kost [21]	Implementasi Bcrypt pada password pengguna	Bcrypt dapat digunakan untuk meningkatkan keamanan password aplikasi	Sejalan, Bcrypt berhasil diterapkan pada modul autentikasi SIMPEG
Penerapan Algoritma Bcrypt untuk Enkripsi Password pada Aplikasi Absensi Pegawai Menggunakan QR Code pada CV RSA Mandiri [22]	Bcrypt pada aplikasi absensi pegawai	Bcrypt relevan untuk sistem yang berkaitan dengan data pegawai	Sejalan, konteksnya dekat dengan SIMPEG sebagai sistem informasi kepegawaian

Might I Get Pwned: A Second Generation Compromised Credential Checking Service [23]	Pemeriksaan kredensial bocor	Kredensial bocor menjadi ancaman serius bagi sistem autentikasi	Mendukung pentingnya pengamanan password pada SIMPEG
Evaluasi Keamanan Penyimpanan Password Menggunakan Algoritma Hash MD5, SHA-1, dan Bcrypt [24]	Perbandingan hash lama dan Bcrypt	Bcrypt lebih sesuai untuk penyimpanan password dibandingkan MD5 dan SHA-1	Sejalan, Bcrypt pada penelitian ini digunakan sebagai algoritma hashing adaptif
Pengembangan Sistem Keamanan Data Berbasis Web Menggunakan Kombinasi Algoritma ChaCha20-Poly1305 dan Argon2 [25]	Pengamanan data berbasis web menggunakan Argon2	Argon2 relevan untuk penguatan keamanan data pada sistem berbasis web	Mendukung hasil penelitian ini bahwa Argon2 layak digunakan pada sistem web yang membutuhkan keamanan tinggi
Penelitian Ini	Perbandingan Bcrypt, Argon2, dan PBKDF2 pada SIMPEG berbasis web	Argon2 paling tahan terhadap dictionary attack berbasis GPU	Memberikan evaluasi langsung pada konteks SIMPEG berbasis web

Berdasarkan Tabel 13, hasil penelitian ini sejalan dengan beberapa studi terdahulu. Penggunaan Hashcat dan GPU konsisten dengan penelitian yang menunjukkan bahwa dictionary attack dapat dipercepat melalui pemrosesan paralel. Hasil pengujian juga mendukung penelitian tentang Argon2, yang menunjukkan bahwa algoritma tersebut memiliki keunggulan dalam keamanan password. Selain itu, hasil implementasi Bcrypt dan PBKDF2 dalam penelitian ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya cost factor, salt, dan iterasi dalam memperkuat penyimpanan password. Dengan demikian, penelitian ini memperkuat temuan terdahulu serta memberikan kontribusi tambahan melalui evaluasi langsung dalam konteks SIMPEG berbasis web.

I. Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan. Pertama, dataset password yang digunakan adalah password sintetis sebanyak 200 data, sehingga belum sepenuhnya mencerminkan seluruh variasi password pengguna nyata. Kedua, simulasi serangan hanya menggunakan dictionary attack dengan wordlist sebanyak 5000 password. Wordlist tersebut memang berisi semua password plaintext dari dataset untuk memungkinkan pengujian yang terkontrol, tetapi tidak mencakup skenario serangan lain seperti brute force, hybrid attack, atau menggunakan wordlist skala besar seperti RockYou. Ketiga, pengujian dilakukan di lingkungan lokal dengan spesifikasi perangkat tertentu. Dengan demikian, hasil performa mungkin bervariasi ketika diterapkan pada server produksi dengan konfigurasi perangkat keras yang berbeda, jumlah pengguna aktif, dan beban layanan yang berbeda. Keempat, penelitian ini berfokus pada perlindungan password melalui hashing, dan belum menguji kerentanan lain seperti SQL injection, mask attack maupun ancaman jaringan seperti DDoS. Serangan DDoS juga harus dipertimbangkan karena dapat mempengaruhi ketersediaan layanan sistem berbasis web [26].

J. Rekomendasi Teknis

Berdasarkan hasil pengujian, disarankan untuk menggunakan Argon2 dalam sistem informasi yang menyimpan data sensitif seperti SIMPEG, karena memiliki ketahanan yang lebih tinggi terhadap dictionary attack berbasis GPU. Argon2 memerlukan waktu yang lebih lama untuk melakukan hashing dan verifikasi, namun peningkatan waktu ini memberikan manfaat keamanan dengan membuat proses cracking menjadi lebih mahal secara komputasi. Oleh karena itu, Argon2 lebih cocok untuk sistem yang memprioritaskan keamanan password dan memiliki sumber daya server yang cukup.

Bcrypt dapat digunakan sebagai alternatif jika sistem membutuhkan keseimbangan antara keamanan dan performa. Dengan pengaturan cost yang tepat, Bcrypt tetap dapat memberikan perlindungan yang baik terhadap serangan password cracking, dan menghasilkan waktu respons yang lebih ringan dari Argon2. Namun, PBKDF2 masih dapat diterapkan untuk sistem yang memerlukan kompatibilitas lintas platform. Tetapi harus dikonfigurasi dengan jumlah iterasi yang tinggi dan salt acak untuk menghindari pemrosesan yang mudah dalam skenario dictionary attack [17].

Dalam implementasi, pemilihan algoritma hashing di instansi pemerintah atau lembaga organisasi tidak hanya didasarkan pada kecepatan, tetapi juga pada tingkat sensitivitas data, kapasitas server, jumlah pengguna aktif, dan kemungkinan ancaman keamanan. Sistem harus menggunakan kebijakan password yang kuat, salt unik, pembaruan parameter hashing secara berkala, dan pemantauan keamanan autentikasi selain pemilihan algoritma. Perlindungan password harus dikombinasikan dengan kebijakan keamanan lainnya seperti pembatasan upaya login, deteksi aktivitas mencurigakan, dan evaluasi keamanan akun secara berkala [23]. Risiko kebocoran kredensial masih tetap menjadi

ancaman yang signifikan. Selain memperkuat autentikasi, instansi juga perlu memperhatikan ancaman di lapisan aplikasi dan jaringan, termasuk serangan DDoS yang dapat mengganggu ketersediaan layanan sistem informasi [26].

IV. SIMPULAN

Berdasarkan hasil pengujian, algoritma Bcrypt, Argon2, dan PBKDF2 berhasil diimplementasikan pada modul autentikasi SIMPEG berbasis web untuk proses registrasi dan login. Hasil pengujian performa menunjukkan bahwa PBKDF2 memiliki waktu hashing dan verifikasi terendah yaitu masing-masing 226,007 ms dan 228,536 ms, diikuti oleh Bcrypt dengan waktu hashing 317,610 ms dan verifikasi 320,693 ms, serta Argon2 dengan waktu pemrosesan tertinggi yaitu waktu hashing 1403,172 ms dan verifikasi 1198,050 ms. Namun, hasil simulasi dictionary attack dengan Hashcat menunjukkan ketahanan terbaik pada Argon2, karena memerlukan waktu terlama untuk proses cracking, yaitu 6 jam dan 35 menit, dan juga memiliki hash rate terendah sebesar 19 H/s, dibandingkan dengan Bcrypt pada 317 H/s dan PBKDF2 pada 465 H/s. Oleh karena itu, Argon2 direkomendasikan untuk SIMPEG atau sistem informasi dengan penekanan pada keamanan password dan sumber daya server yang memadai, sementara Bcrypt lebih merupakan kompromi antara keamanan dan performa, dan PBKDF2 tetap relevan untuk kebutuhan kompatibilitas dengan konfigurasi iterasi dan salt yang kuat. Penelitian ini terbatas pada dataset password sintetis, wordlist sebanyak 5.000 password, lingkungan pengujian lokal, dan skenario dictionary attack, sehingga penelitian selanjutnya dapat menggunakan dataset password yang lebih besar, wordlist yang lebih besar, skenario serangan lain seperti brute force, hybrid attack atau mask attack, dan pengujian pada server produksi dengan jumlah pengguna bersamaan yang lebih bervariasi.

DAFTAR PUSTAKA

[1] M. A. Al Hilmi and R. K. Yunan, "Pengujian Keamanan Fitur Upload File pada Sistem Aplikasi Web," *J. Inform. J. Pengemb. IT*, vol. 7, no. 1, pp. 37–42, 2022, doi: 10.30591/jpit.v7i1.3336.

[2] S. Erdi, P. Sohidin, H. Prasetyo Utomo, and A. Ahmadi, "Penerapan Algoritma Bcrypt untuk Pengamanan Password pada Sistem Informasi Akademik (SIAK) (Studi Kasus : Universitas Langlangbuana)," *J. Infosecure*, vol. 6, no. 2, pp. 1–5, 2025.

[3] M. Adri Ramadhan, D. Saputra, D. Iskandar Mulyana, S. Tinggi Ilmu Komputer Cipta Karya Informatika, and D. Jakarta, "Pencegahan Serangan Berbasis Kata Sandi: Studi Komprehensif Tentang Implementasi Hash Pada Aplikasi Web Prevention of Password-Based Attacks: a Comprehensive Study of Hash Implementation in Web Applications," *J. Inf. Technol. Comput. Sci.*, vol. 7, no. 3, pp. 920–925, 2024.

[4] S. C., R. J., and R. R., "Accelerated and Intelligent Password Cracking with Performance Optimization," *Proc. 1st Int. Conf. Res. Dev. Information, Commun. Comput. Technol. (ICRDICCT 2025)*, vol. 4, pp. 741–749, 2025, doi: 10.5220/0013920100004919.

[5] V. Fedorchenko, O. Yeroshenko, O. Shmatko, O. Kolomiitsev, and M. Omarov, "Password Hashing Methods and Algorithms on the .NET Platform," *Adv. Inf. Syst.*, vol. 8, no. 4, pp. 82–92, 2024, doi: 10.20998/2522-9052.2024.4.11.

[6] R. M. Liauren, B. Zaman, and S. Bahri, "Implementasi Algoritma Aes Dan Bcrypt Untuk Pengamanan Data Pengguna Pada Website Jahitku," *KHARISMA Tech*, vol. 20, no. 1, pp. 57–71, 2025, doi: 10.55645/kharismatech.v20i1.535.

[7] N. A. Y. -, D. Kiswanto, S. Davina, and A. D. Sitepu, "Simulasi Hashing Password Menggunakan Argon2 Dan Scrypt Serta Pengembangan Fitur Logging Jaringan Real-Time Berbasis Website," *J. Inform. dan Tek. Elektro Terap.*, vol. 14, no. 1, pp. 8–17, 2026, doi: 10.23960/jitet.v14i1.8183.

[8] T. P. Batubara, S. Efendi, and E. B. Nababan, "Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force," *J. Phys. Conf. Ser.*, vol. 1811, no. 1, p. 012129, 2021, doi: 10.1088/1742-6596/1811/1/012129.

[9] N. A. A. Mustafa, "Analysis attackers' methods with hashing secure password using CSPRNG and PBKDF2," *Wasit J. Eng. Sci.*, vol. 12, no. 2, pp. 60–70, 2024, doi: 10.31185/ejuow.vol12.iss2.502.

[10] P. Tippe and M. P. Berner, "Evaluating Argon2 Adoption and Effectiveness in Real-World Software," *Lect. Notes Comput. Sci.*, vol. 15993 LNCS, pp. 25–46, 2025, doi: 10.1007/978-3-032-00627-1_2.

[11] S. Eum, H. Kim, M. Song, and H. Seo, "Optimized Implementation of Argon2 Utilizing the Graphics Processing Unit," *Appl. Sci.*, vol. 13, no. 16, 2023, doi: 10.3390/app13169295.

[12] I. Listiawan, Z. Zaidir, S. Winardi, and M. Diqi, "Optimising Bcrypt Parameters: Finding the Optimal Number of Rounds for Enhanced Security and Performance," *Compiler*, vol. 13, no. 1, pp. 1–10, 2024, doi: 10.28989/compiler.v13i1.2111.

[13] A. A. S. AlQahtani, "Key Derivation: A Dynamic PBKDF2 Model for Modern Cryptographic Systems," *Cryptography*, vol. 9, no. 2, p. 39, 2025, doi: 10.3390/cryptography9020039.

[14] D. Febrian *et al.*, "Implementation of Bcrypt Algorithm on Website-Based Hashing Generator Using Laravel Framework," *J. Inf. Syst. Informatics Comput.*, vol. 7, no. 2, p. 199, 2023, doi: 10.52362/jisicom.v7i2.1130.

[15] K. Nur, D. Suhartono, M. Thoriq, and A. Qothrunnada, "Implementasi Pengamanan Data Menggunakan Teknik Bcrypt Hashing Password dan Algoritma Advanced Encryption Standard (AES) Implementation of Data Security Using Bcrypt Hashing Password Technique and Advanced Encryption Standard (AES) Algorithm," *J. Sist. dan Teknol. Inf.*, vol. 13, no. 1, pp. 101–108, 2025, doi: 10.26418/justin.v13i1.84997.

[16] R. Patra and S. Patra, "Cryptography: A Quantitative Analysis of the Effectiveness of Various Password Storage Techniques," *J. Student Res.*, vol. 10, no. 3, pp. 1–14, 2021, doi: 10.47611/jsrhs.v10i3.1764.

[17] R. Khande, S. Ramaswami, C. Naidu, and N. Patel, "An Effective Mechanism for Securing and Managing Password Using Aes-256 Encryption & Pbkdf2," *Int. J. Electr. Eng. Technol.*, vol. 12, no. 5, pp. 1–7, 2021, doi: 10.34218/ijeet.12.5.2021.001.

[18] I. Alkhwaja *et al.*, "Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming," *Appl. Sci.*, vol. 13, no. 10, p. 5979, 2023, doi: 10.3390/app13105979.

[19] P. Bagane, M. Sable, A. Panicker, A. Ansh, and O. A. Jebessa, "Passcrack: Cracking, hashing, and strength testing for a secure digital future," *Int. J. Smart Sens. Intell. Syst.*, vol. 18, no. 1, pp. 1–18, 2025, doi: 10.2478/ijssis-2025-0024.

[20] N. R. Rajeswari, R. Buduri, K. Santhi, B. S. Kumar, D. K. S. Rao, and D. K. Pulluru, "Securing Passwords: An Approach Inculcating Argon2 and Three-Fish Algorithm," *Int. J. Eng. Adv. Technol.*, vol. 14, no. 4, pp. 30–35, 2025, doi: 10.35940/ijeat.d4582.14040425.

[21] R. S. Giffary and E. Ramadhani, "Implementasi Bcrypt dengan SHA-256 pada Password Pengguna Aplikasi Golek Kost," *J. Sist. Komput.*

dan Inform., vol. 3, no. 4, p. 543, 2022, doi: 10.30865/json.v3i4.4285.
[22] D. S. Rachmad, "Penerapan Algoritma Bcrypt untuk Enkripsi Password pada Aplikasi Absensi Pegawai Menggunakan QR Code pada CV RSA Mandiri," *INOMATEC J. Inov. dan Kaji. Multidisipliner Kontemporer*, vol. 01, no. 04, pp. 665–672, 2026, doi: 10.70294/ino1087.
[23] B. Pal *et al.*, "Might I Get Pwned: A Second Generation Compromised Credential Checking Service," *Proc. 31st USENIX Secur. Symp. Secur. 2022*, pp. 1831–1848, 2022.
[24] R. Dafi Al Azhar and I. Sholihah Widiati, "Evaluasi Keamanan Penyimpanan Password Menggunakan Algoritma Hash: MD5, SHA-1, dan Bcrypt," *Pros. Semin. Nas. Teknol. Inf. dan Bisnis*, pp. 1302–1305, 2025, doi: 10.47701/q885nj69.
[25] N. T. Jehian *et al.*, "Pengembangan Sistem Keamanan Data Berbasis Web Menggunakan Kombinasi Algoritma ChaCha20-Poly1305 dan Argon2," *JITET (Jurnal Inform. dan Tek. Elektro Terapan)*, vol. 13, no. 3S1, pp. 1958–1968, 2025, doi: 10.23960/jitet.v13i3S1.8151.
[26] M. Q. Syahputra, D. R. Akbi, and D. Risqiwati, "Deteksi Dan Mitigasi Serangan DDoS Pada Software Defined Network Menggunakan Algoritma Decision Tree," *J. Repos.*, vol. 2, no. 11, p. 1491, 2020, doi: 10.22219/repositor.v2i11.795.