ANALISIS KOMPARATIF ALGORITMA PASSWORD HASHING: STUDI LITERATUR PADA BCRYPT

Eka Jiparolim (20230801078)

Program Studi Teknik Informatika

Universitas Esa Unggul

Email: ekajiparolim123@student.esaunggul.ac.id

I. Pendahuluan

Fenomena kebocoran data di dunia siber telah menjadi kekhawatiran serius. Semakin maraknya kasus pencurian data, peretasan sistem, dan pelanggaran privasi menunjukkan adanya kerentanan kritis dalam sistem informasi modern. Salah satu titik rawan paling signifikan terletak pada cara sistem menyimpan kredensial pengguna, terutama password (kata sandi).

Secara historis, fungsi hash kriptografis seperti MD5 dan SHA-1 digunakan untuk mengamankan password ini. Namun, algoritma tersebut tidak pernah dirancang untuk skenario ini (Eastlake & Jones, 2001; Rivest, 1992). Kelemahan fundamental mereka adalah kecepatan komputasi. Dengan hardware modern seperti GPU, kecepatan ini justru menjadi bumerang, memungkinkan serangan brute-force untuk mencoba miliaran tebakan per detik (Dafi et al., 2025). Selain itu, kerentanan mereka terhadap serangan rainbow table (kamus hash yang sudah dihitung sebelumnya) membuat hash tanpa salt dapat dibobol secara instan (Erdi et al., 2025). OWASP Foundation (2021) mengidentifikasi kegagalan kriptografi ini sebagai salah satu risiko keamanan web paling kritis (OWASP Foundation, 2021).

Kelemahan ini memicu pergeseran paradigma. Pengembang beralih dari hash yang cepat (untuk integritas data) ke algoritma hash yang dirancang khusus untuk password, dengan fokus utama yaitu sengaja dibuat lambat (Etese & Adesina, 2025). Algoritma seperti bcrypt dirancang khusus untuk tujuan ini. Berbeda dengan pendahulunya, bcrypt secara fundamental memperkenalkan mekanisme cost factor (faktor biaya). Fitur ini membuatnya adaptif, memungkinkannya untuk mengimbangi peningkatan kecepatan hardware dari waktu ke waktu.

Namun, lanskap keamanan terus berevolusi. Bcrypt kini bukanlah satu-satunya solusi; algoritma yang lebih baru seperti Argon2 (George & Mathew, 2021), (yang dirancang untuk mengatasi kelemahan Bcrypt terhadap *hardware* modern) telah hadir (Etese & Adesina, 2025). Memilih algoritma yang tepat kini menghadirkan trade-off yang kompleks. Oleh karena itu, tujuan dari studi literatur ini adalah untuk melakukan analisis komparatif terhadap algoritma password hashing, dengan fokus khusus pada bcrypt. Penelitian ini akan mengkaji literatur yang ada melalui dua kriteria utama: Keamanan (ketahanan terhadap serangan) dan Performa (dampak latensi dan resource).

II. Konsep Dasar

Untuk memahami analisis komparatif ini, penting untuk membedakan dua kategori fungsi hash yang, meskipun serupa, memiliki tujuan desain yang fundamentalnya berlawanan.

2.1 Definisi Kunci: Hash Cepat vs. Hash Lambat

- Kategori pertama adalah Fungsi Hash Kriptografis (Cepat), seperti MD5 dan SHA-1. Tujuan desain utama dari algoritma ini adalah untuk menjamin integritas data (integritas file) atau berfungsi sebagai checksum (Eastlake & Jones, 2001; Rivest, 1992). Fungsi ini dirancang untuk menjadi secepat mungkin, kecepatan adalah fitur yang diinginkan untuk memverifikasi file besar dengan cepat tanpa perubahan (Sanchia Melati BR Lumbantoruan, 2022).
- Kategori kedua adalah Fungsi Password Hashing (Sengaja Lambat), seperti Bcrypt dan Argon2. Tujuan desain utama mereka adalah untuk keamanan password. Berbeda dengan kategori pertama, algoritma ini sengaja dirancang untuk lambat dan memakan sumber daya (CPU atau memori). Kelambatan ini adalah fitur keamanan inti untuk membuat serangan brute-force menjadi mahal dan tidak praktis bagi peretas (Etese & Adesina, 2025).

2.2 Metode Serangan Relevan

Evolusi dari hash cepat ke hash lambat didorong oleh dua metode serangan utama yang mengeksploitasi kelemahan hash cepat:

- Serangan Brute-Force (Brute-Force Attack) Ini adalah metode di mana penyerang secara sistematis mencoba semua kemungkinan kombinasi password. Terhadap hash cepat seperti MD5, serangan ini menjadi sangat efektif ketika dipercepat oleh hardware modern seperti GPU (Graphics Processing Unit). Karena hash cepat (MD5/SHA-1) dirancang untuk menjadi efisien, GPU dapat menguji miliaran tebakan per detik, membuat password yang tidak kompleks dapat dibobol dalam hitungan menit (Dafi et al., 2025).
- Serangan Rainbow Table (Rainbow Table Attack) Ini adalah metode yang menggunakan "kamus" atau database raksasa berisi miliaran hash yang sudah dihitung sebelumnya (precomputed). Serangan ini sangat efektif terhadap hash yang tidak menggunakan salt (garam). Karena hash MD5 bersifat deterministik (input '123' akan selalu menghasilkan hash 'abc'), peretas hanya perlu mencari hash yang dicuri di dalam rainbow table mereka untuk menemukan password aslinya (OWASP Foundation, 2021).

2.3 Anatomi Algoritma Password Hashing

Memahami desain setiap algoritma sangat penting untuk analisis komparatif dalam studi ini:

- **MD5 / SHA-1 (Algoritma Usang)** Algoritma ini dirancang murni untuk kecepatan [sitasi: Rivest, 1992]. Kelemahan fatal desain mereka untuk password adalah: (1) Terlalu Cepat, yang membuatnya sangat rentan terhadap Brute-Force GPU (Dafi et

al., 2025), dan dan (2) Deterministik (tidak ada salt bawaan), yang membuatnya rentan terhadap Rainbow Table (OWASP Foundation, 2021).

- **Bcrypt (Solusi Adaptif)** Bcrypt dirancang khusus untuk mengatasi kelemahan tersebut, didasarkan pada algoritma enkripsi Blowfish (Provos & Mazieres, 1999). Keunggulannya terletak pada dua mekanisme: pertama, Mekanisme Salt Otomatis. Bcrypt secara otomatis menghasilkan salt acak yang unik untuk setiap password. Salt ini disimpan bersama hash, memastikan bahwa dua password yang sama akan memiliki hash yang berbeda. Ini secara efektif mengalahkan serangan Rainbow Table (Dafi et al., 2025). Kedua, Mekanisme Cost Factor. Ini adalah "senjata" utama Bcrypt melawan Brute-Force. Ini adalah parameter yang dapat diatur yang secara sengaja memperlambat komputasi. Karena peretas harus melakukan proses lambat ini untuk setiap tebakan, serangan brute-force menjadi sangat mahal dan tidak praktis (Provos & Mazieres, 1999). Fitur ini bersifat adaptif, memungkinkan cost factor dinaikkan di masa depan untuk mengimbangi hardware yang lebih cepat.
- **Argon2 (Standar Modern Memory-Hard)** Argon2 adalah pemenang dari Password Hashing Competition (PHC), Algoritma ini dirancang untuk mengatasi kelemahan teoretis yang masih dimiliki Bcrypt. Bcrypt dirancang untuk menjadi CPU-hard (membuat CPU sibuk), namun masih dapat dioptimalkan dengan hardware khusus (GPU/ASIC) (George & Mathew, 2021). Argon2 memperkenalkan desain Memory-Hard. Ini berarti Argon2 tidak hanya membutuhkan waktu CPU, tetapi juga sengaja dirancang untuk mengisi sejumlah besar RAM selama proses hash. Serangan berbasis GPU/ASIC menjadi sangat mahal karena hardware tersebut biasanya memiliki RAM yang terbatas per unit pemrosesan, sehingga sulit untuk melakukan paralelisasi serangan secara efisien (George & Mathew, 2021).

III. Tinjauan Penelitian Terdahulu

Peneliti	**Metode/Algoritma**	**Tujuan Penelitian**	**Hasil & Temuan**	**Kelemahan / Keterbatasan**
(Dafi et al., 2025)	MD5, SHA-1, Bcrypt	Mengevaluasi dan membandingkan keamanan penyimpanan *password* menggunakan ketiga algoritma tersebut.	Menemukan perbedaan kecepatan komputasi yang drastis. MD5 (0.00002s) dan SHA-1 (0.00001s) "terlalu cepat". Bcrypt (0.28270s) secara sengaja lambat , membuktikan efektivitasnya	Fokus murni pada perbandingan teknis, tidak membahas tantangan implementasi di sisi server.

			melawan *brute-force*.	
(Erdi et al., 2025)	Bcrypt	Menerapkan bcrypt untuk mengamankan *password* pada Sistem Informasi Akademik (SIAK) Universitas Langlangbuana.	Mengatasi masalah spesifik seperti "perubahan nilai mahasiswa tanpa izin". Implementasi *salt* pada bcrypt berhasil memastikan "nilai *hash* yang unik" untuk setiap *password*, meningkatkan integritas data.	Studi kasus terbatas pada satu lingkungan akademik spesifik.
(Nur Isnaini et al., 2025)	Bcrypt dan AES	Mengamankan data pada *web service* menggunakan kombinasi *hashing* bcrypt dan enkripsi AES.	Bcrypt dipilih secara spesifik untuk mengatasi "penyalahgunaan data pribadi" saat *login*. Fitur *cost factor* terbukti esensial untuk "memperlambat serangan *brute force*" secara efektif	Penelitian menggabungkan dua algoritma (Bcrypt dan AES), sehingga sulit mengisolasi kontribusi performa bcrypt secara murni.
(George & Mathew, 2021)	Bcrypt vs. Argon2	Menganalisis kelemahan Bcrypt terhadap *hardware* modern dan mengusulkan Argon2	- Bcrypt (CPU-hard) rentan terhadap optimalisasi GPU/ASIC. - Argon2 (Memory-hard) dirancang untuk menahan serangan GPU. - Menganalisis penggunaan memori sebagai metrik pertahanan baru, tidak hanya kecepatan CPU	Fokus pada analisis teoretis/desain algoritma, mungkin kekurangan data benchmark performa head-to-head di hardware yang sama.

(Etese & Adesina, 2025)	MD5, SHA-1, Bcrypt, Argon2. (Metode: Tinjauan Literatur Sistematis).	Menganalisis evolusi hash dari MD5/SHA-1 ke Bcrypt & Argon2, membandingkan desain, performa, dan keamanannya.	MD5/SHA-1 tidak aman (rentan collision & rainbow table), Bcrypt tahan brute-force (via cost factor) tapi rentan GPU (tidak memory-hard), Argon2 dirancang memory-hard untuk menahan serangan GPU/ASIC	Paper ini adalah tinjauan literatur, bukan benchmark primer baru.

IV. Analisis Sintesis

Tinjauan literatur di Bagian 3 menunjukkan evolusi yang jelas dalam password hashing sebagai respons langsung terhadap ancaman yang berkembang. Analisis komparatif terhadap temuan-temuan ini dapat disintesis ke dalam dua kriteria utama: keamanan dan performa.

4.1 Analisis Komparatif Aspek Keamanan

Analisis keamanan mengungkapkan perlombaan senjata yang konstan antara pengembang algoritma dan peretas.

- Evolusi dari Algoritma Cepat (MD5/SHA-1) Literatur secara konsisten menyimpulkan bahwa algoritma lama seperti MD5 dan SHA-1 sama sekali tidak aman untuk penyimpanan password modern. Kelemahan fatal mereka ada dua:
  - Kerentanan terhadap Rainbow Table: Karena algoritma ini bersifat deterministik dan tidak memiliki mekanisme salt (garam) bawaan, mereka sangat rentan terhadap serangan precomputed hash (kamus hash jadi). Studi kasus seperti (Erdi et al., 2025). membuktikan bahwa implementasi salt adalah langkah mitigasi esensial terhadap ancaman ini.
  - Kerentanan terhadap Brute-Force: Kecepatan, yang dulu dianggap sebagai keunggulan desain untuk integritas file, kini menjadi "kewajiban" (liability) keamanan terbesar mereka. Studi benchmark seperti (Dafi et al., 2025) menunjukkan bahwa hardware modern (GPU) dapat menguji miliaran hash ini per detik.
- Bcrypt sebagai Solusi Adaptif (CPU-Hard) Bcrypt dirancang khusus untuk memecahkan kedua masalah tersebut. (Provos & Mazieres, 1999) menunjukkan desainnya mengatasi:
  - Rainbow Table: Dengan mewajibkan salt otomatis untuk setiap hash, membuat kamus precomputed tidak berguna.
  - Brute-Force (CPU): Dengan sengaja menjadi lambat melalui cost factor yang adaptif, membuat serangan menjadi mahal secara komputasi.

Namun, literatur modern ((Etese & Adesina, 2025) dan (George & Mathew, 2021)) kini mengidentifikasi kelemahan teoretis Bcrypt: ia dirancang untuk menjadi CPU-hard (mengikat CPU), bukan memory-hard (mengikat memori) . Ini berarti serangan yang dioptimalkan menggunakan hardware paralel khusus (GPU/ASIC) masih dapat memproses hash Bcrypt secara efisien.

- Argon2 sebagai Solusi Modern (Memory-Hard) Argon2 (pemenang Password Hashing Competition) dirancang secara eksplisit untuk mengatasi kelemahan Bcrypt. Dengan menjadi memory-hard, Argon2 tidak hanya membutuhkan waktu CPU tetapi juga sejumlah besar RAM. Desain ini secara signifikan lebih efektif dalam menahan serangan paralel berbasis GPU/ASIC, karena hardware tersebut biasanya memiliki RAM terbatas per unit pemrosesan.

4.2 Analisis Komparatif Aspek Performa

Analisis performa dalam konteks password hashing sangat unik: "performa baik" berarti "performa lambat" dari sudut pandang keamanan.

- Mendefinisikan Ulang "Performa" Studi benchmark (Dafi et al., 2025)) secara kuantitatif membuktikan hal ini: performa "cepat" MD5 (misal: 0.00002 detik) adalah performa keamanan yang buruk. Sebaliknya, performa "lambat" Bcrypt (misal: 0.28 detik) adalah performa keamanan yang baik. Literatur (Etese & Adesina, 2025) mendukung ini dengan menyatakan bahwa kecepatan SHA-256 (yang secara kriptografis kuat) masih menjadi "kewajiban" (liability) karena terlalu cepat untuk password.
- Trade-off Bcrypt vs. Argon2 Analisis trade-off performa menunjukkan perbedaan utama dalam konfigurasi:
  - Bcrypt: Performa (kelambatan) diatur oleh satu parameter: Cost Factor. Literatur menunjukkan cost 12 menghasilkan latensi ~300ms di CPU modern, yang dapat diterima untuk login pengguna. Namun, performa ini tidak cukup untuk menghentikan GPU (mencapai 184.000 H/s).
  - Argon2: Menawarkan trade-off yang lebih baik dengan tiga parameter: Time Cost, Memory Cost, dan Parallelism. Ini memungkinkan administrator untuk mengonfigurasi hash agar tidak hanya lambat di CPU (Time Cost), tetapi juga sangat mahal untuk dijalankan di GPU (Memory Cost) .

4.3 Sintesis Komparatif Akhir

Sintesis dari kedua aspek tersebut jelas: Bcrypt adalah lompatan keamanan fundamental dari MD5/SHA-1 dan tetap menjadi solusi yang cukup aman untuk banyak aplikasi. Namun, Argon2 dirancang untuk mengatasi lanskap ancaman modern (serangan GPU/ASIC) yang tidak diantisipasi saat Bcrypt dibuat.

Pilihan antar keduanya adalah trade-off antara:

- Bcrypt: Kematangan (Maturity), kemudahan implementasi (satu parameter), dan dukungan library yang universal.

- Argon2: Keamanan teoretis superior terhadap ancaman hardware modern, tetapi dengan kompleksitas implementasi yang lebih tinggi (tiga parameter yang perlu disetel).

V. Arah dan Peluang Penelitian

Berdasarkan tinjauan dan sintesis literatur yang telah dilakukan, beberapa celah penelitian (research gap) dan peluang penelitian di masa depan dapat diidentifikasi. Fokus penelitian saat ini telah beralih dari "apakah Bcrypt lebih baik dari MD5" menjadi "bagaimana perbandingan Bcrypt dengan standar modern seperti Argon2 dalam menghadapi ancaman hardware saat ini".

Peluang penelitian selanjutnya dapat difokuskan pada area-area berikut:

1. Benchmark Komparatif pada Hardware Modern (GPU/ASIC) Banyak literatur, seperti (Dafi et al., 2025). berfokus pada perbandingan Bcrypt dengan algoritma usang (MD5). Sementara studi seperti (Etese & Adesina, 2025) dan (George & Mathew, 2021) telah mengidentifikasi kelemahan teoretis Bcrypt terhadap GPU, masih terdapat celah untuk studi benchmark kuantitatif yang head-to-head. Penelitian di masa depan dapat melakukan perbandingan performa cracking antara Bcrypt dan Argon2 pada hardware GPU/ASIC modern (misalnya, seri NVIDIA RTX terbaru) untuk mengukur secara pasti seberapa besar keunggulan keamanan praktis yang ditawarkan oleh desain memory-hard Argon2 .
2. Pengembangan Pedoman Konfigurasi (Tuning) Argon2 Literatur menunjukkan bahwa keunggulan Argon2 diimbangi dengan kompleksitas konfigurasinya (tiga parameter: time cost, memory cost, parallelism), berbeda dengan Bcrypt yang hanya memiliki satu cost factor . Terdapat peluang penelitian untuk mengembangkan kerangka kerja (framework) atau model pedoman yang jelas bagi pengembang. Penelitian ini dapat menjawab: "Bagaimana konfigurasi parameter Argon2 yang optimal untuk berbagai skenario aplikasi (misal: web server high-traffic, aplikasi seluler, atau secure vault)?"
3. Analisis Keamanan terhadap Ancaman Baru (Side-Channel & Quantum) Seperti yang disoroti oleh (Etese & Adesina, 2025), lanskap ancaman terus berevolusi. Penelitian masa depan harus mulai mengeksplorasi:
   - Analisis Side-Channel: Menyelidiki kerentanan implementasi library Bcrypt dan Argon2 yang populer terhadap serangan side-channel (misal: timing attack atau power analysis).
   - Implikasi Komputasi Kuantum: Meskipun algoritma hash saat ini tidak secara langsung rentan terhadap serangan kuantum (seperti halnya enkripsi asimetris), penelitian diperlukan untuk menganalisis implikasi jangka panjang dan perlunya "kriptografi yang tangkas" (cryptographic agility) pada arsitektur penyimpanan password.

VI. Kesimpulan

Studi literatur ini telah melakukan analisis komparatif terhadap algoritma password hashing, dengan fokus pada evolusi dari MD5 ke Bcrypt dan standar modern Argon2. Tinjauan ini mengonfirmasi adanya pergeseran paradigma fundamental dalam keamanan

password: dari algoritma yang dirancang untuk kecepatan (integritas data) menjadi algoritma yang dirancang untuk sengaja lambat (keamanan password).

Temuan utama dari sintesis literatur adalah sebagai berikut:

1. Kegagalan Algoritma Usang (MD5/SHA-1): Terdapat konsensus akademis yang kuat bahwa algoritma seperti MD5 dan SHA-1 sama sekali tidak aman untuk penyimpanan password modern. Kelemahan fatal mereka adalah (a) kecepatan komputasi mereka, yang memungkinkan serangan brute-force berbasis GPU, dan (b) sifat deterministik (tanpa salt bawaan), yang membuat mereka rentan terhadap rainbow table.
2. Bcrypt sebagai Solusi Adaptif (CPU-Hard): Bcrypt hadir sebagai solusi pertama yang dirancang khusus untuk password. Dengan salt otomatis dan cost factor yang adaptif, Bcrypt berhasil memitigasi ancaman rainbow table dan brute-force berbasis CPU . Bcrypt terbukti merupakan lompatan keamanan yang esensial dan masih menjadi pilihan yang matang dan mudah diimplementasikan.
3. Kelemahan Bcrypt & Kebangkitan Argon2 (Memory-Hard): Literatur modern kini mengidentifikasi kelemahan teoretis Bcrypt: desainnya yang CPU-hard tidak secara eksplisit dirancang untuk menahan serangan paralel skala besar menggunakan hardware modern (GPU/ASIC) . Argon2, pemenang Password Hashing Competition, dirancang secara spesifik untuk mengatasi celah ini dengan memperkenalkan desain memory-hard. Fitur ini membutuhkan sejumlah besar RAM, sehingga secara signifikan lebih efektif dalam memperlambat serangan berbasis GPU.

Sebagai kesimpulan, meskipun Bcrypt tetap menjadi solusi yang jauh lebih aman daripada MD5/SHA-1 dan masih layak untuk sistem warisan (legacy), Argon2 (khususnya Argon2id) terbukti secara teoretis lebih unggul dan direkomendasikan sebagai standar modern untuk semua sistem baru yang membutuhkan tingkat keamanan tertinggi. Pilihan di antara keduanya mewakili trade-off antara kematangan dan kemudahan implementasi (Bcrypt) versus keamanan superior terhadap ancaman hardware modern (Argon2).

VII. Daftar Pustaka

Dafi, R., Azhar, A., & Widiati, I. S. (2025). Evaluasi Keamanan Penyimpanan Password Menggunakan Algoritma Hash: MD5, SHA-1, dan Bcrypt. *Seminar Nasional Teknologi Informasi Dan Bisnis (SENATIB)*, 2025.

Eastlake, D., & Jones, P. (2001). *US Secure Hash Algorithm 1 (SHA1)*. IETF. https://datatracker.ietf.org/doc/html/rfc3174

Erdi, S., Sohidin, P., Prasetyo Utomo, H., & Ahmadi, A. (2025). Penerapan Algoritma Bcrypt untuk Pengamanan Password pada Sistem Informasi Akademik (SIAK) (Studi Kasus : Universitas Langlangbuana). In *Jurnal Infosecure* (Vol. 6, Issue 2).

Etese, O., & Adesina, A.-A. (2025). *A Review and Comparative Analysis of Password Hashing Techniques: Evaluating Bcrypt and Argon2*.

George, A. T., & Mathew, J. (2021). *Argon2: The Secure Password Hashing Function*. https://doi.org/10.5281/zenodo.5091700

Nur Isnaini, K., Suhartono, D., Thoriq Jamil, M., & Qothrunnada Khoirunnita, A. (2025). *Implementasi Pengamanan Data Menggunakan Teknik Bcrypt Hashing Password dan Algoritma Advanced Encryption Standard (AES)*.

OWASP Foundation. (2021). *A02:2021 – Cryptographic Failures*. OWASP Top 10:2021. https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

Provos, N., & Mazieres, D. (1999). A Future-Adaptable Password Scheme. *Proceedings of the 1999 USENIX Annual Technical Conference*. https://www.usenix.org/legacy/event/usenix99/provos/provos.pdf

Rivest, R. (1992). *The MD5 message-digest algorithm*. IETF. https://www.ietf.org/rfc/rfc1321.txt

Sanchia Melati BR Lumbantoruan, G. (2022). Perancangan Aplikasi Duplicate Image Scanner Menerapkan Algoritma MD5. *Journal of Computing and Informatics Research*, *1*(3), 88–94. https://doi.org/10.47065/comforch.v1i3.347