Jurnal Ilmiah Informatika (JIF)
| ISSN (Print) 2337-8379 | ISSN (Online) 2615-1049 |
Jurnal online di akses di http://ejournal.upbatam.ac.id/index.php/jif

Security Testing (*White Box Penetration Testing*) Pada *Authentication* Sistem *Login Website*

Amar Luthfi[1], E. Haodudin Nurkifli[2], Iqbal Maulana[3]

[1,2,3]Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang Karawang, Jawa Barat, Indonesia 41361

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 08-08-2025
Revisi Akhir: 16-08-2025
Diterbitkan *Online*: 10-09-2025

KATA KUNCI

Autentication

MD5

Penetration

Security

KORESPONDENSI

E-mail: amar.luthfi.18154@student.unsika.ac.id

A B S T R A C T

In 2024, according to the 2024 Indonesian Cybersecurity Landscape by the National Cyber Security Agency (BSSN), Indonesia experienced a total of 330,527,636 anomalous traffic. The highest traffic occurred in December with 112,085,045 traffic, and the lowest in May with 12,273,078 traffic. The impact of this leak is very broad, causing the risk of phishing, account hijacking, identity theft, ransomware attacks on companies, and misuse of business email. Based on this background, this study aims to minimize the many attacks that occur in the cyber world, especially on website applications. By using the Security Testing method (White Box Penetration Testing). The MD5 algorithm is still effective in protecting passwords in databases from SQL injection attacks that use common username and password combinations and on websites that use PHP without JavaScript is proven safe from Cross-Site Scripting attacks.

1. PENDAHULUAN

Pada tahun 2024, bersumber pada Lanskap Keamanan Siber Indonesia 2024 oleh BSSN Negara Indonesia mendapatkan total trafik anomali 330.527.636 trafik. Trafik tertinggi terjadi pada bulan Desember dengan jumlah 112.085.045 trafik, dan terendah pada bulan Mei dengan jumlah 12.273.078 trafik. Aktivitas ini dapat berdampak pada penurunan efisiensi perangkat dan jaringan. Pencurian data sensitif, bisa merusak reputasi dan kepercayaan terhadap suatu organisasi [1].

Pada artikel berjudul "Alarm Siber Global: 16 Miliar *Username* & *Password* Bocor" dari CSIRT.or.id juga melaporkan insiden kebocoran data *login* terbesar yang di temukan oleh tim siber *Cybernews* pada pertengahan Juni 2025. Sekitar 16 miliar kredensial (*username* dan *password*) ditemukan tersebar luas di internet, yang disebut sebagai "cetak biru untuk eksploitasi massal." [2].

Rekapan kebocoran tersebut berada pada penyimpanan *database* pada server yang tidak aman. Kebocoran yang mencakup kredensial untuk layanan populer seperti Facebook, Google, Apple, GitHub, Telegram, bahkan platform pemerintah dengan sebagian besar data masih aktif. Format data yang bocor menunjukkan bahwa banyak di antaranya adalah hasil curian *malware infostealer*, berisi URL *login*, *username*/*email*, dan *password plaintext*.

Dampak kebocoran ini sangat luas, menyebabkan risiko *phishing*, pembajakan akun, pencurian identitas, serangan *ransomware* pada perusahaan, dan penyalahgunaan *email* bisnis. Pola kerja peretas kini bergeser ke penyimpanan data dalam basis data raksasa untuk eksploitasi otomatis. Meskipun perusahaan besar diatas membantah kebocoran dari server mereka, keberadaan ribuan *login* yang mengarah ke situs mereka mengindikasikan bahwa akun diretas melalui *malware* di perangkat pengguna.

Sebagai langkah perlindungan, pada artikel tersebut merekomendasikan langkah-langkah pengamanan seperti penggunaan pengelola kata sandi, penggantian kata sandi secara rutin, pengaktifan autentikasi dua faktor (2FA), pemindaian perangkat dari *malware*, da pemantauan akun. Selain itu, pada pihak pengembang juga diperlukan pemahaman secara teoritis dan praktis mengenai keamanan pada aplikasi atau platform yang dikembangkan.

Berdasarkan latarbelakang tersebut penelitian ini bertujuan untuk meminimalisir banyak serangan yang terjadi pada dunia siber lebih khususnya pada aplikasi *website*. Dengan menggunakan metode *Security Testing* (*White Box Penetration Testing*) dengan melakukan simulai penyerangan pada *website* sehingga sebelum situs diakses secara masal, situs web sudah diketahui titik kerentanannya dan diperbaiki terlebih dahulu. Sehingga situs web yang dapat diakses sudah terjamin keamanannya.
Metode *Penetration testing* digunakan untuk menguji lapisan dasar pada pengamanan sebuah aplikasi atau sistem dengan melihat dokumen terbuka dari *source code*. Salah satu jenis *Penetration Testing* yaitu *White Box Testing*, pengujian keamanan yang sangat detail dan mendalam, memanfaatkan pengetahuan internal sistem untuk mengungkap kerentanan yang mungkin tersembunyi jauh di dalam kode dan arsitektur aplikasi.

2. TINJAUAN PUSTAKA

2.1. *Security Testing*

Security testing adalah metode terintegrasi yang menawarkan alat pendukung pada seluruh proses pengujian, diawali pemetaan dan analisis serangan pada permukaan aplikasi sampai menemukan dan mengeksploitasi kerentanan keamanan [3].
Security Testing adalah salah satu metode yang mengungkap apakah perangkat lunak dan fungsi keamanannya berjalan dengan benar saat terjadi serangan berbahaya [4].

2.2. *Penetration Testing*

Penetration testing adalah alat penilaian yang digunakan baik untuk bisnis maupun operasinya karena *penetration testing* membantu membentuk strategi keamanan informasi dengan mengidentifikasi kerentanan yang cepat dan akurat [5].
Penetration testing adalah salah satu metode *ethical hacking* untuk menguji dan melindungi keamanan informasi dalam rangka mengevaluasi sistem keamanan dengan cara melakukan simulasi serangan yang mendapat persetujuan legal dari pemilik *platform* [6].

2.3. *Website*

Website adalah wadah informasi bagi masyarakat mengenai hasil atau produk dari perusahaan atau organisasi tentang jasa hingga pemerintahan agar masyarakat mengetahui perkembangan informasi [7].
Website adalah sebuah laman yang menyajikan dokumen berisi informasi yang dapat dijangkau oleh *smartphone* ataupun *desktop* yang dapat membantu dan diakses sehari-hari [8].

2.4. *User Authentication*

Otentikasi Pengguna adalah proses dalam memverifikasi suatu otoritas atau hak terhadap seseorang untuk memenuhi berbagai syarat yang dibutuhkan bagi autentikasi agar dapat melakukan pengaksesan suatu sistem layanan aplikasi untuk meningkatkan tingkat keamanan data [9].
User Authentication adalah proses *checking* identitas pengguna pada proses *login* ke dalam sebuah sistem untuk mengetahui apakah pengguna memiliki otoritas atas sistem, atau mungkin aplikasi yang berjalan pada sistem [10].

2.5. *Email*

Electronic-mail adalah fasilitas internet yang paling banyak digunakan digunakan di internet dilengkapi fitur yang tidak hanya berbetuk teks tapi juga dapat dalam bentuk file audio, video, photo dan file ektensi lainnya [11].
Email adalah fasilitas di internet maupun jaringan komputer yang dapat mengirim dan menerima surat secara elektronis berupa berita, gambar atau data yang bersifat *Asynchoronous Communication Mode* [12].

2.6. *Password*

Password adalah suatu mekanisme yang umum digunakan untuk verifikasi pengguna yang sah yang memiliki akses akun ke sebuah platform [13].
Password adalah serangkaian kode rahasia berfungsi sebagai autentikasi pengguna untuk mendapatkan akses ke sistem, aplikasi, file data, atau server jaringan [14].
.

3. METODOLOGI

Penelitian ini akan menggunakan metodologi *security testing* (*white box penetration testing*). *White box penetration testing* adalah metode pengujian perangkat lunak yang memeriksa struktur internal, desain, dan kode dari suatu aplikasi. Pada metode ini memiliki tahapan penelitian ini pada gambar 1.

Gambar 1 Metodelogi Penelitian

Tahapan pertama yaitu mengumpulkan semua informasi yang relevan tentang sistem, menentukan sistem/aplikasi yang akan diuji, dan mendefinisikan tujuan keamanan spesifik (misalnya, mencari kerentanan injeksi SQL dan XSS). Selanjutnya melakukan identifikasi potensi kerentanan keamanan tanpa menjalankan kode. *Manual Code Review*: *Pen-tester* secara manual meninjau kode sumber untuk mencari pola kerentanan umum (misalnya, *input validation* yang buruk, penggunaan fungsi yang tidak aman, *hardcoded credentials*, kesalahan logika). Seperti pada gambar 2 yang dapat dilihat penggunaan *function* untuk menutup kolom saat terjadi kesalahan *input* pada akses pengguna,

```
// --- Konfigurasi Lockout ---
$max_login_attempts = 3; // Maksimal percobaan login gagal
$lockout_duration = 300; // Durasi lockout dalam detik (300 detik = 5 menit)
// --- End Konfigurasi Lockout ---

// Cek apakah user sudah login, jika iya redirect ke halaman utama
```

Gambar 2 Batas Maksimal Kesalahan *Input*

Selain itu, pada kolom *input*-an juga perlu dilakukan verifikasi *type input*-an seperti pada gambar 3 *input*-an *Email* mengharuskan pengguna menggunakan simbol @ sehingga jika asal maka *website* tidak akan *loading* atau melakukan proses verifikasi akun.

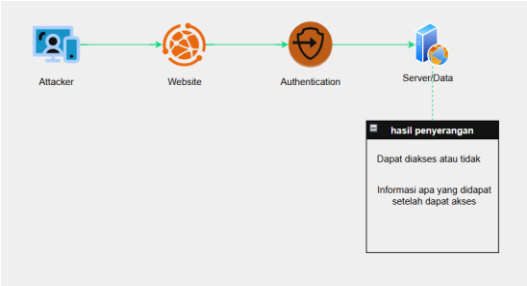
```
    <label>Email</label>
    <input type="email" name="email" value="<?php echo $email; ?>" <?php echo $is_locked ? 'disabled' : ''; ?>>
    <span class="error"><?php echo $email_err; ?></span>
</div>
<div>
    <label>Password</label>
    <input type="password" name="password" <?php echo $is_locked ? 'disabled' : ''; ?>>
    <span class="error"><?php echo $password_err; ?></span>
</div>
<div>
    <input type="submit" value="Login" <?php echo $is_locked ? 'disabled' : ''; ?>>
```

Gambar 3 Tipe Input Email

Sedangkan pada bagian *password* diterapkan metode MD5 pada *database* sehingga *password* yang tersimpan adalah bentuk *hash* dari algoritma tersebut.

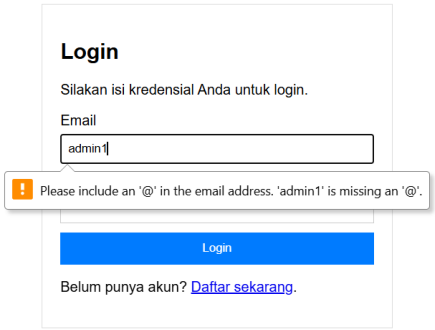
Gambar 4 Algoritma MD5

Pada tahapan ketiga menganalisis konfigurasi lingkungan (*server operating system*, *web server*, *application server*, *database server*), dan mengidentifikasi potensi titik masuk dan keluar data. Selanjutnya melakukan penggambaran skenario pengujian seperti pada gambar 5.



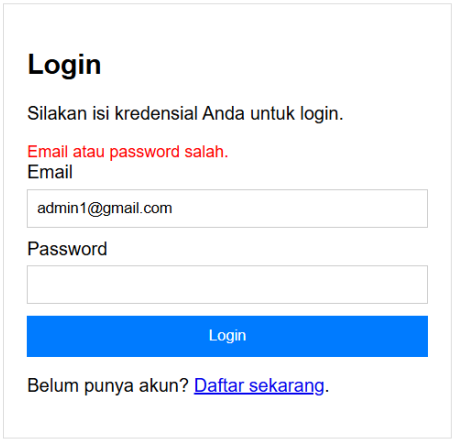
Gambar 5 Skenario Pengujian

Pengujian pertama, memastikan hasil dari pengembangan pada *source code* berfungsi dengan baik. Untuk hasil dari pengujian pertama dapat dilihat pada gambar 6 sebagai contoh pengguna tidak menggunakan simbol @.



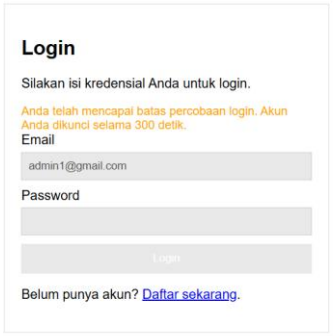
Gambar 6 Pengguna Harus Menggunakan @

Selanjutnya pada gambar 7 adalah saat pengguna menggunakan *email* tapi akun tidak tidak ditemukan.



Gambar 7 Pengguna Tidak Ditemukan

Terakhir, pengujian untuk mengetahui apakah sistem blok kolom *input*-an berjalan dengan baik atau tidak saat pengguna melakukan kesalahan *input*-an lebih dari tiga kali. Hasilnya dapat dilihat pada gambar 8.



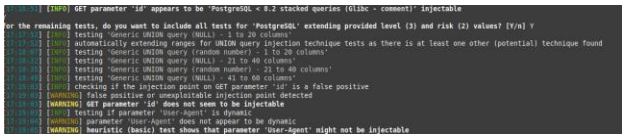
Gambar 8 Akses di *Block* Selama 300 Detik

Setelah mengetahui aplikasi berjalan dengan sesuai, langkah selanjutnya adalah melakukan eksploitasi. Jenis serangan yang digunakan adalah *sql injection* dan *cross-site scipting*. Dua metode tersebut dipilih karena kecocokannya dengan metode

penelitian ini. Dimana, *Security Testing* (*White Box Penetration Testing*) adalah metode pengujian awal pada aplikasi, apakah sistem keamanan yang dikembangkan berfungsi saat terjadi percobaan penyerangan. Sedangkan SQLi akan menyerang bagian hak akses pada *website* dengan cara memanfaatkan kombinasi *username* dengan *password* dan XSS akan mengambil kerentanan pada url untuk dimanfaatkan sebagai akses masuk ke *website*.

Penyerangan pertama dengan menggunakan SQLmap untuk melakukan *injection* pada *website*, dengan metode temper pada gambar 9, dan meningkatkan *level* dan *risk* pengujian pada gambar 10.

Gambar 9 Penyerangan *Temper*

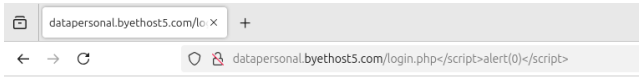


Gambar 10 Penyerangan *Level* 3 dan *Risk* 2

Dari dua cara penyerangan diatas mendapatkan hasil yang sama, Dimana *log* menunjukan *website* tidak menunjukan paramater dapat di injeksi. Hasil dapat dilihat pada gambar 11 dibawah ini.

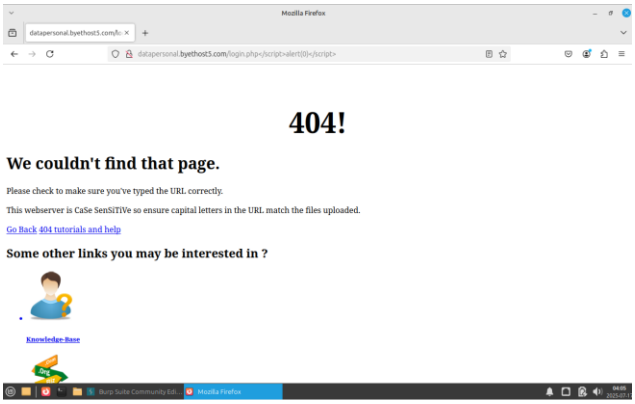
Gambar 11 SQLi gagal dilakukan

Setelah melakukan serangan pertama, selanjutnya melakukan serangan dengan metode XSS dengan menyusupkan *script* pada bagian akhir url seperti pada gambar 12.



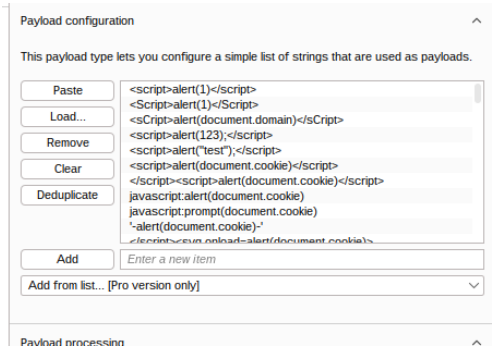
Gambar 12 *Input Script* XSS pada URL

Dari *script* diatas menunjukkan hasil bahwa halaman tidak diketahui atau 404 seperti pada gambar 13.



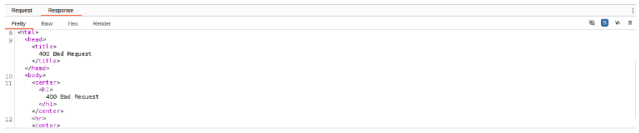
Gambar 13 Hasil Penyusupan *Script*

Ini adalah salah satu ciri bahwa serangan XSS tidak dapat dilakukan pada *website*, untuk memastikannya. Bisa menggunakan Burpsuite untuk menyisipkan *script* pada *website*. Seperti pada gambar 14.



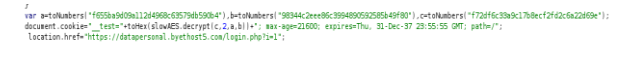
Gambar 14 *Scripting* dengan Burpsuite

Hasil dari penyerangan XSS pada Burpsuite juga menampilkan bahwa *website* tidak dapat dijangkau. Dapat dilihat pada gambar 15.



Gambar 15 Hasil-1 *Scripting* dengan Burpsuite

Dan tidak ditemukan url hasil dari *scripting* yang dapat dijadikan sebagai akses untuk masuk ke *website* dari cara *payload*.



Gambar 16 Hasil-2 Scripting dengan Burpsuite

4. HASIL DAN PEMBAHASAN

Setelah melakukan eksploitasi pada sistem dengan cara *Security Testing* (*White Box Penetration Testing*) didapat hasil dari analisis pengerjaan penelitian sebagai berikut.

Tabel 1 Tabel Hasil Uji

Jenis Serangan	**Informasi yang didapat**	**Keterangan Serangan & Kualitas Serangan** **(S/T) - (H/M/L/0)**
SQL *injection*	Tidak mendapatkan informasi kredensial dari *website*	T – 0
XSS	Tidak mendapatkan informasi kredensial dari *website* Injeksi mendapatkan *respond 404 not found*	T – 0

*keterangan:

S = sukses H = *High* L = *Low*

T = tidak M = *Medium* 0 = gagal

Maka dapat disimpulkan hasil dari *Security Testing* (White Box *Penetration Testing*) terhadap *website* https://datapersonal.byethost5.com/ berhasil membuktikan *website* tersebut aman dari serangan SQLi dan XSS. Pada serangan SQLi serangan gagal karena pada aplikasi Sqlmap tidak menunjukan aktifitas *log* yang mendapatkan kata sandi dan nama pengguna atau *email* pada kasus penelitian ini. Penggunaan algoritma MD5 untuk hashing berhasil mengubah kata sandi *plaint text* menjadi *encrypted* sehingga memperkuat keamanan data. Selain itu, penggunaan kode ‘ type=email ‘ pada pembuatan *website* juga membuat bagian *input* tidak bisa diisi sembarang data. Jika tidak menggunakan simbol @ maka tidak ada proses *loading page*, justru memunculkan notifikasi kesalahan input.

Sedangkan, pada serangan XSS gagal karena pengembangan *website* tidak menggunakan JavaScript untuk memuat maupun memproses data. *Website* dikembangkan dengan bahasa PHP untuk melakukan proses *input* dan *output*. Sehingga injeksi atau menyisipkan *script* tidak berpengaruh untuk mendapatkan *copy link* untuk membuat *website* tiruan dan aksesnya.

5. KESIMPULAN DAN SARAN

Kesimpulan pada penelitian ini adalah sebagai berikut:

1. Algoritma MD5 masih efektif untuk melindungi password pada database dari serangan SQL injection yang menggunakan kombinasi username dan password umum.
2. Website yang menggunakan PHP tanpa JavaScript terbukti aman dari serangan Cross-Site Scripting.
3. Penggunaan hash pada password tidak mengharuskan pengguna mengingat banyak pola atau sandi keamanan. Karena sandi disimpan dalam bentuk enkripsi.
4. Pengembang perlu mengelola kode sumber (manage source) dengan rapi untuk menjaga keamanan.
5. Metode White Box Testing direkomendasikan untuk melakukan uji keamanan sistem atau aplikasi karena memungkinkan penguji untuk memahami logika dan interkoneksi aplikasi (XSS).

Adapun saran untuk penelitian selanjutnya adalah:

1. Gunakan metode penyerangan yang lain, yang bersifat krusial.
2. Gunakan tambahan metode keamanan seperti *two-factor authentetication* atau *multi-factor authentication*.

DAFTAR PUSTAKA

[1] Badan Siber dan Sandi Negara (BSSN). (2024). LANSKAP KEAMANAN SIBER 2024. *https://www.bssn.go.id/wp-content/uploads/2025/02/LANSKAP-KEAMANAN-SIBER-2024-1.pdf*

[2] CSIRT Indonesia. (2025). Alarm Siber Global: 16 Miliar Username & Password Bocor. *https://csirt.or.id/berita/siber-global-16-miliar-password-bocor*

[3] Erdogan, G. (2009). *Security Testing of Web Based Applications*. *July*.

[4] Κωνσταντίνου, Δ. Χ. (2024). *Penetration Testing Methodology Μεθοδολογία Δοκιμών Διείσδυσης*. *February*. https://doi.org/10.26262/heal.auth.ir.356613

[5] Hasibuan, M., & Elhanafi, A. M. (2022). *Penetration Testing* Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box. *Sudo Jurnal Teknik Informatika*, *1*(4), 171–177. https://doi.org/10.56211/sudo.v1i4.160

[6] Widi Linggih Jaelani, Yanto, Y., & Khoirunnisa, F. (2023). *Penetration Testing Website* Dengan Metode Black Box Testing Untuk Meningkatkan Keamanan *Website* Pada Instansi (Redacted). *Naratif : Jurnal Nasional Riset, Aplikasi Dan Teknik Informatika*, *5*(1), 1–8. https://doi.org/10.53580/naratif.v5i1.180

[7] Risky, M. A. Z., & Yuhandri, Y. (2021). Optimalisasi dalam Penetrasi Testing Keamanan *Website*

Menggunakan Teknik SQL Injection dan XSS. *Jurnal Sistim Informasi Dan Teknologi*, *3*, 215–220. https://doi.org/10.37034/jsisfotek.v3i4.68

[8] Wardhana, A. W., & Seta, H. B. (2021). Analisis Keamanan Sistem Pembelajaran Online Menggunakan Metode ISSAF pada *Website* Universitas XYZ. *Informatik : Jurnal Ilmu Komputer*, *17*(3), 226. https://doi.org/10.52958/iftk.v17i3.3653

[9] Fadlullah Fadlullah, Muhlis Tahir, Briliant Pijar Bintari, Mia Liana Dewi, Muhammad Fahri Ilmy, Syafi' Syafi', & Rama Ardiansyah. (2023). Implementasi Algoritma AES pada Autentikasi *Log*in Sistem Informasi. *Jurnal Bintang Pendidikan Indonesia*, *1*(2), 251–263. https://doi.org/10.55606/jubpi.v1i2.1420

[10] Rusdan, M., & Sabar, M. (2020). Analisis dan Perancangan Jaringan Wireless Dengan Wireless Distribution System Menggunakan User Authentication Berbasis Multi-Factor Authentication. *JOINT (Journal of Information Technology)*, *02*(01), 17–24.

[11] Fauzi, M. K., & Setiawan, A. (2024). Implementasi Algoritma Vigenere Chiper atau Caesar Chiper Untuk Pengamanan Password Dalam Penerimaan Siswa Baru. Jurnal Info Digit) eISSN29880289 Vo l. *Jid*, *2*(3), 1083. http://kti.potensi-utama.ac.id/index.php/JID

[12] Wahyu Hidayat M, Nurhayi Musdira, Natatsa Rasyid, Miftahul Khairi S, & Muh Juharman. (2023). Analisis Ancaman Terhadap Keamanan Data Pribadi pada *Email*. *Jurnal Pendidikan Terapan*, *01*, 7–12. https://doi.org/10.61255/jupiter.v1i2.73

[13] Fauzi, F. M., Indonesia, U. K., Afrianto, I., & Indonesia, U. K. (2023). *Implementasi Algoritma Md5 Untuk Keamanan Login Website Implementasi Algoritma Md5 Untuk Keamanan*. *d*(August), 1–5

[14] Yamin, M., Malethi, T. T., Monica, Jodhika, & Natali, S. (2023). Evaluasi Risiko Pada Penggunaan *Password* Yang Lemah: Analisis Kasus Penggunaan *Password* Umum. *Jurnal Ilmiah Multidisiplin Ilmu Komputer*, *1*(1), 41–48. https://doi.org/10.61674/jimik.v1i1.112

[15] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?," IEEE Trans. Commun., vol. 67, no. 10, pp. 7331–7376, 2019, doi: 10.1109/TCOMM.2019.2924010.

BIODATA PENULIS

Amar Luthfi
Merupakan mahasiswa Informatika, Fakultas Ilmu Komputer. Universitas Singaperbangsa Karawang

E. Haodudin Nurkifli
Merupakan Dosen Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

Iqbal Maulana
Merupakan Dosen Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang