**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN IMPLEMENTATION WEB AUTHENTICATION API (WebAuthn)**

*Laporan ini disusun untuk memenuhi tugas 1 mata kuliah Pengembangan Web (Teori)*



Disusun oleh kelompok B4:

Asri Husnul Rosadi 221524035

Faris Abulkhoir 221524040

Mahardika Pratama 221524044

Muhamad Fahri Yuwan 221524047

Najib Alimudin Fajri 221524053

Sarah 221524059

Septyana Agustina 221524060

Dosen Pengampu:

Joe Lian Min, M.Eng.

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc176232797)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc176232798)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc176232799)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc176232800)

[**D.** **PELAKSANAAN EKSPERIMEN** 2](#_Toc176232801)

[**E.** **ANALISIS HASIL EKSPERIMEN** 2](#_Toc176232802)

# **IDENTIFIKASI PROBLEM**

Menerapkan sistem autentikasi berbasis WebAuthn untuk pendaftaran dan autentikasi pengguna pada aplikasi web.

# **DESKRIPSI PROBLEM**

Autentikasi adalah salah satu aspek penting dalam keamanan aplikasi web. Dengan meningkatnya kebutuhan akan sistem autentikasi yang lebih aman, WebAuthn (Web Authentication API) muncul sebagai solusi yang mengizinkan penggunaan berbagai metode autentikasi seperti security key, biometrik, dan platform authenticators. Namun, implementasi WebAuthn sering kali menghadapi tantangan dalam konfigurasi dan integrasi dengan sistem yang ada. Tujuan dari eksperimen ini adalah untuk menguji dan menerapkan WebAuthn dalam skenario pendaftaran dan autentikasi pengguna dengan server Express dan aplikasi web sederhana.

# **METODOLOGI EKSPERIMEN**

1. Pengaturan Lingkungan:

* Server: Menggunakan Express.js untuk menyediakan endpoint pendaftaran dan autentikasi.
* Klien: Halaman HTML dengan formulir untuk pendaftaran dan autentikasi menggunakan JavaScript dan WebAuthn API.

1. Pendaftaran Pengguna:

* Menyiapkan endpoint untuk menerima kredensial dari browser dan menyimpan informasi pengguna di memori server.
* Menggunakan WebAuthn API untuk menghasilkan credential baru di sisi klien.

1. Autentikasi Pengguna:

* Menyiapkan endpoint untuk menerima assertion dari browser dan memverifikasi kredensial.
* Menggunakan WebAuthn API untuk mendapatkan assertion dari klien.

1. Pengujian:

* Mendaftarkan pengguna baru dengan memasukkan username dan menyelesaikan pendaftaran menggunakan perangkat autentikasi.
* Mengautentikasi pengguna dengan memasukkan username dan menyelesaikan autentikasi menggunakan perangkat autentikasi..

# **PELAKSANAAN EKSPERIMEN**

1. Pengaturan Server:

* Kode server menggunakan Express.js dengan endpoint /register dan /authenticate untuk menangani pendaftaran dan autentikasi.
* Data pengguna disimpan di memori untuk tujuan eksperimen.

1. Pengaturan Klien:

* Halaman HTML berisi formulir untuk pendaftaran dan autentikasi.
* JavaScript di halaman HTML menangani pembuatan credential dan assertion menggunakan WebAuthn API.

1. Uji Coba:

* Pendaftaran: Pengguna memasukkan username, browser memunculkan jendela Windows Security (atau metode autentikasi lain), dan kredensial dikirim ke server.
* Autentikasi: Pengguna memasukkan username, browser menyiapkan challenge, dan assertion dikirim ke server untuk verifikasi.

# **ANALISIS HASIL EKSPERIMEN**

1. Hasil:

* Pendaftaran Berhasil: Jendela Windows Security muncul dan kredensial berhasil dikirim dan disimpan di server.
* Autentikasi Berhasil: Pengguna berhasil mengautentikasi diri dengan username yang terdaftar dan assertion diverifikasi.

1. Pros dan Cons

Pros:

* Keamanan Tinggi: WebAuthn menyediakan tingkat keamanan yang lebih tinggi dibandingkan dengan sistem autentikasi berbasis password, menggunakan metode autentikasi kuat seperti security key dan biometrik.
* Pengalaman Pengguna: Metode autentikasi yang nyaman seperti biometrik dan security key meningkatkan pengalaman pengguna dengan mengurangi kebutuhan untuk mengingat password.
* Standar Terbuka: WebAuthn adalah standar terbuka yang didukung oleh berbagai browser dan platform, memudahkan integrasi.

Cons:

* Keterbatasan Perangkat: Tidak semua perangkat mendukung semua metode autentikasi WebAuthn. Pengguna mungkin memerlukan perangkat tambahan seperti security key.
* Kompleksitas Implementasi: Meskipun standar WebAuthn cukup matang, implementasinya memerlukan penanganan detail teknis yang kompleks, seperti manajemen challenge dan verification.
* Keamanan Data: Data pengguna disimpan di memori server dalam eksperimen ini. Dalam aplikasi nyata, data harus disimpan dengan aman di database dan ditangani sesuai kebijakan keamanan.

1. Kesimpulan

Eksperimen ini berhasil menerapkan sistem autentikasi WebAuthn, dengan pendaftaran dan autentikasi pengguna berjalan sesuai rencana. Munculnya jendela Windows Security dan kemampuan untuk menyelesaikan proses pendaftaran dan autentikasi mengindikasikan bahwa WebAuthn telah diintegrasikan dengan benar. Meskipun eksperimen menunjukkan hasil positif, ada beberapa kekurangan yang perlu diperhatikan, seperti keterbatasan perangkat dan kompleksitas implementasi. Penerapan WebAuthn dalam aplikasi nyata akan memerlukan perhatian lebih pada keamanan data dan kompatibilitas perangkat.

.