**PENGEMBANGAN WEB (TEORI)**

**LAPORAN EKSPERIMEN MENGENAI PROBLEM MENCEGAH SERANGAN CROSS-SITE SCRIPTING (XSS)**

**DENGAN PENYELESAIAN MENGGUNAKAN SOLUSI VALIDASI INPUT PADA WEBSITE NATIVE PHP**

*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

Septyana Agustina 221524058

Sarah 221524059

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](#_Toc175484086)

[**A.** **IDENTIFIKASI PROBLEM** 1](#_Toc175484087)

[**B.** **DESKRIPSI PROBLEM** 1](#_Toc175484088)

[**C.** **METODOLOGI EKSPERIMEN** 1](#_Toc175484089)

[**D.** **PELAKSANAAN EKSPERIMEN** 1](#_Toc175484090)

[**E.** **ANALISIS HASIL EKSPERIMEN** 2](#_Toc175484091)

Link chatGPT : https://chatgpt.com/share/7da9d47d-9a37-4889-8ff8-1193e1ab0131

# **IDENTIFIKASI PROBLEM**

Dalam dunia pengembangan web, keamanan aplikasi adalah aspek krusial yang harus diperhatikan. Salah satu kerentanan umum yang sering ditemui adalah **Cross-Site Scripting** (XSS). XSS memungkinkan penyerang untuk menyuntikkan skrip berbahaya ke dalam halaman web yang ditampilkan kepada pengguna lain, yang dapat menyebabkan pencurian data, perusakan integritas situs, atau serangan phishing. Oleh karena itu, penting untuk memahami dan menerapkan teknik yang efektif untuk mencegah XSS.

# **DESKRIPSI PROBLEM**

Pada eksperimen ini, kami fokus pada kerentanan XSS pada sebuah aplikasi web sederhana yang dibangun menggunakan PHP. Aplikasi ini memiliki fitur komentar, di mana pengguna dapat memasukkan nama dan komentar mereka. Tanpa perlindungan yang memadai, input pengguna ini berpotensi disalahgunakan untuk menyuntikkan skrip berbahaya. Saya ingin menguji efektivitas dari metode validasi input dalam menangkal XSS dan membandingkannya dengan solusi lainnya.

# **METODOLOGI EKSPERIMEN**

Untuk mengevaluasi efektivitas validasi input dalam mencegah XSS, saya melakukan eksperimen dengan langkah-langkah berikut:

1. **Pengaturan Lingkungan:** Saya membuat aplikasi web sederhana dengan PHP yang memungkinkan pengguna memasukkan nama dan komentar.
2. **Implementasi Validasi Input:** Saya menerapkan validasi panjang input dan memeriksa adanya tag **<script>** pada input nama dan komentar. Jika input tidak sesuai, sistem akan menolak data tersebut.
3. **Sanitasi Output:** Setelah validasi, saya menggunakan fungsi **htmlspecialchars()** untuk mengamankan data yang akan ditampilkan.
4. **Pengujian:** Saya menguji aplikasi dengan berbagai input, termasuk skrip XSS, untuk mengevaluasi apakah validasi dan sanitasi yang diterapkan efektif.

# **PELAKSANAAN EKSPERIMEN**

Pada implementasi yang telah saya lakukan:

* **Validasi Input:** Saya memeriksa panjang input dan memastikan tidak ada tag HTML atau skrip yang disertakan.
* **Sanitasi Output:** Data yang dianggap aman kemudian ditampilkan dengan fungsi **htmlspecialchars()** untuk menghindari interpretasi HTML dan JavaScript oleh browser.

Setelah penerapan metode ini, saya menguji dengan menyertakan skrip **<script>alert('XSS');</script>** dalam nama dan komentar. Validasi input mencegah input tersebut disimpan, dan sanitasi output memastikan data yang ditampilkan tetap aman.

# **ANALISIS HASIL EKSPERIMEN**

Dari hasil eksperimen, dapat disimpulkan bahwa penerapan validasi input efektif dalam mencegah serangan XSS. Ketika input berbahaya seperti **<script>alert('XSS');</script>** dimasukkan, aplikasi dapat mendeteksi dan menolak input tersebut. Sebagai hasilnya, tidak ada skrip yang dieksekusi di browser, sehingga aplikasi tidak rentan terhadap serangan XSS.

**Temuan Penting:**

* **Validasi Input:** Validasi input sangat penting untuk mencegah XSS. Memeriksa apakah input mengandung tag HTML atau skrip berbahaya sebelum disimpan atau ditampilkan dapat mencegah eksekusi kode berbahaya.
* **Fungsi Sanitasi:** Penggunaan fungsi sanitasi seperti htmlspecialchars() tetap diperlukan untuk menambah lapisan keamanan saat menampilkan input pengguna, bahkan setelah validasi dilakukan.
* **Kombinasi Teknik Keamanan:** Validasi input harus digabungkan dengan sanitasi output dan penggunaan header keamanan untuk memberikan perlindungan yang lebih komprehensif terhadap serangan XSS.

**Pro dan Kontra :**

A black background with white text

Description automatically generated

**Kesimpulan:**

**Validasi input** adalah teknik penting dalam strategi keamanan web yang membantu mencegah data berbahaya masuk ke dalam sistem dan memberikan kontrol atas data yang diterima. Namun, meskipun validasi input sangat berguna, ia tidak sepenuhnya menggantikan kebutuhan untuk teknik keamanan tambahan seperti sanitasi output dan Content Security Policy (CSP). Validasi input harus diterapkan dengan hati-hati dan sering digabungkan dengan langkah-langkah keamanan lainnya untuk perlindungan yang lebih komprehensif.

Perbedaan dalam penanganan validasi input antara aplikasi native dan menggunakan framework terletak pada pendekatan, fitur bawaan, pembaruan keamanan, dan konsistensi. Framework menyediakan alat dan pustaka yang mempermudah proses, menawarkan solusi yang lebih konsisten dan terintegrasi, sementara aplikasi native memerlukan pengembangan dan pemeliharaan manual yang lebih intensif. Pilihan antara menggunakan framework atau aplikasi native sering kali tergantung pada kebutuhan proyek, keterampilan tim pengembang, dan preferensi arsitektur.