MANAJEMEN KEAMANAN SIBER

Dosen Pengampu :

Dr. Niskarto Zendrato S.Kom., M.Kom

DISUSUN OLEH :

Grant Gabriel Tambunan (221402057)

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

SEPTEMBER 2025

Dokumentasi Tugas:

https://github.com/niskarto89/Aplikasi-Web-Verifikasi-Unduhan-PDF-Bebras-Challenge-2025-

analisis source code diatas, web diatas bisa di hosting di localhost atau di dewacloud atau hosting lainnya
Buat source code baru yang sudah di perbaiki,

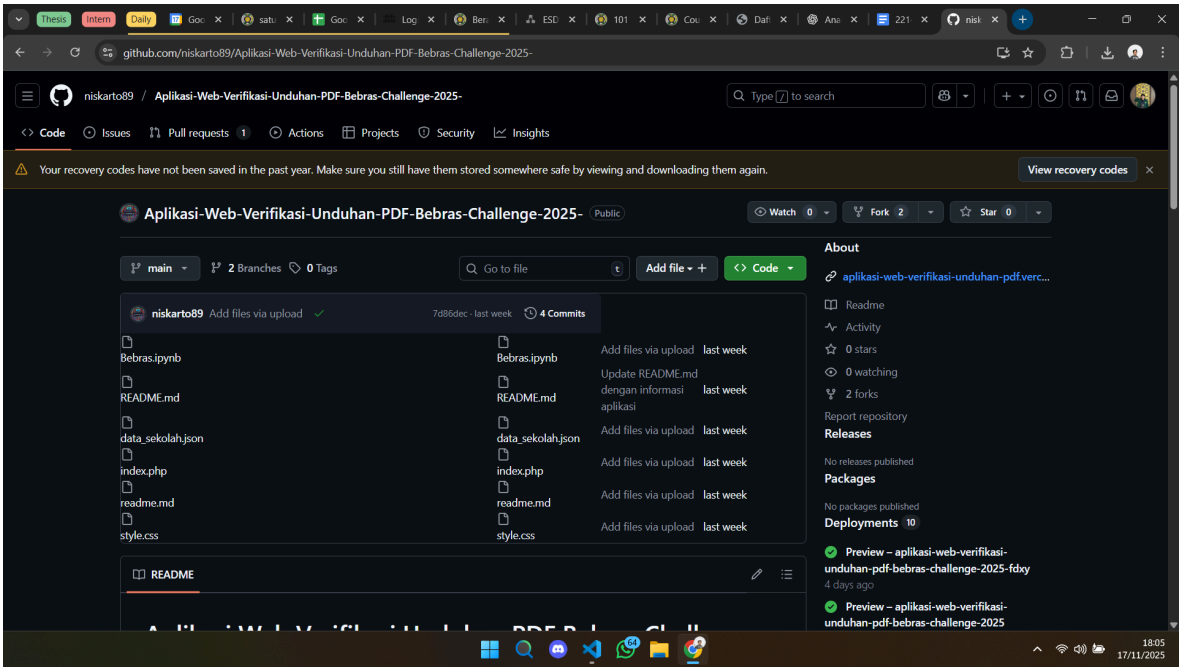
Salah satu celah security di file json dan index nya sudah ada terdapat kode verifikasi
buat laporan mitigasinya terkait analisis yang anda dapatkan dan pencegahannya

kumpulkan link githubnya dengan code yang sudah diperbaiki di penugasan ini, dan report mitigasinya lampirkan bentuk pdf

Contioh laporan mitigasi ada di pertemuan sebelumnya

Write Up:

Diberikan sebuah repository github, kurang lebih berisi seperti ini

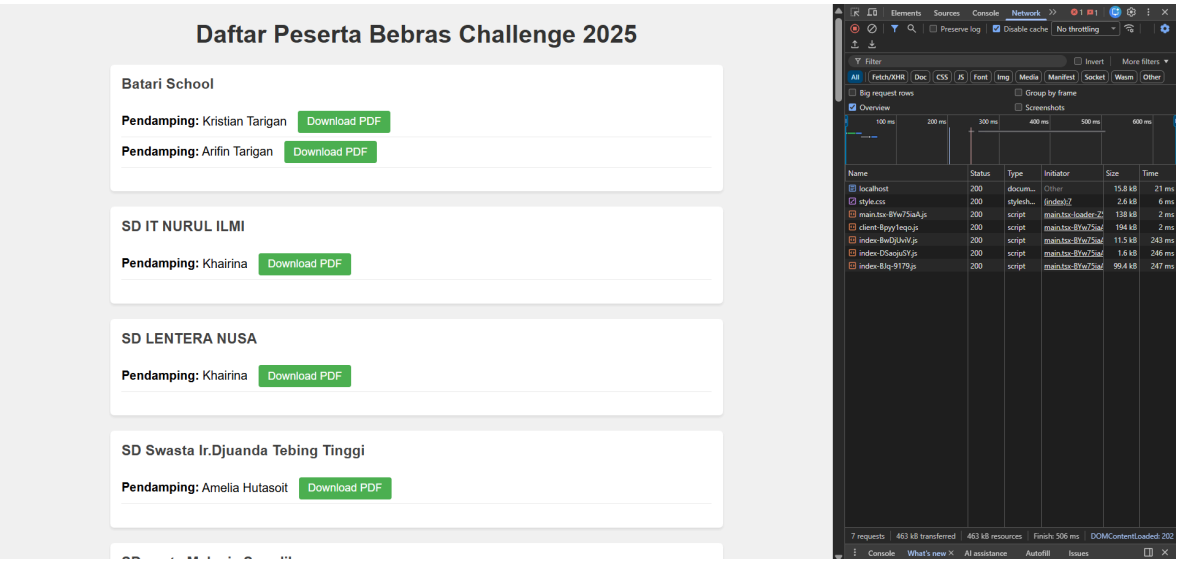


Code tersebut dilakukan hosting lokal, lalu dianalisa kerentanan dalam file .json dan index.php nya, lalu lakukan patch kerentanan itu, commit githubnya dan laporannya dibuat disini.

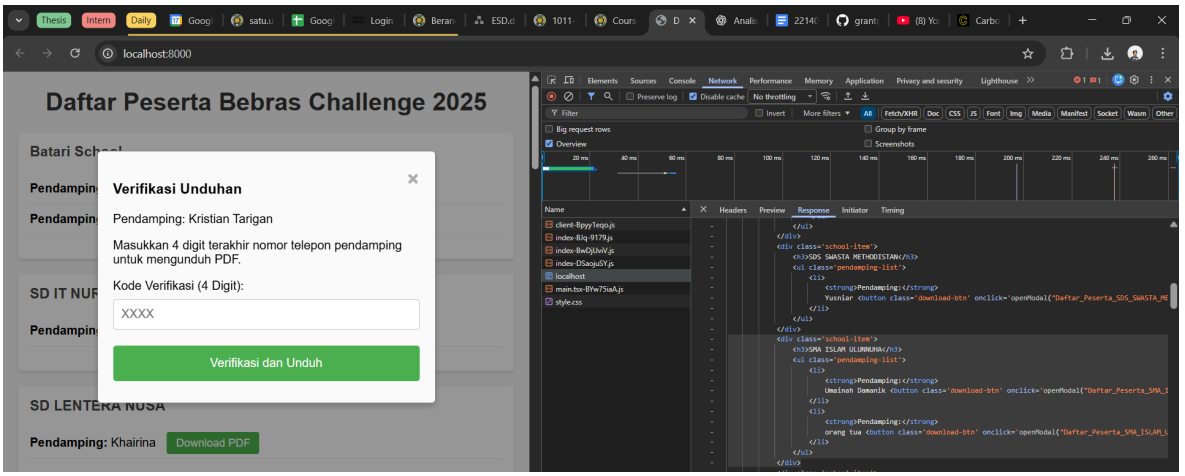
Pertama, setelah melakukan fork repository ke github saya, saya mengclone githubnya dengan perintah:

D:\Kulyeh\Materials And Homework\Semester 7\MKS\> git clone https://github.com/grantgabriel/Aplikasi-Web-Verifikasi-Unduhan-PDF-Bebras-Challenge-2025.git

Selanjutnya, saya menjalankan web secara lokal dengan perintah php -S localhost:8000. Lalu web akan berjalan di index:8000.



Dokumentasi tugas menyatakan bahwa terdapat celah pada file .json, dan index.php, dimana terdapat kode verifikasi. Setelah melihat network changes dari web, file data_sekolah.json tidak ada di fetch, yang menandakan file tersebut diolah secara client-side. Selanjutnya, saya membuka file docs dari localhost yang saya jalankan. Dan ternyata benar, 4 kode verifikasinya muncul di client-side




```
<div class='school-item'>
    <h3>Batari School</h3>
    <ul class='pendamping-list'>
      <li>
          <strong>Pendamping:</strong>
          Kristian Tarigan <button class='download-btn' onclick='openModal("Daftar_Peserta_Batari_School.pdf", "7865", "Kristian
Tarigan")'>Download PDF</button>
      </li>
      <li>
          <strong>Pendamping:</strong>
          Arifin Tarigan <button class='download-btn' onclick='openModal("Daftar_Peserta_Batari_School.pdf", "2758", "Arifin
Tarigan")'>Download PDF</button>
      </li>
    </ul>
</div>
```

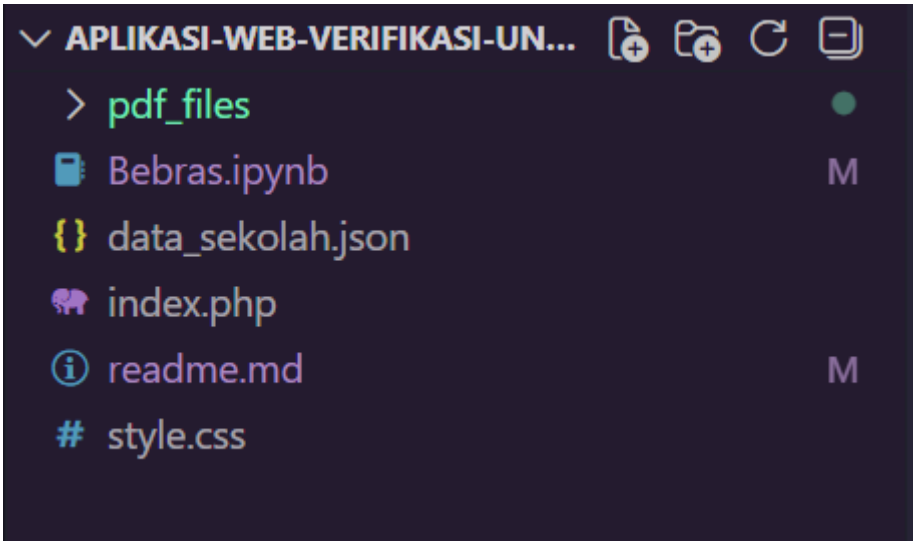
Kode verifikasi ini penting, untuk mengunduh file PDF terkait sekolah tersebut, dan pengguna dapat dengan mudah mendapatkan kode nya dengan meng-inspect website tersebut. Ini menunjukkan bahwa sepertinya proses verifikasi dilakukan secara client-side, padahal verifikasi ini seharusnya berjalan di server-side. Akibatnya, dapat terjadi kebocoran data. Secara singkat, analisa yang saya dapatkan adalah:

- data_sekolah.json berisi data sekolah + nomor kontak dan/atau kode verifikasi.
- index.php (implementasi client-side) melakukan verifikasi 4-digit di client dengan membandingkan data JSON → kode verifikasi terekspos ke siapa pun yang membuka source (serious security issue).
- Potensi masalah lain: file PDF dapat diakses jika attacker menebak nama file → Insecure Direct Object Reference (IDOR) dan data leakage.

Kerentanan ini berersiko untuk:

- Pembocoran kode verifikasi & nomor kontak.
- Siapa saja bisa mendownload PDF jika tahu nama file.
- Data sensitif tersimpan di repo publik.

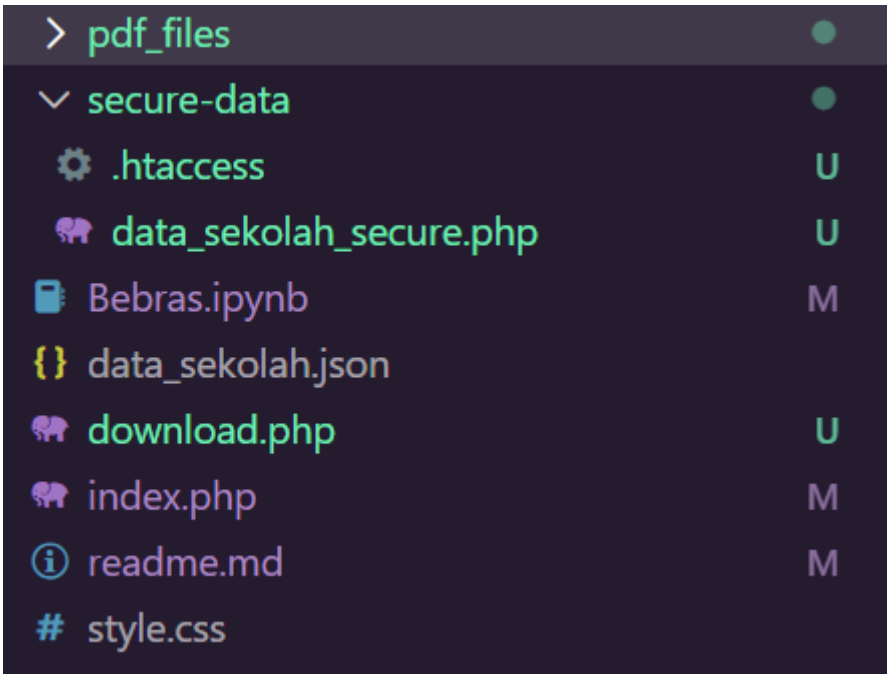
Mitigasi dan solusi yang tepat bisa dilakukan untuk mencegah hal tersebut terjadi. Sekarang, saya akan melakukan patching terhadap sistem tersebut. Saya menjalankan isi dari python notebook Bebras.ipynb secara keseluruhan untuk mendapatkan semua pdf files. Lalu saya membuat direktori pdf_files dan memindahkan nya ke direktori proyek seperti ini



Masuk ke tahap patch, saat ini kita bisa melihat bahwa di index.php, data files di load langsung, dan juga di load data nya dari data_sekolah.json.

```
<div class="school-list">
    <?php
    // Baca file JSON
    $json_file = 'data_sekolah.json';
    $json_data = file_get_contents($json_file);
    if ($json_data === false) {
        die("Error: Tidak dapat membaca file $json_file.");
    }
    $data = json_decode($json_data, true);
    if (json_last_error() !== JSON_ERROR_NONE) {
        die("Error: Format JSON tidak valid di $json_file. " . json_last_error_msg());
    }
```

Sekarang, mitigasi yang akan saya lakukan ada dua, pertama mengubah logic dari verifikasi file dari client-side ke server-side. Selanjutnya juga mendefine file .htaccess agar tidak bisa mengakses file melalui nama file nya saja, harus dari verifikasi. Berikut adalah gambaran akhir dari patch saya.



Saya akan menjelaskan beberapa mitigasi saya. Pertama .htaccess di folder pdf_files, berisi seperti ini

deny from all

Gunanya untuk mengdeny semua akses ke folder tersebut tanpa jalur yang jelas. Selanjutnya .htaccess dalam folder secure_data

Require all denied

Berguna untuk blokir semua akses langsung ke file sensitif. Selanjutnya file data_sekolah_secure.php yang berisi data-data dari data_sekolah.json dalam bentuk file PHP. Data tersebut nanti dikelola secara server side. Berikut adalah contoh penggalan dari data tersebut dalam bentuk array PHP.

```
"28" => [
    "nama_sekolah" => "SMAS Islam Ulun Nuha Medan",
    "pendamping" => [
         "1" => ["nama" => "Aulia Fitri", "kode_verifikasi" => "9399",
"pdf_file" => "Daftar_Peserta_SMAS_Islam_Ulun_Nuha_Medan.pdf"],
          "2" => ["nama" => "Dwi Indah Permata", "kode_verifikasi" =>
"2695",                          "pdf_file"                          =>
"Daftar_Peserta_SMAS_Islam_Ulun_Nuha_Medan.pdf"],
    ]
],
```

Selanjutnya, ada file download.php, disini saya memindahkan logic dari unduh berkas pdf dari client-side ke server-side. Jadi ketika pengguna memasukkan 4 angka verifikasi, file download.php akan dijalankan dan logic mengunduh akan dijalankan disini, tidak akan masuk ke network changes di sisi client. Berikut adalah kode dari download.php

```php
<?php

// Pastikan request hanya dari POST
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    header("Location: index.php?error=1");
    exit;
}

$schoolId     = $_POST['school_id'] ?? "";
$pendampingId = $_POST['pendamping_id'] ?? "";
$kodeInput    = $_POST['kode_verifikasi'] ?? "";

// Validasi format dasar kode (harus 4 digit)
if (!preg_match('/^[0-9]{4}$/', $kodeInput)) {
    header("Location: index.php?error=1");
    exit;
}

// Load data aman (tidak dikirim ke browser)
```

```php
$data = require __DIR__ . "/secure-data/data_sekolah_secure.php";

// Validasi ID sekolah
if (!isset($data[$schoolId])) {
    header("Location: index.php?error=1");
    exit;
}

// Validasi ID pendamping pada sekolah tsb
if (!isset($data[$schoolId]['pendamping'][$pendampingId])) {
    header("Location: index.php?error=1");
    exit;
}

$entry = $data[$schoolId]['pendamping'][$pendampingId];

// Verifikasi kode dengan aman
if (!hash_equals($entry['kode_verifikasi'], $kodeInput)) {
    header("Location: index.php?error=1");
    exit;
}

// Ambil file PDF yang sesuai
$pdf  = $entry['pdf_file'];
$path = __DIR__ . "/pdf_files/" . $pdf;

// Pastikan file ada
if (!is_file($path)) {
    header("Location: index.php?error=1");
    exit;
}

// Kirim file ke browser
header("Content-Type: application/pdf");
header("Content-Disposition:      attachment;      filename=\""      .
basename($pdf) . "\"");
header("Content-Length: " . filesize($path));

readfile($path);
exit;
```

Saya juga menyiapkan berbagai mitigasi, seperti ketika file tidak ada, ketika kode verifikasi salah, dan lain lain. Ketika hal tersebut terjadi, pengguna akan diarahkan ke page error dan dikembalikan ke page awal. Terakhir, saya mengubah logic dari index.php agar tidak melakukan verifikasi secara sisi client, dan menghilangkan kode verifikasi yang ada di inspect. Berikut kode akhirnya

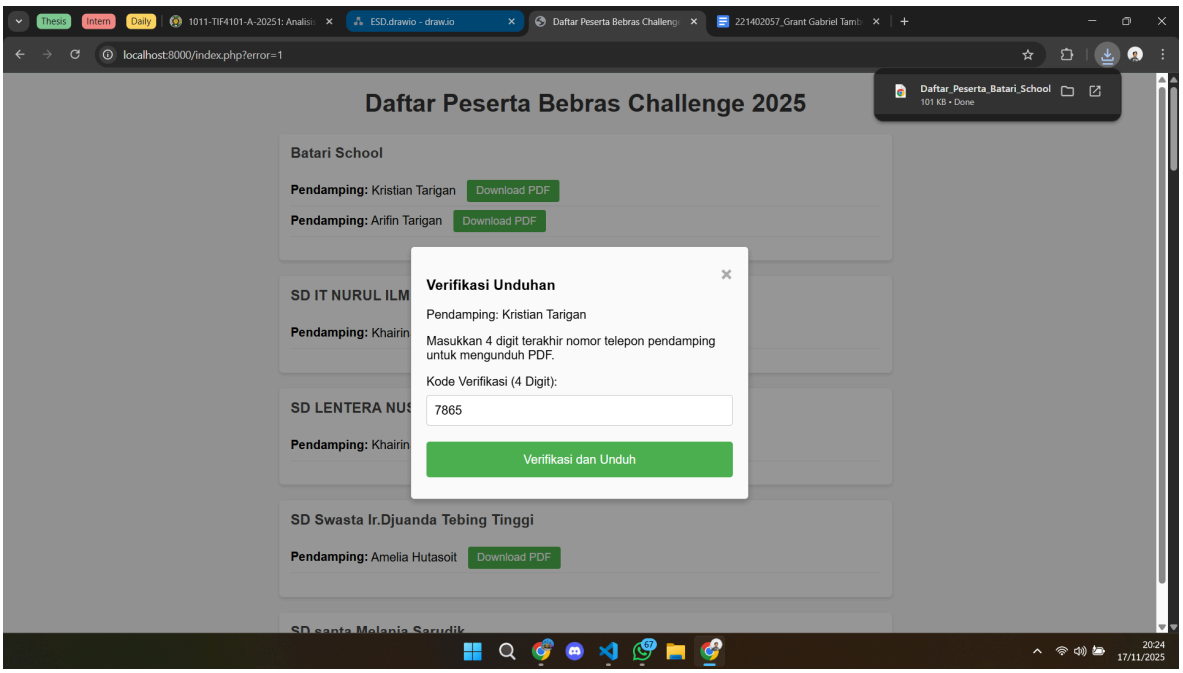
```
<?php
// Load data sekolah aman
$data = require __DIR__ . "/secure-data/data_sekolah_secure.php";

// Urutkan berdasarkan nama sekolah (alfabet)
$sorted = [];
foreach ($data as $id => $row) {
      $sorted[$row['nama_sekolah']] = ["id" => $id, "pendamping" =>
$row['pendamping']];
}
ksort($sorted);
?>

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
                <meta    name="viewport"    content="width=device-width,
initial-scale=1.0">
    <title>Daftar Peserta Bebras Challenge 2025</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

<h1>Daftar Peserta Bebras Challenge 2025</h1>

<div class="school-list">

<?php
// Loop sekolah (UI tetap sama)
foreach ($sorted as $school_name => $info):
    $schoolId = $info['id'];
    $pendamping_list = $info['pendamping'];
?>
    <div class='school-item'>
        <h3><?= htmlspecialchars($school_name) ?></h3>
        <ul class='pendamping-list'>
```

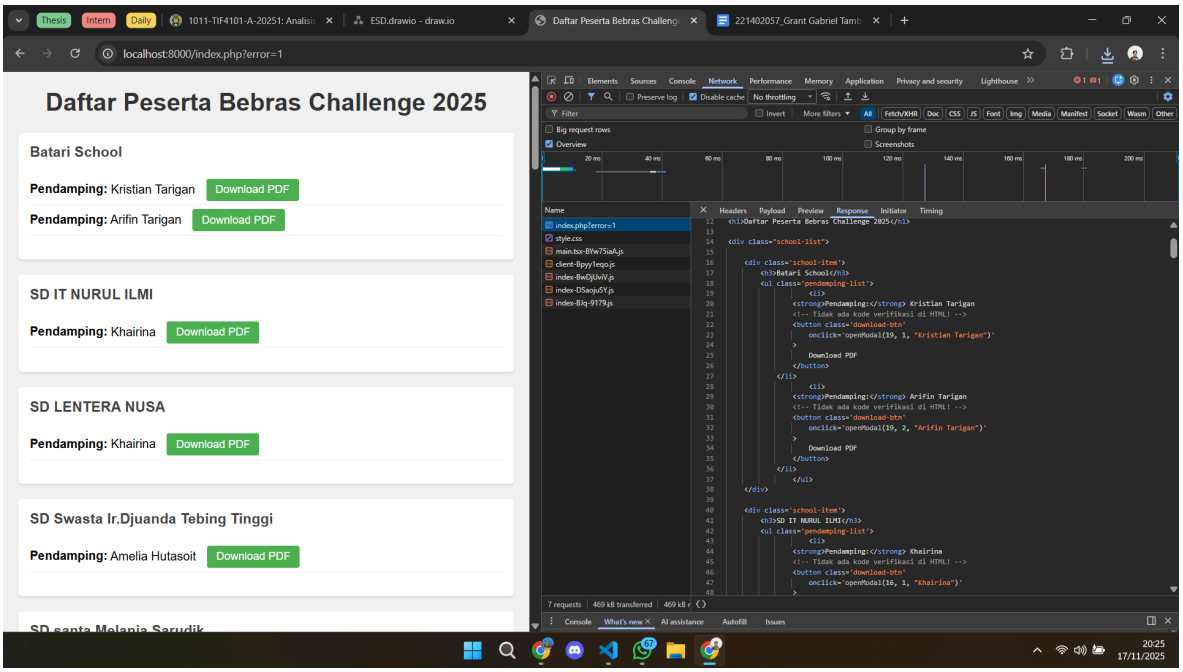
```
                <?php foreach ($pendamping_list as $pendampingId =>
$pendamping): ?>
            <li>
                                        <strong>Pendamping:</strong> <?=
htmlspecialchars($pendamping['nama']) ?>

                <!-- Tidak ada kode verifikasi di HTML! -->
                <button class='download-btn'
                    onclick='openModal(<?= json_encode($schoolId) ?>,
<?=            json_encode($pendampingId)            ?>,            <?=
json_encode($pendamping["nama"]) ?>)'
                >
                    Download PDF
                </button>
            </li>
        <?php endforeach; ?>
        </ul>
    </div>

<?php endforeach; ?>

</div>

<!-- The Modal -->
<div id="myModal" class="modal">
    <div class="modal-content">
        <span class="close" onclick="closeModal()">&times;</span>
        <h3>Verifikasi Unduhan</h3>
        <p id="modal-pendamping-name"></p>
          <p>Masukkan 4 digit terakhir nomor telepon pendamping untuk
mengunduh PDF.</p>

        <!-- Form menuju download.php -->
                        <form id="verificationForm" method="POST"
action="download.php">
            <input type="hidden" id="schoolIdInput" name="school_id">
                        <input type="hidden" id="pendampingIdInput"
name="pendamping_id">

                    <label for="verificationCode">Kode Verifikasi (4
Digit):</label>
                            <input type="text" id="verificationCode"
name="kode_verifikasi" placeholder="XXXX" maxlength="4" required>
```

```
            <div id="errorMessage" class="error-message"></div>

            <button type="submit">Verifikasi dan Unduh</button>
        </form>
    </div>
</div>

<script>
    const modal = document.getElementById("myModal");

    function openModal(schoolId, pendampingId, pendampingName) {
        document.getElementById("schoolIdInput").value = schoolId;
                document.getElementById("pendampingIdInput").value  =
pendampingId;
          document.getElementById("modal-pendamping-name").textContent
= "Pendamping: " + pendampingName;

        document.getElementById("verificationCode").value = "";
        document.getElementById("errorMessage").textContent = "";

        modal.style.display = "block";
    }

    function closeModal() {
        modal.style.display = "none";
    }

    window.onclick = function(event) {
        if (event.target == modal) closeModal();
    };

     // Client-side validation ONLY (UI), verifikasi tetap dilakukan
server-side

document.getElementById("verificationForm").addEventListener("submit"
, function(event) {
                                                const      code      =
document.getElementById("verificationCode").value.trim();
        const errorDiv = document.getElementById("errorMessage");

        errorDiv.textContent = "";
```

```
        if (code.length !== 4 || isNaN(code)) {
                errorDiv.textContent = "Kode harus berupa 4 digit
angka.";
            event.preventDefault();
            return;
        }
    });
</script>

</body>
</html>
```

Pada akhirnya sekarang, sistem tetap berkerja, file pdf tetap bisa terunduh jika kode verifikasi benar, dan kode verifikasi tidak akan terlihat secara sisi klien dan pengguna tidak akan bisa mengakses file bagaimanapun tanpa melakukan verifikasi





Dengan demikian, berakhir mitigasi dari sistem tersebut. Saya telah melakukan patching dan update system agar tidak celah tersebut tidak bisa dieksploitasi. Kode lengkap dapat dilihat disini

https://github.com/grantgabriel/Aplikasi-Web-Verifikasi-Unduhan-PDF-Bebras-Challenge-2025