LAPORAN PRAKTIKUM 8

SISTEM KEAMANAN INFORMASI

Praktikum Cross Site Scripting

Untuk Memenuhi Salah Satu Tugas

Mata Kuliah Sistem Keamanan Informasi

Dosen Pengampu: Lindung Siswanto, S.Kom., M.Eng.

Disusun Oleh:

Sany Adika Prayata

NIM 3202116097

No. Presensi 20

PROGRAM STUDI D-III TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI PONTIANAK

2024

KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Allah SWT atas rahmat dan karunia-Nya sehingga laporan yang berjudul “Praktikum Cross Site Scripting” dapat terselesaikan dengan baik. Laporan ini merupakan salah satu tugas yang diberikan oleh dosen pengampu mata kuliah Sistem Keamanan Informasi kepada mahasiswa Program Studi D3 Teknik Informatika Jurusan Teknik Elektro sebagai salah satu bagian dari komponen penilaian akademis.

Laporan ini membahas terkait percobaan penyerangan dengan menggunakan cross site scripting dan pencegahannya. Demikian Laporan ini saya buat, semoga bermanfaat.

Pontianak, 16 Juli 2024

Penyusun,

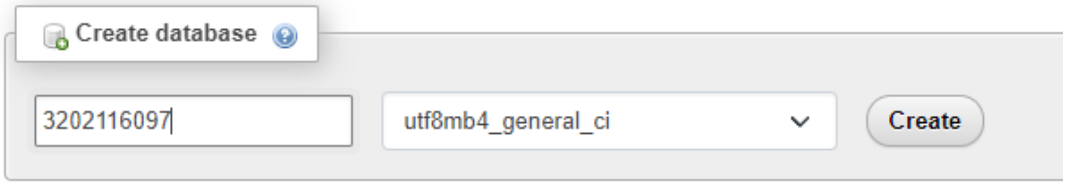
(Sany Adika Prayata)

DAFTAR ISI

DAFTAR GAMBAR

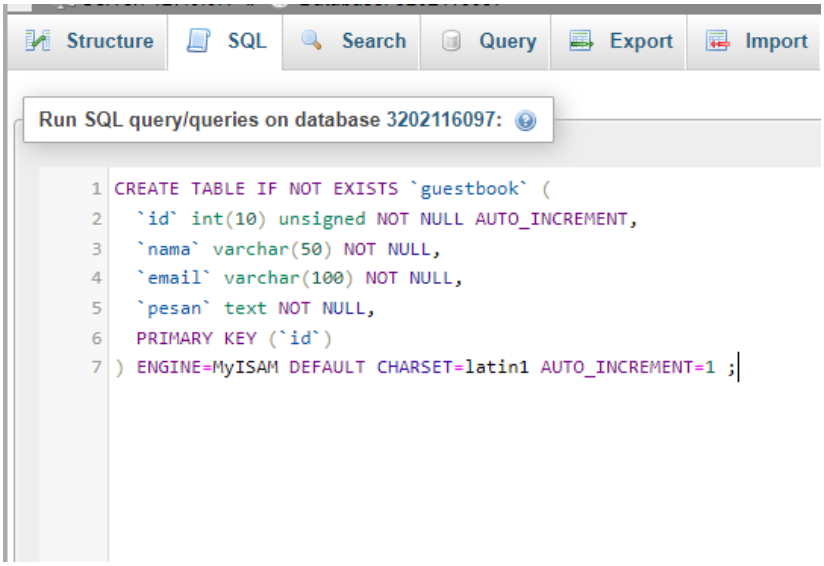
1. Percobaan XSS *Non Persistent*

a. Buat *database* sesuai NIM pada phpmyadmin



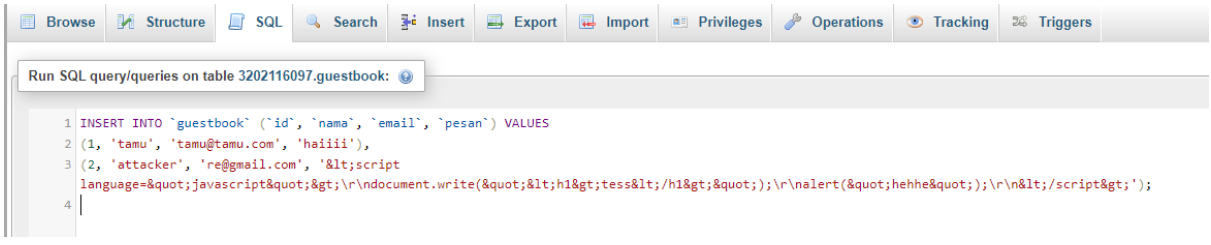
Gambar 1. Pembuatan *Database* pada phpmyadmin

b. Buat tabel pada *database* melalui *tab* SQL



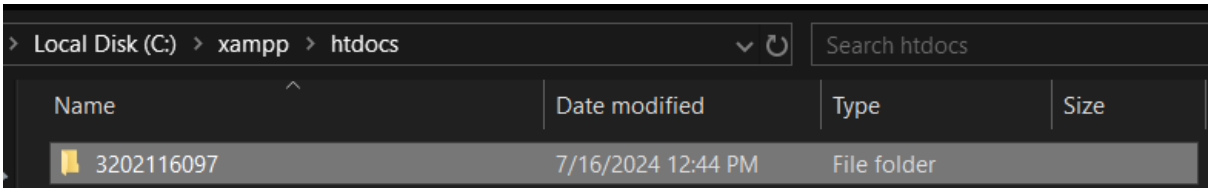
Gambar 2. Pembuatan Tabel dalam *Database*

c. Tambah data pada tabel



Gambar 3. Penambahan Data ke Dalam Tabel

d. Buat direktori sesuai NIM pada direktori root



Gambar 4. *Folder* NIM pada Direktori htdocs

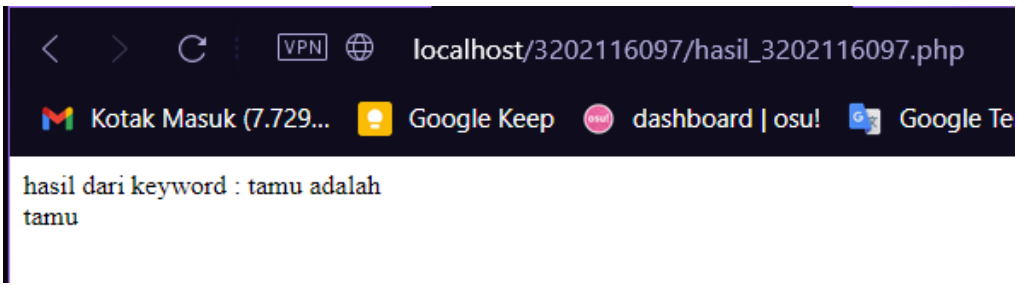
e. Buat *file* search.php berisikan kode seperti berikut

```
<form action="hasil_3202116097.php" method="GET">
Search : <input type="text" name="q">
<br>
<input type="submit" name="submit" value="Cari">
</form>
```

f. Buat *file* hasil_3202116097.php berisikan kode seperti berikut

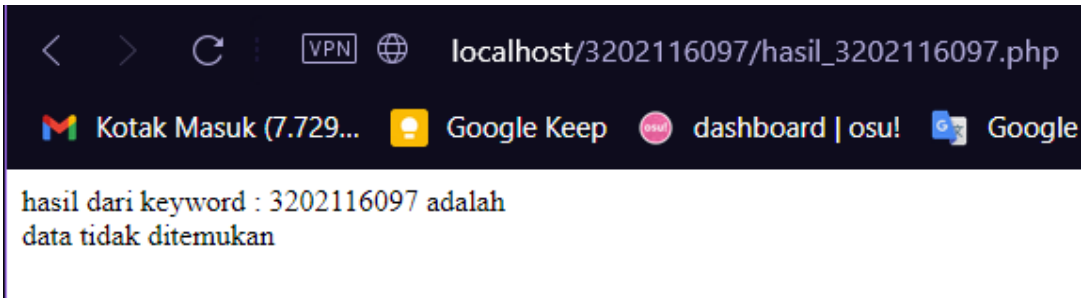
```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "3202116097";

// Membuat koneksi
$conn = new mysqli($servername, $username, $password, $dbname);

// Cek koneksi
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}

$q = @$_GET['q'];
if(isset($_GET['submit'])) {
    $sql = "SELECT * FROM guestbook WHERE nama LIKE '%$q%'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        while($data = $result->fetch_assoc()) {
            echo "hasil dari keyword : $q adalah <br>";
            echo $data['nama'] . "<br>";
        }
    } else {
        echo "hasil dari keyword : $q adalah <br>";
        echo "data tidak ditemukan";
    }
} else {
    echo "tidak ada data yang dicari";
}

$conn->close();
?>
```

g. Mencoba mencari data yang ada pada *database*



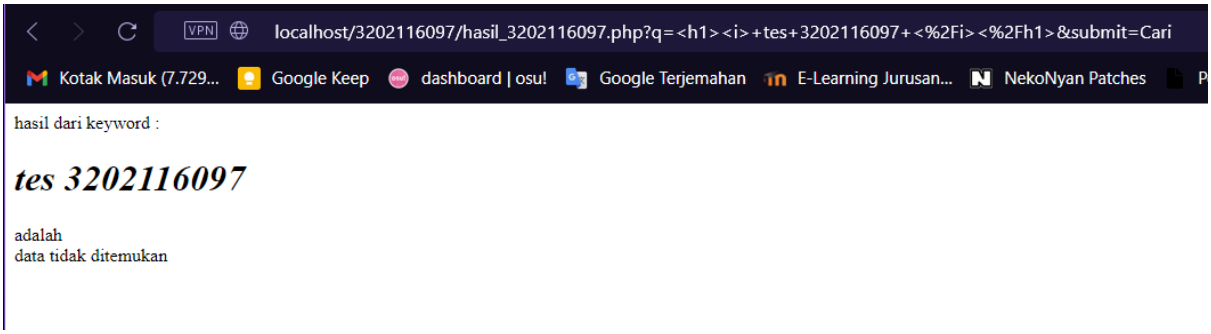
Gambar 5. Contoh *Output* Pencarian Data yang Ada pada *Database*

h. Mencoba mencari data yang tidak ada pada *database*



Gambar 6. Contoh *Output* Pencarian Data yang Tidak Ada pada *Database*

i. Mencoba memasukkan kode HTML pada kolom *search*



Gambar 7. *Output* yang Dihasilkan Ketika Memasukkan Kode HTML

j. Pencegahan

1. Mengganti *method* GET pada *file* search.php menjadi *method* POST

Kode sebelumnya:

```
<form action="hasil_3202116097.php" method="GET">
```

Kode setelahnya:

```
<form action="hasil_3202116097.php" method="POST">
```

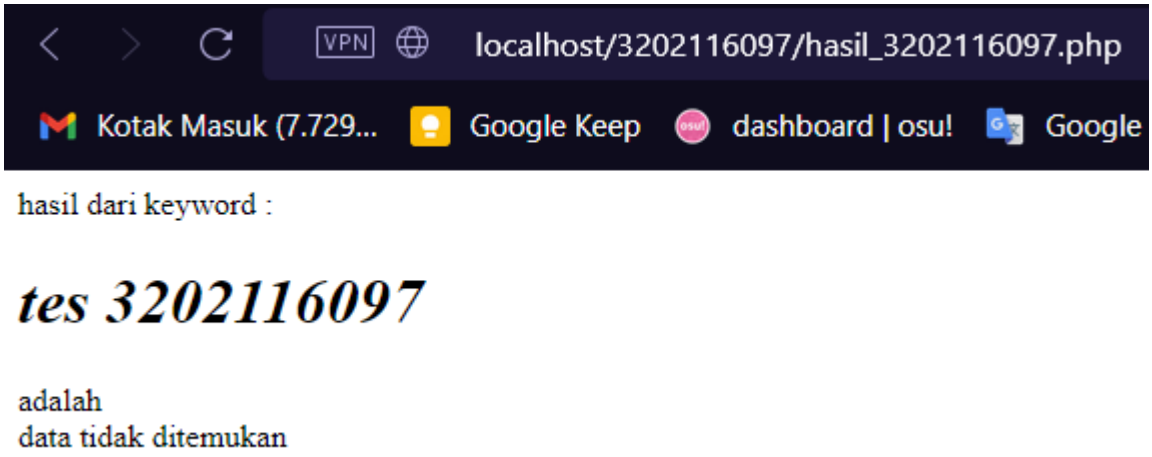
2. Mengganti *method* GET pada *file* hasil_3202116097.php menjadi *method* POST

Kode sebelumnya:

```
$q = @$_GET['q'];
if(isset($_GET['submit']))
```

Kode setelahnya:

```
$q = @$_POST['q'];
if(isset($_POST['submit']))
```



Gambar 8. Tampilan URL Setelah Menggunakan Method POST

Hasil analisa:

Method GET lebih rentan karena data yang dikirim menggunakan *method* GET akan ditambahkan sebagai parameter *query* yang dapat dilihat dan di-*edit* oleh pengguna. Sedangkan data yang dikirim menggunakan *method* POST dikirim sebagai request.body sehingga data lebih aman karena tidak dapat terlihat pada URL. Untuk pencegahan serangan XSS, selain menggunakan *method* POST dapat juga menambahkan validasi *input* dengan *function* htmlspecialchars dan filter_input.

2. Percobaan XSS *Persistent*

a. Buat *file* koneksi_3202116097.php berisikan kode seperti berikut

```php
<?php
$host = "localhost";
$uname = "root";
$pass = "";
$db = "3202116097";

$koneksi = new mysqli($host, $uname, $pass, $db);

if ($koneksi->connect_error) {
    die("Connection failed: " . $koneksi->connect_error);
}

echo "Connected successfully";
?>
```

b. Buat *file* indeks.php berisikan kode seperti berikut

```
<title>Buku Tamu</title>
<p class="judul"> Buku Tamu</p>
<form method="post" action="proses_3202116097.php">
  Nama <br>  <input class="kotak"
type="text" name="nama" size="34"><br>
  Email <br>  <input class="kotak"
type="text" name="email" size="34"><br>
  Pesan <br>  <textarea class="kotak"
name="pesan" rows="8" cols="40"></textarea>
<br>  (*) Wajib diisi <br>
<p>  <input class="tombol" type="reset"
value="Reset" name="kosongkan">
<input type="submit" class="tombol" value="Kirim" name="kirim">
</form>
<p>  [ <a href="lihatgb_3202116097.php">Lihat Data Tamu</a>]
```

c. Buat *file* proses_3202116097.php berisikan kode seperti berikut

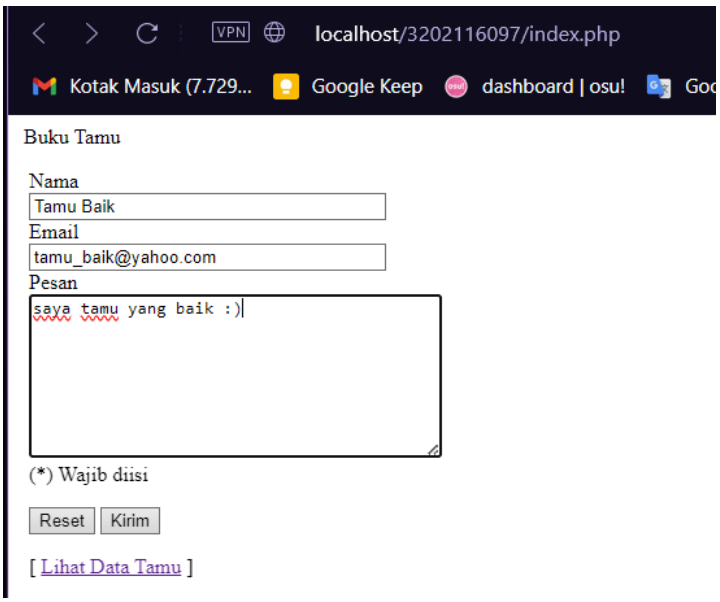
```
<style type="text/css">
body { background:#fff;
       color:#000;
       font-family:tahoma;
       font-size:15px;
}
</style>
<?php
include 'koneksi_3202116097.php';

if(isset($_POST['kirim']) && ($_POST['kirim'] === 'Kirim')) {
    $nama = $_POST['nama'];
    $email = $_POST['email'];
    $pesan = $_POST['pesan'];

    if(empty($nama) || empty($email) || empty($pesan)) {
        $error = "Data tidak lengkap.<br>";
    }

    if(empty($error)) {
        $stmt = $koneksi->prepare("INSERT INTO guestbook(nama,
 email, pesan) VALUES (?, ?, ?)");
        $stmt->bind_param("sss", $nama, $email, $pesan);

        if($stmt->execute()) {
            echo "<b>Guestbook berhasil disimpan</b>";
            echo "<meta http-equiv=\"refresh\" content=\"2;
url=./\">";
        } else {
            echo "<font color=red><b>Data gagal disimpan</b>
</font>";
            echo "<meta http-equiv=\"refresh\" content=\"2;
url=./\">";
        }

        $stmt->close();
    }
}

if(isset($error)) {
    echo "<font color=red><b>$error</b></font>";
    echo "<meta http-equiv=\"refresh\" content=\"2; url=./\">";
}
?>
```

d. Buat *file* lihatgb_3202116097.php berisikan kode seperti berikut

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Data Tamu</title>
    <style type="text/css">
        body {
            background:#fff;
            border:1px solid #444444;
            font-family:tahoma;
            font-size:12px;
            color:#000;
        }
        .kotak {
            background:#e3ebf3;
            border:1px solid #444444;
            color:#000;
            font-family:tahoma;
            font-size:13px;
        }
        .kotak:hover {
            border-bottom:1px solid #aac7e5;
            border-top:1px solid #aac7e5;
            background : #2b4661;
        }
        td {
            background:#e3ebf3;
            border:1px solid #444444;
            color:#000;
            font-family:tahoma;
            font-size:13px;
        }
        td:hover {
            background : #dedede;
        }
        a{
            color:#000;
            text-decoration:none;
        }
        a:hover {
            color:#2b4661;
            text-decoration:none;
        }
    </style>
</head>
```

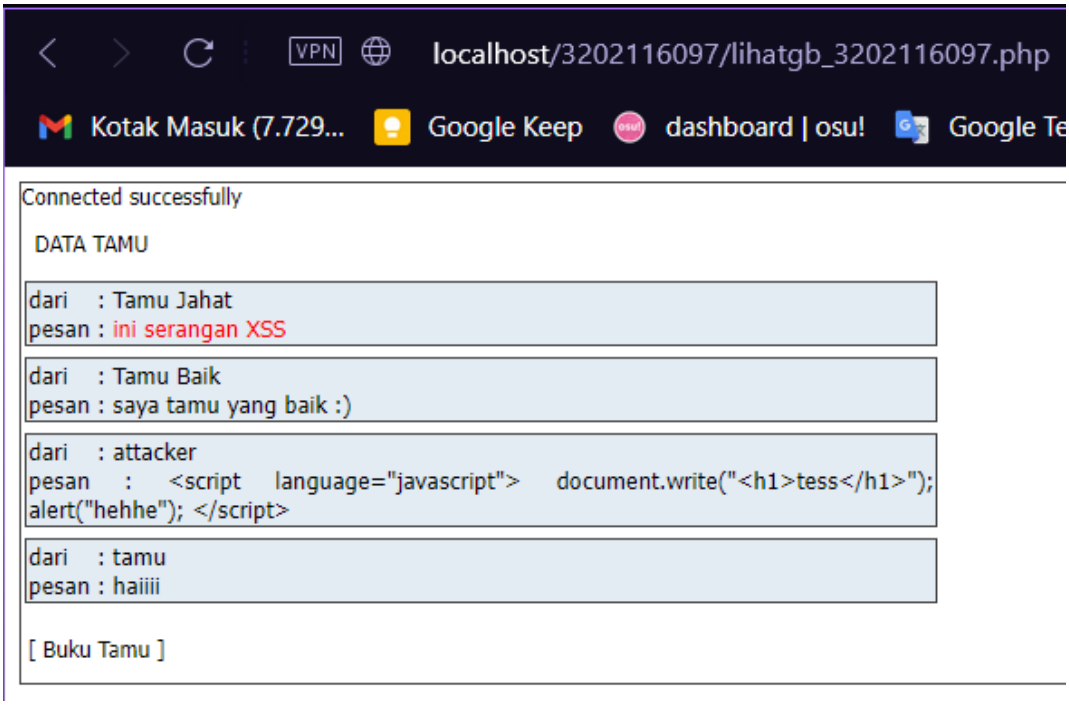
```
<body>
    <table>
    <?php
    // Ubah ke penggunaan MySQLi
    include 'koneksi_3202116097.php';
    $queryGb = mysqli_query($koneksi, "SELECT * FROM guestbook");
    $jmlGb = mysqli_num_rows($queryGb);
    if ($jmlGb == 0) {
        echo "Data tamu masih kosong";
    } else {
        $queryTampil = mysqli_query($koneksi, "SELECT * FROM
 guestbook ORDER BY id DESC");
        echo "<p>  DATA TAMU</p>";
        echo "<table width=\"508\">";
        while ($dataGb = mysqli_fetch_array($queryTampil)) {
            echo "<tr><td align=\"justify\">dari   
 : <a href=\"mailto:$dataGb[2]\">$dataGb[1]</a><br>";
            echo "pesan : $dataGb[3]</td></tr>";
            echo "<tr><th></th></tr>";
        }
        echo "</table>";
    }
    ?>
    <p> [ <a href=".">Buku Tamu</a> ]</p>
    </table>
</body>
</html>
```

e. Masukkan data seperti contoh berikut



Gambar 9. Contoh Isi Data Tamu Baik

Gambar 10. Contoh Isi Data Tamu Jahat



Gambar 11. Tampilan Isi Data Buku Tamu

f. Coba masukkan kode HTML pada *form* pesan

1. *Script* A

```
<script language="javascript">document.write("<h2>Attacked by
3202116097</h2>");alert("Saya berhasil...!!!");</script>
```

Gambar 12. Dampak *Script* A

Dampak yang dihasilkan oleh kode di atas adalah muncul *alert* pada *browser*.

2. *Script* B

Buat *file* baru pada *root directory* bernama hacked_3202116097.php berisikan kode berikut

```
<font color=red><h1>Anda sudah di HACKED oleh 3202116097
</h1></font>
```

Kemudian isi *form* pesan dengan kode berikut

```
<iframe src="hacked_3202116097.php" height=200 width=400
frameborder=1 align=center></iframe>
```

Gambar 13. Dampak *Script* B

Dampak yang dihasilkan oleh kode di atas adalah *tag* iframe akan menampilkan isi konten dari *file* hacked_3202116097.php .

3. *Script C*

```
<script>alert(document.cookie)</script>
```

Gambar 14. Dampak *Script* C

Dampak pada kode di atas adalah akan membuat *alert* yang menampilkan *cookie*.

g. Pencegahan

1. Menggunakan htmlentities

Ubah kode menjadi seperti berikut

```
$nama = trim(htmlentities($_POST['nama']));
$email = trim(htmlentities($_POST['email']));
$pesan = trim(htmlentities($_POST['pesan']));
```

Output yang dihasilkan:

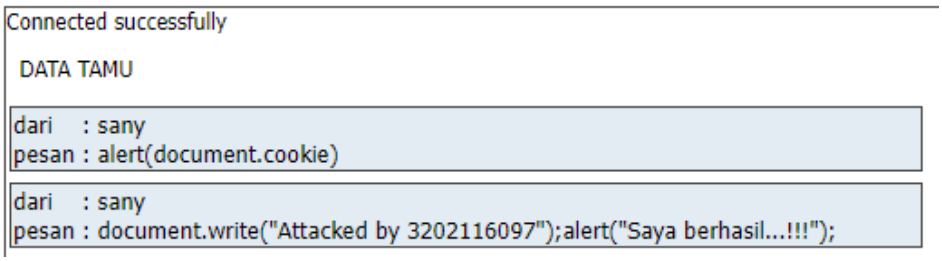
Gambar 15. Tampilan Data yang Tersimpan Ketika Menggunakan htmlentities

Ketika menggunakan htmlentities kode html tidak akan dieksekusi, melainkan mengubah entitas html menjadi sebuah string biasa.

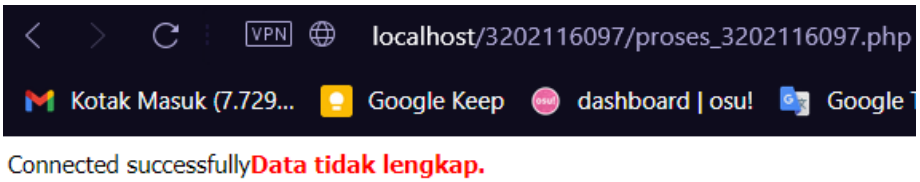
2. Menggunakan strip_tags

```
$nama = trim(strip_tags($_POST['nama']));
$email = trim(strip_tags($_POST['email']));
$pesan = trim(strip_tags($_POST['pesan']));
```

Output:



Gambar 16. Tampilan Data yang Tersimpan Ketika Menggunakan strip_tags



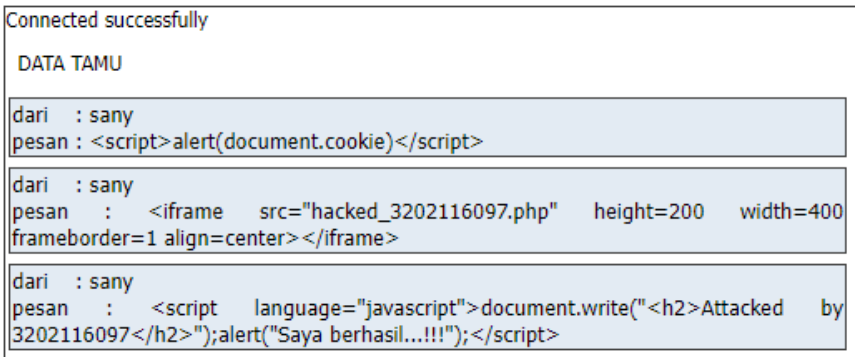
Gambar 17. *Output* yang Didapat Ketika Mengirimkan *Script* B

Ketika menggunakan strip_tags, segala tag html akan dihapus.

3. Menggunakan htmlspecialchars

```
$nama = trim(htmlspecialchars($_POST['nama']));
$email = trim(htmlspecialchars($_POST['email']));
$pesan = trim(htmlspecialchars($_POST['pesan']));
```

Output:



Gambar 18. Tampilan Data yang Tersimpan Ketika Menggunakan htmlspecialchars

Mirip dengan htmlentities, namun htmlspecialchars tetap akan mengkonversi html entity.