

# LAPORAN PRAKTIKUM PEMROGRAMAN WEB SERVER SIDE

## Modul 6

Judul Praktikum : Sanitasi dan Validasi Input dari Form  
Tanggal Praktikum : 26 April 2025  
Tempat : Lab Manajemen Ritel



NIM : 235410075  
Nama Mahasiswa : Wildan Wigenta  
Dosen Pengampu : Ir. M. Guntara, M.T.  
Asisten : -

**PRODI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**LABORATORIUM**  
**UTDI**

Sem. Genap TA. 2024/2025

# Analisis

## PRAKTIK

### A. Indexed htmlspecialchars() Praktik

Program:

```
testchar.php X
mod6 > testchar.php > ...
2  <!-- form.html -->
3  <form method="post" action="">
4      Nama: <input type="text" name="nama">
5      <input type="submit" value="Kirim">
6  </form>
7
8  <?php
9      if ($_SERVER["REQUEST_METHOD"] == "POST") {
10         $nama = $_POST["nama"];
11         echo "Tanpa htmlspecialchars:<br>";
12         echo $nama;
13
14         echo "<br>Dengan htmlspecialchars:<br>";
15         echo htmlspecialchars($nama);
16     }
17
18     echo "<br><br>Contoh penggunaan ".htmlspecialchars("&lt; dan &gt;"). "<br>";
19     $teks_mentah = "<script>alert('Halo!');</script> Teks biasa > lainnya.";
20     echo "Teks Mentah: ". $teks_mentah . "<br>";
21     $teks_aman = htmlspecialchars($teks_mentah);
22     echo "Teks Aman (setelah htmlspecialchars): ". $teks_aman;
23     ?>
```

Output:

localhost/pwss/mod6/testchar.php

Pengaruh penggunaan htmlspecialchars

Nama:

Tanpa htmlspecialchars:

Dengan htmlspecialchars:  
<input type="text">

Contoh penggunaan &lt; dan &gt;

Teks Mentah: Teks biasa > lainnya.  
Teks Aman (setelah htmlspecialchars): <script>alert('Halo!');</script> Teks biasa > lainnya.

localhost/pwss/mod6/testchar.php

Pengaruh penggunaan htmlspecialchars

Nama:

Tanpa htmlspecialchars:  
WILDAN WIGENTA

Dengan htmlspecialchars:  
WILDAN WIGENTA

Contoh penggunaan &lt; dan &gt;

Teks Mentah: Teks biasa > lainnya.  
Teks Aman (setelah htmlspecialchars): <script>alert('Halo!');</script> Teks biasa > lainnya.

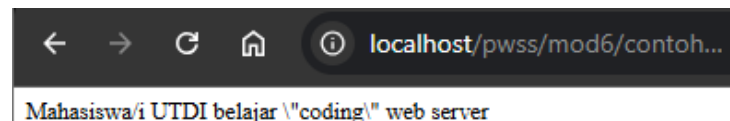
Pembahasan: Program tersebut membandingkan hasil input formulir yang ditampilkan secara langsung dan dengan htmlspecialchars(). Fungsi ini digunakan untuk mencegah karakter khusus seperti < dan > dieksekusi sebagai kode HTML atau JavaScript, sehingga melindungi dari serangan XSS. Program juga menampilkan contoh perbedaan teks mentah dan teks yang telah diamankan.

## B. Addslashes pada PHP

Program:

```
contohadd.php X
mod6 > contohadd.php > ...
1  <?php
2      $str = addslashes('Mahasiswa/i UTDI belajar "coding" web server');
3      echo $str;
4  ?>
```

Output:



Pembahasan: Program ini menggunakan fungsi addslashes() untuk menambahkan karakter escape (\) sebelum karakter khusus seperti tanda kutip satu ('), kutip dua ("), dan garis miring (/) dalam string. Dalam contoh tersebut, string 'Mahasiswa/i UTDI belajar "coding" web server' akan diubah menjadi MahasiswaVi UTDI belajar \"coding\" web server, lalu ditampilkan. Fungsi ini berguna untuk mencegah error atau serangan injeksi saat menyimpan data ke database.

## C. Validasi dan Sanitasi

Program:

```
validasi.php X
mod6 > validasi.php > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style>
5          .error {color: red; #FF0000;}
6      </style>
7  </head>
8  <body>
9
10     @var string $name_pesan
11     $name_pesan = $email_pesan = "";
12     $nama = $email = "";
13
14     if ($_SERVER["REQUEST_METHOD"] == "POST") {
15         if (empty($_POST["nama"])) {
16             $name_pesan = "Nama harus diisi";
17         } else {
18             $nama = htmlspecialchars($_POST["nama"]);
19         }
20
21         if (empty($_POST["email"])) {
22             $email_pesan = "Email harus diisi";
23         } else {
24             $email = htmlspecialchars($_POST["email"]);
25         }
26     }
27 }
28 <h2>Form Input Data</h2>
29
```

```

30 <form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">
31     Nama: <input type="text" name="nama">
32     <span class="error">* <?php echo $name_pesan; ?></span>
33     <br><br>
34     Email: <input type="text" name="email">
35     <span class="error">* <?php echo $email_pesan; ?></span>
36     <br><br>
37     <input type="submit" name="submit" value="Submit">
38 </form>
39
40 <?php
41     echo "<h2>Yang kamu inputkan:</h2>";
42     echo "Nama: " . $nama;
43     echo "<br>";
44     echo "E-Mail: " . $email;
45 ?>
46 </body>
47 </html>

```

Output:

The first screenshot shows a web browser at `localhost/pwss/mod6/validasi....` displaying a form titled "Form Input Data". The form has two text input fields: "Nama:" and "Email:". The "Nama:" field has a red error message "\* Nama harus diisi" next to it. The "Email:" field has a red asterisk "\*" next to it. Below the fields is a "Submit" button. Below the form, the text "Yang kamu inputkan:" is followed by "Nama:" and "E-Mail: wildanwigenta52@gmail.com".

The second screenshot shows the same browser after the form has been submitted. The "Nama:" field now has a red asterisk "\*" next to it, and the "Email:" field has a red error message "\* Email harus diisi" next to it. The "Submit" button is still present. Below the form, the text "Yang kamu inputkan:" is followed by "Nama: WILDAN WIGENTA" and "E-Mail:".

Pembahasan: Program tersebut adalah form input sederhana yang memvalidasi data nama dan email. Jika pengguna men-submit formulir tanpa mengisi salah satu atau kedua field, akan ditampilkan pesan error berwarna merah di samping field tersebut. Jika data diisi, nilai input akan difilter menggunakan `htmlspecialchars()` untuk mencegah eksekusi kode berbahaya, lalu ditampilkan kembali di bawah form sebagai hasil input. Program ini juga menggunakan `$_SERVER['PHP_SELF']` agar data diproses oleh file yang sama.

## LATIHAN

Tambahkan form dengan NIM

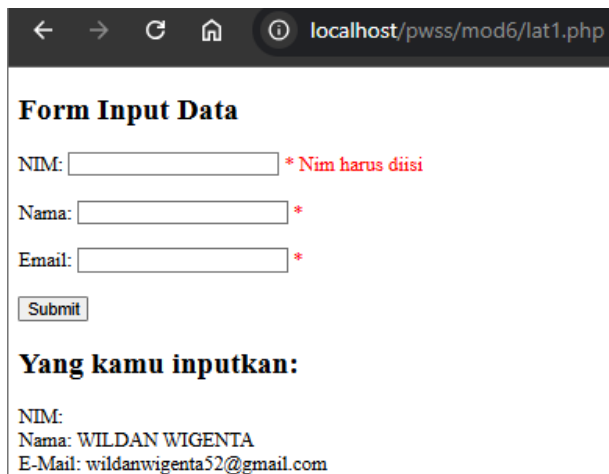
Bila NIM tidak diisi akan muncul pesan "NIM harus diisi"

Program:

```
lat1.php x
mod6 > lat1.php > html > body > form
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  .error {color: red; #FF0000;}
6  </style>
7  </head>
8  <body>
9
10 <?php
11 $nim = $nim_pesan = "";
12 $name_pesan = $email_pesan = "";
13 $nama = $email = "";
14
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16     if (empty($_POST["nim"])) {
17         $nim_pesan = "Nim harus diisi";
18     } else {
19         $nim = htmlspecialchars($_POST["nim"]);
20     }
21
22     if (empty($_POST["nama"])) {
23         $name_pesan = "Nama harus diisi";
24     } else {
25         $nama = htmlspecialchars($_POST["nama"]);
26     }
27
28     if (empty($_POST["email"])) {
29         $email_pesan = "Email harus diisi";
30     } else {
31         $email = htmlspecialchars($_POST["email"]);
32     }
33 }
34
35 <h2>Form Input Data</h2>
36 <form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>">
37     NIM: <input type="text" name="nim">
38     <span class="error">* <?php echo $nim_pesan; ?></span>
39     <br><br>
40     Nama: <input type="text" name="nama">
41     <span class="error">* <?php echo $name_pesan; ?></span>
42     <br><br>
43     Email: <input type="text" name="email">
44     <span class="error">* <?php echo $email_pesan; ?></span>
45     <br><br>
46     <input type="submit" name="submit" value="Submit">
47 </form>
48
49 <?php
50     echo "<h2>Yang kamu inputkan:</h2>";
51     echo "NIM: " . $nim;
52     echo "<br>";
53     echo "Nama: " . $nama;
54     echo "<br>";
55     echo "E-Mail: " . $email;
56 ?>
57 </body>
58 </html>
```

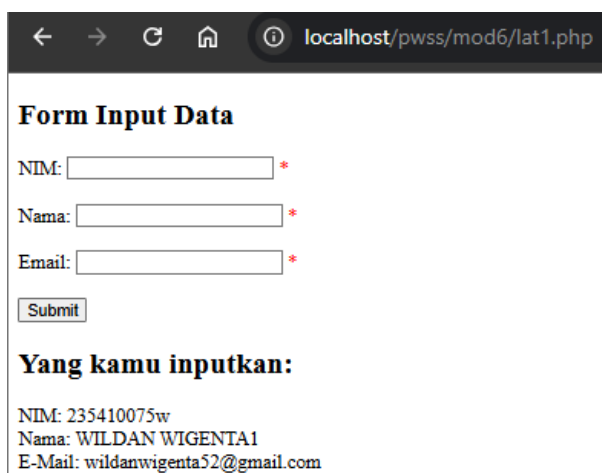
Output:

Bila NIM tidak diisi akan muncul pesan "NIM harus diisi"



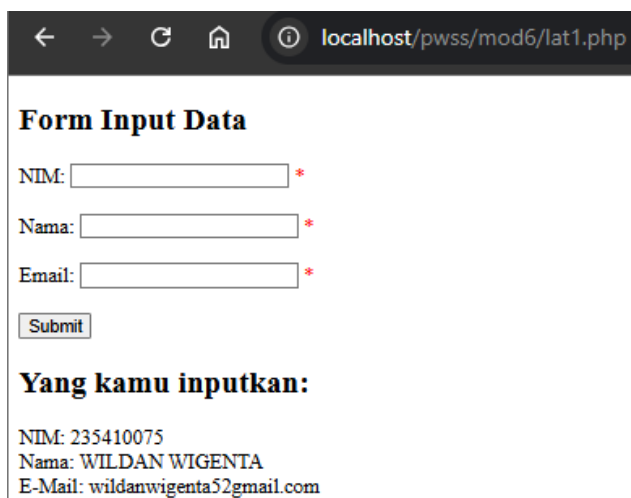
The screenshot shows a web browser window with the address bar displaying 'localhost/pwss/mod6/lat1.php'. The page title is 'Form Input Data'. Below the title, there are three input fields: 'NIM:', 'Nama:', and 'Email:'. The 'NIM:' field is empty and has a red asterisk next to it with the text '\* NIM harus diisi' in red. The 'Nama:' and 'Email:' fields are also empty but have red asterisks next to them. Below the input fields is a 'Submit' button. Under the heading 'Yang kamu inputkan:', the following text is displayed: 'NIM:', 'Nama: WILDAN WIGENTA', and 'E-Mail: wildanwigenta52@gmail.com'.

NIM, Nama diisi sembarang ada angka dan huruf



The screenshot shows the same web browser window as before, but now the 'NIM:' field contains the value '235410075w' and the 'Nama:' field contains 'WILDAN WIGENTA1'. Both fields have red asterisks next to them. The 'Email:' field is still empty with a red asterisk. The 'Submit' button is still present. Under the heading 'Yang kamu inputkan:', the following text is displayed: 'NIM: 235410075w', 'Nama: WILDAN WIGENTA1', and 'E-Mail: wildanwigenta52@gmail.com'.

Email diisi sembarang tanpa simbol @



The screenshot shows the same web browser window as before, but now the 'Email:' field contains the value 'wildanwigenta52gmail.com'. It has a red asterisk next to it. The 'NIM:' and 'Nama:' fields are still filled with '235410075w' and 'WILDAN WIGENTA1' respectively, both with red asterisks. The 'Submit' button is still present. Under the heading 'Yang kamu inputkan:', the following text is displayed: 'NIM: 235410075', 'Nama: WILDAN WIGENTA', and 'E-Mail: wildanwigenta52gmail.com'.

## TUGAS

Menggunakan hasil latihan diatas, lengkapi validasi :

- NIM, harus angka, bila ada karakter/huruf ada pesan "NIM harus angka semua"
- Nama, tidak boleh ada angka, bila ada angka muncul pesan "Nama harus huruf semua"
- email , harus ada simbol @ ditengah nama email (cari referensi untuk validasi email), bila tidak ada simbol @ muncul pesan "Penulisan email belum benar"

Program:

```
tugas.php X
mod6 > tugas.php > html > body > form
2  <html>
6  <body>
8  <?php
9  $nim = $nama = $email = "";
10 $nim_pesan = $name_pesan = $email_pesan = "";
11
12 if ($_SERVER["REQUEST_METHOD"] == "POST") {
13     $nim = htmlspecialchars($_POST["nim"]);
14     $nama = htmlspecialchars($_POST["nama"]);
15     $email = htmlspecialchars($_POST["email"]);
16
17     if (empty($nim)) $nim_pesan = "NIM harus diisi";
18     elseif (!ctype_digit($nim)) $nim_pesan = "NIM harus angka semua";
19
20     if (empty($nama)) $name_pesan = "Nama harus diisi";
21     elseif (preg_match('/\d/', $nama)) $name_pesan = "Nama harus huruf semua";
22
23     if (empty($email)) $email_pesan = "Email harus diisi";
24     elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) $email_pesan = "Penulisan email belum benar";
25 }
26 ?>
27
28 <h2>Form Input Data</h2>
29 <form method="post">
30     NIM: <input type="text" name="nim" value="<?=$nim ?>">
31     <span class="error">* <?=$nim_pesan ?></span><br><br>
32
33     Nama: <input type="text" name="nama" value="<?=$nama ?>">
34     <span class="error">* <?=$name_pesan ?></span><br><br>
35
36     Email: <input type="text" name="email" value="<?=$email ?>">
37     <span class="error">* <?=$email_pesan ?></span><br><br>
38
39     <input type="submit" value="Submit">
40 </form>
41
42 <?php if ($_SERVER["REQUEST_METHOD"] == "POST" && !$nim_pesan && !$name_pesan && !$email_pesan): ?>
43     <h2>Yang kamu inputkan:</h2>
44     NIM: <?=$nim ?><br>
45     Nama: <?=$nama ?><br>
46     Email: <?=$email ?><br>
47 <?php endif; ?>
```

Output:

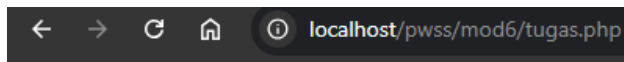
localhost/pwss/mod6/tugas.php

### Form Input Data

NIM:  \* NIM harus angka semua

Nama:  \* Nama harus huruf semua

Email:  \* Penulisan email belum benar



### Form Input Data

NIM:  \*

Nama:  \*

Email:  \*

### Yang kamu inputkan:

NIM: 235410075  
Nama: WILDAN WIGENTA  
Email: wildanwigenta52@gmail.com

Pembahasan: Program di atas form input data yang melakukan validasi untuk tiga field: NIM, Nama, dan Email. Saat pengguna mengirimkan form, program memeriksa apakah NIM hanya berisi angka, Nama hanya terdiri dari huruf tanpa angka, dan Email ditulis dengan format yang benar (mengandung simbol @). Jika ada kesalahan, pesan error ditampilkan di samping masing-masing input; jika semua valid, data yang dimasukkan akan ditampilkan di bawah form.

## Kesimpulan

Setelah saya mempelajari materi sanitasi dan validasi input dari form di PHP, saya memahami pentingnya menjaga keamanan dan keakuratan data yang dikirimkan pengguna. Sanitasi membantu membersihkan input dari karakter berbahaya, sementara validasi memastikan data sesuai dengan format yang diharapkan, sehingga dapat mencegah serangan seperti XSS (Cross Site Scripting) dan kesalahan logika dalam aplikasi. Melalui latihan ini, saya juga belajar membuat form yang lebih aman, responsif terhadap kesalahan input, serta menulis kode PHP yang lebih rapi dan efisien.