Jurusan Teknologi Informasi Politeknik Negeri Malang



**Jobsheet-8: PHP - Form Proccessing**

**Mata Kuliah Desain dan Pemrograman Web**

Pengampu: Tim Ajar Desain dan Pemrograman Web

*Oktober 2024*

**Topik**

- Konsep Form Proccessing dengan PHP

# Tujuan

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu membuat form menggunakan PHP
2. Mahasiswa mampu membuat form menggunakan jQuery

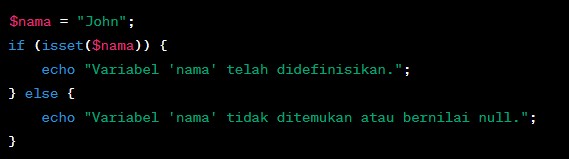
# Perhatian

Jobsheet ini harus dikerjakan step-by-step sesuai langkah-langkah praktikum yang sudah diberikan. Soal dapat dijawab langsung di dalam kolom yang disediakan dengan menggunakan PDF Editor.

# Pendahuluan

## Function isset

**isset()** adalah sebuah fungsi dalam PHP yang digunakan untuk memeriksa apakah suatu variabel telah didefinisikan (ada) atau tidak. Fungsi ini mengembalikan **true** jika variabel telah didefinisikan dan memiliki nilai, dan **false** jika variabel tidak ada atau bernilai **null**. Berikut adalah contoh script dan penjelasan minimal mengenai **isset()**:

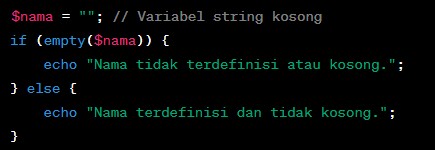


Kode di atas memeriksa apakah variabel **$nama** telah didefinisikan. Jika ya, maka pesan "Variabel 'nama' telah didefinisikan." akan ditampilkan, jika tidak, maka pesan "Variabel 'nama' tidak ditemukan atau bernilai null." akan ditampilkan.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Praktikum 1.** | | | | **Function Isset** | |
| **Langkah** | | **Keterangan** | | | |
| 1 | | Buatlah satu file baru di dalam direktori dasarWeb, beri nama isset.php. | | | |
| 2 | | Ketikkan ke dalam file isset.php tersebut kode di bawah ini. | | | |
| 3 | |  | | | |
| 4 | | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ isset.php. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.1)  Jawab : karena variabel umur belum di beri nilai, maka akan muncul hasil seperti berikut : | |  | |
| 5 | | Tambahkan isi dari file isset.php tersebut dengan kode di bawah ini. | |  | |
| 6 | | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/isset.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan isset pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 1.2)  Jawab : dikarenakan yang dipanggil hanya nama dalam array, maka hasilnya pun akan muncul namanya saja. | | | |

## Function empty

Fungsi **empty()** pada PHP digunakan untuk memeriksa apakah suatu variabel kosong atau tidak terdefinisi. Fungsi ini mengembalikan **true** jika variabel kosong atau tidak terdefinisi, dan **false** jika variabel memiliki nilai atau telah didefinisikan. Berikut adalah penjelasan dan contoh penggunaan **empty()**:



Fungsi **empty()** dapat digunakan untuk memeriksa apakah suatu string kosong atau tidak.

**Praktikum 2. Function empty**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama empty.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam empty.php |
| 2 |  |
| 3 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ empty.php. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.1)  Jawab : dikarenakan nilai dalam array belum diisi maka hasilnya pun seperti dibawah ini.    Program tetap dapat dijalankan karena penggunaan empty pada baris program. |
| 4 | Tambahkan isi dari file empty.php tersebut dengan kode di bawah ini. |
| 5 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ empty.php buat hasil tidak menjadi 1 baris, jadi tampilan dari echo harus terpisah. Apa yang anda pahami dari penggunaan empty pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 2.2)  Jawab : sama seperti sebelumnya, hasil dari output akan seperti ini.    Meskipun array tidak di instansiasi, namun dengan adanya fungsi empty, program masih tetap bisa dijalankan. |

**Praktikum Bagian 3 : Form Input PHP**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form.php |

|  |  |
| --- | --- |
| 2 |  |
| 3 | Buat satu file baru bernama proses\_form.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam proses\_form.php |
| 4 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ proses\_form.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.1)  Jawab : setelah mengisi pada kotak yang ada dan meng-submitnya, maka hasil outputnya akan sesuai dengan apa yang telah diisi. |
| 5 | Buat satu file baru bernama form\_self.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_self.php |
|  |  |
| 6 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/ form\_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda. (soal no 3.2)  Jawab : setelah mengisi data pada textbox, maka akan muncul tanda berhasil setelah submit, dan jika dikosongkan dan disubmit, maka akan muncul text untuk mengisi textbox tersebut. |

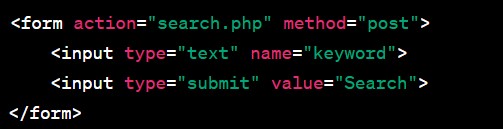
## HTML Injection

HTML injection (atau disebut juga "HTML injection attack" atau "client-side injection") adalah serangan keamanan yang terjadi ketika seorang penyerang memasukkan kode HTML atau JavaScript yang berbahaya ke dalam input yang diterima oleh sebuah aplikasi web. Kode berbahaya tersebut akan dieksekusi oleh browser pengguna yang melihat halaman web yang terpengaruh, yang dapat mengakibatkan akses ilegal ke data, perubahan tampilan halaman, atau serangan lainnya.

Seorang penyerang dapat mencoba menyisipkan kode HTML atau JavaScript yang berbahaya ke dalam input yang diterima oleh aplikasi web. Jika aplikasi web tersebut tidak melakukan sanitasi atau escape input dengan benar, maka kode berbahaya tersebut akan dieksekusi oleh browser pengguna, yang dapat menyebabkan masalah keamanan.

Contoh HTML Injection:

Misalnya, kita memiliki formulir pencarian sederhana di situs web yang mencari kata kunci di antara ulasan pengguna:



Sekarang, jika aplikasi web tidak memvalidasi input dengan benar, seorang penyerang dapat memasukkan input berbahaya seperti ini:



Jika aplikasi web tidak menghindari atau membersihkan input ini sebelum menampilkannya di halaman hasil pencarian, maka hasilnya akan menjadi seperti ini:

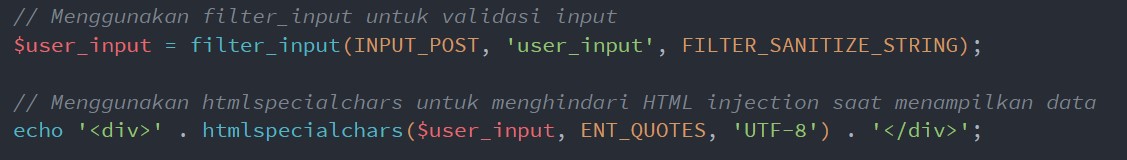


Kode JavaScript berbahaya tersebut akan dieksekusi oleh browser pengguna dan akan memunculkan kotak peringatan "You have been hacked!". Ini adalah contoh HTML injection yang sederhana. Sebagai pengembang web, penting untuk selalu melakukan sanitasi dan validasi input dari pengguna, serta menghindari menampilkan input pengguna langsung di halaman web tanpa escaping atau sanitasi yang tepat untuk menghindari serangan semacam ini.

**Cara Mengangani HTML Injection**

Cara menangani HTML injection adalah dengan memvalidasi, menyaring, dan menghindari menampilkan input pengguna tanpa escaping yang tepat. Berikut adalah beberapa langkah untuk melindungi aplikasi Anda dari HTML injection:

1. **Validasi Input**: Validasi input dari pengguna untuk memastikan bahwa hanya data yang valid yang diterima. Anda bisa menggunakan fungsi seperti filter\_input() atau filter\_var() untuk melakukan validasi.
2. **Saring Input**: Saring input pengguna untuk menghapus atau menggantikan karakter yang berpotensi berbahaya, seperti <, >, &, dan lainnya. Anda bisa menggunakan fungsi seperti htmlspecialchars() atau strip\_tags() untuk ini.
3. **Parameterized Statements (Query)**: Jika Anda menghasilkan SQL query dengan input pengguna, gunakan parameterized statements atau prepared statements untuk mencegah SQL injection, yang bisa menjadi bentuk serangan yang mirip dengan HTML injection.
4. **Content Security Policy (CSP)**: Terapkan Content Security Policy (CSP) pada header HTTP Anda untuk mengontrol sumber daya yang dapat digunakan dalam halaman web Anda.
5. **Escape Output**: Ketika Anda menampilkan data di halaman web, pastikan Anda menghindari injeksi JavaScript dengan menggunakan htmlspecialchars() atau metode serupa. Berikut contoh sederhana dalam PHP untuk mengatasi HTML injection:



Dengan menggunakan langkah-langkah ini, Anda dapat memitigasi risiko HTML injection dalam aplikasi web Anda. Selalu penting untuk memvalidasi input, membersihkan data yang masuk, dan menghindari menampilkan data pengguna tanpa escaping yang benar.

## Praktikum 4 : HTML Injection

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama html\_aman.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam html\_aman.php |
| 2 |  |
| 3 | Tambahkan script lainya supaya langkah 2 bisa berjalan normal. Lalu simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html\_aman.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.1)  Jawab : baris kode tersebut berguna untuk menghindari injeksi javascript. |
| 5 | Ketikkan kode tambahan pada langkah 6 di dalam html\_aman.php |
| 6 |  |
| 7 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/html\_aman.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 4.2)  Jawab : jika pengisian data pada textbox sesuai, maka data akan berhasil disimpan. |

## Regular Expression (Regex)

Regex adalah alat yang kuat untuk mencari, mencocokkan, atau memanipulasi teks berdasarkan pola tertentu. Anda dapat menggunakannya untuk berbagai tujuan seperti validasi input, pencarian teks, penggantian teks, dan banyak lagi.

**Basic Regex Patterns:**

* **Karakter Tunggal**: Karakter apa pun akan cocok dengan dirinya sendiri dalam Regex, kecuali karakter khusus yang harus dihindari dengan karakter pelarian \.

Contoh: Pola /a/ akan cocok dengan huruf "a" dalam teks.

* **Karakter Spesifik**: Anda dapat mencocokkan karakter tertentu dengan menyebutkannya dalam pola.

Contoh: Pola /hello/ akan cocok dengan teks "hello" dalam teks.

* **Karakter Khusus**: Beberapa karakter khusus dalam Regex harus dihindari dengan karakter pelarian (\) jika Anda ingin mencocokkannya secara harfiah. Contoh karakter khusus: ., \*, +, ?, |,

[, ], (, ), {, }, ^, $, \.

* **Karakter Set**: Anda dapat mencocokkan karakter dari satu set karakter dengan menggunakan [ ]. Misalnya, /[aeiou]/ akan mencocokkan salah satu huruf vokal.
* **Rentang Karakter**: Anda dapat menggunakan - dalam karakter set untuk menentukan rentang karakter. Misalnya, /[a-z]/ akan mencocokkan huruf kecil apa pun.
* **Kuantifier**: Anda dapat mengatur berapa kali karakter atau kelompok karakter sebelumnya harus muncul. Contoh kuantifier:

\*: 0 atau lebih kali

+: 1 atau lebih kali

?: 0 atau 1 kali

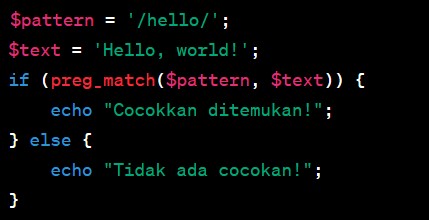
{n}: Persis n kali

{n,}: Setidaknya n kali

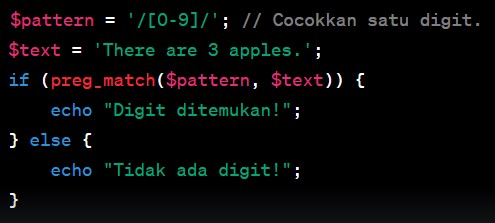
{n,m}: Setidaknya n kali, maksimal m kali

**Contoh Penggunaan Regex:**

1. Mencocokan pola



1. Mencocokan karakter set



## Praktikum 5 : Penggunaan Regex Pada PHP

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama regex.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam regex.php |

|  |  |
| --- | --- |
| 2 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ regex.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.1)  Jawab : setelah program dijalankan, akan memeriksa tiap huruf dalam teks, apakah terdapat huruf kecil didalamnya. |
| 5 | Ketikkan kode tambahan pada langkah 6 di dalam regex.php |
| 6 |  |
| 7 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.2)  Jawab : setelah program dijalankan, akan memeriksa apakah terdapat kesamaan pada teks dengan angka. |
| 9 | Ketikkan kode tambahan pada langkah 10 di dalam regex.php |
| 10 |  |
| 11 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 12 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.3)  Jawab : setelah program dijalankan, maka akan memeriksa mana kata apel yang nantinya akan digantikan dengan banana. |
| 13 | Ketikkan kode tambahan pada langkah 14 di dalam regex.php |
| 14 |  |
| 15 | Lengkapi kode program di atas sehingga hasilnya rapi. Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/regex.php |
| 16 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.4)  Jawab : setelah program dijalankan akan memeriksa mana yang sesuai dengan pattern yang telah disiapkan. Dan karena yang pertama ditemukan kecocokannya adalah god, maka hasil outputnya adalah god. |
| 17 | Soal 5.5: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan `?` (0 atau 1 kali). Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.5)  Jawab : setelah progra dijalankan, maka akan memeriksa adakah kata god pada text. |
| 18 | Soal 5.6: Buatlah script langkah 14 dengan variable pattern yang di ubah menggunakan  `{n,m}`. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 5.6)  Jawab : setelah program dijalankan, akan memeriksa adakah huruf n atau m pada text. Dan jika tidak ditemukan makan outputnya akan “tidak ada yang cocok”. |

## Praktikum 6 : Form Lanjut

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama form\_lanjut.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_lanjut.php |

|  |  |
| --- | --- |
| 2 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_lanjut.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.1)  Jawab : setelah mengisi form yang ada dan di submit, maka hasilnya akan muncul sesuai dengan pilihan dalam form sebelumnya. |
| 5 | Buat satu file baru bernama form\_ajax.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_ajax.php |

|  |  |
| --- | --- |
|  |  |
| 6 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_ajax.php. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 6.2)  Jawab : setelah form diisi dan disubmit, hasilnya akan segera muncul dalam laman yang sama tanpa berganti. |

## Praktikum 7 : Validasi Form

Langkah-langkah Praktikum :

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat satu file baru bernama form\_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 2 di dalam form\_validasi.php |
| 2 |  |
| 3 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_validasi.php |
| 4 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.1)  Jawab : setelah form diisi dan disubmit, maka data akan berhasil disimpan juga data akan kembali ditampilkan untuk mengkonfirmasi kembali. |
| 5 | Kembangkan file bernama form\_validasi.php di dalam folder dasarWeb. Ketikkan kode pada langkah 6 di dalam form\_validasi.php |
| 6 |  |
| 7 | Simpan file , kemudian buka browser dan jalankan/refresh localhost/dasarWeb/ form\_validasi.php |
| 8 | Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.2)  Jawab : sama seperti sebelumnya, setelah form diisi dan disubmit, maka data akan berhasil disimpan juga data akan kembali ditampilkan untuk mengkonfirmasi kembali. Jika terdapat kesalahan dalam pengisian data, maka akan muncul text peringatan agar mengisi data dengan sesuai. |
| 9 | Soal 7.3: Buatlah script langkah 6 dengan menggunakan ajax. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.3)  Jawab : jika form tidak diisi dan langsung dikirim, maka akan mengulang hingga form diisi dengan benar agar bisa dikirim dan disimpan datanya. |
| 10 | Soal 7.4: Tambahkan validasi untuk password dengan ketentuan minimal 8 karakter dengan menggunakan jQuery dan PHP tambahkan pada langkah 9. Catat di sini apa yang anda amati dari penambahan kode program di atas. (soal no 7.4)  Jawab : akan dilakukan pengecekan panjang password yang mana ketika password tidak sesuai yang ditentukan, akan false. |

**Tugas Praktikum;**

Buatlah sebuah formulir kontak sederhana yang memungkinkan pengunjung website mengirimkan pesan. Formulir ini akan menggunakan PHP untuk memproses pengiriman email, sementara jQuery akan digunakan untuk melakukan validasi input data secara real-time.

**Ketentuan :**

* **Elemen Form:**
  + Nama lengkap (teks)
  + Alamat email (email)
  + Subjek (teks)
  + Pesan (textarea)
  + Tombol kirim
* **Validasi dengan jQuery:**
  + Nama lengkap minimal 3 karakter
  + Alamat email harus dalam format yang benar
  + Subjek dan pesan tidak boleh kosong
  + Tampilkan pesan error di samping setiap field yang tidak valid
* **Pengiriman Email dengan PHP:**
  + Gunakan fungsi mail() di PHP untuk mengirimkan email.
  + Email berisi data yang diinputkan oleh pengguna.
  + Tampilkan pesan sukses setelah email berhasil dikirim.

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Formulir Kontak</title>

    <script *src*="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <style>

*.error* {

        color: red;

    }

    </style>

</head>

<body>

    <h1>Formulir Kontak</h1>

    <form *id*="kontak-form">

        <label *for*="nama">Nama Lengkap:</label>

        <input *type*="text" *id*="nama" *name*="nama">

        <span *id*="nama-error" *class*="error"></span><br>

        <label *for*="email">Alamat Email:</label>

        <input *type*="email" *id*="email" *name*="email">

        <span *id*="email-error" *class*="error"></span><br>

        <label *for*="subjek">Subjek:</label>

        <input *type*="text" *id*="subjek" *name*="subjek">

        <span *id*="subjek-error" *class*="error"></span><br>

        <label *for*="pesan">Pesan:</label>

        <textarea *id*="pesan" *name*="pesan"></textarea>

        <span *id*="pesan-error" *class*="error"></span><br>

        <input *type*="submit" *value*="Kirim">

    </form>

    <script>

    $**(**document**)**.ready**(**function() {

        $**(**"#kontak-form"**)**.submit**(**function(event) {

            event.preventDefault**()**;

            var nama = $**(**"#nama"**)**.val**()**;

            var email = $**(**"#email"**)**.val**()**;

            var subjek = $**(**"#subjek"**)**.val**()**;

            var pesan = $**(**"#pesan"**)**.val**()**;

            var valid = true;

*// Validasi Nama*

            if **(**nama.length < 3**)** {

                $**(**"#nama-error"**)**.text**(**"Nama harus minimal 3 karakter."**)**;

                valid = false;

            } else {

                $**(**"#nama-error"**)**.text**(**""**)**;

            }

*// Validasi Email*

            var emailRegex = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

            if **(**!emailRegex.test**(**email**))** {

                $**(**"#email-error"**)**.text**(**"Alamat email harus dalam format yang benar."**)**;

                valid = false;

            } else {

                $**(**"#email-error"**)**.text**(**""**)**;

            }

*// Validasi Subjek*

            if **(**subjek == ""**)** {

                $**(**"#subjek-error"**)**.text**(**"Subjek tidak boleh kosong."**)**;

                valid = false;

            } else {

                $**(**"#subjek-error"**)**.text**(**""**)**;

            }

*// Validasi Pesan*

            if **(**pesan == ""**)** {

                $**(**"#pesan-error"**)**.text**(**"Pesan tidak boleh kosong."**)**;

                valid = false;

            } else {

                $**(**"#pesan-error"**)**.text**(**""**)**;

            }

            if **(**valid**)** {

                $.ajax**(**{

**type:** "POST",

**url:** "proses\_kontak.php",

**data:** $**(**"#kontak-form"**)**.serialize**()**,

**success:** function(data) {

                        console.log**(**data**)**;

                        alert**(**"Email berhasil dikirim!"**)**;

                    }

                }**)**;

            }

        }**)**;

    }**)**;

    </script>

</body>

</html>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $nama = $\_POST["nama"];

    $email = $\_POST["email"];

    $subjek = $\_POST["subjek"];

    $pesan = $\_POST["pesan"];

*// Kirim email*

    $to = "tujuan@email.com"; *// Ganti dengan alamat email tujuan*

    $subject = $subjek;

    $message = "Nama: $nama\nEmail: $email\nPesan: $pesan";

    $headers = "From: $email\r\n";

    if (mail($to, $subject, $message, $headers)) {

        echo "Email berhasil dikirim!";

    } else {

        echo "Gagal mengirim email.";

    }

}

?>

