

LAPORAN PRAKTIKUM WEB

MINGGU 7 - PHP 4



TI-2F

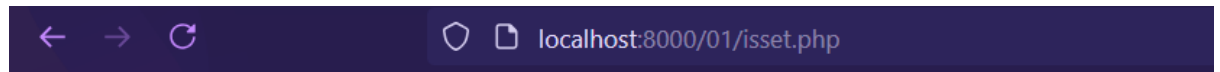
Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

1. Function ISSET

Soal 1.1 - 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.

Jawab: Berikut hasil yang dirender oleh browser:

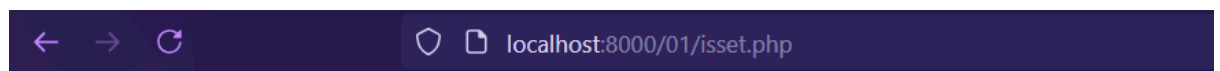


Anda belum dewasa atau variabel 'umur' tidak ditemukan!

Teks tersebut dihasilkan oleh operasi `if else` yang bakal menentukan teks apa yang dimunculkan ke pengguna. Berdasarkan kode yang ada. Variabel `$umur` sudah dibuat namun belum diberikan nilai atau belum dilakukan inisialisasi. Dengan demikian, dia akan masuk ke dalam kondisi yang `false` saat dilakukan pengecekan kondisi variabel `$umur`. Fungsi `isset()` di sana berfungsi untuk mengecek apakah variabel `$umur` tidak `null` dan telah didefinisikan. Meski variabel `$umur` sudah ada, tapi masih belum ada nilainya. Ini membuat nilai `$umur = null`.

Soal 1.2 - 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!

Jawab: Berikut hasil yang dirender oleh browser:



Anda belum dewasa atau variabel 'umur' tidak ditemukan!

Nama: Jane

Kode yang dibuat agar baris pertama dengan kedua tidak menyatu adalah dengan menambahkan satu baris berikut:

```
11 echo "<br><br><br>";
```

2. Function EMPTY

Soal 2.1 - 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!

Jawab: Hasil kode yang dirender di browser:



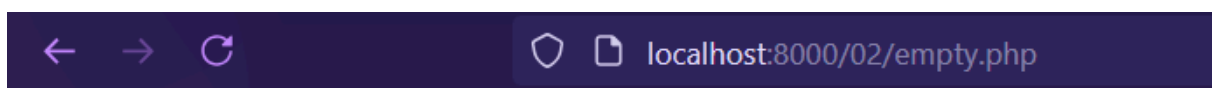
Variabel tidak terdefinisi atau kosong

Kode yang ada memanfaatkan fungsi `empty()` untuk mengecek apakah suatu variabel kosong. Jika variabel adalah array maka dia akan dianggap kosong ketika tidak memiliki elemen sama sekali. Pada kasus tersebut, variabel `$myArray` tidak memiliki anggota elemen sama sekali. Maka dari itu, akan dicetak sebuah teks “Variabel tidak terdefinisi atau kosong”.

Soal 2.2 - 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 `isset` pada file tersebut. Catat di bawah ini pemahaman anda!

Jawab: Hasilnya sebagai berikut dengan tambahan satu baris kode untuk membuat tampilan menjadi dua baris berbeda.



Array tidak terdefinisi atau kosong

Variabel tidak terdefinisi atau kosong

Berikut tambahan kodenya:

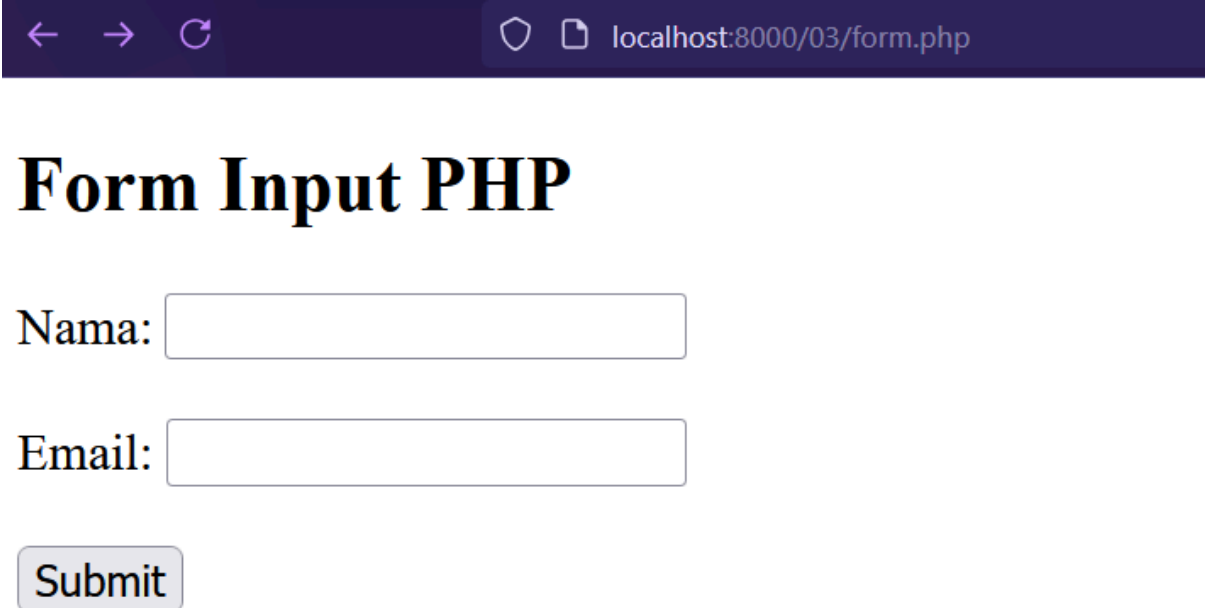
```
11 echo "<br><br><br>";
```

Muncul sebuah teks “Variabel tidak terdefinisi atau kosong” karena variabel `$nonExistentVar` masih belum dibuat sehingga saat dicek menggunakan fungsi `empty()` hasilnya adalah `true`.

3. Form Input PHP

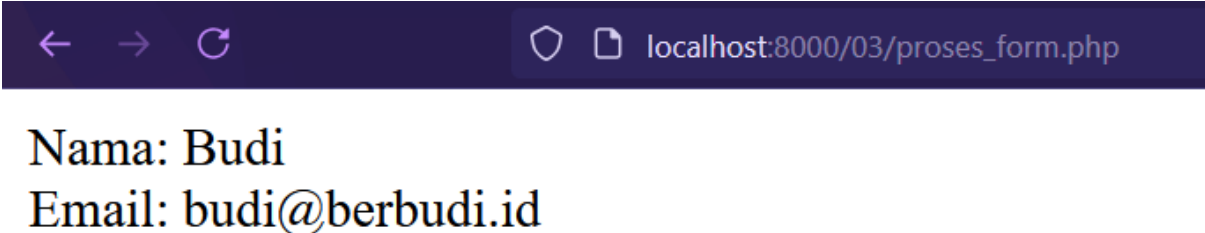
Soal 3.1 - Simpan file tersebut, kemudian buka browser dan jalankan `localhost/dasarWeb/proses_form.php`. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda!

Jawab: Tampilan awal web sebagai berikut:



The screenshot shows a web browser window with the address bar displaying `localhost:8000/03/form.php`. The page title is "Form Input PHP". Below the title, there are two input fields: "Nama:" followed by a text input box, and "Email:" followed by a text input box. Below these fields is a "Submit" button.

Kemudian apabila disubmit, maka bisa mendapatkan tampilan halaman baru:

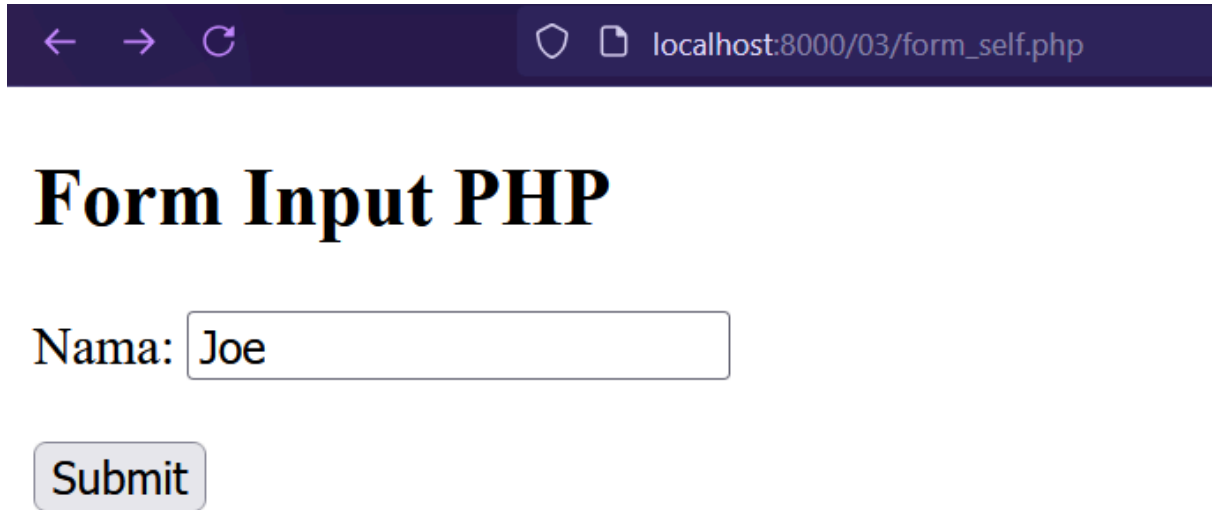


The screenshot shows the same web browser window, but the address bar now displays `localhost:8000/03/proses_form.php`. The page content has changed to show the submitted data: "Nama: Budi" and "Email: budi@berbudi.id".

Jadi, data yang dikirimkan dari `form.php` akan di kontrol oleh kode program yang berada di file `proses_form.php`. Data yang diambil berupa nama dan email. Setelah mendapatkan datanya maka data-data tersebut bisa ditampilkan ke pengguna.

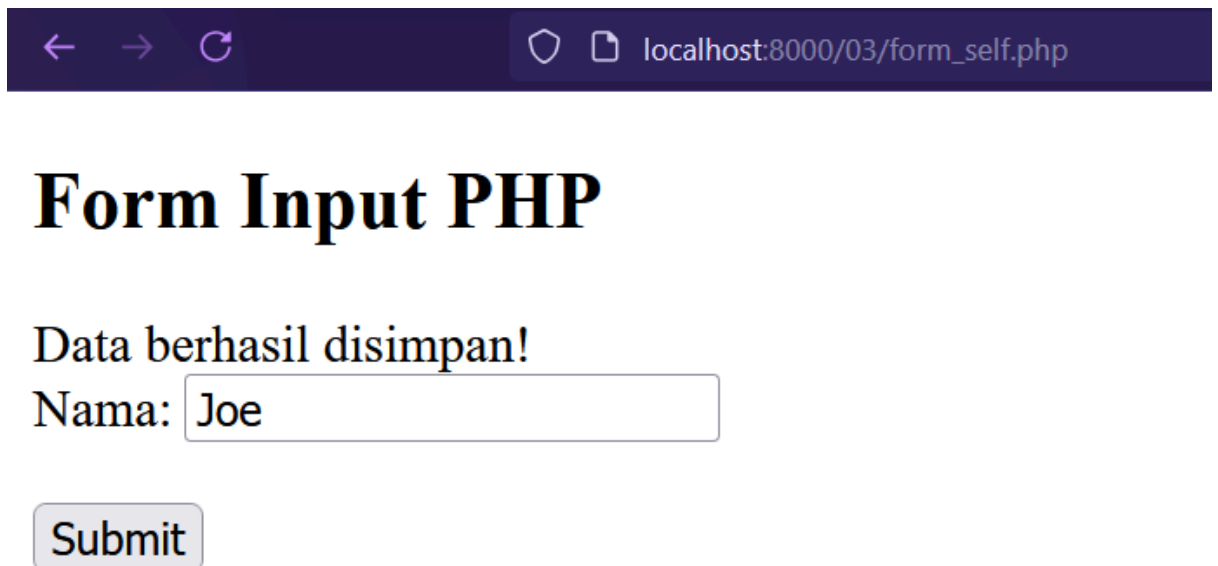
Soal 3.2 - Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb/form_self.php. Apa yang anda pahami dari penggunaan form pada file tersebut. Catat di bawah ini pemahaman anda!

Jawab: Tampilan awal dari halaman web ketika pertama kali dirender:



The screenshot shows a web browser window with a dark purple header bar. On the left, there are navigation icons: a back arrow, a forward arrow, and a circular refresh icon. On the right, there is a shield icon, a document icon, and the address bar text "localhost:8000/03/form_self.php". Below the header, the main content area has a white background. At the top, the text "Form Input PHP" is displayed in a large, bold, black serif font. Below this, the text "Nama:" is followed by a text input field containing the value "Joe". Underneath the input field is a light blue button with rounded corners and the text "Submit" in a bold, black sans-serif font.

Dan berikut adalah tampilan ketika berhasil disubmit:



The screenshot shows the same web browser window as before, but the content has changed. The "Form Input PHP" title remains at the top. Below it, the text "Data berhasil disimpan!" is displayed in a bold, black serif font. Underneath this message, the text "Nama:" is followed by a text input field containing the value "Joe". At the bottom, the same light blue "Submit" button is present.

Secara konsep cara kerja file form_self.php dan percobaan sebelumnya. Namun, untuk mekanisme handle data form dilakukan di satu file yang sama. Pada kode PHP akan mengecek apakah data yang dikirimkan nama memiliki nilai atau tidak. Jika memiliki, maka akan data tersebut akan disimpan dan ditampilkan teks "Data berhasil

disimpan!”. Jika data nama kosong, maka teks yang dimunculkan adalah “Nama harus diisi”.

4. HTML Injection

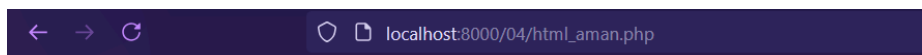
Soal 4.1 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Berikut adalah penambahan kode agar program bisa berjalan dengan lancar:

```
04 > html_aman.php > ...
1  <?php
2
3  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
4      $input = $_POST['input'];
5      $input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
6  }
7  
```

Hal yang ditambahkan adalah memberikan pengecekan kondisi jika request method dari client berupa POST, maka akan mengeksekusi dua baris kode sebelumnya.

Jika dilihat di browser maka hasilnya sebagai berikut:



Hasilnya berupa halaman web kosong. Ini karena file html_aman.php tidak mempunyai sebuah handler untuk menangani sebuah request dengan metode GET. Secara default, setiap kali kita masuk ke URL tertentu di browser itu akan secara otomatis menggunakan metode GET request.

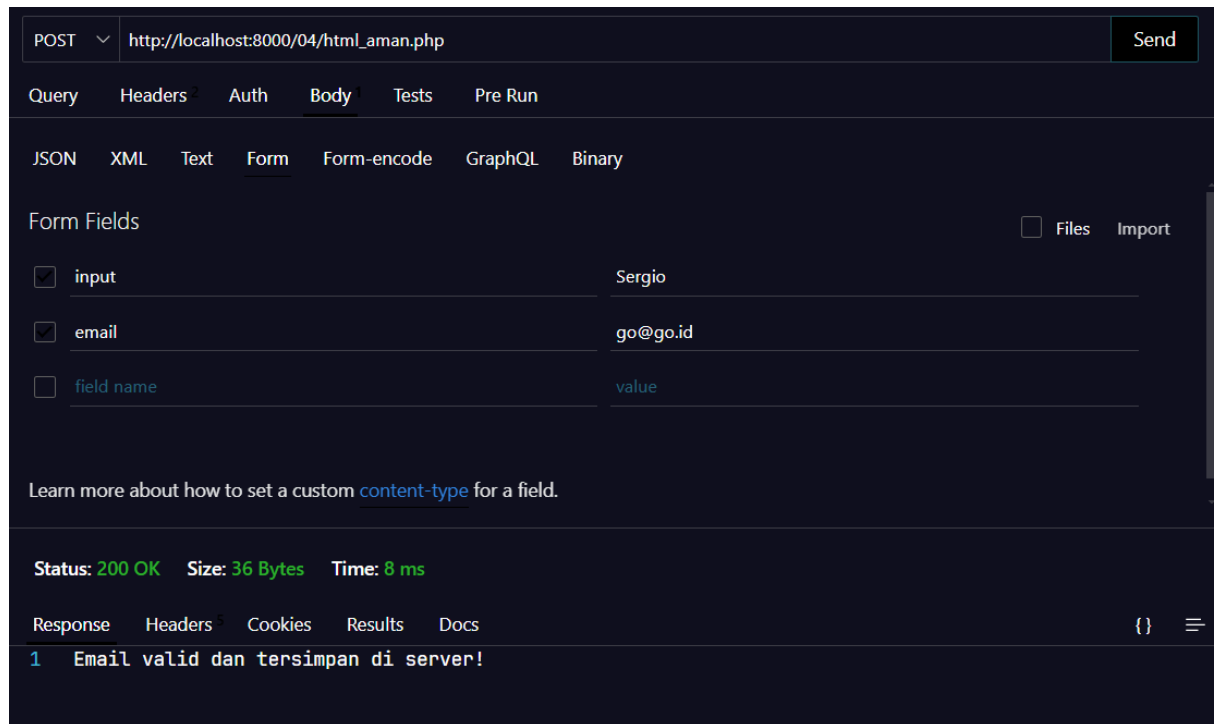
Soal 4.2 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Kode tambahan:

```
04 > html_aman.php > ...
1  <?php
2
3  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
4      $input = $_POST['input'];
5      $input = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
6
7      $email = $_POST['email'];
8      if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
9          echo "Email valid dan tersimpan di server!";
10     } else {
11         $email = null;
12         echo "Email tidak valid!";
13     }
14 }
15 
```

Kode program yang berada di dalam blok if (filter_var(\$email, FILTER_VALIDATE_EMAIL)) berfungsi untuk melakukan validasi data email yang diinputkan dari sebuah form yang ada di client dengan metode POST. Jika validasi berhasil, maka data email akan disimpan di server sedangkan jika tidak akan diberikan pesan “Email tidak valid”.

Contoh dari percobaan request dengan metode POST:



POST http://localhost:8000/04/html_aman.php Send

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

Form Fields ☐ Files Import

☒ input Sergio

☒ email go@go.id

☐ field name value

Learn more about how to set a custom [content-type](#) for a field.

Status: 200 OK Size: 36 Bytes Time: 8 ms

Response Headers Cookies Results Docs {} ≡

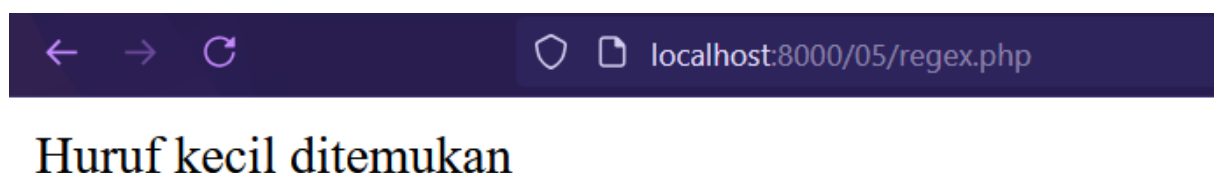
1 Email valid dan tersimpan di server!

Jika dilihat proses request data tersebut dinyatakan berhasil yang ditunjukkan dengan hasil response 200 OK dengan response body “Email valid dan tersimpan di server!”. Ini berarti request yang diajukan ke server berhasil.

5. Penggunaan REGEX pada PHP

Soal 5.1 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Berikut halaman web yang berhasil dirender



Halaman web tersebut menampilkan sebuah teks “Huruf kecil ditemukan”. Teks tersebut ditampilkan karena dari hasil pengecekan pada variabel \$text, memiliki pola

yang sesuai dengan regex di variabel \$pattern. Pada variabel \$pattern menyimpan sebuah pola untuk menampung huruf-huruf kecil dari a hingga z. Ditambah dengan fungsi preg_match() maka bisa dilakukan sebuah proses pengecekan sebuah pola dari suatu variabel.

Soal 5.2 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Hasilnya sebagai berikut:



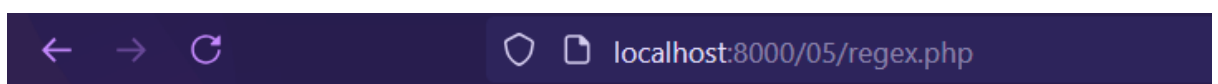
Huruf kecil ditemukan

Cocokkan: 123

Kode yang dibuat untuk merender halaman web tersebut adalah memanfaatkan fungsi preg_match(). Fungsi tersebut akan mencocokkan sebuah string dengan pola yang sudah ditentukan. Jika terdapat teks yang sesuai, maka akan ditampung di variabel \$matches. Variabel tersebut berupa array untuk menampung banyak data variabel yang sesuai dengan pola yang ditentukan.

Soal 5.3 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Hasilnya sebagai berikut:



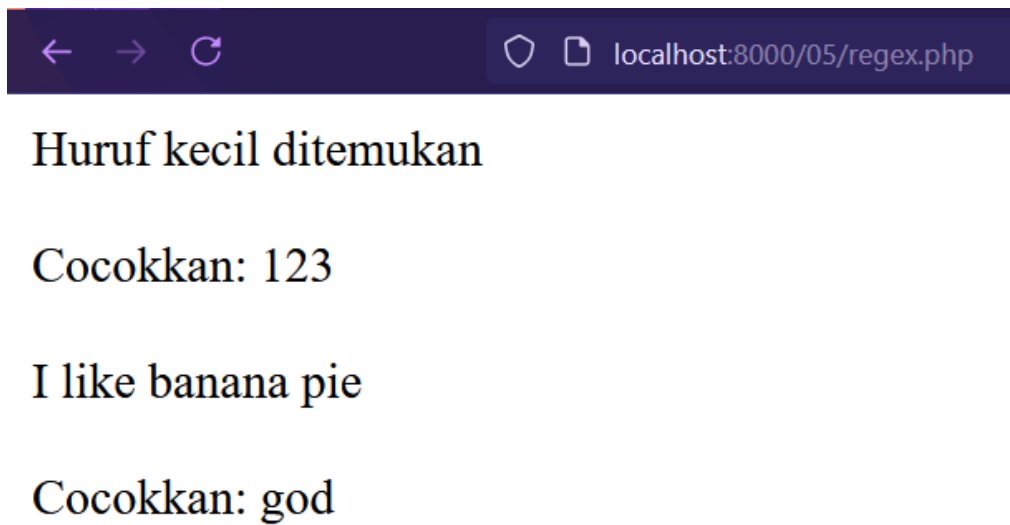
Huruf kecil ditemukan

I like banana pie

Teks yang seharusnya menampilkan tulisan “I like apple pie” diganti menjadi “I like banana pie”. Untuk melakukan hal itu semua maka bisa dengan menggunakan fungsi preg_replace(). Fungsi tersebut akan memproses pola yang akan dideteksi, teks yang akan diganti, dan basis teks yang akan dirubah.

Soal 5.4 - Catat di sini apa yang anda amati dari penambahan kode program yang telah dibuat!

Jawab: Hasilnya sebagai berikut:



Kode yang digunakan untuk menampilkan halaman web di atas adalah dengan berusaha mencari string yang cocok dengan pola yang telah ditentukan. Pada pola yang diinputkan sebagai argumen pada fungsi `preg_match()` akan mencocokkan dengan pola string `"/go*od/"`. Di situ terdapat tanda bintang yang menunjukkan akan menyeleksi karakter `o` berapapun jumlahnya. Jadi, misalkan terdapat string `"god"`, `"good"`, dan `"gooooo"` masih dianggap sesuai dengan pola. Pada halaman tersebut hanya menampilkan string pertama yang cocok.

Soal 5.5 - 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!

Jawab: Kode yang dibuat sebagai berikut:

```
43 $pattern = '/go?od/';
44 $text = 'god is good.';
45 if (preg_match($pattern, $text, $matches)) {
46     echo "Cocokkan: " . $matches[0];
47 } else {
48     echo "Tidak ada yang cocok!";
49 }
```

Maka output yang dihasilkan juga masih sama dengan sebelumnya:



Huruf kecil ditemukan

Cocokkan: 123

I like banana pie

Cocokkan: god

Cocokkan: god

Tanda tanya pada variabel pattern tersebut berarti menunjukkan bahwa karakter “o” setelah “g” adalah opsional. Jadi, jika terdapat karakter o ataupun tidak setelah karakter g, maka itu akan dianggap benar.

Soal 5.6 - 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!

Jawab: Kode yang dibuat sebagai berikut:

```
53 $pattern = '/go{1,2}d/';  
54 $text = 'god is good.';  
55 if (preg_match($pattern, $text, $matches)) {  
56     echo "Cocokkan: " . $matches[0];  
57 } else {  
58     echo "Tidak ada yang cocok!";  
59 }
```

Maka hasilnya adalah sebagai berikut:



Huruf kecil ditemukan

Cocokkan: 123

I like banana pie

Cocokkan: god

Cocokkan: god

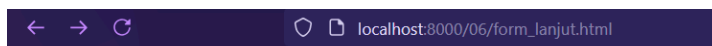
Cocokkan: god

Pattern tersebut berarti akan mencocokkan apakah karakter o ada minimal 1 karakter dan maksimal 2 karakter. Jika memenuhi, maka hasilnya akan ditampung di variabel \$matches

6. Form Lanjut

Soal 6.1 - Catat di sini apa yang anda amati dari penambahan kode program di atas!

Jawab: Hasil yang dirender di web sebagai berikut:



Contoh Form

Pilih buah:

Pilih warna favorit:

☒ Merah

☐ Biru

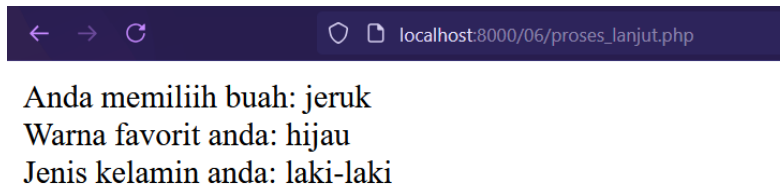
☒ Hijau

Pilih jenis kelamin:

☒ Laki-laki

☐ Perempuan

Jika submit maka hasilnya sebagai berikut:



Soal 6.2 - 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.

Jawab: Tampilan awal akan seperti berikut:

← → ↻ localhost:8000/06/form_ajax.html

Form Contoh

Pilih Buah Apel ▼

Pilih Warna Favorit

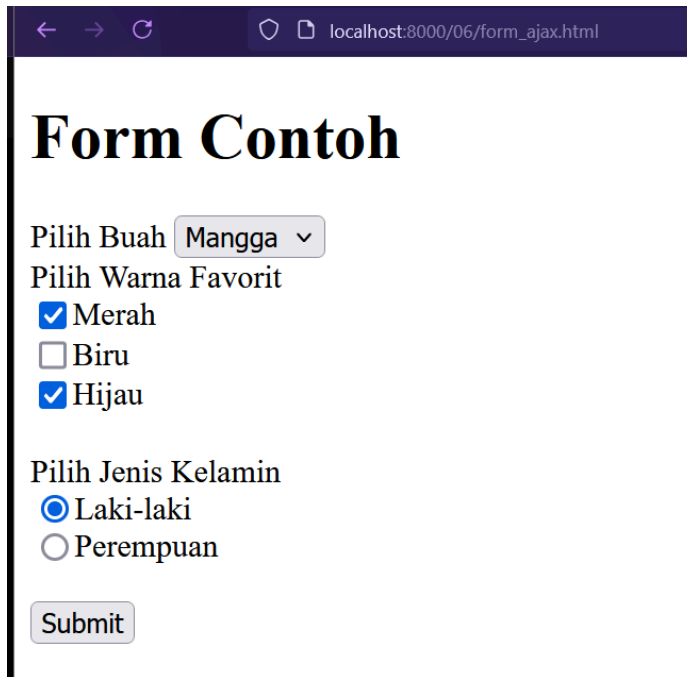
☐ Merah
☐ Biru
☐ Hijau

Pilih Jenis Kelamin

☐ Laki-laki
☐ Perempuan

Submit

Kemudian berikut adalah simulasi pengiriman sebuah form



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/06/form_ajax.html'. The page has a title 'Form Contoh'. Below the title, there are three sections: 'Pilih Buah' with a dropdown menu showing 'Mangga'; 'Pilih Warna Favorit' with three checkboxes: 'Merah' (checked), 'Biru' (unchecked), and 'Hijau' (checked); and 'Pilih Jenis Kelamin' with two radio buttons: 'Laki-laki' (selected) and 'Perempuan' (unselected). At the bottom of the form is a 'Submit' button.

Maka hasil form bisa dilihat di bawah tombol “submit”

Submit
Anda memilih buah: mangga
Warna favorit anda: hijau
Jenis kelamin anda: laki-laki

Proses tersebut dilakukan oleh jQuery yang berusaha melakukan request ke server. Server yang handle tugas tersebut adalah kode program PHP. Dari kode program PHP akan diolah data-data yang ada, kemudian data response akan dikirim kembali ke pengguna. jQuery handle-nya dengan menambahkan teks yang menjadi teks hasil response server di bawah form yang telah dibuat.

7. Validasi Form

Soal 7.1 - Catat di sini apa yang anda amati dari penambahan kode program di atas!

Jawab: Berikut adalah tampilan awal dari halaman web yang dibuat:



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/07/form_validasi.html'. The page has a title 'Form Input dengan Validasi'. Below the title, there are two text input fields: 'Nama:' with the value 'Budiman' and 'Email:' with the value 'budi@berbudi.id'. Below the input fields is a 'Submit' button.

Ketika data berhasil dikirimkan dan data benar, maka pengguna akan mendapatkan teks berikut:



Data berhasil dikirim:

1. Nama: Budiman
2. Email: budi@berbudi.id

Berikut adalah contoh ketika terjadi kesalahan pengiriman form:



Nama harus diisi
Email harus diisi

Soal 7.2 - Catat di sini apa yang anda amati dari penambahan kode program di atas!

Jawab: Tampilan awal akan seperti berikut:

A screenshot of a web browser showing a form titled 'Form Input dengan Validasi'. The form has two input fields: 'Nama:' and 'Email:'. Below the 'Email:' field is a 'Submit' button. The browser's address bar shows 'localhost:8000/07/form_validasi.html'.

Dan berikut adalah simulasi pengisian form dengan data yang salah

A screenshot of a web browser showing a form titled 'Form Input dengan Validasi'. The form has two input fields: 'Nama:' and 'Email:'. The 'Nama:' field contains the text 'bejo'. The 'Email:' field is empty. To the right of the 'Email:' field, there is a red error message: 'Email harus diisi!'. Below the 'Email:' field is a 'Submit' button. The browser's address bar shows 'localhost:8000/07/form_validasi.html'.

Bisa dilihat bahwa validasi juga tak hanya dilakukan oleh PHP. jQuery juga turut serta dalam melakukan validasi data pengguna. Jika kondisi tersebut diterapkan, server bisa lebih ringan kerjanya karena sebagian tugas sudah diserahkan di client. Jika terdapat kesalahan data di client, maka request akan dibatalkan sehingga server tidak perlu melakukan apapun.

Soal 7.3 - Buatlah script langkah 6 dengan menggunakan ajax. Catat di sini apa yang anda amati dari penambahan kode program di atas!

Jawab: Kode program yang dibuat hanya melakukan modifikasi sederhana pada JavaScript berikut:



```
<script>
$(document).ready(() => {
    $("#myForm").submit(function (e) {
        e.preventDefault()

        const nama = $("#nama").val()
        const email = $("#email").val()
        let valid = true

        if (nama === "") {
            $("#nama-error").text("Nama harus diisi!")
            valid = false
        }

        if (email === "") {
            $("#email-error").text("Email harus diisi!")
            valid = false
        }

        if (valid) {
            $.ajax({
                url: 'proses_validasi.php',
                type: 'POST',
                data: $(this).serialize(),
                success: (res) => {
                    $("#hasil").html(res)
                }
            })
        }
    })
})
</script>
```

Hal yang paling membedakan adalah eksekusi yang harus dilakukan ketika data-data yang dikirimkan sudah valid. jQuery akan melakukan request ke server dan dia akan handle response yang akan diterima.

Berikut adalah contoh ketika data berhasil dikirimkan:



Form Input dengan Validasi

Nama:

Email:



Form Input dengan Validasi

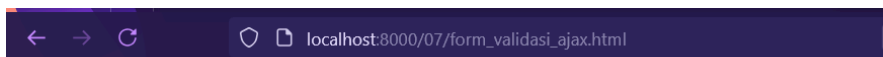
Nama:

Email:

Data berhasil dikirim:

1. Nama: bagus
2. Email: bagus@sekali.com

Berikut adalah contoh ketika terdapat data yang salah kirim



Form Input dengan Validasi

Nama:

Email:

Email harus diisi!

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!

Jawab: Kode program yang ditambahkan sebagai berikut:

HTML:

```
<label for="email">Paassword: </label>
<input type="password" id="password" name="password">
<span id="password-error" style="color: red"></span>
<hr>
```

jQuery:

```
if (password.length < 8) {
    $("#password-error").text("Password harus diisi minimal 8 karakter")
    valid = false
} else {
    $("#password-error").text("")
}
```

PHP:

```
if (empty($password)) {
    $errors[] = "Password harus diisi";
}
```

Berikut adalah contoh pengisian data sesuai dengan petunjuk:

← → ↻ localhost:8000/07/form_validasi_ajax.html 150%

Form Input dengan Validasi

Nama:

Email:

Paassword:

Data berhasil dikirim:

1. Nama: Budi
2. Email: budi@baik.com
3. Password: admin123-anti-hack

Dan berikut adalah kondisi di mana pengguna menginputkan password kurang dari 8 karakter:

← → ↻ localhost:8000/07/form_validasi_ajax.html 150% ☆

Form Input dengan Validasi

Nama:

Email:

Paassword: Password harus diisi minimal 8 karakter