**LAPORAN PRAKTIKUM**

**PEMROGRAMAN WEB I**



**Disusun oleh :**

PUPUT SURYA NINGTYAS

V3523015

**Dosen**

**Masbahah, S.pd., M.pd.**

**PSDKU D-III TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**2024**

**BAB 1**

**PENDAHULUAN**

**1.1 Latar Belakang**

Bahasa pemrograman PHP (Hypertext Preprocessor) merupakan salah satu bahasa pemrograman server-side yang paling populer untuk membangun website dinamis. Salah satu fitur penting dalam pengembangan web dengan PHP adalah kemampuan untuk menangani formulir (form handling). Formulir memungkinkan pengguna untuk berinteraksi dengan website dan memasukkan data yang kemudian dapat diproses dan disimpan oleh server.

**1.2 Tujuan**

* Mempelajari cara menangani formulir dalam pemrograman web dengan PHP
* Mempelajari berbagai atribut form yang dapat digunakan untuk mengontrol dan memvalidasi data input
* Memahami berbagai elemen form yang dapat digunakan untuk membangun formulir interaktif.

**1.3 Manfaat**

* Memahami konsep dasar form handling dalam PHP
* Mampu membangun formulir interaktif dengan PHP
* Mampu memvalidasi data input yang dikirimkan dari formulir
* Mampu memproses dan menyimpan data formulir dengan PHP

**1.4 Alat dan Bahan**

* Komputer dengan sistem operasi Windows, MacOS, atau Linux
* Visual Studio Code
* XAMPP
* Browser web

**BAB 2**

**TINJAUAN PUSTAKA**

Form Handling adalah proses pengelolaan data yang dikirimkan melalui formulir HTML, yang biasanya melibatkan validasi, penyimpanan, dan pengiriman data. Form Attributes merujuk pada atribut-atribut yang dapat ditambahkan ke elemen <form> dalam HTML untuk mengontrol perilaku dan tampilan formulir. Form Elements adalah elemen-elemen HTML seperti input, textarea, dan select yang digunakan untuk mengumpulkan data dari pengguna. Sedangkan Input Types adalah jenis-jenis input yang dapat digunakan dalam elemen <input>, seperti text, password, checkbox, radio, dan lain-lain. Memahami konsep-konsep ini penting untuk membangun aplikasi web yang interaktif dan dinamis dengan menggunakan PHP. Dengan memanfaatkan penanganan formulir PHP, atribut formulir, elemen formulir, dan jenis-jenis input, pengembang web dapat membuat pengalaman pengguna yang lebih baik dan memproses data dengan lebih efisien.

**BAB 3**

**HASIL DAN PEMBAHASAN**

1. **Kode**

**Kode yang ditambahkan pada halaman 1**

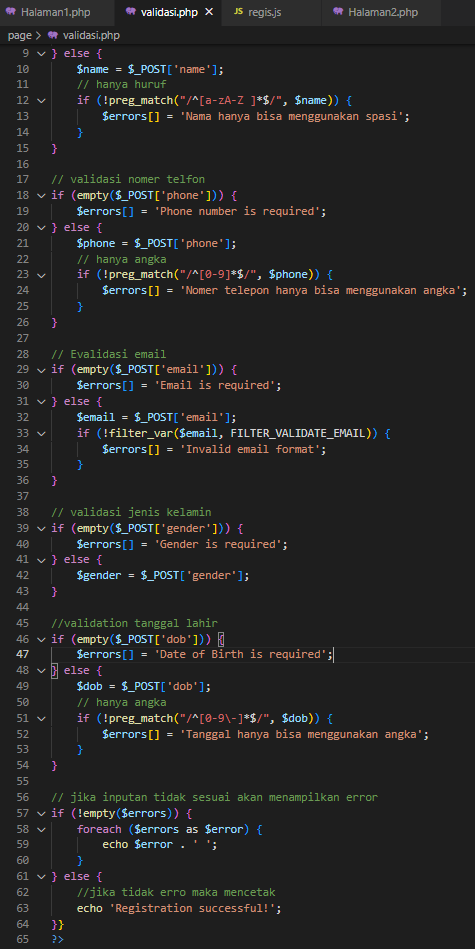
****

**Penjelasan kode :**

Kode di atas adalah formulir HTML untuk registrasi anggota dengan beberapa atribut dan elemen penting. Mari kita jelaskan dengan lebih rinci:

1. **`<section id="registration">`:** Ini adalah sebuah elemen section HTML yang memiliki atribut id dengan nilai "registration". Ini digunakan untuk mengelompokkan dan menandai bagian formulir registrasi dalam halaman web.
2. **`<h2>Member Registration</h2>**`: Ini adalah heading level 2 (h2) yang menampilkan teks "Member Registration", memberikan judul atau label untuk bagian registrasi.
3. **`<form id="registration-form" action="validasi.php" method="POST">`:** Ini adalah elemen form HTML yang memiliki atribut id dengan nilai "registration-form". Atribut action menunjukkan ke alamat URL di mana data formulir akan dikirimkan untuk divalidasi,ke "validasi.php". Atribut method menunjukkan metode pengiriman data menggunakan metode POST.
4. `**<label for="name">Nama Lengkap:</label>` dan `<input type="text" id="name" name="name" required><br><br>`:** Ini adalah elemen label dan input untuk memasukkan nama lengkap pengguna. Atribut "for" pada label terhubung dengan id input sehingga ketika label diklik, fokus akan berpindah ke input tersebut. Input memiliki tipe "text", id "name", dan atribut "required" yang menunjukkan bahwa input ini harus diisi sebelum pengiriman formulir.
5. **`<label for="phone">Nomor Telepon:</label>` dan `<input type="tel" id="phone" name="phone" pattern="[0-9]{12}" required><br><br>`:** Ini adalah elemen label dan input untuk memasukkan nomor telepon pengguna. Input memiliki tipe "tel", id "phone", dan atribut "pattern" yang membatasi input hanya menerima angka dengan panjang tepat 12 digit. Input juga memiliki atribut "required".
6. **`<label for="email">Email:</label>` dan `<input type="email" id="email" name="email" required><br><br>`**: Ini adalah elemen label dan input untuk memasukkan alamat email pengguna. Input memiliki tipe "email", id "email", dan atribut "required" untuk memastikan pengguna memasukkan alamat email yang valid sebelum mengirimkan formulir.
7. **`<label for="gender">Jenis Kelamin:</label>` dan `<select id="gender" name="gender" required>...`:** Ini adalah elemen label dan select untuk memilih jenis kelamin pengguna. Select memiliki id "gender" dan name "gender", dan atribut "required" untuk memastikan pengguna memilih jenis kelamin sebelum mengirimkan formulir.
8. **`<label for="dob">Tanggal Lahir:</label>` dan `<input type="date" id="dob" name="dob" required><br><br>`:** Ini adalah elemen label dan input untuk memasukkan tanggal lahir pengguna. Input memiliki tipe "date", id "dob", name "dob", dan atribut "required".
9. **`<input type="submit" value="Register" id="submit-button">`**: Ini adalah tombol submit untuk mengirimkan formulir. Ketika tombol ini diklik, data dari formulir akan dikirimkan ke "validasi.php" untuk diproses.
10. **`<div id="notification"></div>`:** Ini adalah elemen div yang digunakan untuk menampilkan pemberitahuan atau pesan kepada pengguna setelah pengiriman formulir. Pesan-pesan tersebut dapat ditampilkan menggunakan JavaScript setelah validasi formulir dilakukan di sisi server.

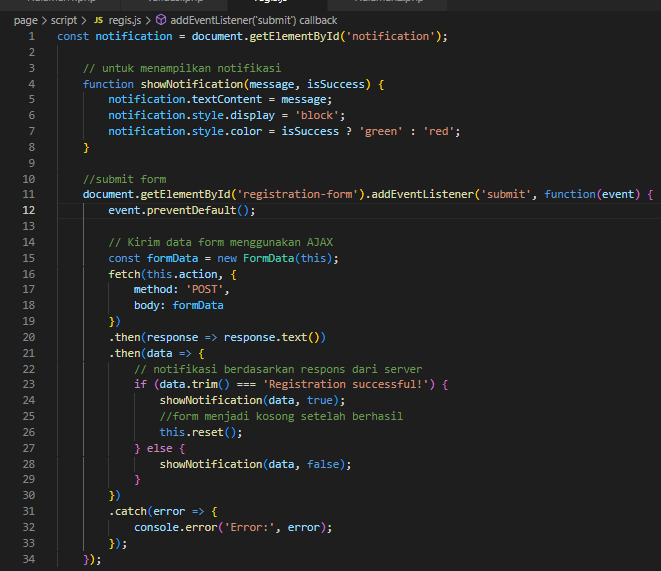
**Kode php untuk validasi form inputan**



**Penejelasan kode:**

1. **if ($\_SERVER["REQUEST\_METHOD"] == "POST") { ... }**: Ini adalah kondisi yang memeriksa apakah metode permintaan HTTP adalah POST. Jika benar, artinya formulir telah disubmit.
2. **// Validasi inputs**: Ini adalah komentar yang menandakan awal dari blok kode untuk validasi input.
3. **$errors = array();**: Ini adalah variabel yang digunakan untuk menyimpan pesan kesalahan validasi.
4. Validasi Nama:
   * Pertama, dilakukan pemeriksaan apakah input nama kosong. Jika kosong, maka pesan kesalahan "Name is required" akan ditambahkan ke dalam array $errors.
   * Jika tidak kosong, maka input nama akan diperiksa apakah hanya berisi huruf dan spasi. Jika tidak, pesan kesalahan "Nama hanya bisa menggunakan spasi" akan ditambahkan ke dalam array $errors.
5. Validasi Nomor Telepon:
   * Sama seperti validasi nama, nomor telepon juga diperiksa apakah kosong atau tidak. Jika kosong, pesan kesalahan "Phone number is required" akan ditambahkan ke dalam array $errors.
   * Jika tidak kosong, nomor telepon akan diperiksa apakah hanya berisi angka. Jika tidak, pesan kesalahan "Nomer telepon hanya bisa menggunakan angka" akan ditambahkan ke dalam array $errors.
6. Validasi Email:
   * Input email akan diperiksa apakah kosong atau tidak. Jika kosong, pesan kesalahan "Email is required" akan ditambahkan ke dalam array $errors.
   * Jika tidak kosong, email akan diperiksa apakah formatnya valid menggunakan fungsi **filter\_var()** dengan parameter **FILTER\_VALIDATE\_EMAIL**. Jika tidak valid, pesan kesalahan "Invalid email format" akan ditambahkan ke dalam array $errors.
7. Validasi Jenis Kelamin:
   * Jenis kelamin akan diperiksa apakah kosong atau tidak. Jika kosong, pesan kesalahan "Gender is required" akan ditambahkan ke dalam array $errors.
8. Validasi Tanggal Lahir:
   * Tanggal lahir akan diperiksa apakah kosong atau tidak. Jika kosong, pesan kesalahan "Date of Birth is required" akan ditambahkan ke dalam array $errors.
   * Jika tidak kosong, tanggal lahir akan diperiksa apakah hanya berisi angka dan tanda hubung (-). Jika tidak, pesan kesalahan "Tanggal hanya bisa menggunakan angka" akan ditambahkan ke dalam array $errors.
9. Setelah semua validasi selesai, dilakukan pengecekan apakah terdapat pesan kesalahan dalam array $errors. Jika ada, pesan-pesan kesalahan tersebut akan dicetak. Jika tidak ada, akan dicetak pesan "Registration successful!", menandakan bahwa proses registrasi berhasil.

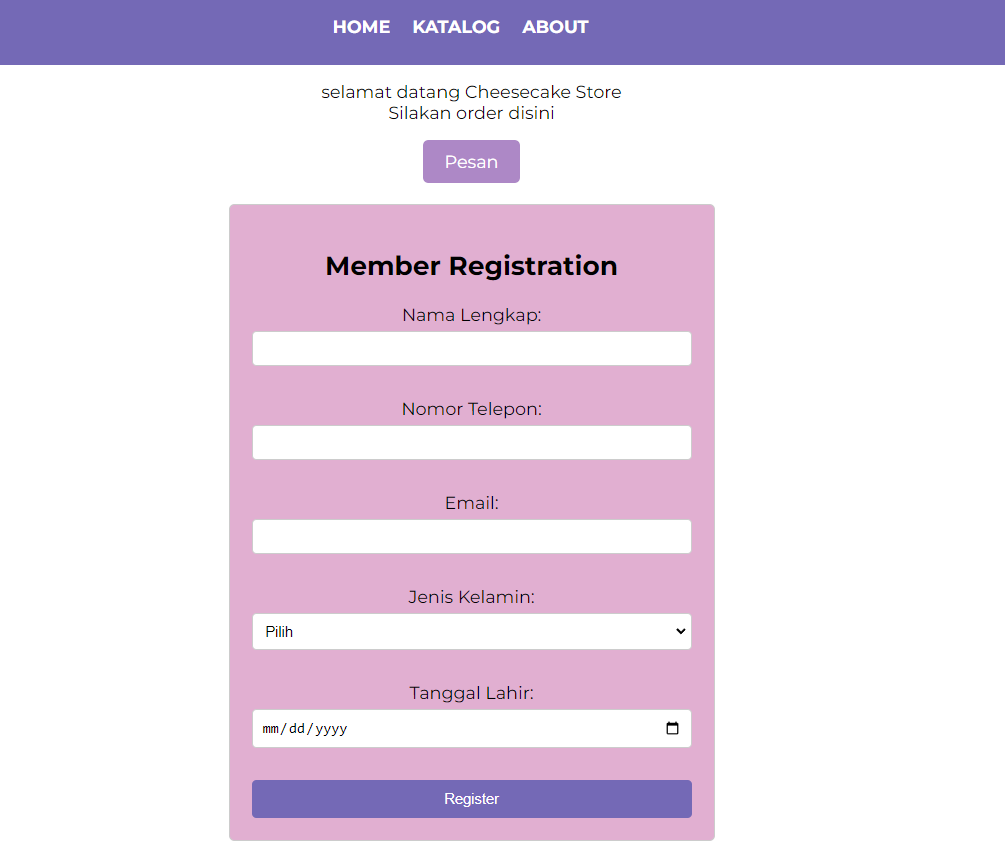
**Kode js untuk notifikasi registrasi berhasil atau tidak**

****

**Penjelasan kode :**

1. **const notification = document.getElementById('notification');**: Ini adalah konstanta yang menunjuk ke elemen HTML dengan id "notification", yang digunakan untuk menampilkan pesan notifikasi kepada pengguna.
2. **function showNotification(message, isSuccess) { ... }**: Ini adalah fungsi yang menerima dua parameter, yaitu pesan yang akan ditampilkan dan sebuah boolean yang menunjukkan apakah operasi berhasil atau tidak. Fungsi ini mengubah teks dan warna notifikasi sesuai dengan parameter yang diterima.
3. **document.getElementById('registration-form').addEventListener('submit', function(event) { ... });**: Ini adalah event listener yang menangkap saat formulir dengan id "registration-form" disubmit. Ketika formulir disubmit, fungsi callback dijalankan.
4. **event.preventDefault();**: Ini adalah perintah untuk mencegah perilaku default browser saat formulir disubmit, sehingga pengiriman formulir akan ditangani menggunakan JavaScript.
5. **const formData = new FormData(this);**: Ini adalah objek FormData yang digunakan untuk mengumpulkan data formulir.
6. **fetch(this.action, { ... })**: Ini adalah API Fetch yang mengirimkan data formulir menggunakan metode POST ke alamat yang ditentukan dalam atribut "action" dari formulir.
7. **.then(response => response.text())**: Ini adalah metode untuk mengubah respons dari server menjadi teks.
8. **.then(data => { ... })**: Ini adalah blok then yang akan dijalankan setelah respons dari server diterima. Di dalamnya, notifikasi akan ditampilkan berdasarkan respons yang diterima. Jika respons adalah "Registration successful!", notifikasi akan ditampilkan dengan warna hijau, dan formulir akan direset. Jika tidak, notifikasi akan ditampilkan dengan warna merah.
9. **.catch(error => { ... })**: Ini adalah blok catch yang menangani kesalahan jika terjadi kesalahan dalam pengiriman atau respons dari server.

**2. Hasil**



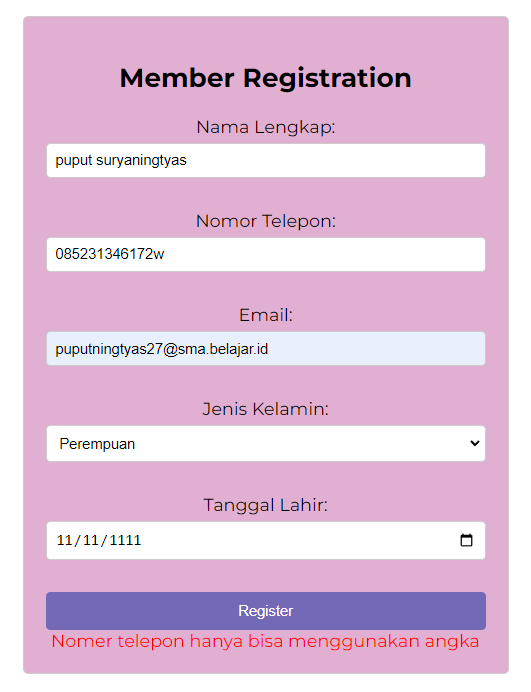
* Error jika tidak memilih jenis kelamin

****

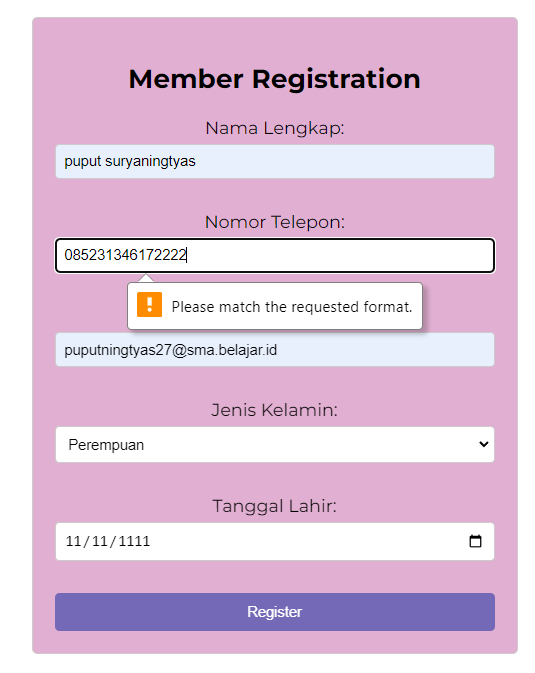
* Error jika memasukan nama menggunakan angka, akan muncul notifikasi “nama hanya bisa menggunakan spasi



* Error jika memasukan nomor telepon menggunakan huruf, akan muncu notifikasi “nomer telepon hanya bisa menggunkaan angka



* Error jika memasukkan angka lebih dari 12

****

* Jika registrasi berhasil maka akan muncul notifikasi “registrasi successful”, dan otomatis menghapus inputan pengguna

****

**BAB 4**

**KESIMPULAN**

**4.1 Kesimpulan**

Pemahaman yang mendalam tentang pengelolaan formulir HTML dan validasi data sangat penting dalam pengembangan aplikasi web. Dengan memanfaatkan PHP sebagai bahasa pemrograman server-side, pengembang dapat membuat formulir yang interaktif dan aman, yang memungkinkan pengguna untuk mengirimkan data dengan mudah dan menghindari masalah keamanan. Validasi data di sisi server juga sangat penting untuk memastikan keabsahan data sebelum data tersebut diproses lebih lanjut atau disimpan ke dalam basis data.

**4.2 Saran**

Untuk memperkaya pengalaman praktikum, disarankan untuk menyediakan proyek-proyek praktis yang melibatkan pengembangan aplikasi web dari awal hingga akhir. Proyek-proyek tersebut dapat mencakup pembuatan formulir pendaftaran, aplikasi manajemen konten sederhana, atau aplikasi e-commerce sederhana. Dalam praktikum, mahasiswa dapat belajar tentang konsep-konsep penting seperti pengelolaan sesi, validasi formulir, interaksi dengan basis data, dan aspek keamanan web.

**DAFTAR PUSTAKA**

[https://www.w3schools.com/php/php\_forms.asp](https://www.w3schools.com/php/php_forms.asp" \t "_blank)