

**LAPORAN PRAKTIKUM LAB D1**  
**PENGEMBANGAN BERBASIS PLATFORM**  
“Pemrosesan Form”



**DISUSUN OLEH:**  
RESMA ADI NUGROHO  
24060121120021

**DEPARTEMEN INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**2023**

# BAB I

## PENDAHULUAN

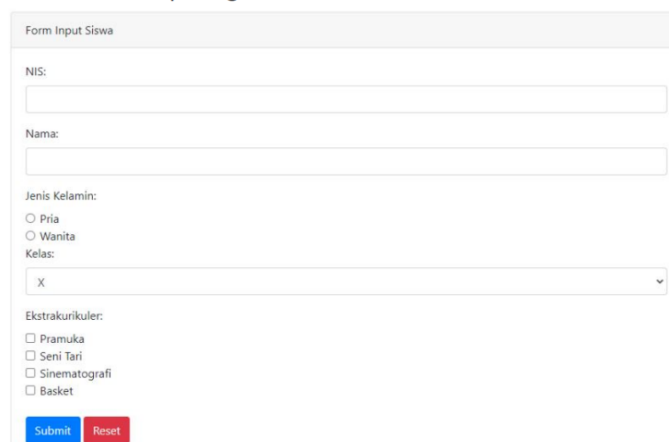
### 1.1 Tujuan

1. Membaca isi form dengan PHP menggunakan method GET maupun POST
2. Melakukan validasi input data dengan PHP
3. Menampilkan isi kembali form

### 1.2 Rumusan Masalah

1. Buatlah sebuah form pengisian dengan ketentuan sebagai berikut:

Buatlah form seperti gambar berikut:



The image shows a web form titled "Form Input Siswa". It contains the following fields and controls:

- NIS:** A text input field.
- Nama:** A text input field.
- Jenis Kelamin:** Two radio buttons labeled "Pria" and "Wanita".
- Kelas:** A dropdown menu with "X" selected.
- Ekstrakurikuler:** Three checkboxes labeled "Pramuka", "Seni Tari", and "Basket".
- Buttons:** A blue "Submit" button and a red "Reset" button.

Aturan validasi:

- Semua field harus diisi.
- NIS terdiri atas 10 karakter dan hanya boleh berisi angka 0..9.
- Jika siswa kelas X atau XI, maka program menampilkan pilihan ekstrakurikuler. Siswa wajib memilih kegiatan ekstrakurikuler yang diminati, minimal 1 maksimal 3. Jika kelas XII siswa tidak boleh mengikuti kegiatan ekstrakurikuler, sehingga program tidak perlu menampilkan kegiatan ekstrakurikuler.

## **BAB II**

### **DASAR TEORI**

#### **2.1 HTTP Method**

HTTP (Hypertext Transfer Protocol) adalah protokol yang digunakan untuk mengirimkan data antara klien (misalnya, browser web) dan server web. Ada dua metode HTTP yang umum digunakan untuk mengirimkan data: GET dan POST. Keduanya memiliki perbedaan dalam cara mereka mengirimkan dan mengelola data. Berikut adalah penjelasan singkat tentang perbedaan antara metode GET dan POST:

##### **HTTP GET**

1. **Penggunaan:** Metode GET digunakan untuk mengambil data dari *server*. Ini adalah metode yang umum digunakan untuk menampilkan halaman web atau mengambil data dari *database*.
2. **Parameter:** Data dikirimkan sebagai bagian dari URL, biasanya dalam bentuk string query. Misalnya, <http://example.com/page?name=John&age=30>.
3. **Tampilan di URL:** Parameter terlihat dalam URL, sehingga data dapat dengan mudah dilihat oleh siapa saja yang melihat URL.
4. **Caching:** Data dapat di-cache oleh browser, yang berarti hasil permintaan sebelumnya dapat disimpan untuk penggunaan berikutnya.

##### **HTTP POST**

1. **Penggunaan:** Metode POST digunakan untuk mengirim data ke *server*. Ini digunakan ketika Anda ingin mengirim data yang sensitif atau data yang akan dimasukkan ke dalam *database*.
2. **Parameter:** Data dikirim sebagai bagian dari tubuh permintaan HTTP, dan tidak terlihat dalam URL. Oleh karena itu, data POST lebih aman dari mata pengguna.
3. **Tampilan di URL:** Parameter tidak terlihat dalam URL, sehingga lebih aman dan cocok untuk mengirim data sensitif seperti kata sandi.
4. **Caching:** Data tidak di-cache oleh browser, sehingga data tidak dapat disimpan untuk penggunaan berikutnya.

## BAB III

### PEMBAHASAN

#### 3.1 Form Input Siswa

Kode file PHP untuk file user\_form\_siswa.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- cdn bootstrap -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.
css" integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
    <title>Form Input Siswa</title>
    <style>
        .error {color: #FF0000;}
    </style>
</head>
<body>
<?php
    if(isset($_POST['submit'])) {
        $nama = test_input($_POST['nama']);
        if(empty($nama)){
            $error_nama = "Nama harus diisi";
        } elseif (!preg_match("/^[a-zA-Z ]*$/",$nama)) {
            $error_nama = "Nama hanya dapat berisi huruf dan spasi";
        }

        $nis = test_input($_POST['nis']);
        if(empty($nis)){
            $error_nis = "NIS harus diisi";
        } elseif (!preg_match('/^[0-9]{10}$/', $nis)) {
            $error_nis = "Format NIS salah";
        }

        $kelas = test_input($_POST['kelas']);
        if(empty($kelas)){
            $error_kelas = "Kelas harus diisi";
        }
        if(!isset($_POST['jenis_kelamin'])){
            $error_jenis_kelamin = "Jenis kelamin harus diisi";
        }
        if ($kelas != 'XII') {
            // Jika kelas X atau XI, validasi ekstrakurikuler
            if(!isset($_POST['ekskul']) || count($_POST['ekskul']) < 1 ||
count($_POST['ekskul']) > 3) {
                $error_ekskul = "Pilih 1-3 ekstrakurikuler";
            }
        }
    }
    function test_input($data){
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }
?>
```

```

<div class="card m-5">
<div class="card-header">
    Form Input Siswa
</div>
<form action="" autocomplete="on" method="POST" class="m-4">

<div class="form-group">
    <label for="nis">NIS:</label>
    <input type="number" name="nis" id="nis" class="form-control"
minlength="10" maxlength="10" value="<?php if(isset($nis)) {echo
$nis;}?>">
    <div class="error"><?php if(isset($error_nis)) echo $error_nis?></div>
</div>

<div class="form-group">
    <label for="nama">Nama:</label>
    <input type="text" name="nama" id="nama" class="form-control"
maxlength="50" value="<?php if(isset($nama)) {echo $nama;}?>">
    <div class="error"><?php if(isset($error_nama)) echo
$error_nama?></div>
</div>

<label>Jenis Kelamin</label>
<div class="form-check">
    <label class="form-check-label">
        <input type="radio" class="form-check-input" name="jenis_kelamin"
value="pria" <?php if(isset($jenis_kelamin) && $jenis_kelamin=="pria")
{echo "checked";}?>>Pria
    </label>
</div>
<div class="form-check">
    <label class="form-check-label">
        <input type="radio" class="form-check-input" name="jenis_kelamin"
value="wanita" <?php if(isset($jenis_kelamin) && $jenis_kelamin=="wanita")
{echo "checked";}?>>Wanita
    </label>
</div>
<div class="error"><?php if(isset($error_jenis_kelamin)) echo
$error_jenis_kelamin?></div>

<div class="form-group">
    <label for="kelas">Kelas</label>
    <select name="kelas" id="kelas" class="form-control"
onchange="toggleEkstrakurikuler()">
        <option value="X" <?php if(isset($kelas) && $kelas=="X") {echo
'selected="true"';}?>>X</option>
        <option value="XI" <?php if(isset($kelas) && $kelas=="XI") {echo
'selected="true"';}?>>XI</option>
        <option value="XII" <?php if(isset($kelas) && $kelas=="XII") {echo
'selected="true"';}?>>XII</option>
    </select>
    <div class="error"><?php if(isset($error_kelas)) echo
$error_kelas?></div>
</div>

<div id="ekstrakurikuler">
    <label>Ekstrakurikuler</label>
    <div class="form-check">
        <label class="form-check-label">
            <input type="checkbox" class="form-check-input"
name="ekskul[]" value="Pramuka">Pramuka
        </label>
    </div>

    <div class="form-check">
        <label class="form-check-label">
            <input type="checkbox" class="form-check-input"
name="ekskul[]" value="Seni Tari">Seni Tari
        </label>
    </div>

```

```

        <div class="form-check">
            <label class="form-check-label">
                <input type="checkbox" class="form-check-input"
name="ekskul[]" value="Sinematografi">Sinematografi
            </label>
        </div>

        <div class="form-check">
            <label class="form-check-label">
                <input type="checkbox" class="form-check-input"
name="ekskul[]" value="Basket">Basket
            </label>
            <div class="error"><?php if(isset($error_ekskul)) echo
$error_ekskul?></div>
        </div>
    </div>

    <br>
    <!-- submit, reset dan button -->
    <button type="submit" class="btn btn-primary" name="submit"
value="submit">Submit</button>
    <button type="reset" class="btn btn-danger">Reset</button>
</form>
</div>

<?php
    if(isset($_POST["submit"])) {
        echo "<h3>Your Input</h3>";
        echo "NIS: " . $_POST["nis"]. "<br/>";
        echo "Nama: " . $_POST["nama"]. "<br/>";
        if(!isset($_POST["jenis_kelamin"])) {
            echo "Jenis kelamin belum dipilih<br/>";
        } else {
            echo "Jenis kelamin: " . $_POST["jenis_kelamin"]. "<br/>";
        }
        echo "Kelas: " . $_POST["kelas"]. "<br/>";

        if ($kelas != 'XII') {
            // Jika kelas X atau XI, tampilkan ekstrakurikuler yang
dipilih
            if(!isset($_POST['ekskul']) || count($_POST['ekskul']) < 1 ||
count($_POST['ekskul']) > 3) {
                echo "Ekstrakurikuler belum dipilih";
            } else {
                echo "Ekstrakurikuler: ";
                foreach($_POST['ekskul'] as $ekskul) {
                    echo $ekskul . ", ";
                }
            }
        }
    }
?>
<script>
function toggleEkstrakurikuler() {
    var kelasSelect = document.getElementById("kelas");
    var ekstrakurikulerDiv = document.getElementById("ekstrakurikuler");

    if (kelasSelect.value == "XII") {
        ekstrakurikulerDiv.style.display = "none";
    } else {
        ekstrakurikulerDiv.style.display = "block";
    }
}
</script>
</body>
</html>

```

Kode diatas merupakan kode untuk menampilkan sebuah form input siswa dengan beberapa field input seperti nama, nis, kelas, jenis kelamin dan ekskul. Agar inputan dapat sesuai dengan ketentuan validasi yang sudah ditentukan maka terdapat beberapa konfigurasi atau validasi pada inputnya yang diimplementasikan baik pada HTML maupun PHP.

1. NIS akan memiliki validasi nilainya tidak boleh kosong dan panjangnya harus 10 dan input ini merupakan sebuah number atau bilangan dari 0 sampai 9 maka dapat diberikan sebuah validasi pada htmlnya dengan min dan max length serta jenis tipe inputnya menjadi number. Namun untuk memeriksa lebih lanjut lagi maka pada kode PHP perlu ditambahkan juga validasi untuk nilai input dengan menggunakan regular expression seperti dibawah ini:

```
$nis = test_input($_POST['nis']);
if(empty($nis)){
    $error_nis = "NIS harus diisi";
} elseif (!preg_match('/^[0-9]{10}$/', $nis)) {
    $error_nis = "Format NIS salah";
}
```

2. Nama akan memiliki validasi nilainya tidak boleh kosong dan hanya dapat diisi dengan huruf dan spasi. Validasi ini juga dilakukan pada HTML dan PHP, dengan bertipe sebuah text dan validasi pada PHP menggunakan regular expression seperti dibawah ini:

```
$nama = test_input($_POST['nama']);
if(empty($nama)){
    $error_nama = "Nama harus diisi";
} elseif (!preg_match("/^[a-zA-Z ]*$/", $nama)) {
    $error_nama = "Nama hanya dapat berisi huruf dan spasi";
}
```

3. Jenis kelamin akan sama memiliki validasi bahwa nilai tidak boleh kosong.

```
if(!isset($_POST['jenis_kelamin'])){
    $error_jenis_kelamin = "Jenis kelamin harus diisi";
}
```

4. Kelas memiliki validasi yang sama juga dengan nilainya tidak boleh kosong dan terdapat fungsi perubahan atau dalam HTML bernama onchange yang akan memanggil sebuah fungsi javascript untuk mengecek apakah nilai kelas sama dengan XII, jika false maka checkbox ekskul akan muncul. Beberapa catatan pada bagian ini apabila ditujukan untuk menampilkan dan menyembunyikan checkbox dengan menggunakan kode HTML dan PHP saja maka dibutuhkan sebuah tombol trigger untuk mengecek apakah input atau value dari kelas bernilai XII. Namun berdasarkan atau spesifikasi yang diberikan, maka tombol trigger tidak digunakan

dan menggunakan sebuah fungsi yang ditempatkan pada onchange untuk menampilkan dan menyembunyikan checkbox sesuai dengan ketentuan sebelumnya.

```
$kelas = test_input($_POST['kelas']);  
if(empty($kelas)){  
    $error_kelas = "Kelas harus diisi";  
}
```

Sedangkan untuk fungsi menampilkan dan menyembunyikan checkbox ekstrakurikuler dapat dilihat pada kode dibawah:

```
function toggleEkstrakurikuler() {  
    var kelasSelect = document.getElementById("kelas");  
    var ekstrakurikulerDiv = document.getElementById("ekstrakurikuler");  
  
    if (kelasSelect.value == "XII") {  
        ekstrakurikulerDiv.style.display = "none";  
    } else {  
        ekstrakurikulerDiv.style.display = "block";  
    }  
}
```

5. Ekstrakurikuler akan memiliki validasi nilainya tidak boleh kosong secara spesifik bahwa siswa harus memilih 1 sampai 3 ekstrakurikuler yang tersedia. Namun hal ini berlaku hanya apabila siswa bukan kelas XII.

```
if ($kelas != 'XII') {  
    // Jika kelas X atau XI, validasi ekstrakurikuler  
    if(!isset($_POST['ekskul']) || count($_POST['ekskul']) < 1 ||  
count($_POST['ekskul']) > 3) {  
        $error_ekskul = "Pilih 1-3 ekstrakurikuler";  
    }  
}
```

Setelah semua field berhasil diinput dan divalidasi maka proses akan berlanjut untuk menampilkan hasil dari input tersebut, dan untuk metode HTTP yang digunakan adalah POST karena data yang dikirimkan bersifat data pribadi pengguna sehingga proses transfer data apabila data akan dikirim ke database dapat lebih aman.



## **BAB IV**

### **PENUTUP**

Implementasi HTTP GET dan POST adalah aspek kunci dalam pengembangan aplikasi web. Metode HTTP GET digunakan untuk mengambil data dari server berdasarkan URL yang telah ditentukan, sementara metode HTTP POST digunakan untuk mengirim data ke server. Keduanya memiliki peran yang penting dalam berinteraksi dengan sumber daya web, dengan HTTP GET sering digunakan untuk mengambil data yang telah ada, seperti halaman web atau informasi publik, dan HTTP POST digunakan untuk mengirim data yang diperbarui atau sensitif ke server, seperti ketika mengirimkan formulir atau data pengguna.

Penggunaan metode HTTP GET dan POST harus dipertimbangkan dengan bijak dalam pengembangan web. HTTP GET cenderung lebih cocok untuk operasi baca atau tampilan halaman, sementara HTTP POST lebih sesuai untuk operasi yang memengaruhi atau memodifikasi data, seperti mengirimkan data formulir. Keamanan juga menjadi pertimbangan penting, karena data yang dikirim melalui HTTP GET dapat terlihat dalam URL, sementara data yang dikirim melalui HTTP POST tersembunyi.

Selain itu, keamanan data yang dikirim dan diterima harus dijaga dengan baik, dengan penggunaan HTTPS (HTTP Secure) yang disarankan untuk melindungi komunikasi antara klien dan server. Penggunaan metode HTTP GET dan POST yang benar dan aman akan membantu memastikan keberhasilan dan keamanan aplikasi web.

## **DAFTAR PUSTAKA**

"HTTP/1.1 Semantics and Content" - Dokumen resmi dari World Wide Web Consortium (W3C) yang menjelaskan rincian protokol HTTP, termasuk metode GET dan POST.

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

"HTTP - Methods" - Dokumen MDN Web Docs yang memberikan penjelasan lebih lanjut tentang metode HTTP, termasuk GET dan POST.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

"HTTP GET vs POST: What's the Difference?" - Artikel yang memberikan penjelasan singkat tentang perbedaan antara metode GET dan POST.

<https://www.keycdn.com/support/http-methods>