
PEMROGRAMAN WEB

MODUL AJAR

“OPERASI FILE – Create, Read, Update and Delete (CRUD)”

Kelas XI - Jurusan Rekayasa Perangkat Lunak

SMK Negeri 1 Kepanjen

TUJUAN PEMBELAJARAN

1. Menjelaskan prosedur operasi file dalam aplikasi web
2. Menerapkan operasi file dalam halaman web
3. Membuat kode program operasi file dalam halaman web
4. Menguji program aplikasi operasi file dalam halaman web

ALAT DAN BAHAN

1. Alat
 - Laptop atau PC
2. Bahan
 - XAMPP (atau sejenisnya)
 - Text Editor (VSCode, Sublime, Notepad++, dll)

MATERI PEMBELAJARAN

Dalam pemrograman web, Kita mengenal istilah *Client-side* dan *Server-side Scripting*. 2 istilah itu digunakan untuk mengelompokkan dimana kode atau bahasa pemrograman itu dijalankan, sisi client atau sisi server.

Jika diartikan secara bahasa, *Client-side* berarti sisi pengguna dan *Server-side* berarti sisi server/penyedia.

Server-side Scripting

Server-side Scripting adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer server/penyedia. Jadi, setiap kali sebuah web dikunjungi, server akan mengirimkan data-data yang diminta dari database yang kemudian akan ditampilkan di web.

Server-side Script biasanya hanya terdapat pada web dinamis saja. *Server-side Script* memengaruhi berat-tidaknya loading sebuah website bergantung pada kecepatan & spesifikasi komputer server.

Kode program *Server-side* tidak bisa kita lihat karena sifatnya yang rahasia untuk pengguna. Contoh dari *Server-side Script* adalah PHP, ASP, ASP.Net, Java, dan masih banyak lagi.

Bahasa Pemrograman PHP

PHP adalah singkatan dari "PHP: Hypertext Preprocessor", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, nama bahasa ini diubah menjadi "PHP: Hypertext Preprocessor" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke-7.

PHP digunakan karena untuk membuat website dinamis bisa digunakan untuk menyimpan data ke dalam database, membuat halaman yang dapat berubah-ubah sesuai dengan input user, memproses form, dll.

Syntax PHP

Pengertian Syntax PHP adalah aturan penulisan agar mampu dimengerti dengan benar oleh compiler saat membaca bahasa pemrograman. Dalam penulisan PHP yang benar diawali dengan "<?php" dan diakhiri dengan "?>". Dan di dalam File PHP juga dapat berisi tag seperti HTML dan skrip sisi klien seperti JavaScript.

Contoh penerapan syntax PHP:

```
<?php echo "Hello World!"; ?>
```

MySQL

MySQL adalah sistem manajemen database relasional *open source* (RDBMS) dengan *client-server* model. Sedangkan RDBMS merupakan software untuk membuat dan mengelola database berdasarkan pada model relasional.

MySQLi

Dalam bahasa pemrograman (khususnya PHP), jika kita menggunakan MySQL, terdapat dua extension yang bisa kita gunakan, yaitu MySQL extension dan MySQLi extension. Keduanya sama-sama menghubungkan MySQL dengan bahasa pemrograman yang kita pakai.

Penggunaan MySQL extension saat ini sudah tidak direkomendasikan. Maka dari itu dikenalkan MySQLi extension sebagai penggantinya. Pengembangan MySQLi bertujuan untuk mendukung berbagai macam fitur terbaru MySQL server versi 4.1 dan setelahnya.

API

API (*Application Programming Interface*) merupakan sebuah class, metode, fungsi, dan variabel yang dibutuhkan sebuah aplikasi untuk meminta aplikasi lain untuk melakukan tugas. Pada kasus PHP dan MySQL, API dipakai untuk mengomunikasikan PHP dengan database yang ada di MySQL menggunakan PHP extensions.

Mengembangkan aplikasi PHP terkadang membutuhkan koneksi dengan MySQL server. kita tidak perlu khawatir karena di PHP terdapat beberapa opsi API yang tersedia. kita hanya tinggal menyesuaikannya dengan aplikasi yang sedang dikembangkan.

Perbedaan Object Oriented dan Procedural

MySQLi mempunyai dua tipe penggunaan, yaitu *object-oriented* dan *procedural*.

Jika sebelumnya sudah menggunakan MySQL dan ingin migrasi menggunakan MySQLi, ada dua alternatif yang bisa kita pakai, yaitu menggunakan *object oriented* dan *procedural*.

- **MySQLi Tipe *Procedural***

Cara penulisan ini mirip dengan MySQL *extension*. Jadi penulisan fungsi yang dipakai sama dengan penggunaan fungsi-fungsi ketika mengakses database.

Contohnya sama dengan yang sudah dibahas tadi, yaitu ketika MySQL *extension* menggunakan “mysql_connect()”, di MySQL *extension* menggunakan “mysqli_connect()”.

- **MySQLi Tipe *Object Oriented***

Jika menggunakan cara penulisan ini, penggunaannya harus sesuai dengan aturan pemrograman berbasis objek (*object oriented programming*) agar bisa menghubungkan sistem dengan MySQL server.

Contohnya ketika Anda ingin menghubungkan sistem dengan MySQL server. Fungsi yang Anda tulis adalah “mysql_connect()” ketika menggunakan MySQL *extension*, sedangkan jika menggunakan MySQLi dengan penulisan berbasis objek menjadi “\$mysqli->query”.

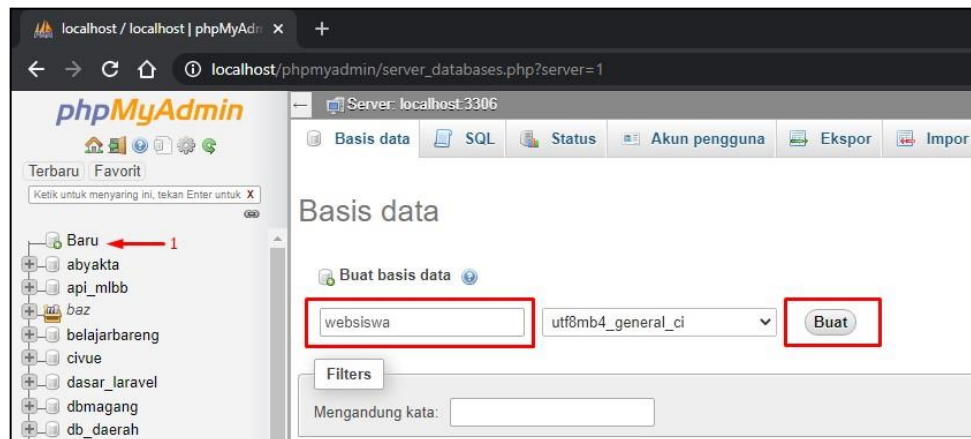
Kita dapat menggunakan salah satu cara ini untuk menuliskan MySQLi di dalam bahasa pemrograman yang dipakai.

PRAKTIKUM

Pada praktikum ini akan dijelaskan langkah-langkah menghubungkan PHP dan MySQL menggunakan API mysqli dengan metode prosedural :

1. Menyiapkan Database

- 1) Pertama, kita buat terlebih dahulu database-nya. Disini diberikan contoh database bernama **websiswa**



Selain melalui GUI, kita dapat membuat database baru dengan menggunakan perintah:

```
CREATE DATABASE `websiswa`;
```

- 2) Selanjutnya buat tabel baru dalam database tersebut dengan nama **siswa**



Disini kita akan membuat sebuah tabel dengan nama siswa, yang didalamnya kita akan membuat sejumlah 5 kolom awal

- 3) Buat struktur tabel siswa seperti pada gambar dibawah

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Terilai	Indeks	A_1	Komentar
id	INT	3	Tidak ada			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
nama	VARCHAR	100	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
email	VARCHAR	100	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
telepon	VARCHAR	15	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	
kelamin	ENUM	'Laki-Laki','Perempuan'	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	

Keterangan:

Nama Kolom	Jenis	Panjang	Keterangan
------------	-------	---------	------------

id	INT	3	Primary Key, Auto Increment
nama	VARCHAR	100	-
email	VARCHAR	100	-
telepon	VARCHAR	15	-
kelamin	ENUM	['Laki-Laki', 'Perempuan']	-

- 4) Kemudian klik save untuk mulai proses pembuatan tabel
 - 5) Setelah tabel dibuat, kita coba isi dengan satu record (data)
 - 6) Isi record sesuai field nya, kemudian klik kirim/go untuk memulai operasi penambahan data
- Untuk menambahkan data baru, pada phpmyadmin klik menu **Tambahkan**, kemudian isi field baru dengan contoh sebagai berikut:



Kolom	Jenis	Fungsi	Tak Ternilai	Nilai
id	int(3)			
nama	varchar(100)			Andi
email	varchar(100)			andiandiandi@gmail.com
telepon	varchar(15)			089518555555
kelamin	enum	--		Laki-Laki

Kirim

Selain dengan cara seperti diatas, kita dapat menggunakan kode MySQL secara langsung sebagai berikut:

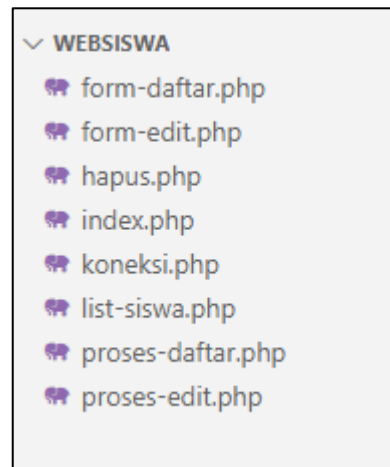
```
INSERT INTO `siswa` (`id`, `nama`, `email`, `telepon`, `kelamin`) VALUES (NULL, 'Andi', 'andiandiandi@gmail.com', '089518555555', 'Laki-Laki');
```

2. Membuat Proyek Baru

- 1) Untuk memulai membuat proyek baru, masuk kedalam direktori **htdocs** dan buat direktori baru

- 2) Pada direktori tersebut, buat file-file PHP yang dibutuhkan

Pada praktikum ini, daftar file yang dibutuhkan adalah sebagai berikut:



Keterangan:

- koneksi.php : sebagai tempat koneksi PHP dengan MySQL, terdapat konfigurasi database, user, dan password pengguna database
- index.php : halaman utama program
- list-siswa.php : halaman untuk menampilkan daftar siswa baru
- form-daftar.php : halaman untuk mengisi formulir pendaftaran
- proses-daftar.php : di dalamnya terdapat operasi untuk memproses pendaftaran siswa baru
- form-edit.php : halaman untuk melakukan edit data siswa
- proses-edit.php : di dalamnya terdapat operasi untuk memproses update data siswa
- hapus.php : di dalamnya terdapat operasi untuk proses menghapus data siswa

3. Koneksi ke Database MySQL

- 1) Buka file **koneksi.php**, kemudian isikan dengan kode berikut

```
<?php

$server = "localhost";
$user = "root";
$password = "";
$db_name = "isi_dengan_nama_database_kalian";

$conn = mysqli_connect($server, $user, $password, $db_name);

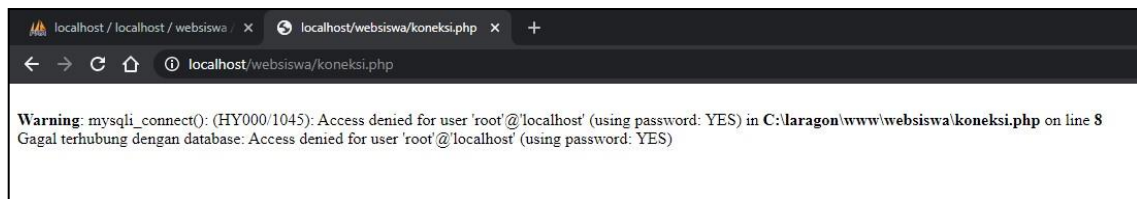
if (!$conn) {
    die("Gagal terhubung dengan database: " . mysqli_connect_error());
}
```

Untuk menghubungkan PHP dengan MySQL, digunakan fungsi **mysqli_connect()** dengan parameter alamat server, user, password, dan nama database. Isi password sesuai dengan yang dibuat di servernya. Jika tidak menggunakan password, maka dikosongkan saja.

Program akan menjalankan koneksi dari variabel **\$conn**, sehingga jika koneksi gagal maka **\$conn** akan bernilai **0 (false)**. Untuk itu diperlukan pengecekan terhadap variabel **\$conn** untuk melihat kondisi koneksi MySQL, jika **\$conn** bernilai 0 maka akan muncul pesan error.

```
if (!$conn) {  
    die("Gagal terhubung dengan database: " . mysqli_connect_error());  
}
```

Contoh tampilan jika koneksi yang dijalankan gagal adalah sebagai berikut:



4. Membuat Halaman Index

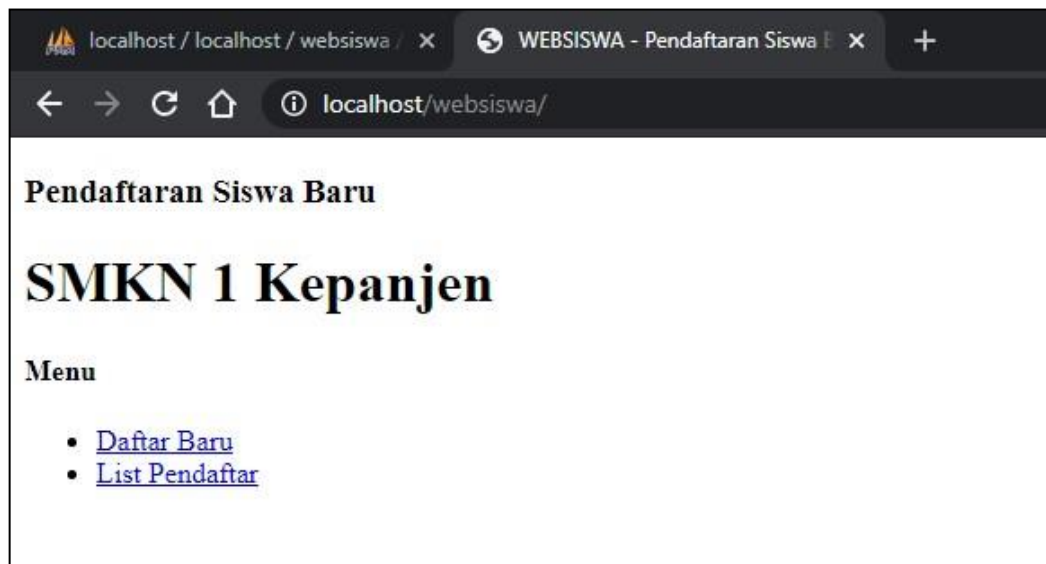
- 1) Buka file **index.php**, kemudian isikan dengan kode berikut

```
<!DOCTYPE html>  
<html lang="en">  
  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>WEBSISWA - Pendaftaran Siswa Baru</title>  
  </head>  
  
  <body>  
    <header>  
      <h3>Pendaftaran Siswa Baru</h3>  
      <h1>SMKN 1 Kepanjen</h1>  
    </header>  
  
    <h4>Menu</h4>  
  
    <nav>  
      <ul>  
        <li><a href="form-daftar.php">Daftar Baru</a></li>  
        <li><a href="list-siswa.php">List Pendaftar</a></li>  
      </ul>  
    </nav>
```

```
<h3>
    <?php if (isset($_GET['status'])) : ?>
        <?php if ($_GET['status'] == 'sukses') : ?>
            Pendaftaran Siswa Baru Berhasil!!!
        <?php else : ?>
            Pendaftaran Siswa Baru Gagal!!!
        <?php endif; ?>
    <?php endif; ?>
</h3>
</body>

</html>
```

2) Tampilan dari file index.php akan menjadi seperti berikut



5. Menampilkan Data Tabel MySQL

1) Buka file **list-siswa.php**, kemudian isikan dengan kode berikut

```
<?php include("koneksi.php"); ?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>WEBSISWA - List Pendaftar Baru</title>
</head>

<body>
```



```

<header>
    <h3>List Pendaftar Baru</h3>
</header>

<nav>
    <a href="form-daftar.php">[+] Daftarkan Siswa Baru</a>
</nav>

<br>

<table border="1">
    <thead>
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Email</th>
            <th>Telepon</th>
            <th>Jenis Kelamin</th>
            <th>Tindakan</th>
        </tr>
    </thead>
    <tbody>

        <?php
        $sql = "SELECT * FROM nama_tabel_kalian";
        $query = mysqli_query($conn, $sql);
        $i = 1;
        ?>

        <?php while ($siswa = mysqli_fetch_array($query)) : ?>
            <tr>
                <td><?php echo $i ?></td>
                <td><?php echo $siswa['nama'] ?></td>
                <td><?php echo $siswa['email'] ?></td>
                <td><?php echo $siswa['telepon'] ?></td>
                <td><?php echo $siswa['kelamin'] ?></td>
                <td>
                    <a href="form-edit.php?id=<?php echo $siswa['id']
?>">Edit</a> |
                    <a href="hapus.php?id=<?php echo $siswa['id']
?>">Hapus</a>
                </td>
            </tr>
            <?php $i++ ?>
        <?php endwhile; ?>
    </tbody>
</table>

```

```

        </tbody>

    </table>

    <p>Total Pendaftar Baru: <?php echo mysqli_num_rows($query) ?></p>
</body>

</html>

```

Pertama kita membutuhkan koneksi ke database, maka kita harus mengimpor file koneksi.php agar variabel \$conn dapat dibaca.

```
<?php include("koneksi.php"); ?>
```

Selanjutnya, kita melakukan query untuk mengambil data siswa pada tabel siswa.

```

<?php
    $sql = "SELECT * FROM nama_tabel_kalian";
    $query = mysqli_query($conn, $sql);
    $i = 1; // iterator untuk nomor urut pada tabel
    ?>

```

Hasil dari query tersebut, kemudian kita ubah menjadi array dengan fungsi **mysqli_fetch_array()** dan hasilnya disimpan dalam variabel \$siswa.

Perulangan while akan terus berulang selama masih ada isi dalam tabel. Kemudian, selama perulangan, data hasil kueri akan ditampilkan.

```

<?php while ($siswa = mysqli_fetch_array($query)) : ?>
    <tr>
        <td><?php echo $i ?></td>
        <td><?php echo $siswa['nama'] ?></td>
        <td><?php echo $siswa['email'] ?></td>
        <td><?php echo $siswa['telepon'] ?></td>
        <td><?php echo $siswa['kelamin'] ?></td>
        <td>
            <a href="form-edit.php?id=<?php echo $siswa['id'] ?>">Edit</a> |
            <a href="hapus.php?id=<?php echo $siswa['id'] ?>">Hapus</a>
        </td>
    </tr>
    <?php $i++ ?>
<?php endwhile; ?>

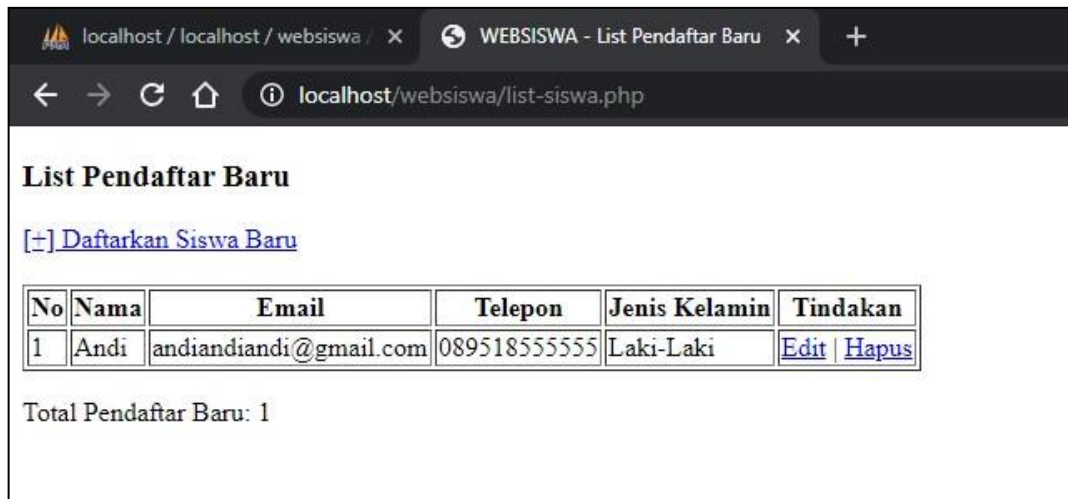
```

Perhatikan array \$siswa, indeksnya sesuai dengan nama kolom pada database dan bersifat *case sensitive* (\$siswa['nama'], \$siswa['email'], dst.).

Terakhir, kita menampilkan total data yang ada di dalam tabel dengan fungsi **mysqli_num_rows()**.

```
<p>Total Pendaftar Baru: <?php echo mysqli_num_rows($query) ?></p>
```

2) Tampilan dari file list-siswa.php akan menjadi seperti berikut



6. Menambahkan Data ke Tabel MySQL

Terdapat 2 file yang dibutuhkan untuk menambahkan data baru kedalam Database MySQL, yaitu **form-daftar.php** dan **proses-daftar.php**

Kenapa dibuat dalam 2 file terpisah? Penerapan file yang terpisah tersebut bertujuan untuk membedakan bagian view sebagai tampilan program dan bagian controller untuk mengelola proses yang terjadi didalamnya.

1) Buka file **form-daftar.php**, kemudian isikan dengan kode berikut

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>WEBSISWA - Tambah Pendaftar Baru</title>
</head>

<body>
    <header>
```

```

<h3>Formulir Pendaftaran Siswa Baru</h3>
</header>

<form action="proses-daftar.php" method="POST">

    <fieldset>

        <p>
            <label for="nama">Nama: </label>
            <input type="text" name="nama" placeholder="Nama Lengkap" />
        </p>
        <p>
            <label for="nama">Email: </label>
            <input type="text" name="email" placeholder="Alamat Email" />
        </p>
        <p>
            <label for="nama">Telepon: </label>
            <input type="text" name="telepon" placeholder="Nomor Telepon"
/>
        </p>
        <p>
            <label for="kelamin">Jenis Kelamin: </label>
            <label><input type="radio" name="kelamin" value="Laki-Laki">
Laki-laki</label>
            <label><input type="radio" name="kelamin" value="Perempuan">
Perempuan</label>
        </p>
        <p>
            <input type="submit" value="Daftar" name="daftar" />
        </p>

    </fieldset>

</form>
</body>

</html>

```

Perhatikan pada tag pembuka **<form>**, di sana kita menggunakan atribut **action** dan **method**.

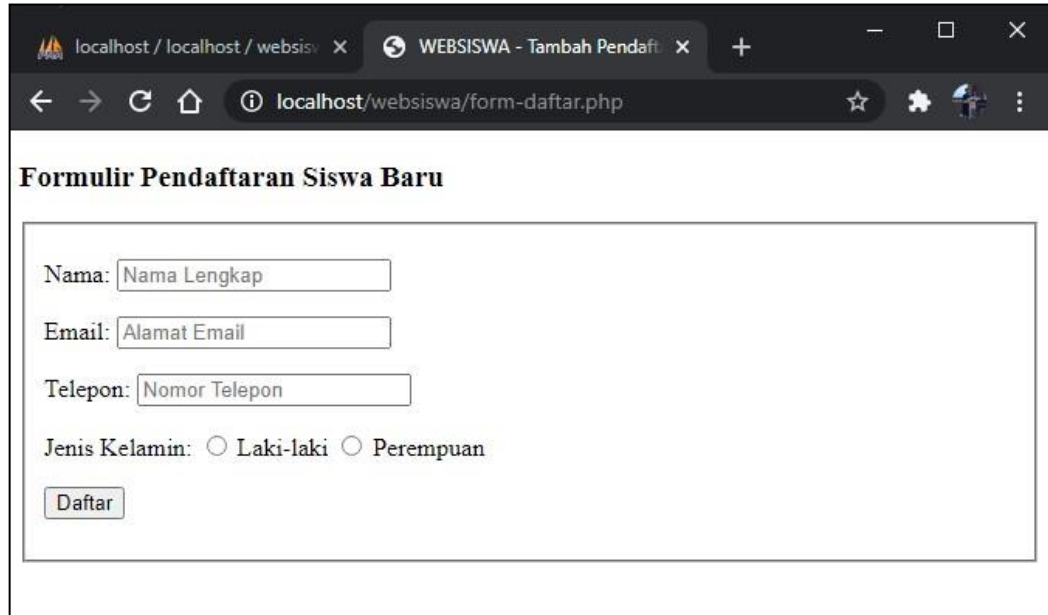
```

...
<form action="proses-daftar.php" method="POST">
...

```

Atribut **action** digunakan untuk menentukan kemana form tersebut akan diproses. Sedangkan atribut **method** untuk menentukan metode pengiriman datanya.

Perhatikan juga pada setiap field input-nya, disana ada atribut **name**. Atribut ini untuk menentukan kunci indeks dari array **\$_POST** dan/atau **\$_GET**.



2) Kemudian buka file **proses-daftar.php**, kemudian isikan dengan kode berikut

```
<?php

include("koneksi.php");

// cek apakah tombol daftar sudah diklik atau blum?
if (isset($_POST['daftar'])) {

    // ambil data dari formulir
    $nama = htmlspecialchars($_POST['nama']);
    $email = htmlspecialchars($_POST['email']);
    $telepon = htmlspecialchars($_POST['telepon']);
    $kelamin = htmlspecialchars($_POST['kelamin']);

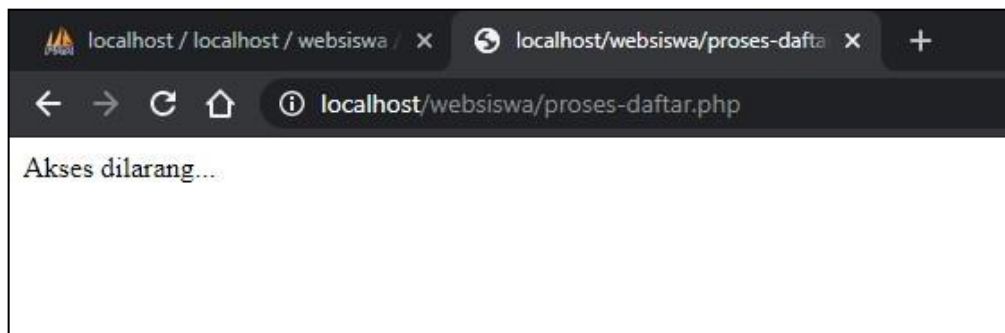
    // buat query
    $sql = "INSERT INTO nama_tabel_kalian (nama, email, telepon, kelamin)
    VALUE ('$nama', '$email', '$telepon', '$kelamin')";
    $query = mysqli_query($conn, $sql);

    // apakah query simpan berhasil?
    if ($query) {
        // kalau berhasil alihkan ke halaman index.php dengan status=sukses
        header('Location: index.php?status=sukses');
    } else {
```

```
// kalau gagal alihkan ke halaman index.php dengan status=gagal
header('Location: index.php?status=gagal');

}
} else {
    die("Akses dilarang...");
}
```

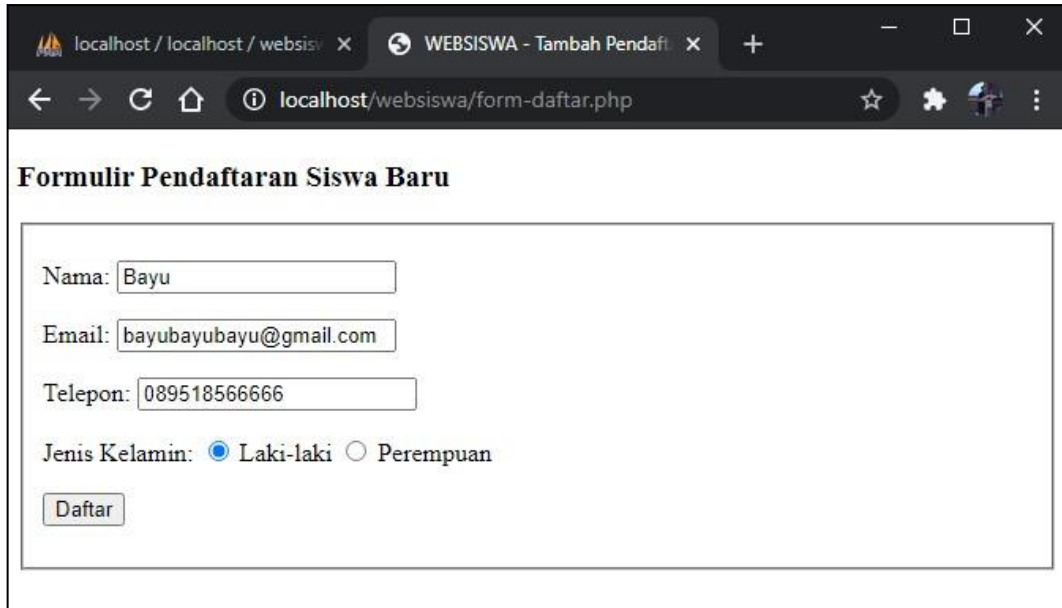
Pada kode diatas sudah dilakukan pengecekan apakah sebelumnya tombol daftar sudah di klik atau belum, jika belum di klik maka nilai `$_POST['daftar']` menjadi 0, dan akan menampilkan pesan error



- 3) Untuk melakukan pemeriksaan apakah data tersebut berhasil disimpan dalam database atau tidak, program akan memeriksa nilai `$_GET['status']` hasil dari proses pendaftaran. Pada halaman **index.php** tambahkan kode berikut dibawah tag `</nav>`

```
<h3>
    <?php if (isset($_GET['status'])) : ?>
        <?php if ($_GET['status'] == 'sukses') : ?>
            Pendaftaran Siswa Baru Berhasil!!
        <?php else : ?>
            Pendaftaran Siswa Baru Gagal!!
        <?php endif; ?>
    <?php endif; ?>
</h3>
```

- 4) Untuk mencoba operasi penambahan data yang sudah dibuat, buka halaman web form-daftar.php untuk mencoba menambahkan siswa baru



Formulir Pendaftaran Siswa Baru

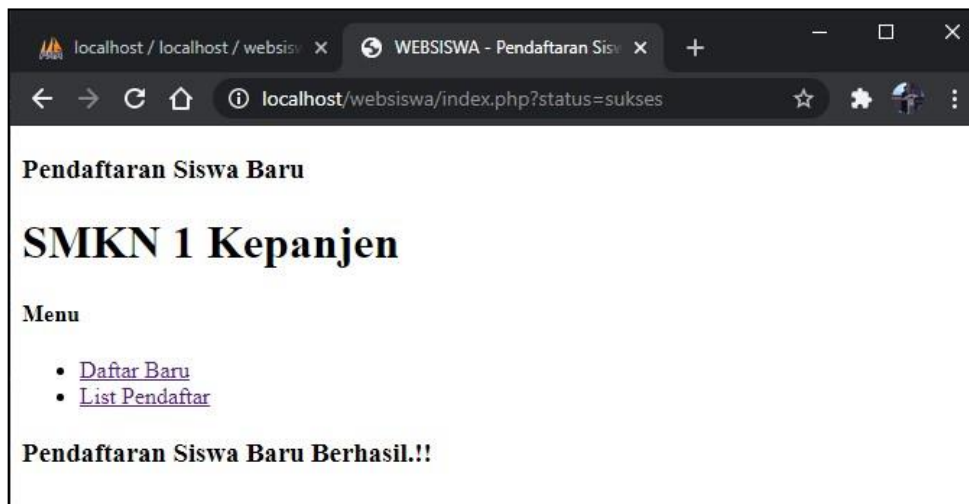
Nama:

Email:

Telepon:

Jenis Kelamin: ☒ Laki-laki ☐ Perempuan

- 5) Jika data baru berhasil disimpan ke dalam database, maka pada website akan kembali ke halaman index.php dan akan menampilkan pesan berhasil



Pendaftaran Siswa Baru

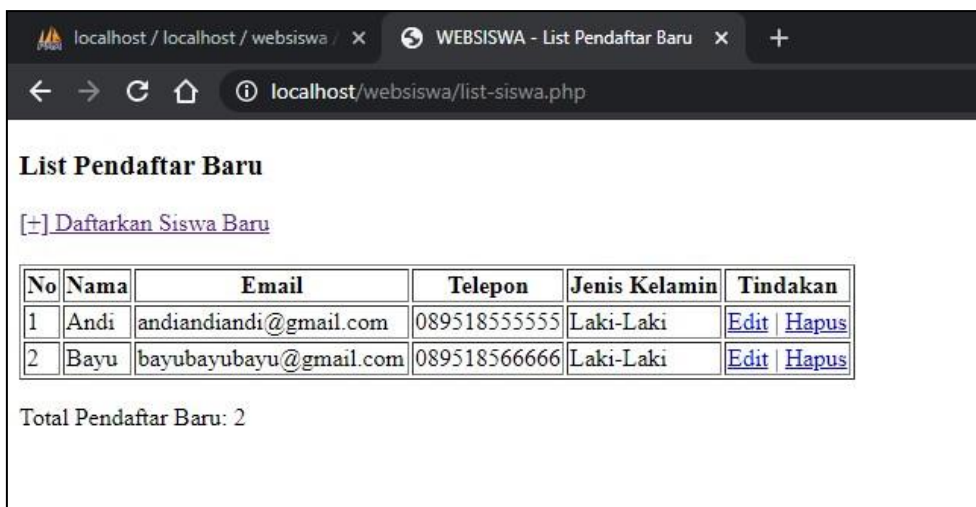
SMKN 1 Kepanjen

Menu

- [Daftar Baru](#)
- [List Pendaftaran](#)

Pendaftaran Siswa Baru Berhasil!!

- 6) Halaman **list-siswa.php** setelah penambahan data baru



List Pendaftar Baru

[\[+\] Daftarkan Siswa Baru](#)

No	Nama	Email	Telepon	Jenis Kelamin	Tindakan
1	Andi	andiandiandi@gmail.com	089518555555	Laki-Laki	Edit Hapus
2	Bayu	bayubayubayu@gmail.com	089518566666	Laki-Laki	Edit Hapus

Total Pendaftar Baru: 2

7. Melakukan Update Data

Setelah kita melakukan proses penambahan data, adakalanya pada data yang sudah dimasukkan tersebut terdapat kesalahan. Dalam penerapannya pengguna aplikasi web dinamis perlu untuk mengubah konten yang ada dalam database.

Pada project yang sudah dibuat sebelumnya, pada bagian list-siswa.php terdapat baris sebagai berikut:

```
<a href="form-edit.php?id=<?php echo $siswa['id'] ?>">Edit</a>
```

Ketika tombol “Edit” tersebut di klik, maka program akan mengarahkan pengguna ke halaman form-edit.php, dengan memberikan parameter berupa id siswa yang dituliskan dengan sintaks \$siswa['id'].

1. Buat file baru dalam project dengan nama **form-edit.php** dan **proses-edit.php**.
2. Buka file form-edit.php dan isi dengan kode berikut:

```
<?php

include("koneksi.php");

// kalau tidak ada id di query string
if (!isset($_GET['id'])) {
    header('Location: list-siswa.php');
}

//ambil id dari query string
$id = $_GET['id'];

// buat query untuk ambil data dari database
$sql = "SELECT * FROM nama_tabel_kalian WHERE id=$id";
$query = mysqli_query($conn, $sql);
$siswa = mysqli_fetch_assoc($query);

// jika data yang di-edit tidak ditemukan
if (mysqli_num_rows($query) < 1) {
    die("data tidak ditemukan...");
}

?>

<!DOCTYPE html>
<html lang="en">
```



```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>WEBSISWA - Edit Pendaftar</title>
</head>

<body>
    <header>
        <h3>Formulir Edit Siswa</h3>
    </header>

    <form action="proses-edit.php" method="POST">

        <fieldset>

            <input type="hidden" name="id" value="<?php echo $siswa['id'] ?>"
/>

            <p>
                <label for="nama">Nama: </label>
                <input type="text" name="nama" placeholder="Nama lengkap"
value="<?php echo $siswa['nama'] ?>" />
            </p>
            <p>
                <label for="email">Email: </label>
                <input type="email" name="email" placeholder="Email"
value="<?php echo $siswa['email'] ?>" />
            </p>
            <p>
                <label for="telepon">Telepon: </label>
                <input type="telepon" name="telepon" placeholder="Nomor
Telepon" value="<?php echo $siswa['telepon'] ?>" />
            </p>
            <p>
                <label for="kelamin">Jenis Kelamin: </label>
                <?php $jk = $siswa['kelamin']; ?>
                <label><input type="radio" name="kelamin" value="Laki-Laki"
<?php echo ($jk == 'Laki-Laki') ? "checked" : "" ?>> Laki-laki</label>
                <label><input type="radio" name="kelamin" value="Perempuan"
<?php echo ($jk == 'Perempuan') ? "checked" : "" ?>> Perempuan</label>
            </p>
            <p>
                <input type="submit" value="Simpan" name="simpan" />
            </p>

```

```

        </fieldset>

    </form>

</body>

</html>

```

Ada beberapa hal yang perlu diperhatikan dalam kode tersebut.

Pertama kita mengubah data dari hasil query menjadi array dengan menggunakan fungsi `mysqli_fetch_assoc()`.

Kenapa menggunakan fungsi tersebut?

Karena data yang akan kita ambil adalah satu baris dan agar indeks array-nya menggunakan nama kolom.

Contoh hasilnya seperti ini:



```

Array
(
    [id] => 100
    [nama] => "Candra"
    [email] => "Candra@gmail.com"
    [telepon] => "082229920011"
    [kelamin] => "Laki-Laki"
)

```

Kemudian data tersebut kita tampilkan untuk menjadi nilai default pada form.

Jangan lupa untuk membuat input id dengan tipe hidden untuk menyimpan id data yang akan di-update.

```

<input type="hidden" name="id" value="<?php echo $siswa['id'] ?>" />

```

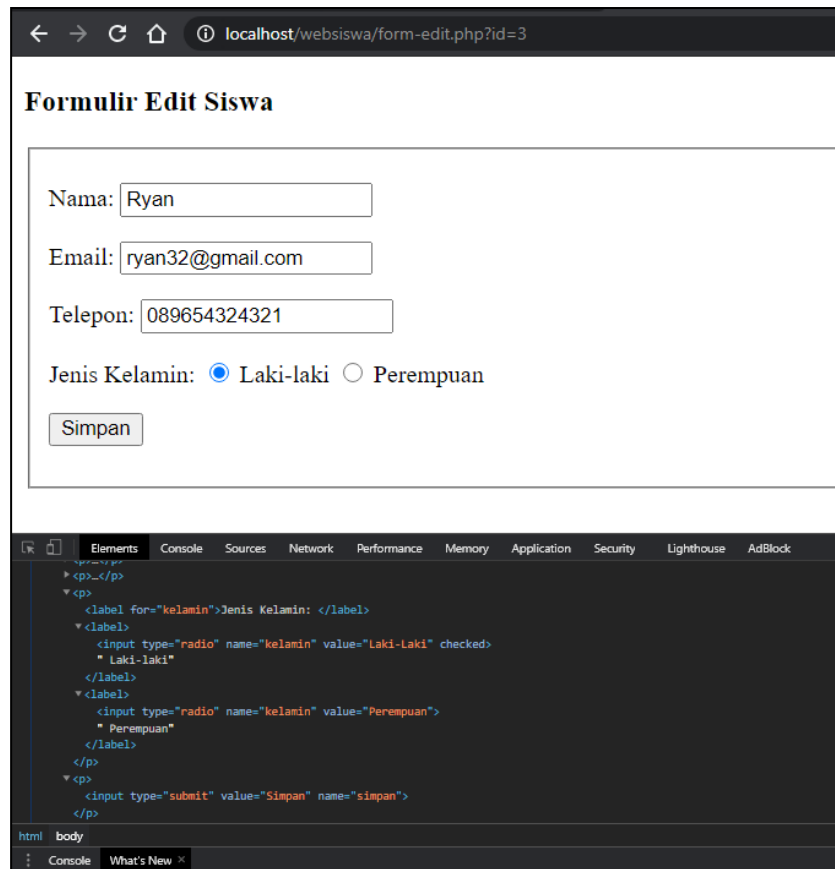
Untuk input radio seperti pada field jenis kelamin, kita menggunakan operator ternary agar bisa membuat atribut *checked*.

```

<p>
    <label for="kelamin">Jenis Kelamin: </label>
    <?php $jk = $siswa['kelamin']; ?>
    <label><input type="radio" name="kelamin" value="Laki-Laki" <?php echo ($jk
== 'Laki-Laki') ? "checked" : "" ?>> Laki-laki</label>
    <label><input type="radio" name="kelamin" value="Perempuan" <?php echo ($jk
== 'Perempuan') ? "checked" : "" ?>> Perempuan</label>
</p>

```

Ketika kita lihat dari *inspect elements*, maka akan tampil seperti ini:



Jika pengguna selesai melakukan proses edit dan klik pada tombol simpan, maka program akan mengalihkan ke **proses-edit.php**.

3. Isi file **proses-edit.php** dengan kode berikut:

```
<?php

include("koneksi.php");

// cek apakah tombol simpan sudah diklik atau blum?
if (isset($_POST['simpan'])) {

    // ambil data dari formulir
    $id = $_POST['id'];
    $nama = $_POST['nama'];
    $email = $_POST['email'];
    $telepon = $_POST['telepon'];
    $jk = $_POST['kelamin'];

    // buat query update
    $sql = "UPDATE nama_tabel_kalian SET nama='$nama', email='$email',
telepon='$telepon', kelamin='$jk' WHERE id=$id";
```

```
$query = mysqli_query($conn, $sql);

// apakah query update berhasil?
if ($query) {
    // kalau berhasil alihkan ke halaman list-siswa.php
    header('Location: list-siswa.php');
} else {
    // kalau gagal tampilkan pesan
    die("Gagal menyimpan perubahan...");
}
} else {
    die("Akses dilarang...");
}
```

4. Simpan file tersebut, kemudian coba untuk melakukan proses edit data.

Disini terdapat 2 data, akan dilakukan perubahan data siswa dengan nama Ryan:

List Pendaftar Baru

[\[+\] Daftarkan Siswa Baru](#)

No	Nama	Email	Telepon	Jenis Kelamin	Tindakan
1	Andi	andiandiandi@gmail.com	089518555557	Laki-Laki	Edit Hapus
2	Ryan	ryan32@gmail.com	089654324321	Laki-Laki	Edit Hapus

Total Pendaftar Baru: 2

Setelah di klik tombol edit, maka akan dialihkan ke form edit data. Perhatikan pada bagian URL, disana tertulis id siswa yaitu **3**, karena id bersifat unik maka id digunakan sebagai kunci utama untuk mengubah data. Disini akan dilakukan perubahan dari nama “Ryan” menjadi “Ryanti”, dan mengubah Jenis Kelamin dari “Laki-Laki” menjadi “Perempuan”:

← → ↻ ⬆️ ⓘ localhost/websiswa/form-edit.php?id=3

Formulir Edit Siswa

Nama:

Email:

Telepon:

Jenis Kelamin: ☐ Laki-laki ☒ Perempuan

Jika proses update data berjalan dengan baik, maka data list siswa yang baru akan berubah menjadi seperti berikut:

List Pendaftar Baru

[\[+\] Daftarkan Siswa Baru](#)

No	Nama	Email	Telepon	Jenis Kelamin	Tindakan
1	Andi	andiandiandi@gmail.com	089518555557	Laki-Laki	Edit Hapus
2	Ryanti	ryan32@gmail.com	089654324321	Perempuan	Edit Hapus

Total Pendaftar Baru: 2

8. Melakukan Delete Data

Kita membutuhkan id untuk menentukan data yang akan dihapus. Karena itu, kita mengirimkan id melalui query string.

Contoh:

```
http://localhost/websiswa/hapus.php?id=1
```

Query string-nya adalah ?id=1. Query string akan tersimpan dalam variabel \$_GET.

Untuk lebih jelasnya, silahkan buka file hapus.php dan isi dengan kode berikut:

```
<?php

include("koneksi.php");

if (isset($_GET['id'])) {

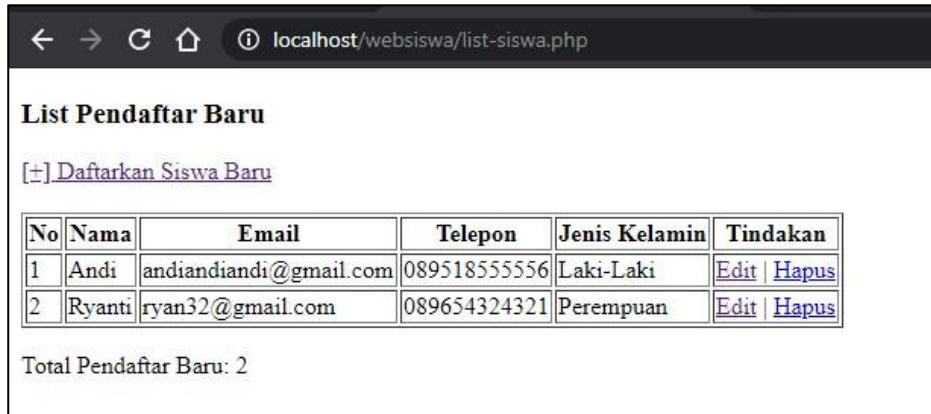
    // ambil id dari query string
    $id = $_GET['id'];

    // buat query hapus
    $sql = "DELETE FROM nama_tabel_kalian WHERE id=$id";
    $query = mysqli_query($conn, $sql);

    // apakah query hapus berhasil?
    if ($query) {
        header('Location: list-siswa.php');
    } else {
        die("gagal menghapus...");
    }
} else {
    die("akses dilarang...");
}
```

Sekarang coba klik link hapus, maka datanya akan langsung terhapus.

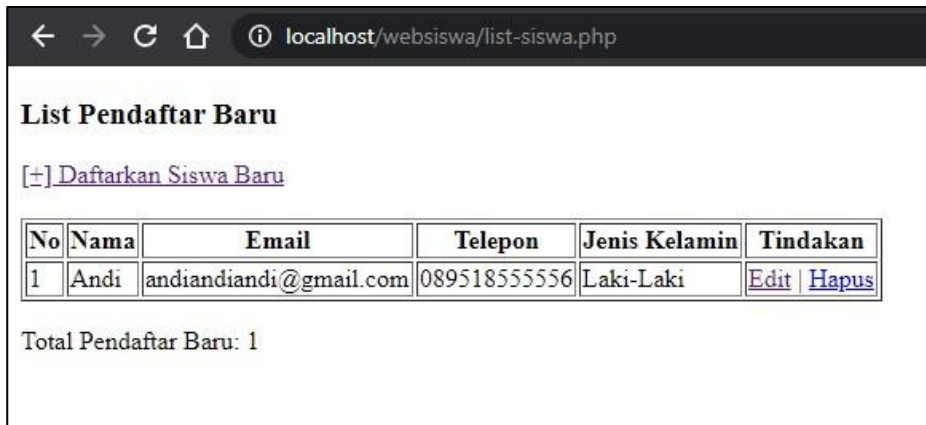
Data sebelum dihapus:



No	Nama	Email	Telepon	Jenis Kelamin	Tindakan
1	Andi	andiandiandi@gmail.com	089518555556	Laki-Laki	Edit Hapus
2	Ryanti	ryan32@gmail.com	089654324321	Perempuan	Edit Hapus

Total Pendaftar Baru: 2

Data setelah dihapus:



No	Nama	Email	Telepon	Jenis Kelamin	Tindakan
1	Andi	andiandiandi@gmail.com	089518555556	Laki-Laki	Edit Hapus

Total Pendaftar Baru: 1

Sebenarnya cara ini kurang aman, karena kita tidak memberikan dialog konfirmasi kepada pengguna. Misalnya seperti menanyakan ‘Apakah yakin akan menghapus?’