

PRAKTIKUM 8

Materi:

1. Menghubungkan PostgreSQL dengan PHP
2. Query ke database postgresQL dari PHP
3. Membuat aplikasi website berbasis PHP dan database postgresQL

Tujuan Praktikum :

Mahasiswa mampu membuat aplikasi website berbasis PHP yang terkoneksi dengan database

A. Penyajian dan Latihan

Untuk memulai membuat aplikasi website, software yang dibutuhkan antara lain :

- XAMPP sebagai server PHP
- DBMS (dalam contoh ini adalah PostgreSQL)
- Text Editor

Menyambungkan PHP dengan PostgreSQL

1. Pastikan komputer anda sudah terinstall xampp untuk menjalankan script php
2. Pada folder instalasi xampp, buka pada path xampp/php/php.ini. Edit file php.ini tersebut untuk menghilangkan tanda ; (tanda comment) pada baris berikut
`;extension=pgsql`
3. Restart servis apache pada xampp

Pada contoh ini akan dibuat suatu program PHP yang tersambung ke database untuk aplikasi Pendaftaran Siswa

Langkah-langkah :

1. Buat Database baru pada postgres dengan nama 'sekolah'

```
CREATE DATABASE sekolah;
```

```
\c sekolah;
```

2. Buat table baru pada database sekolah dengan script berikut

```
CREATE TABLE calonsiswa(  
id serial,  
nama VARCHAR(64) NOT NULL,  
alamat VARCHAR(20) NOT NULL,  
jenis_kelamin VARCHAR(16) NOT NULL,  
sekolah_asal VARCHAR(64) NOT NULL,  
CONSTRAINT calonsiswa_id_PK PRIMARY KEY(id));
```

3. Buat folder baru pada XAMPP/htdocs dengan nama 'LKP8'
4. Download file-file program pada link berikut <https://bit.ly/2F5njD1> dan letakkan pada folder LKP8
5. Buka Text Editor(mis.Atom) dan buka folder LKP8

6. Lihat pada file config.php, file ini berisi kode php untuk melakukan koneksi dengan database 'sekolah' pada PostgreSQL dengan username dan password yang diberikan. Pastikan nama database, username, dan password pada script sudah sesuai dengan settingan di postgres anda.

```
<?php
$db=pg_connect('host=localhost dbname=sekolah user=postgres password=postgres');
if( !$db ){
    die("Gagal terhubung dengan database: " . pg_connect_error());
}
?>
```

7. Lihat pada file formdaftar.php, file ini berisi script html berupa form. Form ini dilengkapi dengan tombol submit yang jika diklik maka akan mengirimkan data isian form untuk diproses di file prosespendaftaran.php. Pengiriman data dilakukan dengan metode POST sehingga data yang dikirim tidak terlihat pada url. Metode pengiriman data selain dengan POST yaitu dengan GET dimana data yang dikirimkan akan terlihat pada url.

```
<form action="prosespendaftaran.php" method="POST">
  <fieldset>
    <p>
      <label for="nama">Nama: </label>
      <input type="text" name="nama" placeholder="nama lengkap" />
    </p>
    <p>
      <label for="alamat">Alamat: </label>
      <textarea name="alamat"></textarea>
    </p>
    <p>
      <label for="jenis_kelamin">Jenis Kelamin: </label>
      <label><input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki</label>
      <label><input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan</label>
    </p>
    <p>
      <label for="sekolah_asal">Sekolah Asal: </label>
      <input type="text" name="sekolah_asal" placeholder="nama sekolah" />
    </p>
    <p>
      <input type="submit" value="Daftar" name="daftar" />
    </p>
  </fieldset>
</form>
```

8. Sebagaimana didefinisikan pada form di formdaftar.php, prosespendaftaran.php ini akan memproses data form yang dikirimkan user (setelah mengklik tombol submit) dari form. Data tersebut selanjutnya akan dimasukkan ke table di database sebagaimana yang ditunjukkan pada script berikut :

```

include("config.php");
// cek apakah tombol daftar sudah diklik atau blum?
if(isset($_POST['daftar'])){

    // ambil data dari formulir
    $nama = $_POST['nama'];
    $alamat = $_POST['alamat'];
    $jk = $_POST['jenis_kelamin'];
    $sekolah = $_POST['sekolah_asal'];

    // buat query
    $query = pg_query("INSERT INTO calonsiswa (nama, alamat, jenis_kelamin, sekolah_asal) VALUES ('$nama', '$alamat', '$jk', '$sekolah')");

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

```

9. Pastikan servis Apache sudah di-start dan jalankan aplikasi dengan membuka browser dan akses alamat <http://location/LKP8>

Menambahkan Fasilitas Delete

1. Pertama, tambahkan tombol delete pada file daftarsiswa.php

```

while($siswa = pg_fetch_array($query)){
    echo "<tr>";

    echo "<td>".$siswa['id']. "</td>";
    echo "<td>".$siswa['nama']. "</td>";
    echo "<td>".$siswa['alamat']. "</td>";
    echo "<td>".$siswa['jenis_kelamin']. "</td>";
    echo "<td>".$siswa['sekolah_asal']. "</td>";

    echo "<td>";
    echo "<a href='hapus.php?id=".$siswa['id']."'>Hapus</a>";
    echo "</td>";

    echo "</tr>";
}

```

2. Setelah itu buat file baru dengan nama hapus.php, yang berisi query menghapus data dengan id tertentu. Ketikkan script berikut lalu simpan.

```

<?php
include("config.php");
if( isset($_GET['id']) ){

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

    // buat query hapus
    $query = pg_query("DELETE FROM calonsiswa WHERE id=$id");

    // apakah query hapus berhasil?
    if( $query ){
        header('Location: daftarsiswa.php');
    } else {
        die("gagal menghapus...");
    }

} else {
    die("akses dilarang...");
}
?>

```

Pada contoh di atas, pengiriman parameter id yang akan dihapus dilakukan dengan method **GET**.

LEMBAR KERJA PRAKTIKUM 8

Nama:	Tanggal Praktikum:
NRP :	Waktu Praktikum:
Nilai :	Nama Asisten :

Lengkapi program di atas sehingga bisa digunakan untuk mengedit data.

- Tambahkan tombol 'edit' daftarsiswa.php

Tambahkan 2 file berikut :

- formedit.php : menampilkan form untuk mengedit data
- prosesedit.php : memproses data pada form untuk dilakukan perintah update table pada database