

Nama : Putri Naftali Manurung

NIM : 121140027

Kelas : RA

```
<!-- Pesan status penambahan data -->
<div class="message">
    <?php
        $host = "localhost";
        $username = "root";
        $password = "";
        $database = "data_mahasiswa";
        $koneksi = new mysqli($host, $username, $password, $database);

        if ($koneksi->connect_error) {
            die("Koneksi database gagal: " . $koneksi->connect_error);
        }

        if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['tambah'])) {
            $nim = $_POST['nim'];
            $nama = $_POST['nama'];
            $prodi = $_POST['prodi'];

            $query_tambah = "INSERT INTO mahasiswa (nim, nama, prodi) VALUES ('$nim', '$nama', '$prodi')
            ON DUPLICATE KEY UPDATE nama = '$nama', prodi = '$prodi'";

            if ($koneksi->query($query_tambah) === TRUE) {
                echo "Data mahasiswa berhasil ditambahkan.";
                echo "<script>setTimeout(function(){document.querySelector('.message').style.display = 'none';}, 3000);</script>";
            } else {
                echo "Error: " . $query_tambah . "<br>" . $koneksi->error;
            }
        }
    ?>
</div>
```

Nama : Putri Nafati Manung
NIM : 121140027
Kelas : RA

- `'$host', '$username', '$password', dan '$database'`
⇒ digunakan untuk koneksi ke database MySQL.
- `'new mysqli()'`
⇒ digunakan untuk membuat objek koneksi baru ke database menggunakan informasi yang telah didefinisikan sebelumnya.
- `if ($koneksi -> connect_error) {
die("Koneksi database gagal : ". $koneksi -> connect_error); }`
⇒ memeriksa apakah terjadi kesalahan saat menghubungkan ke database.
Jika terjadi kesalahan, pesan error akan ditampilkan dan skrip akan dihentikan menggunakan fungsi `'die()'`.
- `'$_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['tambah'])'`
⇒ memeriksa apakah permintaan yang diterima adalah metode POST dan apakah terdapat variabel POST dengan nama 'tambah'. Ini digunakan untuk menangani form submission.
- `'$nim', '$nama', '$prodi'`
⇒ Variabel yang diambil dari input form dengan method POST.
- `'$query-tambah'`
⇒ Query SQL yang akan dieksekusi untuk menambahkan data mahasiswa ke database.
Penggunaan `'ON DUPLICATE KEY UPDATE'` berarti jika ada duplikat kunci (biasanya kunci unik seperti primary key), maka data akan di update.
- `'$koneksi->query($query-tambah)'`
⇒ Melakukan eksekusi query ke database.
- `<script> setTimeout(function () { document.querySelector('.message').style.display = 'none'; }, 3000); </script>`
⇒ digunakan untuk menyembunyikan pesan setelah 3 detik dengan mengubah style dari elemen yang memiliki kelas 'message' menjadi `'display: none'`.

```
<!-- Tabel untuk menampilkan data mahasiswa -->
<?php
$query_tampilkan = "SELECT * FROM mahasiswa";
$result = $koneksi->query($query_tampilkan);

if ($result->num_rows > 0) {
    echo "<table border='1'>";
    echo "<tr><th>NIM</th><th>Nama</th><th>Program Studi</th><th>Aksi</th></tr>";
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>".$row['nim']. "</td>";
        echo "<td>".$row['nama']. "</td>";
        echo "<td>".$row['prodi']. "</td>";
        echo "<td><a href='edit.php?nim=".$row['nim']."'>Edit</a> | <a href='hapus.php?nim=".$row['nim']."'>Hapus</a></td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "<p>Tidak ada data mahasiswa.</p>";
}
?>
```

- `<?php>`
⇒ Tanda awal dan akhir dari blok kode PHP.
- `$query_tampilkan = "SELECT * FROM mahasiswa";`
⇒ Mendefinisikan sebuah variabel `$query_tampilkan` yang berisi perintah SQL untuk mengambil semua data dari table 'mahasiswa'.
- `$result = $koneksi->query($query_tampilkan);`
⇒ menjalankan query yang telah didefinisikan sebelumnya menggunakan objek `$koneksi`. Query dieksekusi dan hasilnya disimpan dalam variabel `$result`.
- `if ($result->num_rows) > 0 { ... } else { ... }`
⇒ Memeriksa apakah hasil query mengembalikan setidaknya satu baris data. Jika hasilnya lebih dari 0, maka blok kode dalam kurung kurawal pertama akan dieksekusi.
- `echo "table border = '1'>"`
⇒ Menampilkan awal tag table HTML dengan border set ke 1 untuk menampilkan tabel data mahasiswa.
- `echo "<tr><th>NIM</th><th>Nama</th><th>Program Studi</th><th>Aksi</th></tr>"`
⇒ Menampilkan baris judul table dengan header kolom NIM, Nama, Program Studi, dan Aksi.
- `while ($row = $result->fetch_assoc()) { ... }`
⇒ loop while yang akan berjalan selama ada baris data yang diambil dari hasil query. Setiap baris data dimasukkan ke dalam array asosiatif `$row`.
- `echo "<tr>";`
⇒ Mencetak awal baris dalam tabel
- `echo "<td>". $row['nim']. "</td>";`
⇒ Mencetak isi kolom NIM dari baris saat ini di dalam loop.
- `echo "<td>". $row['nama']. "</td>";`
⇒ Mencetak isi kolom Nama dari baris saat ini di dalam loop
- `echo "<td>". $row['prodi']. "</td>";`
⇒ Mencetak isi kolom Program Studi dari baris saat ini di dalam loop
- `echo "<td>Edit | Hapus</td>";`
⇒ Mencetak kolom Aksi dalam bentuk Hyperlink untuk mengedit dan menghapus data. Hyperlink ini akan mengarah ke halaman 'edit.php' atau 'hapus.php' dengan parameter nim sebagai identifikasi data yang akan diubah atau dihapus

```

<!-- Pencarian Mahasiswa -->
<h2>Cari Mahasiswa berdasarkan Program Studi</h2>
<form action="" method="get">
    <select name="prodi">
        <option value="">Pilih Program Studi</option>
        <?php
            $query_program_studi = "SELECT DISTINCT prodi FROM mahasiswa";
            $result_program_studi = $koneksi->query($query_program_studi);

            while ($row = $result_program_studi->fetch_assoc()) {
                echo "<option value='" . $row['prodi'] . "'>" . $row['prodi'] . "</option>";
            }
        ?>
    </select>
    <button type="submit" name="cari">Cari</button>
</form>

<!-- Pencarian data mahasiswa berdasarkan program studi yang dipilih -->
<?php
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['cari']) && isset($_GET['prodi'])) {
    $prodi = $koneksi->real_escape_string($_GET['prodi']);

    $query_cari = "SELECT * FROM mahasiswa WHERE prodi = '$prodi'";
    $result_cari = $koneksi->query($query_cari);

    if ($result_cari->num_rows > 0) {
        echo "<h2>Hasil Pencarian</h2>";
        echo "<table border='1'>";
        echo "<tr><th>NIM</th><th>Nama</th><th>Program Studi</th></tr>";
        while ($row = $result_cari->fetch_assoc()) {
            echo "<tr>";
            echo "<td>".$row['nim'].</td>";
            echo "<td>".$row['nama'].</td>";
            echo "<td>".$row['prodi'].</td>";
            echo "</tr>";
        }
        echo "</table>";
    } else {
        echo "<p>Tidak ditemukan data mahasiswa pada program studi '$prodi'.</p>";
    }
}
}

```


- `<form action="" method="get">`
- ⇒ Membuka sebuah form HTML dengan metode GET yang mengarah ke halaman itu sendiri saat formulir telah disubmit.
- `<select name="prodi">...</select>`
- ⇒ Membuat dropdown/select box yang memungkinkan pengguna memilih program studi. Opsi-opsi program studi diisi melalui query ke database.
- `$query-program-studi = "SELECT DISTINCT prodi FROM Mahasiswa"`
- ⇒ Query untuk mengambil daftar program studi yang unik dari tabel mahasiswa.
- ~~\$query~~ `$result-program-studi = $koneksi->query($query-program-studi);`
- ⇒ Eksekusi query untuk mendapatkan hasil daftar program studi.
- `while ($row = $result-program-studi->fetch_assoc()) { ... }`
- ⇒ Melatikan header melalui hasil query program studi dan membuat opsi dropdown / select box untuk setiap program studi yang ditemukan.
- `if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['cari']) && isset($_GET['prodi'])) { ... }`
- ⇒ Mengecek apakah permintaan merupakan metode GET, dan apakah terdapat parameter 'cari' dan 'prodi' yang dikirimkan.
- `$prodi = $koneksi->real_escape_string($_GET['prodi']);`
- ⇒ Mengambil nilai program studi dari inputan form dan membersihkannya untuk mencegah sql injection dengan menggunakan 'real_escape_string'.
- `$query_cari = "SELECT * FROM mahasiswa WHERE prodi = '$prodi'";`
- ⇒ Membuat query untuk mencari mahasiswa berdasarkan program studi yang dipilih.
- `$result_cari = $koneksi->query($query_cari);`
- ⇒ Mencari mahasiswa berdasarkan program studi yang dipilih.
- `if ($result_cari->num_rows > 0) { ... } else { ... }`
- ⇒ memeriksa apakah hasil pencarian mengembalikan setidaknya satu baris data.
- `$koneksi->close();`
- ⇒ Menutup koneksi ke database setelah selesai menggunakan data.

```

if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['nim'])) {
    $nim = $_GET['nim'];

    $query_edit = "SELECT * FROM mahasiswa WHERE nim = '$nim'";
    $result_edit = $koneksi->query($query_edit);

    if ($result_edit->num_rows > 0) {
        $row = $result_edit->fetch_assoc();
        $nim_edit = $row['nim'];
        $nama = $row['nama'];
        $prodi = $row['prodi'];
    } else {
        echo "Data mahasiswa tidak ditemukan.";
        exit();
    }
}

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['update'])) {
    $nim_edit = $_POST['nim_edit'];
    $nim_baru = $_POST['nim'];
    $nama = $_POST['nama'];
    $prodi = $_POST['prodi'];

    $query_check_nim = "SELECT * FROM mahasiswa WHERE nim = '$nim_baru'";
    $result_check_nim = $koneksi->query($query_check_nim);

    if ($result_check_nim->num_rows > 0 && $nim_baru != $nim_edit) {
        echo "NIM sudah ada dalam database. Harap gunakan NIM yang berbeda.";
    } else {
        $query_update = "UPDATE mahasiswa SET nim = '$nim_baru', nama = '$nama', prodi = '$prodi' WHERE nim = '$nim_edit'";

        if ($koneksi->query($query_update) === TRUE) {
            echo "Data mahasiswa dengan NIM $nim_edit berhasil diperbarui.";
            header("Location: index.php");
            exit();
        } else {
            echo "Error: " . $query_update . "<br>" . $koneksi->error;
        }
    }
}
?>

```

`if ($SERVER["REQUEST_METHOD"] == "GET" || isset($_GET['nim'])) {`
ini adalah sebuah kondisi yang mengecek apakah permintaan adalah tipe GET
(`$SERVER["REQUEST_METHOD"] == "GET"`) dan apakah terdapat parameter 'nim' yang
diterima melalui URL (`isset($_GET['nim'])`). Jika kondisi ini terpenuhi, maka kode didalam
blok akan dijalankan.

`$nim = $_GET['nim'];`

Baris ini mengambil nilai dari parameter 'nim' yang diterima melalui url dan menyimpan
ke dalam variabel '\$nim'.

`$query-edit = "SELECT * FROM mahasiswa WHERE nim = '$nim'";`

digunakan untuk mengambil data mahasiswa dari data base berdasarkan NIM yang diterima
dari parameter URL.

`$result-edit = mysqli_query($koneksi, $query-edit);`

~~\$result-edit~~ digunakan untuk mencari data mahasiswa berdasarkan nim yang diberikan parameter URL

`if ($result-edit->num_rows > 0) {`

Kondisi untuk memeriksa query sebelumnya mengembalikan satu basis data.

`$row = $result-edit->fetch_assoc();`

`$nim-edit = $row['nim'];`

`$nama = $row['nama'];`

`$prodi = $row['prodi'];`

Nim yang diberikan di temukan, maka informasi dari basis pertama hasil query yang disimpan.

`$else {`
`echo " ";`

`exit();`

`}`

Jika nama nim tidak di temukan maka "data mahasiswa tidak ditemukan"

`if ($SERVER["REQUEST_METHOD"] == "POST" || isset($_POST['update'])) {`

`$nim-edit = $_POST['nim-edit'];`

`$nim-baru = $_POST['nim'];`

`$nama = $_POST['nama'];`

`$prodi = $_POST['prodi'];`

Menyimpan ke dalam Variabel yang sesuai.

`$query-check-nim = "SELECT * FROM mahasiswa WHERE nim = '$nim-baru'";`

`$result-check-nim = mysqli_query($koneksi, $query-check-nim);`

Memeriksa Nim yang baru di inputkan masuk ke database.

`if ($result-check-nim->num_rows > 0 || $nim-baru == $nim-edit) {`

Memeriksa nim yang sudah ada di data base dan apakah nim tersebut berbeda dengan nim yang sebelumnya.

`$query-update = "UPDATE mahasiswa SET nim = '$nim-baru', nama = '$nama', prodi`

digunakan untuk melakukan update mahasiswa dengan Nim.


```
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['nim'])) {  
    $nim = $_GET['nim'];  
  
    $query_hapus = "DELETE FROM mahasiswa WHERE nim = '$nim'";  
  
    if ($koneksi->query($query_hapus) === TRUE) {  
        header("Location: index.php");  
        exit();  
    } else {  
        echo "Error: " . $query_hapus . "<br>" . $koneksi->error;  
    }  
}  
?>
```

- `'if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['nim']))`
 => memeriksa apakah request yang diterima adalah metode GET dan apakah terdapat parameter 'nim' yang dikirimkan melalui URL.
- `'$nim = $_GET['nim'];'`
 => mengambil nilai dari parameter 'nim'
- `'$query-hapus = "DELETE FROM mahasiswa WHERE nim = '$nim'" ;'`
 => digunakan untuk menghapus data dari tabel 'mahasiswa' dimana nilai kolom 'nim' sama dengan nilai dari variabel '\$nim'.
- `'if ($koneksi -> query($query-hapus) == TRUE`
 => Melakukan eksekusi query ke database menggunakan objek koneksi.
 Jika penghapusan berhasil, maka :
 - `'header("Location : index.php");'`
 => akan mengarahkan ulang ke halaman 'index.php'.
- `'echo 'Error : " . $query-hapus . "
". $koneksi -> error;'`
 => pesan error jika penghapusan data gagal, berisi detail dari query yang gagal dieksekusi beserta pesan error dari koneksi database.