

## **MODUL VI**

### **INTERAKSI DATABASE**

#### **A. TUJUAN**

- Memahami konsep dasar akses dan manipulasi data.
- Mampu mengintegrasikan aplikasi PHP dengan database MySQL.
- Mampu menghasilkan aplikasi web database untuk pengolahan data sederhana.
- Memahami teknik-teknik akses dan manipulasi data lanjutan.
- Mampu melakukan penambahan, pengubahan, dan penghapusan data melalui aplikasi PHP.
- Mampu menghasilkan aplikasi PHP untuk menangani pengolahan data di database.

#### **B. ALOKASI WAKTU**

4 JS (4 x 50 menit)

#### **C. PETUNJUK**

Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.  
Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.  
Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.  
Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

#### **D. DASAR TEORI**

##### **1. Akses dan Manipulasi Data**

Sehubungan dengan pengolahan data secara permanen, PHP menyediakan pendekatan yang variatif sekali. Salah satu pendekatan yang banyak digunakan adalah menginteraksikan aplikasi-aplikasi PHP dengan database. Dalam hal ini, banyak sekali produk database yang didukung oleh PHP, salah satunya yang paling populer adalah MySQL.

Pembahasan pada materi ini akan memfokuskan dasar-dasar akses dan manipulasi data. Secara khusus, di sini akan dijelaskan mengenai pembuatan

koneksi ke data source, pembuatan database dan tabel, serta manipulasi dasar. Selain itu, juga diulas mengenai pengolahan data berbasis tabel (seperti komponen data grid pada aplikasi desktop). Adapun mengenai materi-materi lanjut, akan dibahas di pertemuan berikutnya.

## E. LATIHAN

### 1. Koneksi ke Data Source

Langkah awal dalam membuat aplikasi web database adalah melakukan koneksi ke data source. Baru kemudian setelah data berhasil diakses, dilanjutkan dengan operasi-operasi lain seperti pembacaan, pengubahan, penghapusan data, dan sebagainya. Buat file koneksi.php seperti berikut:

```
<?php
// Catatan:
// Jika perlu, sesuaikan nama user dan password

$host = 'localhost';
$user = 'root';
$pass = '';
$db = 'mysql';

$conn = mysql_connect($host, $user, $pass);
if (!$conn) {
    exit('Koneksi Gagal');
}
$db = mysql_select_db($db);
if (!$db) {
    exit('Gagal Memilih Database');
}

?>
```

Perhatikan, file di atas tidak dimaksudkan untuk dijalankan secara langsung, melainkan di-include (ditempelkan) pada file-file lainnya. Sebagai contoh, cara penggunaannya seperti berikut:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Test Koneksi</title>
</head>

<body>

<?php
require once './koneksi.php';
?>

<h4>Koneksi Berhasil</h4>

</body>
</html>
```

Apabila kode program di atas memunculkan pesan bahwa koneksi berhasil, maka kita dapat melangkah ke tahap selanjutnya. Adapun jika tidak, lakukan pemeriksaan-pemeriksaan, seperti memastikan bahwa database sudah berjalan dan konfigurasi koneksi sudah benar.

## 2. Menciptakan Database dan Tabel

Tak ubahnya di mysql prompt, kita dapat menciptakan database serta tabel melalui kode program. Tekniknya, pernyataan SQL pembuatan database atau tabel kita masukkan sebagai argumen fungsi `mysql_query()`.

- Menciptakan Database

Sebagaimana diketahui, pernyataan untuk menciptakan database baru adalah `CREATE DATABASE`. Adapun dalam konteks script PHP, pernyataan ini akan kita eksekusi dengan memanfaatkan fungsi `mysql_query()`.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Menciptakan Database</title>
</head>
<body>
<?php
require_once './koneksi.php';

$db = 'myweb';

$res = mysql_query('CREATE DATABASE ' . $db);
if ($res) {
  echo 'Database ' . $db. ' Created';
  mysql_close($res);
} else {
  echo mysql_error();
}

?>
</body>
</html>
```

- Menciptakan Tabel

Sama seperti pembuatan database, kita memanfaatkan fungsi `mysql_query()` untuk menciptakan tabel.

Perlu diperhatikan, tabel ini akan diletakkan di dalam database `myweb` yang baru saja kita buat. Oleh karena itu, terlebih dahulu modifikasi file `koneksi.php` untuk menyesuaikan nama database aktif (ubah **mysql** menjadi **myweb**).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Menciptakan Tabel</title>
</head>

<body>

<?php
require once './koneksi.php';

$sql = 'CREATE TABLE mahasiswa (

    nim VARCHAR(12) NOT NULL,
    nama VARCHAR(40) NOT NULL,
    alamat VARCHAR(100),
    PRIMARY KEY (nim)
) ENGINE=MyISAM;';

$res = mysql_query($sql);
if ($res) {
    echo 'Tabel Created';
    mysql_close($res);
} else {
    echo mysql_error();
}

?>

</body>
</html>
```

Selain menggunakan kode program, kita juga bisa memanfaatkan tool grafis untuk memudahkan administrasi dan pengolahan database, misal melalui phpmyadmin.

### 3. Akses dan Manipulasi Data

Dapat dikatakan, operasi database yang sesungguhnya adalah akses dan manipulasi data. Bagaimanapun, operasi ini akan sangat sering kita lakukan nantinya (dibanding pembuatan struktur tabel).

- Seleksi/Pengambilan Data

Secara umum, segala pernyataan SQL akan kita kirimkan ke server dengan memanfaatkan fungsi `mysql_query()`. Adapun untuk operasi yang mengembalikan hasil, seperti pengambilan data, maka perlu kita tangani lagi secara spesifik.

Sebagai contoh, pengambilan data di tabel dapat dilakukan dengan memanfaatkan fungsi `mysql_fetch_row()`. Fungsi ini akan mengembalikan hasil berupa array numerik sehingga perlu kita uraikan melalui iterasi.

```
<?php
// File: seleksi.php

require_once './koneksi.php';

$sql = 'SELECT * FROM mahasiswa';
$res = mysql_query($sql);
if ($res) {
    if (mysql_num_rows($res)) { ?>

        <table border=1 cellspacing=1 cellpadding=5>
        <tr>
            <th>#</th>
            <th width=100>NIM</th>
            <th width=150>Nama</th>

                <th>Alamat</th>
        </tr>
        <?php
        $i = 1;
        while ($row = mysql_fetch_row($res)) { ?>
            <tr>
                <td>
                    <?php echo $i;?>
                </td>
                <td>
                    <?php echo $row[0];?>
                </td>
                <td>
                    <?php echo $row[1];?>
                </td>
                <td>
                    <?php echo $row[2];?>
                </td>
            </tr>
            <?php
            $i++;
        }
        ?>
        </table>
        <?php
        } else {
            echo 'Data Tidak Ditemukan';
        }
        mysql_close($res);
    }
?>
```

Pada contoh ini, karena kita belum pernah menambahkan data maka hasilnya data tidak ditemukan.

- Penambahan Data

Untuk mendukung penambahan data umumnya aplikasi web menyediakan form khusus yang nantinya dapat digunakan berulang-ulang.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Tambah Data</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
<table>
  <tr>
    <td>NIM</td>
    <td><input type="text" name="nim" /></td>
  </tr>
  <tr>
    <td>Nama</td>
    <td><input type="text" name="nama" size=40 /></td>
  </tr>
  <tr>
    <td>Alamat</td>
    <td><input type="text" name="alamat" size=60 /></td>
  </tr>

  <tr>
    <td></td>
    <td><input type="submit" value="Simpan" /></td>
  </tr>
</table>

</form>

<?php
require_once './koneksi.php';

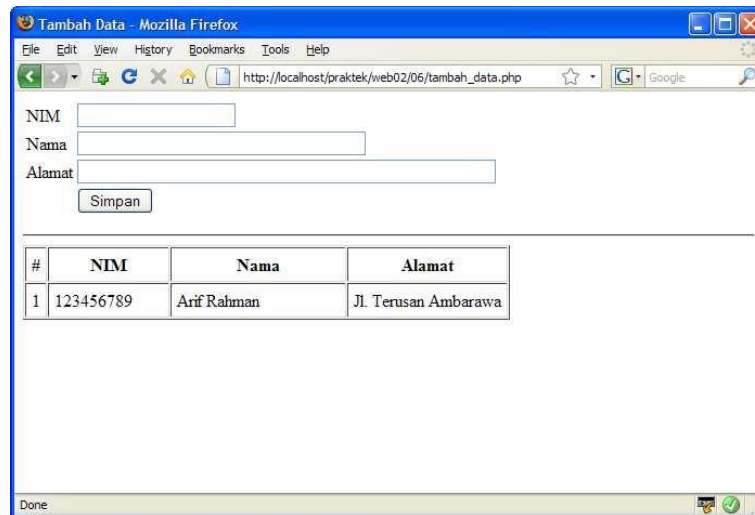
// Jika field nim dan nama diisi lalu disubmit
if (isset($_POST['nim']) && isset($_POST['nama'])) {
  $nim = $_POST['nim'];
  $nama = $_POST['nama'];
  $alamat = $_POST['alamat'];

  // Tambahkan data baru ke tabel
  $sql = "INSERT INTO mahasiswa
        VALUES ('.$nim.', '$nama.', '$alamat.')";

  $res = mysql_query($sql);
  if ($res) {
    echo 'Data Berhasil Ditambahkan';
    mysql_close($res);
  } else {
    echo 'Gagal Menambah Data <br />';
    echo mysql_error();
  }
}

echo '<hr />';
// Memanfaatkan script pengambilan data untuk
// menampilkan hasil
require_once './seleksi.php';
?>

</body>
</html>
```



Gambar 1. Penambahan dan seleksi data

#### 4. Administrasi Data

Untuk menghasilkan halaman akses dan manipulasi data yang terintegrasi, ada beberapa tahap yang perlu dilakukan. Adapun agar lebih mudah dalam melakukan pengelolaan, buat folder baru, misal **aplikasi\_db**.

- Halaman Index

Dalam upaya memudahkan transisi ke aplikasi yang kompleks, kita menyediakan halaman **index.php** sebagai halaman utama. Adapun kode program yang dilibatkan adalah sebagai berikut:

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Akses dan Manipulasi Data</title>
  <style type="text/css">
    .even {
      background: #ddd;
    }
  </style>
</head>

<body>

<?php
ini_set('display_errors',1);

// Meng-include file koneksi dan data handler
require_once './koneksi.php';
require_once './data_handler.php';

// Konstanta nama tabel
define('MHS', 'mahasiswa');
```

```
// Memanggil fungsi data handler
data_handler('?m=data');

?>

</body>
</html>
```

Kode program di atas masih belum bisa dijalankan karena memerlukan file `data_handler.php` yang masih belum dibuat.

Pada prinsipnya, halaman index akan menjadi entry point aplikasi sehingga tidak perlu terdapat banyak deklarasi.

- Koneksi Data Source

Untuk file koneksi ke data source, gunakan kode program **koneksi.php** yang telah dibuat sebelumnya. Di sini tidak diperlukan konfigurasi tambahan apa pun.

- Administrasi Data

Pengelolaan data secara khusus kita letakkan ke dalam sebuah file bernama **data\_handler.php**. File ini berisi segala fungsi yang diperlukan untuk pengolahan data.

Bagian pertama dari data handler ini adalah untuk identifikasi menu pilihan. Kode programnya diperlihatkan sebagai berikut:

```
<?php

/**
 * Fungsi utama untuk menangani pengolahan data
 * @param string root parameter menu
 */
function data_handler($root) {
    if (isset($_GET['act']) && $_GET['act'] == 'add') {
        data_editor($root);
        return;
    }

    $sql = 'SELECT COUNT(*) AS total FROM ' . MHS;
    $res = mysql_query($sql);

    // Jika data di tabel ada
    if (mysql_num_rows($res)) {

        if (isset($_GET['act']) && $_GET['act'] != '') {
            switch($_GET['act']) {
                case 'edit':
                    if (isset($_GET['id']) && ctype_digit($_GET['id'])) {
                        data_editor($root, $_GET['id']);
                    } else {
                        show_admin_data($root);
                    }
                }
            }
        }
        break;
    }
}
```



```
case 'view':
    if (isset($_GET['id']) && ctype_digit($_GET['id'])) {
        data_detail($root, $_GET['id'], 1);
    } else {
        show_admin_data($root);
    }
    break;

case 'del':
    if (isset($_GET['id']) && ctype_digit($_GET['id'])) {
        // Key untuk penghapusan data
        $id = $_GET['id'];

        // Lengkapi pernyataan SQL hapus data

        $res = mysql_query($sql);
        if ($res) { ?>

            // Lengkapi script untuk redireksi ke root

        }
    } else {
        <?php
        } else {
            echo 'Gagal menghapus data';
        }
        } else {
            show_admin_data($root);
        }
    }
    break;
default:
    show_admin_data($root);
}
} else {
    show admin data($root);
}

@mysql_close($res);
} else {
    echo 'Data Tidak Ditemukan';
}
}
```

- Menu Administrasi Data

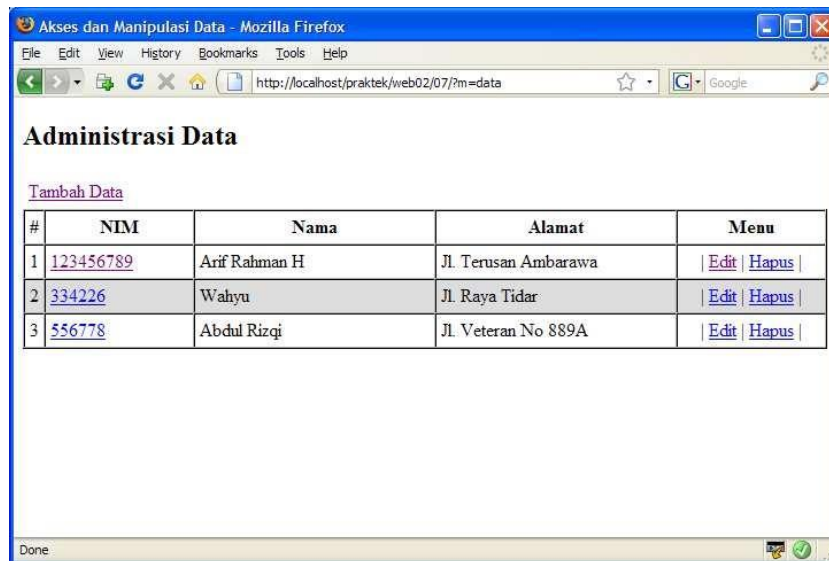
Masih di file yang sama, kita menyediakan halaman administrasi yang direpresentasikan dalam bentuk tabular (seperti data grid). Desain menu seperti ini akan memudahkan operator ketika ingin mengelola data.

```
/**
 * Fungsi untuk menampilkan menu administrasi
 * @param string root parameter menu
 */
function show_admin_data($root) { ?>
    <h2 class="heading">Administrasi Data</h2>
    <?php
    $sql = 'SELECT nim, nama, alamat FROM ' . MHS;
    $res = mysql_query($sql);
```

```

if ($res) {
    $num = mysql_num_rows($res);
    if ($num) {
        ?>
        <div class="tabel">
            <div style="padding:5px;">
                <a href="<?php echo $root;?>&amp;act=add">Tambah Data</a>
            </div>
            <table border=1 width=700 cellpadding=4 cellspacing=0>
                <tr>
                    <th>#</th>
                    <th width=120>NIM</th>
                    <th width=200>Nama</th>
                    <th width=200>Alamat</th>
                    <th>Menu</th>
                </tr>
                <?php
                $i = 1;
                while ($row = mysql_fetch_row($res)) {
                    $bg = (($i % 2) != 0) ? '' : 'even';
                    $id = $row[0]; ?>
                    <tr class="<?php echo $bg;?>">
                        <td width="2%"><?php echo $i;?></td>
                        <td>
                            <a href="<?php echo $root;?>&amp;act=view&amp;id=
                            <?php echo $id;?>"
                            title="Lihat Data"><?php echo $id;?></a>
                        </td>
                        <td><?php echo $row[1];?></td>
                        <td><?php echo $row[2];?></td>
                        <td align="center">
                            | <a href="<?php echo $root;?>&amp;act=edit&amp;id=
                            <?php echo $id;?>"
                            Edit</a> |
                        <!--
                            Lengkapi kode PHP untuk membuat link hapus data
                        -->
                    </td>
                </tr>
                <?php
                $i++;
            }
        ?>
    </table>
    </div>
    <?php
    } else {
        echo 'Belum ada data,
            isi <a href="'. $root. '&amp;act=add">di sini</a>';
    }
    @mysql_close($res);
}
}

```



Gambar 1. Halaman administrasi data

- Menampilkan Detail Data

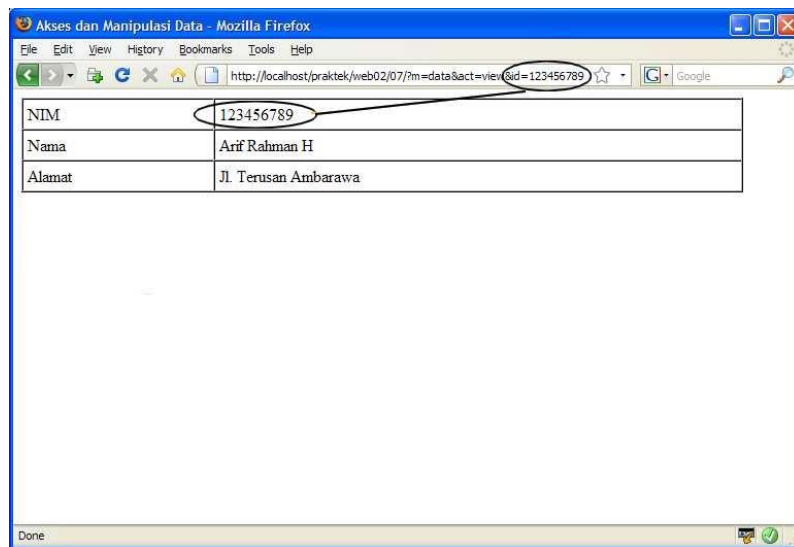
Pembuatan detail data sangat sederhana, di mana kita menampilkan sebuah baris data berdasarkan id nim yang dikirimkan melalui parameter GET.

Berikut kode programnya:

```
/**
 * Fungsi untuk menampilkan detail data mahasiswa
 * @param string root parameter menu
 * @param integer id nim mahasiswa
 */
function data_detail($root, $id) {
    $sql = 'SELECT nim, nama, alamat
          FROM ' . MHS .
          ' WHERE nim=' . $id;
    $res = mysql_query($sql);
    if ($res) {
        if (mysql_num_rows($res)) { ?>
            <div class="tabel">
                <table border=1 width=700 cellpadding=4 cellspacing=0>
                    <tr>
                        <td>NIM</td>
                        <td><?php echo $row[0];?></td>
                    </tr>
                    <tr>
                        <td>Nama</td>
                        <td><?php echo $row[1];?></td>
                    </tr>
                    <tr>
                        <td>Alamat</td>
                        <td><?php echo $row[2];?></td>
                    </tr>
                </table>
            </div>
        }
    }
}
```

```
<?php
} else {
    echo 'Data Tidak Ditemukan';
}
mysqli_close($res);
}
```

Perhatikan, di fungsi sebelumnya (menu administrasi), kita menyediakan link untuk menuju ke menu-menu terkait. Setiap menu ini akan menyertakan id unik sebagai kunci akses data. Dengan demikian—dalam pengambilan detail data ini—kita tinggal memanfaatkan untuk menampilkan data.



Gambar 2. Menampilkan detail data

- Halaman Modifikasi Data

Sama seperti menampilkan data, modifikasi juga menggunakan teknik serupa. Adapun untuk efisiensi, halaman modifikasi sekaligus berfungsi sebagai penambahan dan perubahan data.

```
/**
 * Fungsi untuk menghasilkan form penambahan/pengubahan
 * @param string root parameter menu
 * @param integer id nim mahasiswa
 */
function data_editor($root, $id = 0) {
    $view = true;
    if (isset($POST['nim']) && $POST['nim'] ) {
        // Jika tidak disertai id, berarti insert baru
        if (!$id) {

            // Lengkapi Pernyataan PHP SQL untuk INSERT data
        }
    }
}
```

```
$res = mysql_query($sql);
if ($res) { ?>
  <script type="text/javascript">
    document.location.href="<?php echo $root;?>";
  </script>
<?php
} else {
  echo 'Gagal menambah data';
}
} else {

  // Lengkapi Pernyataan PHP SQL untuk UPDATE data

  $res = mysql_query($sql);
  if ($res) { ?>

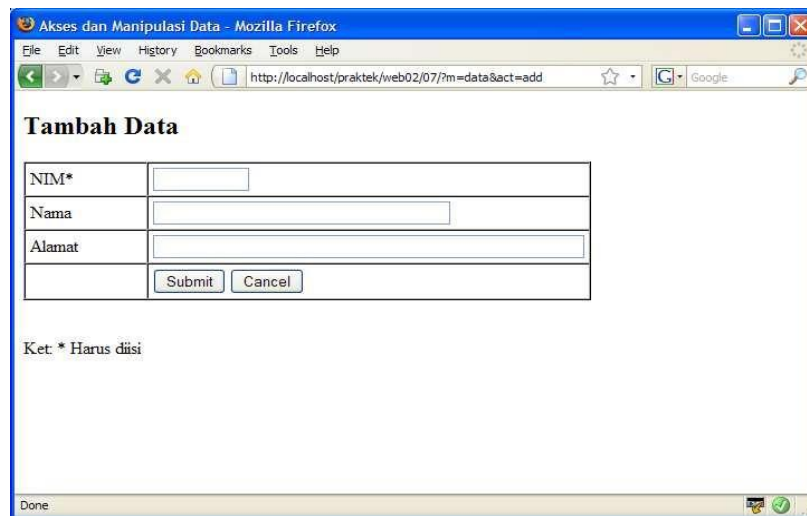
    // Lengkapi script untuk redireksi ke root

    <?php
    } else {
      echo 'Gagal memodifikasi';
    }
  }
}

// Menyiapkan data untuk updating
if ($view) {
  if ($id) {
    $sql = 'SELECT nim, nama, alamat FROM ' . MHS .
      ' WHERE nim=' . $id;
    $res = mysql_query($sql);
    if ($res) {
      if (mysql_num_rows($res)) {
        $row = mysql_fetch_row($res);
        $nim = $row[0];
        $nama = $row[1];
        $alamat = $row[2];
      } else {
        show_admin_data();
        return;
      }
    }
  }
} else {
  $nim = @$_POST['nim'];
  $nama = @$_POST['nama'];
```

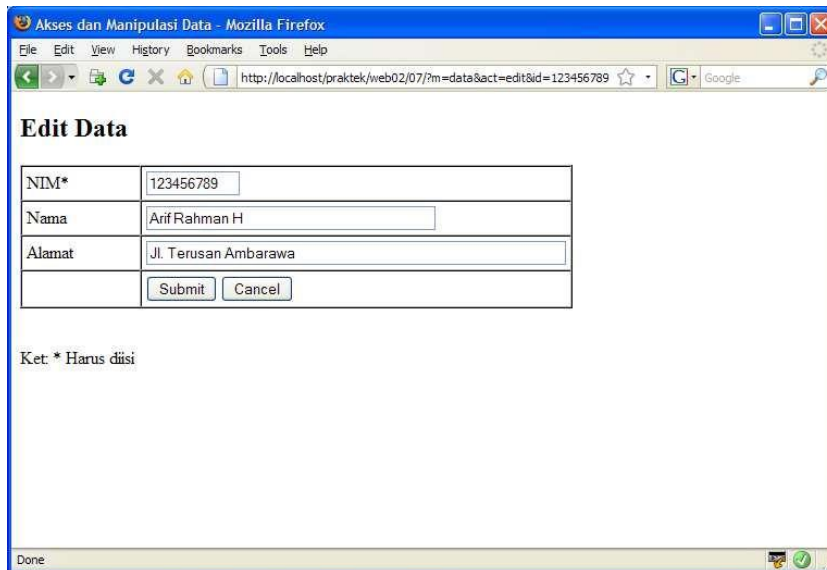
```
$alamat = @$_POST['alamat'];
}
?>
<h2> <?php echo $id ? 'Edit' : 'Tambah';?> Data</h2>
<form action="" method="post">
<table border=1 cellpadding=4 cellspacing=0>
<tr>
<td width=100>NIM*</td>
<td> <input type="text" name="nim" size=10
value="<?php echo $nim;?>" /> </td>
</tr>
<tr>
<td>Nama</td>
<td> <input type="text" name="nama" size=40
value="<?php echo $nama;?>" /> </td>
</tr>
<tr>
<td>Alamat</td>
<td> <input type="text" name="alamat" size=60
value="<?php echo $alamat;?>" /> </td>
</tr>
<tr>
<td> </td>
<td><input type="submit" value="Submit" />

```



Gambar 3. Form penambahan data

Cara membedakan penyajian halaman—apakah penambahan atau pengubahan—sangat sederhana, di mana jika argumen id pada fungsi `data_editor()` tidak diisi maka halaman penambahan ditampilkan, sebaliknya akan menampilkan halaman pengubahan.

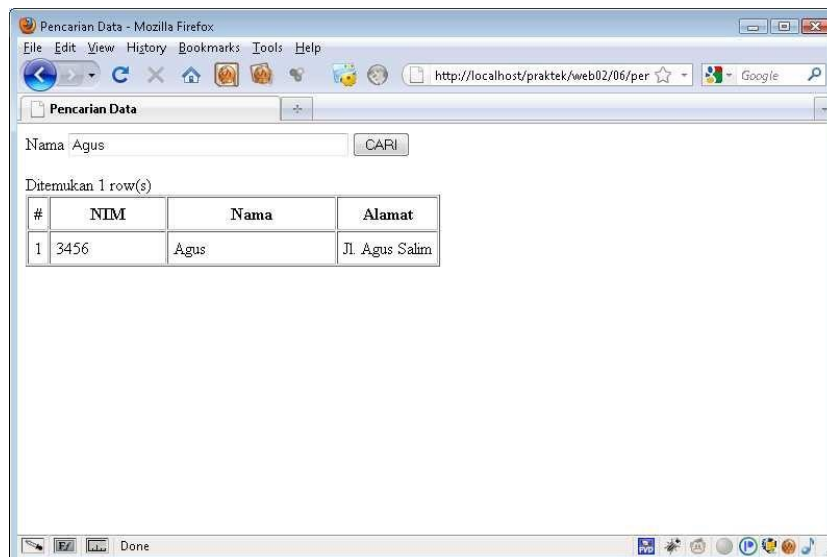


Gambar 4. Form pengubahan data

## 6. Pencarian Data

Operasi ini mutlak diperlukan dalam rangka pengambilan kembali (retrieval) informasi yang sudah tersimpan di database. Melanjutkan pembahasan di praktikum sebelumnya, akan diperlihatkan contoh pencarian data pada tabel mahasiswa. Kondisi pencarian yang digunakan di sini didasarkan pada field nama.

Contoh tampilan hasil pencarian diperlihatkan seperti Gambar 5.

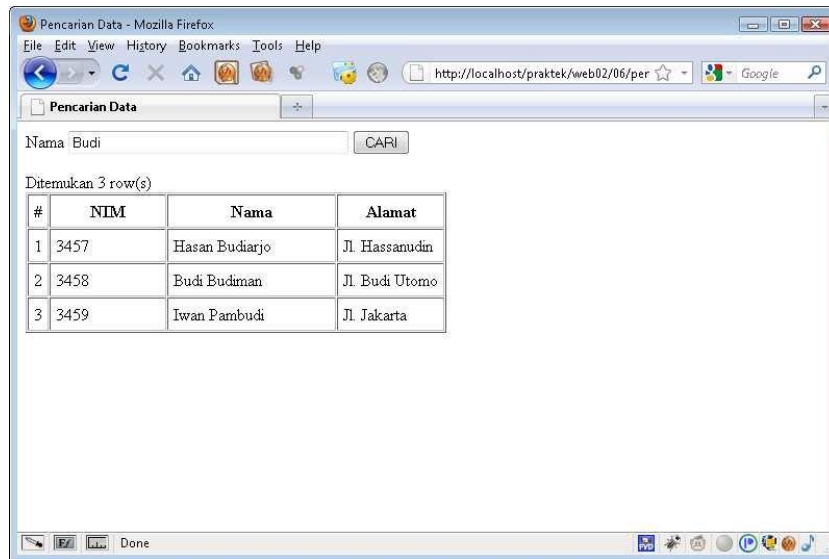


Gambar 5. Tampilan hasil pencarian data

### **Latihan Kecil**

Tambahkan prefilling pada text field sehingga kata kunci tidak hilang ketika halaman di-submit.

Agar kemungkinan hasil pencarian lebih banyak, gunakan klausa “mirip/mendekati” (bukan persis). Selain itu, libatkan lebih dari satu kriteria pencarian, misal nama dan alamat.



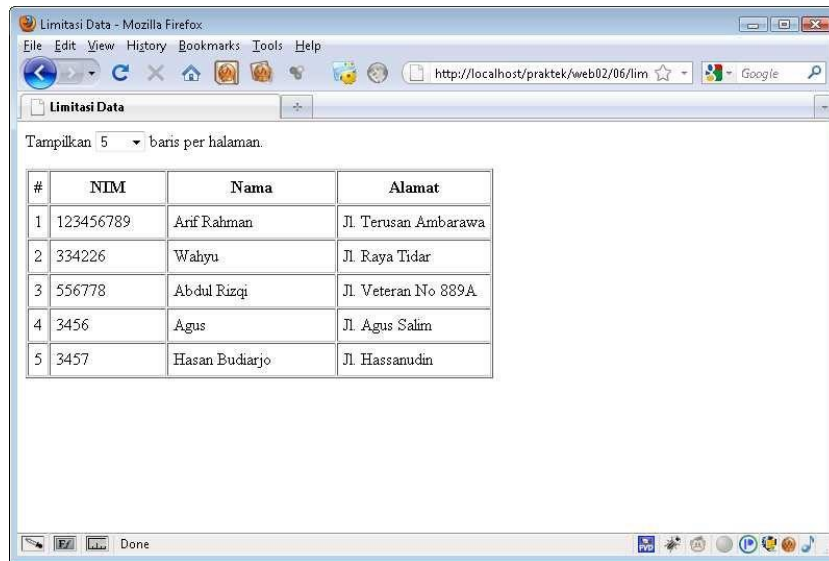
*Gambar 6. Pencarian dengan klausa mendekati*

## **2. Membatasi Tampilan Data**

Salah satu upaya guna meningkatkan efisiensi adalah dengan membatasi jumlah data yang akan ditampilkan. Pendekatan ini sering digunakan pada pengambilan data yang melibatkan jumlah besar. Tekniknya, baris data akan ditampilkan sebanyak batasan yang telah ditetapkan.

Contoh berikut memperlihatkan penempatan nilai batas pada seleksi dan bersifat autosubmit.





Gambar 7. Membatasi tampilan data

### **Latihan Kecil**

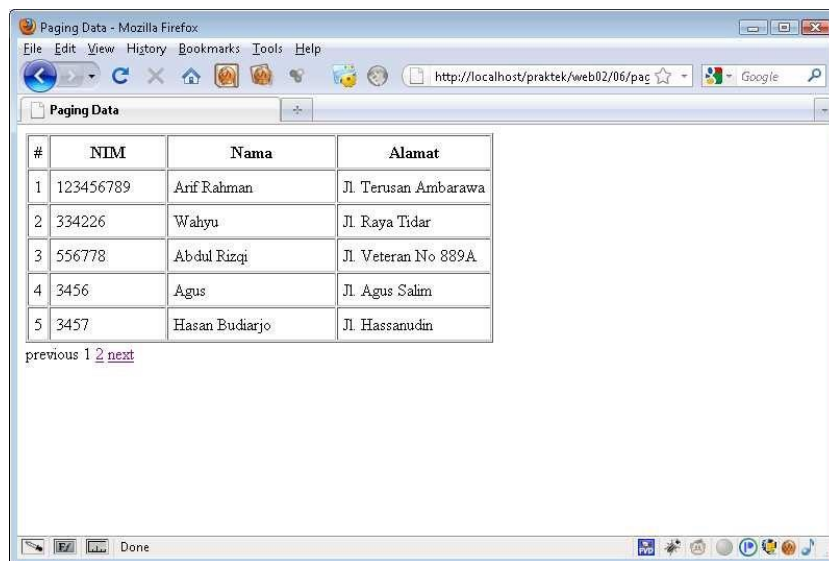
Lengkap kode seleksi dengan menambahkan prefilling sehingga nilai yang terpilih akan tetap ditampilkan begitu halaman di-submit.

### **3. Pemberian Halaman**

Pemberian halaman (paging) merupakan teknik yang banyak digunakan untuk memecah halaman (yang panjang) ke beberapa bagian di mana tiap-tiap bagian dapat diakses melalui pemberian halaman yang telah ditetapkan. Langkah ini sangat efisien sekali dan banyak digunakan untuk halaman pencarian, rekapitulasi, data tabular, deskripsi berita/posting, dan masih banyak lagi.

Teknik pemberian halaman sangat variatif, mulai dari yang paling sederhana hingga kompleks. Kode program berikut memperlihatkan pemberian halaman yang cukup baik, di mana menyediakan link previous, next, dan counter halaman (1, 2, 3, dst).

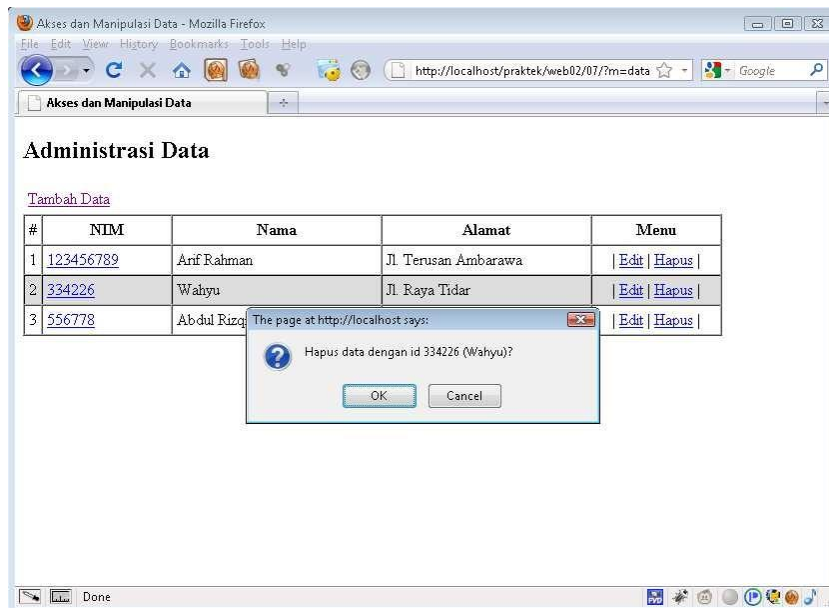
Contoh tampilan hasil pemberian halaman (*paging*) diperlihatkan seperti pada Gambar 8.



Gambar 8. Tampilan hasil paging

## F. STUDI KASUS

1. Sempurnakan operasi penghapusan data dengan menambahkan konfirmasi penghapusan dan memberitahukan id data yang akan dihapus.

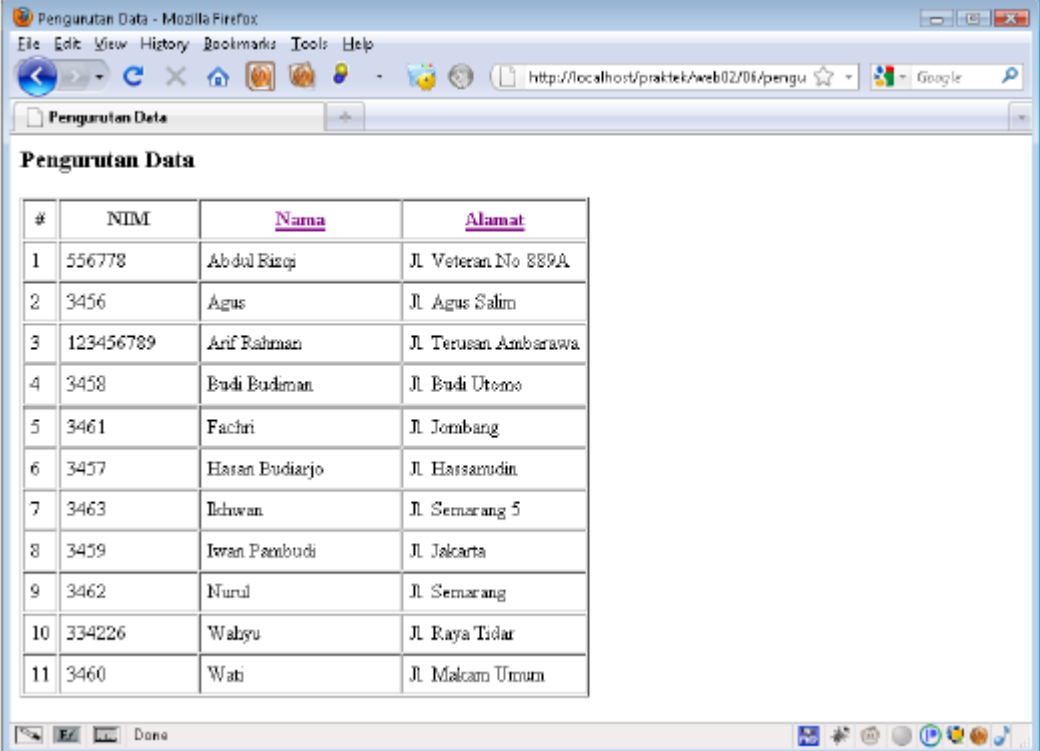


Gambar 9. Konfirmasi penghapusan data

2. Merujuk pada latihan, kombinasikan limitasi data dengan pemberian halaman pada satu tabel.

## G. TUGAS PRAKTIKUM

1. Terkait dengan latihan administrasi data, buat kode untuk melakukan seleksi data pada tabel mahasiswa dan letakkan di folder tertentu. Halaman ini nantinya bisa diakses oleh siapa pun. Selanjutnya, integrasikan kode untuk login (di pertemuan sebelumnya) agar menampilkan halaman administrasi (juga seperti pada latihan). Upload aplikasi database sederhana ini ke hosting masing-masing.
2. Buat seleksi data dengan menambahkan fitur pengurutan (menaik maupun menurun) melalui header tabel. Sebagai contoh, jika header Nama diklik maka akan melakukan pengurutan menaik (ascending); dan begitu diklik lagi, akan melakukan pengurutan menurun (descending). Contoh tampilan hasilnya diperlihatkan seperti Gambar 10.



The screenshot shows a web browser window titled "Pengurutan Data - Mozilla Firefox". The address bar shows the URL "http://localhost/praktek/web02/06/pengu...". The page content is titled "Pengurutan Data" and displays a table with 11 rows of student data. The table has four columns: "#", "NIM", "Nama", and "Alamat". The data is sorted by name in ascending order.

#	NIM	<u>Nama</u>	<u>Alamat</u>
1	556778	Abdul Rizaq	Jl Veteran No 889A
2	3456	Agus	Jl Agus Salim
3	123456789	Arif Rahman	Jl Terusan Ambarawa
4	3458	Budi Budiman	Jl Budi Utomo
5	3461	Fachri	Jl Jombang
6	3457	Hasan Budiarjo	Jl Hassanudin
7	3463	Ikhwan	Jl Semarang 5
8	3459	Iwan Pambudi	Jl Jakarta
9	3462	Nurul	Jl Semarang
10	334226	Wahyu	Jl Raya Tidar
11	3460	Wati	Jl Miskam Umum

Gambar 10. Pengurutan menaik berdasar nama