

## **BAB VI**

### **AKSES DAN MANIPULASI DATA LANJUT**

#### **A. TUJUAN**

- 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 akses dan manipulasi data di MySQL, PHP—setidaknya hingga saat ini—menyediakan dua pendekatan: melalui API MySQL dan MySQL Improved (MySQLi). API yang kedua secara khusus dimaksudkan untuk mendukung implementasi dengan paradigma orientasi objek.

Pembahasan pada materi ini akan menggunakan pendekatan konvensional (melalui API MySQL) untuk melakukan operasi-operasi data. Melanjutkan materi di pembahasan sebelumnya, bagian ini secara khusus memfokuskan pada teknik pengolahan data yang lazim digunakan di dalam aplikasi-aplikasi PHP. Beberapa operasi yang dibahas di antaranya adalah pencarian data, limitasi tampilan data, serta teknik pemberian halaman (paging).

## E. LATIHAN

### 1. 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.

```
<!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>Pencarian Data</title>
</head>

<body>

<form action="<?php $_SERVER['PHP_SELF'];?>" method="get">
  Nama <input type="text" name="nama" size=40 />
  <input type="submit" value="CARI" />
</form>

<?php
if (isset($_GET['nama'])) {
  require_once './koneksi.php';

  // Kata kunci pencarian
  $key = $_GET['nama'];

  // Variabel $sql berisi pernyataan SQL pencarian
  $sql = "SELECT * FROM mahasiswa
        WHERE nama = '$key'";

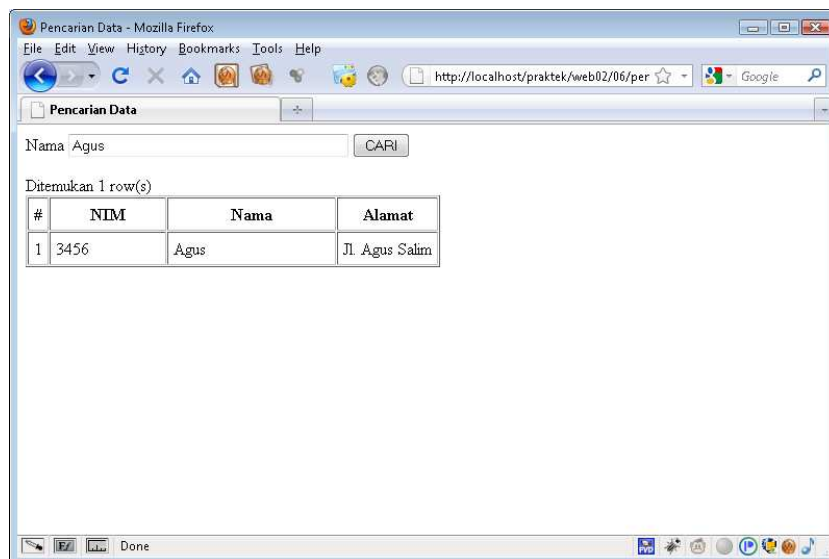
  $res = mysql_query($sql);

  if ($res) {
    $num = mysql_num_rows($res);
    if ($num) {
      echo 'Ditemukan ' . $num . ' row(s)'; ?>

      <table border=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>
      </table>
    }
  }
}
```

```
        </td>
      </tr>
    <?php
      $i++;
    }
  ?>
</table>
<?php
} else {
  echo 'Data Tidak Ditemukan';
}
}
} else {
  echo 'Masukkan kata kunci pencarian';
}
?>
</body>
</html>
```

Contoh tampilan hasil pencarian diperlihatkan seperti Gambar 1.

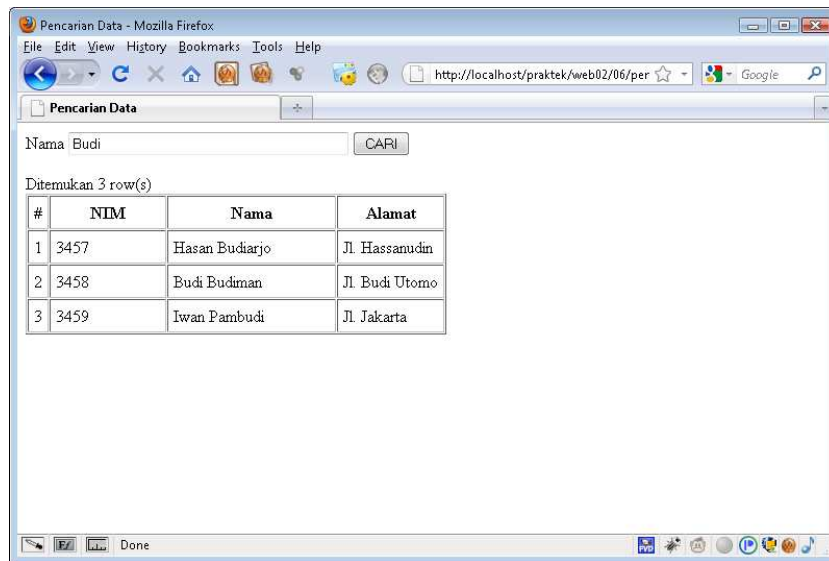


*Gambar 1. Tampilan hasil pencarian data*

### **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 2. 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.

```
<!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>Limitasi Data</title>
</head>

<body>

  <form method="post" action="" name="frm_select">
    Tampilkan
    <select name="row_page"
onchange="document.forms.frm_select.submit();">
      <option>-- pilih --</option>
      <option value="5">5</option>
      <option value="10">10</option>
      <option value="20">20</option>
      <option value="50">50</option>
      <option value="100">100</option>
    </select> baris per halaman.
  </form>

  <?php
if (isset($_POST['row_page']) && $_POST['row_page']) {
  require_once './koneksi.php';

  // Batas baris data
  $row = $_POST['row_page'];
```

```
// LENGKAPI
// Variabel $sql berisi pernyataan SQL retrieve dg limitasi

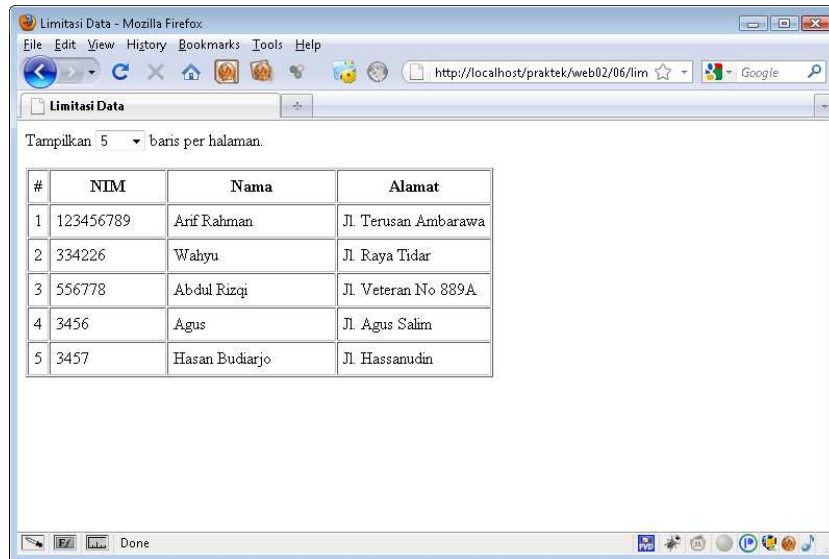
$res = mysql_query($sql);

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';
        }
    }
}

?>

</body>
</html>
```



Gambar 3. 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).

```
<!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>Paging Data</title>
</head>

<body>

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

```
//***** Setup paging
$sql = 'SELECT * FROM mahasiswa';

$self = $_SERVER['PHP_SELF'];
$page = isset($_GET['page']) ? $_GET['page'] : 0;

// Jumlah link counter, misal (prev 1 2 3 next) = 3
$link_num = 5;
// Jumlah record per halaman
$record_num = 5;

// Item pertama yang akan ditampilkan
$offset = $page ? ($page - 1) * $record_num : 0;

$total_rows = mysql_num_rows(mysql_query($sql));
$query = $sql. ' LIMIT ' . $offset . ', ' . $record_num;
$result = mysql_query($query);
$max_page = ceil($total_rows/$record_num);

// Reset jika page tidak sesuai
if ($page > $max_page || $page <= 0) {
    $page = 1;
}

//***** Generate paging
$paging = '';
if($max_page > 1) {
    /*** Render link previous
    if ($page > 1) {
        $paging .= ' <a href="'. $self. '?page=' . ($page-1) . '">previous</a>
';
    } else {
        $paging .= ' previous ';
    }

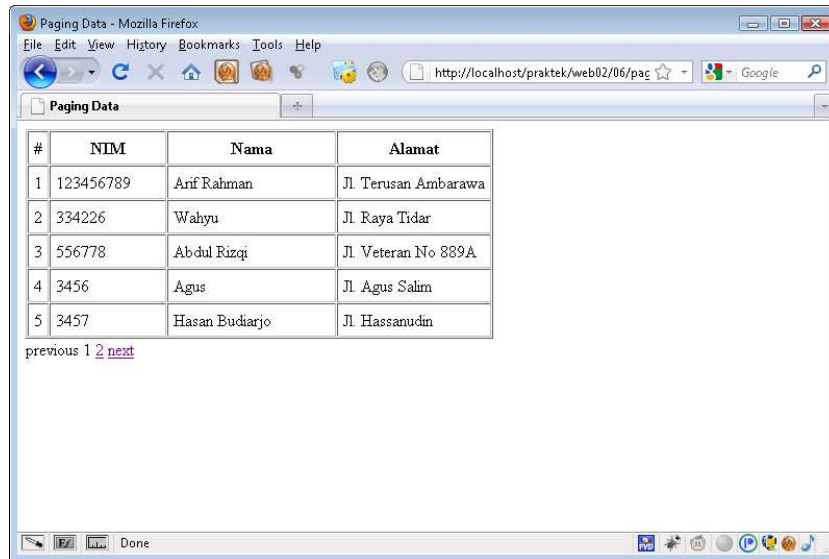
    /*** Render link counter halaman
    for ($counter = 1; $counter <= $max_page; $counter += $link_num) {
        if ($page >= $counter) {
            $start = $counter;
        }
    }
    if ($max_page > $link_num) {
        $end = $start + $link_num;
        if ($end > $max_page) {
            $end = $max_page + 1;
        }
    } else {
        $end = $max_page;
    }
    for ($counter = $start; $counter < $end; $counter++) {
        if ($counter == $page) {
            $paging .= $counter;
        } else {
            $paging .= ' <a href="'. $self. '?page=' . $counter . '">' . $counter.
'</a> ';
        }
    }

    /*** Render link next
    if ($page < $max_page) {
        $paging.= ' <a href="'. $self. '?page=' . ($page+1) . '">next</a> ';
    } else {
        $paging.= ' next ';
    }
}
```

```
}  
}  
?>  
  
<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($result)) { ?>  
  <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  
  
//***** Tampilkan navigasi  
echo $paging;  
  
?>  
  
</body>  
</html>
```

Contoh tampilan hasil pemberian halaman (*paging*) diperlihatkan seperti pada Gambar 4.





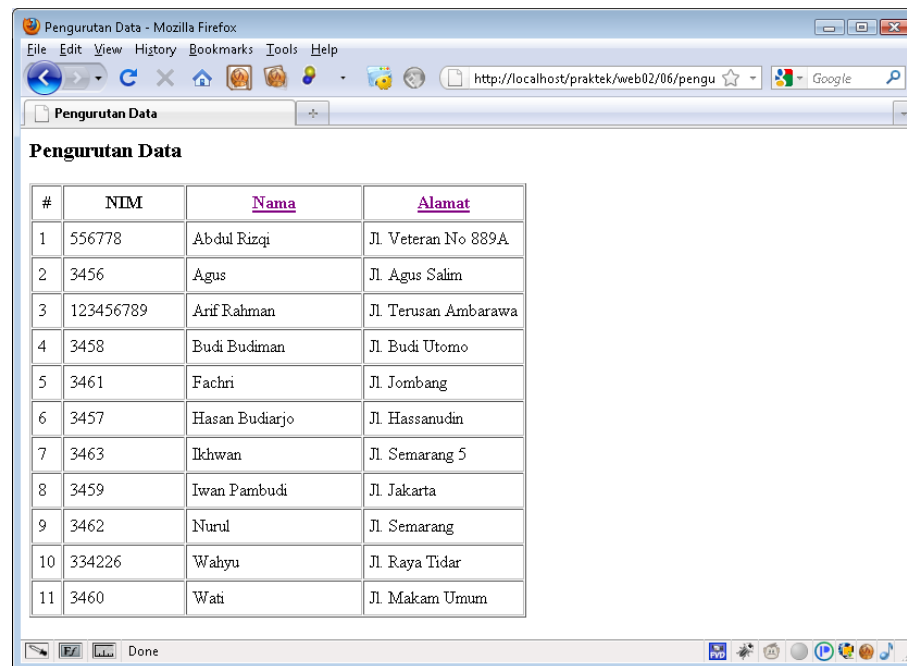
Gambar 4. Tampilan hasil paging

## **F. STUDI KASUS**

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

## G. TUGAS PRAKTIKUM

1. 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 5.



The screenshot shows a web browser window titled 'Pengurutan Data - Mozilla Firefox'. The address bar shows 'http://localhost/praktek/web02/06/pengu'. The page content is titled 'Pengurutan Data' and displays a table with 4 columns: '#', 'NIM', 'Nama', and 'Alamat'. The table contains 11 rows of data, sorted by the 'Nama' column in ascending order.

#	NIM	Nama	Alamat
1	556778	Abdul Rizqi	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. Makam Umum

Gambar 5. Pengurutan menaik berdasar nama