

Mysqli_result Class

Mysqli_result adalah class khusus yang dipakai untuk proses menampilkan data tabel, yakni mengolah hasil dari query `SELECT`. Dibandingkan pembuatan tabel, proses menampilkan data tabel lebih sering kita pakai. **Mysqli_result** memiliki cukup banyak method yang akan kita bahas secara detail.

CATATAN: latihan-latihan yang akan kita lakukan pada pertemuan kali ini ada cukup banyak jadi untuk menghemat penulisan modul, untuk latihan-latihan selanjutnya kalian hanya perlu focus menambahkan bagian **blok try**. Berikut contoh program dengan struktur lengkap nya.

Kita akan menampilkan data yang telah kita simpan dalam database, dengan beberapa cara. Simpan program berikut dengan nama **select1.php** pada direktori **pertemuan9_10**

(ingat untuk latihan2 program selanjutnya saya hanya akan menampilkan bagian blok try, tapi kalian tetap harus menulis secara lengkap agar program dapat berjalan.)

Secara umum program dibawah tidak banyak perubahan dengan program2 sebelumnya, hanya disini kita menggunakan query select karena memang tujuan kita ingin menampilkan data dari database, jadi jangan bingung ya. Saya tidak akan bahas secara detail seluruh baris program, saya akan bahas beberapa bagian baru yang sebelumnya belum kita pelajari.

```
<?php
mysqli_report(MYSQLI_REPORT_STRICT);

try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom"); 1

    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang"; 2
    $result = $mysqli->query($query); 3
    if ($mysqli->error){
        throw new Exception($mysqli->error, $mysqli->errno);
    }
    else {
        echo "<pre>";
        print_r($result); 4
        echo "<pre>";
        $result->free(); 5
    };
}
```

```

catch (Exception $e) {
    echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
    if (isset($mysqli)) {
        $mysqli->close();
    }
}
}

```

Pembahasan.

1. Pembuatan object dari class mysqli dengan tujuan untuk membuka koneksi.
2. Pembuatan variable \$query untuk menampung query select data
3. Pembuatan variable \$result untuk menampung data **mysqli_result object** (penting!!!) hasil eksekusi query dengan fungsi query() yang ada dalam class mysqli.
4. Fungsi print_r(\$result) untuk melihat detail isi variable yang menampung **mysqli_result object**
5. Fungsi free() untuk mengosongkan kembali isi variable \$result yang sebelumnya menampung hasil query

Hasil kode program:

```

mysqli_result Object
(
    [current_field] => 0
    [field_count] => 5
    [lengths] =>
    [num_rows] => 5
    [type] => 0
)

```

Dari hasil tampilan kode program terlihat bahwa **mysqli_result** object memiliki 5 property: **current_field**, **field_count**, **lengths**, **num_rows** dan **type**. **Mysqli_result** merupakan **jenis associative array** (penting!!!)

Dari kelima property ini, yang akan sering kita akses adalah **num_rows** dan **field_count**.

CATATAN PENTING HARUS DIPAHAMI!!!

Property mysqli_result::num_rows berisi informasi total jumlah baris hasil query SELECT (jumlah record). Sedangkan **property mysqli_result::field_count** berisi total kolom tabel hasil dari query SELECT. Dalam contoh ini property num_rows berisi angka 5 karena terdapat 5 record di dalam tabel barang, Begitu juga dengan property field_count yang berisi angka 5 karena tabel barang memiliki 5 kolom.

Sampai sini paham ya?? Baca pelan-pelan. Ok kita akan lanjut dengan melihat bagaimana cara kita mengakses property field_count dan num_rows

Simpan program dibawah ini dengan nama **select2.php**, ingat saya tulis bagian blok try saja, kalian tulis lengkap seperti contoh program sebelumnya.

```
try {  
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");  
  
    // Tampilkan data dari tabel barang  
    $query = "SELECT * FROM barang WHERE id_barang = 1";  
    $result = $mysqli->query($query);  
  
    // Tampilkan jumlah baris dan kolom  
    echo "Terdapat ".$result->field_count." kolom dan ".$result->num_rows." baris";  
    $result->free();  
}
```

Hasil kode program:

Terdapat 5 kolom & 1 baris

Tidak ada perubahan banyak dari program select1.php disini hanya ada perbedaan query dimana kita menambahkan kondisi where id_barang=1, artinya kita hanya akan menampilkan data barang yang memiliki id_barang 1.

Nah bagian **echo** kalian perhatikan, disana kita mengakses property field_count dan num_rows, tujuannya hanya ingin mengetahui berapa jumlah data yang berhasil diambil dari database, terdapat 1 data barang dengan id_barang 1 dan data tersebut memiliki 5 kolom.

Paham ya gaes,

Ok lanjut, latihan dibawah ini kita akan menambahkan contoh validasi, jika data yang dicari tidak ada maka akan menampilkan pesan, nah untuk melakukan fungsi validasi ini kita membutuhkan salah satu dari property mysqli_result yang sebelumnya kita telah pelajari yaitu **num_rows**. Simpan program dengan nama **select3.php**

```
try {  
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");  
  
    // Tampilkan data dari tabel barang  
    $query = "SELECT * FROM barang WHERE id_barang = 100";  
    $result = $mysqli->query($query);
```

```

if ($result->num_rows === 0) {
    echo "Data tidak ditemukan";
}
else {
    echo "Data tersedia";
}
$result->free();
}

```

jika kondisi `if($result->num_rows === 0)` menghasilkan nilai `true`, bisa dipastikan data tidak tersedia di dalam tabel.

Hasil kode program:

Data tidak ditemukan

Menampilkan Tabel Dengan `mysqli_result` Object

Dalam bahasan sebelumnya kita telah lihat bahwa **`mysqli_result`** object dipakai untuk menampung hasil dari perintah query MySQL, terutama query `SELECT`. Data yang tersimpan di dalam **`mysqli_result`** object bisa ditampilkan dalam 4 bentuk:

1. Normal Array (dengan index berupa angka numeric)
2. Associative array (dengan index berupa nama kolom)
3. Normal Array atau associative array (dengan index berupa angka numeric atau nama kolom)
4. Object (dengan property berupa nama kolom)

Untuk mengambil data dari **`mysqli_result`** sebagai *normal array*, kita bisa menggunakan method `mysqli_result::fetch_row()` atau `mysqli_result::fetch_array(MYSQLI_NUM)`. Berikut contoh penggunaannya, simpan dengan nama **`select4.php`**:

Pahami program dibawah dengan pelan-pelan.

```

try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");
    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);
    $row = $result->fetch_row();

    echo "<pre>";
    print_r($row);
    echo "<pre>";
}

```

```
$result->free();  
}
```

Hasil kode program:

```
Array  
(  
    [0] => 1  
    [1] => TV Samsung 43NU7090 4K  
    [2] => 5  
    [3] => 5399000  
    [4] => 2019-01-04 16:59:53  
)
```

Perbedaan paling terlihat dengan program sebelumnya adalah adanya tambahan sebuah variable \$row dimana fungsi dari variable ini adalah untuk menampung data hasil ekstrak dari variable \$result (**penting!!!**, jadi variable \$result yang sebelumnya kita pelajari hanya menampung data secara gelondongan, untuk menampilkan data secara individu kita perlu mengekstraknya salah satunya dengan fungsi fetch_row() diatas.)

perintah **\$row = \$result->fetch_row()**, artinya: "ambil data yang tersimpan di dalam \$result sebagai normal array, kemudian simpan ke variabel \$row".

variabel \$row ini saya tampilkan dengan perintah **print_r(\$row)**. Hasilnya terlihat bahwa \$row berisi sebuah **"array normal"**, maksudnya index dari array ini menggunakan nomor, dimana **\$row[0] berisi "1"**, **\$row[1] berisi "TV Samsung 43NU7090 4K"**, dst.

Program diatas data yang ditampilkan masih dalam bentuk mentah array, lalu bagaimana klo kita ingin menampilkan datanya sesuai dengan keinginan kita.

Berikut contoh programnya, simpan dengan nama **select5.php**

```
try {  
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");  
  
    // Tampilkan semua data di tabel barang  
    $query = "SELECT * FROM barang";  
    $result = $mysqli->query($query);  
  
    $row = $result->fetch_row();  
    echo $row[0]; echo " | ";  
    echo $row[1]; echo " | ";  
    echo $row[2]; echo " | ";  
    echo $row[3]; echo " | ";
```

```
echo $row[4];

$result->free();
}
```

Hasil kode program:

1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-04 16:59:53

Di sini isi dari array `$row` saya tampilkan secara langsung, dimana index array berurutan sesuai nomor kolom dari tabel **barang**. Ingat ya, hasil `fetch_row` berupa array, jadi cara aksesnya dengan memanggil index dari array tersebut.

`echo $row[0];` ini adalah cara mengakses data kolom pertama dari data barang pertama. Jika ingin menampilkan merek barang maka bisa digunakan `echo $row[1];` karena merek barang berada pada variable kedua pada tabel barang.

PAHAM YA, JADI URUTANNYA ADALAH

`$result = $mysqli->query($query);` UNTUK MENAMPUNG HASIL QUERY MENTAH.

LALU

`$row = $result->fetch_row();` UNTUK MENGEKSTRAK DATA DARI HASIL QUERY SEBELUMNYA, SETELAH ITU TINGGAL PROSES MENAMPILKAN DATA SESUAI DENGAN YANG DIINGINKAN.

PERTANYAAN SELANJUTNYA, KOK DATA YANG DITAMPILKAN HANYA 1 KAN DIDALAM TABEL ADA 5, SEDANG QUERI YANG KITA LAKUKAN ADALAH MENGAMBIL SELURUH DATA????

JAWABANNYA ADALAH Method `fetch_row()` hanya memberikan hasil untuk 1 baris saja pada setiap pemanggilan. Jika saya ingin menampilkan data baris kedua, method `fetch_row()` ini harus dijalankan lagi.

Sehingga cara yang paling pas untuk menampilkan semua data adalah dengan membuat sebuah perulangan `while`, seperti contoh berikut simpan dengan nama **select6.php**:

```
try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);
```

```

while ($row = $result->fetch_row()){
    echo $row[0]; echo " | ";
    echo $row[1]; echo " | ";
    echo $row[2]; echo " | ";
    echo $row[3]; echo " | ";
    echo $row[4];
    echo "<br>";
}

$result->free();
}

```

1

Hasil kode program:

```

1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-04 16:59:53
2 | Kulkas LG GC-A432HLHU | 10 | 7600000 | 2019-01-04 16:59:53
3 | Laptop ASUS ROG GL503GE | 7 | 16200000 | 2019-01-04 16:59:53
4 | Printer Epson L220 | 14 | 2099000 | 2019-01-04 16:59:53
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-04 16:59:53

```

Perulangan while akan terus dijalankan untuk setiap baris data yang ada didalam tabel **barang**. Karena ada 5 data maka perulangan akan terjadi sebanyak 5 kali, artinya perintah `fetch_row()` akan dilakukan sebanyak 5 kali.

OK PAHAM YA SEMUA????

OK kita telah mengetahui bahwa `fetch_row()` menghasilkan sebuah data dengan sifat array biasa, sekarang kita akan mencoba cara selanjutnya yaitu `fetch_assoc()`, pada prinsipnya mempunyai fungsi yang sama dengan `fetch_row()` yaitu untuk ekstrak data, tapi bedanya adalah jenis data yang dihasilkan adalah dalam bentuk **associative array**.

Associative array adalah sebutan untuk array dengan *key* yang bisa berisi karakter non-angka, seperti huruf. Simpan program berikut dengan nama **select7.php**

```

try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);

    while ($row = $result->fetch_assoc()){

```

```

    echo $row['id_barang'];    echo " | ";
    echo $row['nama_barang'];  echo " | ";
    echo $row['jumlah_barang']; echo " | ";
    echo $row['harga_barang']; echo " | ";
    echo $row['tanggal_update'];
    echo "<br>";
}

$result->free();
}

```

Hasil kode program:

```

1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-04 16:59:53
2 | Kulkas LG GC-A432HLHU | 10 | 7600000 | 2019-01-04 16:59:53
3 | Laptop ASUS ROG GL503GE | 7 | 16200000 | 2019-01-04 16:59:53
4 | Printer Epson L220 | 14 | 2099000 | 2019-01-04 16:59:53
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-04 16:59:53

```

OK LANGSUNG SAJA, PERBEDAAN DENGAN PROGRAM SEBELUMNYA ADALAH PADA BAGIAN CARA MENGAKSES PROPERTY/ATTRIBUT. Klo sebelumnya kita akses attribute dengan index array biasa misal \$row[0], dengan associative array kita akan ganti dengan \$row['id_barang']; dan seterusnya, paham ya???? Tidak ada perbedaan yang terlalu jauh dengan cara sebelumnya.

Dan pada saat implementasi kalian boleh bebas menentukan menggunakan cara yang mana.

Nah dibawah ini cara penulisan lain dari fetch_assoc().

```

try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);

    while ($row = $result->fetch_array(MYSQLI_ASSOC)){
        echo $row['id_barang'];    echo " | ";
        echo $row['nama_barang'];  echo " | ";
        echo $row['jumlah_barang']; echo " | ";
        echo $row['harga_barang']; echo " | ";
        echo $row['tanggal_update'];
    }
}

```



```

    echo "<br>";
}

$result->free();
}

```

PAHAM YA SEMUA???

OK CARA TERAKHIR DALAM MENGAKSES DATA HASIL QUERY

Cara ketiga untuk menampilkan data dari **mysqli_result** adalah sebagai **object**. Maksudnya, variabel `$row` akan berisi object dengan nama kolom tabel sebagai property. Untuk keperluan ini, kita memakai method **mysqli_result::fetch_object**. Berikut contoh penggunaannya, SIMPAN dengan nama **select8.php**:

PENTING!!! INI AKAN SERING KITA GUNAKAN PADA MATERI SELANJUTNYA, JADI TOLONG PAHAMI

```

try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Tampilkan semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);

    while ($row = $result->fetch_object()){
        echo $row->id_barang;    echo " | ";
        echo $row->nama_barang;  echo " | ";
        echo $row->jumlah_barang; echo " | ";
        echo $row->harga_barang; echo " | ";
        echo $row->tanggal_update;
        echo "<br>";
    }

    $result->free();
}

```

Hasil kode program TETAP SAMA DENGAN PROGRAM SEBELUMNYA:

```
1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-04 16:59:53
2 | Kulkas LG GC-A432HLHU | 10 | 7600000 | 2019-01-04 16:59:53
3 | Laptop ASUS ROG GL503GE | 7 | 16200000 | 2019-01-04 16:59:53
4 | Printer Epson L220 | 14 | 2099000 | 2019-01-04 16:59:53
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-04 16:59:53
```

Seperti sebelumnya perbedaan hanya terjadi pada cara ekstrak data hasil query, kali ini kita menggunakan fungsi `fetch_object()` dimana dengan fungsi ini kita akan mendapatkan hasil ekstrak data berupa object. Dan tentunya saat mengakses data tersebut kita akan memperlakukannya seperti kita mengakses property sebuah object dari class.

Perhatikan baris dibawah ini

```
echo $row->id_barang;
```

itu adalah cara mengakses nilai dari property sebuah object, nah object disini adalah baris data dari tabel yang merupakan hasil dari query.

PAHAM YA. LOGIKA NYA APABILA HASIL QUERY SELECT MENGHASILKAN 5 BUAH DATA, ARTINYA KITA MEMILIKI 5 BUAH OBJECT YANG BISA DITAMPILKAN, MAKANYA DISINI DIGUNAKAN PERULANGAN WHILE KARENA OBJECT YANG DIHASILKAN BISA LEBIH DARI SATU.

Mengambil Data Tabel Dari `mysqli_result` Object

Dalam beberapa situasi, yang kita butuhkan adalah mengambil semua data tabel yang tersimpan di dalam `mysqli_result`, bukan langsung menampilkannya. Untuk keperluan ini tersedia method `mysqli_result::fetch_all()`. Berbeda dengan method `fetch_row()` atau `fetch_assoc()` yang mengambil **baris per baris**, method `fetch_all()` mengambil **seluruh** hasil query SELECT dan menyimpannya sebagai array 2 dimensi.

Simpan program berikut dengan nama **select9.php**

```
try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Ambil semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);

    $arr = $result->fetch_all();
    echo "<pre>";
    print_r($arr);
    echo "</pre>";

    $result->free();
}
```

Perbedaan dari program yang sebelumnya hanya pada fungsi `fetch_all()` yang menyimpan hasilnya pada variable `$arr` (nama variable ini boleh bebas ya!!!).

Lalu bedanya dimana???? Yup dengan fungsi `fetch_all()` apabila kita lihat detail isi dari variable `$arr` hasilnya sebagai berikut:

```
Array
(
    [0] => Array
        (
            [0] => 1
            [1] => TV Samsung 43NU7090 4K
            [2] => 5
            [3] => 5399000
            [4] => 2019-01-04 16:59:53
        )
    [1] => Array
        (
            [0] => 2
            [1] => Kulkas LG GC-A432HLHU
            [2] => 10
            [3] => 7600000
            [4] => 2019-01-04 16:59:53
        )
    ...
    ...
)
```

Menampung seluruh data dalam bentuk array 2 dimensi, tugas kita selanjutnya hanya tinggal menampilkan data tersebut sesuai dengan yang kita inginkan.

Cara dibawah akan lebih sering kita gunakan menggunakan fungsi `fetch_all(MYSQLI_ASSOC)`, dengan fungsi ini data yang dihasilkan dalam bentuk associative array dimana dengan bentuk ini kita akan lebih mudah dalam mengakses datanya.

Simpan program berikut dengan nama **select10.php**

```
try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Ambil semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);

    $arr = $result->fetch_all(MYSQLI_ASSOC);
    echo "<pre>";
    print_r($arr);
}
```

```
echo "</pre>";

$result->free();
}
```

Hasil kode program (bandingkan dengan hasil program sebelumnya, lihat perbedaannya di mana):

```
Array
(
    [0] => Array
        (
            [id_barang] => 1
            [nama_barang] => TV Samsung 43NU7090 4K
            [jumlah_barang] => 5
            [harga_barang] => 5399000
            [tanggal_update] => 2019-01-04 16:59:53
        )
    [1] => Array
        (
            [id_barang] => 2
            [nama_barang] => Kulkas LG GC-A432HLHU
            [jumlah_barang] => 10
            [harga_barang] => 7600000
            [tanggal_update] => 2019-01-04 16:59:53
        )
    ...
    ...
)
```

Perbedaan dengan kode program sebelumnya hanya pada perintah **`$arr = $result->fetch_all(MYSQLI_ASSOC)`**. Seperti yang terlihat, variabel `$arr` berisi *associative array* dimana index atau key array menggunakan nama kolom tabel. Teknik menyimpan keseluruhan hasil query ke dalam sebuah variabel ini sangat berguna, karena kita bisa memisahkan kode PHP yang dipakai untuk mengakses database dengan kode PHP yang nantinya dipakai untuk menampilkan data tersebut. Ini umum dijumpai dalam framework yang menerapkan prinsip MVC (Model-View Controller), dimana kode program yang mengakses database, terpisah dengan kode program yang akan dipakai untuk menampilkan data tersebut. Kita akan lihat bagaimana cara menampilkan data yang ditampung pada halaman HTML sehingga tampilan bisa kita sesuaikan dengan yang kita inginkan.

Simpan dengan nama **select10.php**

```
<?php
mysqli_report(MYSQLI_REPORT_STRICT);
try {
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");

    // Ambil semua data di tabel barang
    $query = "SELECT * FROM barang";
    $result = $mysqli->query($query);
    $arr = $result->fetch_all(MYSQLI_ASSOC);
    $result->free();
}
catch (Exception $e) {
    echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
    if (isset($mysqli)) {
        $mysqli->close(); }
}
?>

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Menampilkan Tabel MySQL</title>
</head>
<body>
<table>
    <tr>
        <th>ID</th>
        <th>Nama</th>
        <th>Stok</th>
        <th>Harga</th>
        <th>Update</th>
    </tr>
```

```

<?php foreach ($arr as $key => $val) {?> //bagian ini paling penting!!!
<tr>
  <td><?php echo $val['id_barang']; ?></td>
  <td><?php echo $val['nama_barang']; ?></td>
  <td><?php echo $val['jumlah_barang']; ?></td>
  <td><?php echo $val['harga_barang']; ?></td>
  <td><?php echo $val['tanggal_update']; ?></td>
</tr>
<?php } ?>
</table>
</body>
</html>

```

Output:

ID	Nama	Stok	Harga	Update
1	TV Samsung 43NU7090 4K	5	5399000	2019-01-04 16:59:53
2	Kulkas LG GC-A432HLHU	10	7600000	2019-01-04 16:59:53
3	Laptop ASUS ROG GL503GE	7	16200000	2019-01-04 16:59:53
4	Printer Epson L220	14	2099000	2019-01-04 16:59:53
5	Smartphone Xiaomi Pocophone F1	25	4750000	2019-01-04 16:59:53

Program diatas kita menggabungkan kode HTML dan PHP dalam satu dokumen secara langsung, tulis program dengan cermat dan teliti.

Bagian utama yang perlu dipahami adalah cara mengambil data dari proses PHP untuk ditampilkan pada halaman HTML. Fungsi foreach digunakan khusus untuk proses looping tipe data array.

```
<?php foreach ($arr as $key => $val)
```

Maksud dari baris diatas adalah, lakukan looping untuk setiap data hasil query yang disimpan dalam variable \$arr, tetapi disini kita perlu membuat variable pembantu yaitu \$val (nama variable boleh bebas).

Dan cara menampilkan data cukup dengan memanggil nama attribute yang diinginkan

```
echo $val['id_barang']; // bila ingin menampilkan attribute id_barang.
```

Menghapus Data Tabel

Mysqli_result object hanya dipakai untuk proses menampilkan hasil tabel (query SELECT). Untuk menghapus data tabel, kita tinggal menjalankan query DELETE dari method `mysqli::query()` seperti contoh berikut simpan dengan nama **delete1.php**:

```
try {  
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");  
  
    // Tampilkan semua data di tabel barang  
    $query = "DELETE FROM barang WHERE id_barang = 1";  
    $mysqli->query($query);  
    echo "Terdapat ".$mysqli->affected_rows." baris yang telah dihapus";  
}
```

Setelah program dijalankan, panggil program **select10.php** untuk memastikan data telah terhapus, atau bisa juga dilihat langsung melalui database server mysql.

Contoh program delete selanjutnya, simpan dengan nama **delete2.php**

```
try {  
    $mysqli = new mysqli("localhost", "root", "", "ilkoom");  
  
    // Tampilkan semua data di tabel barang  
    $query = "DELETE FROM barang WHERE id_barang = 4 OR id_barang = 5";  
    $mysqli->query($query);  
    echo "Terdapat ".$mysqli->affected_rows." baris yang telah dihapus <br>";  
}
```

TO BE CONTINUE...