

PHP dan MySQL [ref >>](#)`mysqli_connect(server, user, pass, db_name)`

Menghubungkan php dengan mysql:

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'db_name');

if (!$conn) {
    die (MySQL Error: ' . mysqli_connect_error());
}
```

`mysqli_query (sql)`

Fungsi untuk menjalankan query MySQL. Fungsi ini harus dijalankan setiap kali menjalankan query, baik SELECT, INSERT, UPDATE maupun DELETE

```
$result = mysqli_query('SELECT * FROM tabel');
```

PENTING: jika data tidak ditemukan, variabel \$result akan bernilai false

`mysqli_fetch_array (result)`

Argumen result merupakan hasil query, salah satunya hasil dari fungsi `mysqli_query`

```
while ($row = mysqli_fetch_array($result)) {
    echo $row[col_name'];
}
```

`mysqli_fetch_all (result, resulttype)`

Fungsi akan mengubah hasil query menjadi bentuk array. nilai argumen resulttype: MYSQLI_ASSOC, MYSQLI_NUM, dan MYSQLI_BOTH

```
mysqli_fetch_all ($result, MYSQLI_ASSOC);
```

WARNING: pemanggilan `mysqli_fetch_xxx` lebih dari sekali akan menghasilkan data kosong

`mysqli_affected_rows ($conn)`

Digunakan untuk mendapatkan jumlah baris yang terpengaruh pada query sebelumnya. Dapat digunakan untuk statemen SELECT, INSERT, UPDATE, dan DELETE

```
$result = mysqli_query($conn, 'SELECT ...');
$affected_rows = mysqli_affected_rows($conn);
```

WARNING: tes dengan `if($affected_rows > 0)` bukan `if($affected_rows)` karena ketika gagal insert atau update data, hasilnya -1 bukan 0

`mysqli_num_rows (result)`

Mendapatkan banyaknya baris hasil query

```
$result = mysqli_query($conn, 'SELECT ...');
$row_count = mysqli_num_rows($result);
```

`mysqli_free_result (result)`

Menghapus hasil query dari memory (RAM), tidak wajib digunakan, gunakan hanya jika hasil query sangat besar. PHP akan otomatis menghapus hasil query pada akhir eksekusi script

```
$result = mysqli_query($conn, 'SELECT ...');
mysqli_free_result($result);
```

`mysqli_close ($conn)`

Menutup koneksi, tidak wajib digunakan. PHP otomatis menutup semua koneksi pada akhir eksekusi script

```
mysqli_close ($conn);
```

`mysqli_insert_id ($conn)`

Mendapatkan nilai terakhir dari auto_increment

```
mysqli_query($conn, 'INSERT INTO ... ');
$last_id = mysqli_insert_id($conn);
```

`mysqli_escape_string ($conn, string)`

Sama dengan `mysqli_real_escape_string (string)` digunakan untuk men-escape tanda kutip

```
$str = "Hari Jum'at";
$str = mysqli_escape_string ($conn, $str);
mysqli_query($conn, 'INSERT INTO tabel VALUES ('.$str.')');
```

`mysqli_connect_error ($conn)`

Untuk mendapatkan pesan error jika koneksi gagal

`mysqli_error ($conn)`

Untuk mendapatkan pesan error pada query terakhir

Transaction [ref >>](#)`mysqli_begin_transaction ($conn)`

Memulai transaksi:

```
mysqli_begin_transaction($conn);
```

`mysqli_commit ($conn) | mysqli_rollback ($conn)`

Commit untuk menerapkan perubahan secara permanen), Rollback untuk membatalkan perubahan

`mysqli_savepoint ($conn, string)`

Membuat savepoint pada transaksi

```
mysqli_savepoint($conn, 'perubahan_1');
```

EXAMPLES

Example 1: SELECT

Contoh script untuk mengambil data

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'toko_buku');

if (!$conn) {
    die ('MySQL ERROR: ' .
    mysqli_connect_error());
}

$result = mysqli_query($conn, 'SELECT * FROM
buku');

if ($result) {

    // Jumlah Data
    $row_count = mysqli_num_rows($result);
    echo 'Jumlah data ' . $row_count;

    // Fetch rows
    while ($row = mysqli_fetch_array($result)) {
        echo $row['judul'];
    }

    // Fetch all
    $all_data = mysqli_fetch_all($result,
                                MYSQLI_ASSOC);

    echo '<pre>';
    print_r($all_data);
    echo '</pre>';
}
mysqli_free_result($result);
mysqli_close($conn);
```

EXAMPLE 2: INSERT

Contoh script untuk memasukkan data

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'toko_buku');

if (!$conn) {
    die ('MySQL ERROR: ' . mysqli_connect_error());
}

$val = 'slash"s';
$val = mysqli_escape_string($conn, $val);
$result = mysqli_query($conn, 'INSERT INTO buku
(judul) VALUES ("'.$val.'")');

if ($result) {
    echo 'Data berhasil dimasukkan sebanyak ' .
    mysqli_affected_rows($conn) . ' baris,
    id_buku: ' . mysqli_insert_id($conn);
} else {
    echo 'Data gagal dimasukkan';
}
```

EXAMPLE 2: UPDATE

Pada contoh berikut, meskipun query berhasil, banyaknya data yang diupdate (mysqli_affected_rows) adalah 0, karena buku dengan id 1 sudah memiliki judul PHP

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'toko_buku');
if (!$conn) {
    die ('MySQL ERROR: ' . mysqli_connect_error());
}

$result = mysqli_query($conn, 'UPDATE buku SET
judul="PHP" WHERE id_buku = 1');

if ($query) {
    echo 'Data berhasil diupdate sebanyak ' .
    mysqli_affected_rows($conn) . ' baris';
} else {
    echo 'Data gagal diupdate';
}
```

EXAMPLE 3: TRANSACTION

Contoh transaksi dengan PHP

```
$conn = @mysqli_connect('localhost', 'root', '',
                        'toko_buku');

if (!$conn) {
    die ('MySQL ERROR: ' . mysqli_connect_error());
}

// Mulai Transaksi
mysqli_begin_transaction($conn);

// Query 1
$error = '';
$result = mysqli_query($conn, 'INSERT INTO buku
(judul) VALUES ("PHP")');
if (!$result) {
    $error .= 'Query 1: ' . mysqli_error($conn) . ', ';
}

// Query 2
$result = mysqli_query($conn, 'INSERT INTO buku
(judul) VALUES ("MySQL")');
if (!$result) {
    $error .= 'Query 2: ' . mysqli_error($conn) . ', ';
}

// Commit - Rollback
if ($error) {
    mysqli_rollback($conn);
    echo 'MySQL ERROR: ' . rtrim($error, ', ');
} else {
    mysqli_commit($conn);
    echo 'Data berhasil dimasukkan';
}
```