**Pertemuan 6  
Database**

1. **Koneksi ke database**

Untuk melakukan koneksi ke database, file database.php di application/config/ harus dirubah sesuai dengan kebutuhan kita, Di file database.php, ubah:

$db['default']['hostname'] = "localhost";

$db['default']['username'] = "";

$db['default']['password'] = "password";

$db['default']['database'] = "latihanCI";

$db['default']['dbdriver'] = "mysqli";

ubah file application/config/autoload.php dan tambahkan ‘database’ pada saat autoload libraries, supaya library database dapat langsung digunakan.

$autoload[‘libraries’] = array('database');

1. **Non Active Record**

Kita menggunakan syntax SQL biasa yang kita gunakan, dan kemudian di-execute dengan command Codeigniter. Command Codeigniter yang digunakan adalah :

$this->db->query();

Untuk menjalankan query, gunakan fungsi berikut:

$sql = “SELECT \* FROM some\_table”;

$this->db->query(‘SQL Command’);

**Query Bindings**

Digunakan untuk menyederhanakan query dengan membiarkan sistem yang melakukannya.

$sql = "SELECT \* FROM some\_table WHERE field1 = ? AND field2 = ? ";

$this->db->query($sql, array(13, 'terdaftar’ ));

1. **Active Record**

Codeigniter menciptakan versi pola Active Record, yang memungkinkan informasi untuk diambil, dimasukkan, atau diupdate ke dalam database dengan menggunakan script yang minimal. Dengan Active Record ini, kita dapat membuat aplikasi yang independen terhadap database, karena syntax query digenerate dari database adapter yang disiapkan oleh Codeigniter.

Menjalankan Query

$this->db->select(‘\*’);

$this->db->from(‘some\_table’);

$this->db->get();

atau hanya

$this->db->get(‘some\_table’);

hasilnya akan sama dengan metode Non-Active Record.

Ada banyak syntax Active Record yang disediakan oleh Codeigniter, misalnya:

$this->db->select();

$this->db->from();

$this->db->where();

$this->db->join();

$this->db->select\_max();

$this->db->select\_avg();

$this->db->select\_sum();

$this->db->orderby();

**Komparasi Non-Active Record dan Active Record :**

Misalkan perintah SQL :

Select field1, field2, field3 from some\_table

Active Record di CI bias digantikan dengan perintah :

$this->db->select(‘field1, field2, field3’);

$this->db->from(‘some\_table’);

Misalkan perintah SQL :

Select field1, field2, sum(field3)

from some\_table

group by field1, field2

Active Record di CI bias digantikan dengan perintah:

$this->db->select(‘field1, field2, sum(field3)’);

$this->db->from(‘some\_table’);

$this->db->group\_by(‘field1, field2’);

Misalkan perintah SQL :

Select a.field1, field2, sum(field3)

from some\_table a

inner join some\_table\_lain b

on a.field1 = b.field1

group by field1, field2

Active Record di CI bias digantikan dengan perintah:

$this->db->select(‘field1, field2, sum(field3)’);

$this->db->from(‘some\_table a’);

$this->db->join(‘some\_table\_lain b’, ‘a.field1 = b.field1, ‘inner’);

$this->db->group\_by(‘field1, field2’);

**Metode Pengambilan hasil query**

Dengan perintah result(), mengembalikan hasil query sebagai array dari object atau array kosong apabila gagal. Metode yang digunakan foreach loop:

$query = $this->db->query("YOUR QUERY");

foreach ($query->result() as $row)

{

echo $row->title;

echo $row->name;

echo $row->body;

}

Dengan perintah result\_array(), mengembalikan hasil query sebagai array murni. Metode yang digunakan foreach loop :

$query = $this->db->query("YOUR QUERY");

foreach ($query->result\_array() as $row)

{

echo $row['title'];

echo $row['name'];

}

Dengan perintah row(), mengembalikan satu baris hasil sebagai object, meskipun hasil dari query lebih dari satu row, yang dikembalikan adalah row pertama hasil query.

$query = $this->db->query("YOUR QUERY");

if ($query->num\_rows() > 0)

{

$row = $query->row();

echo $row->title;

echo $row->name;

}

Dengan perintah row\_array() sama dengan row, tetapi yang dikembalikan bukan object, melainkan array murni:

$query = $this->db->query("YOUR QUERY");

if ($query->num\_rows() > 0)

{

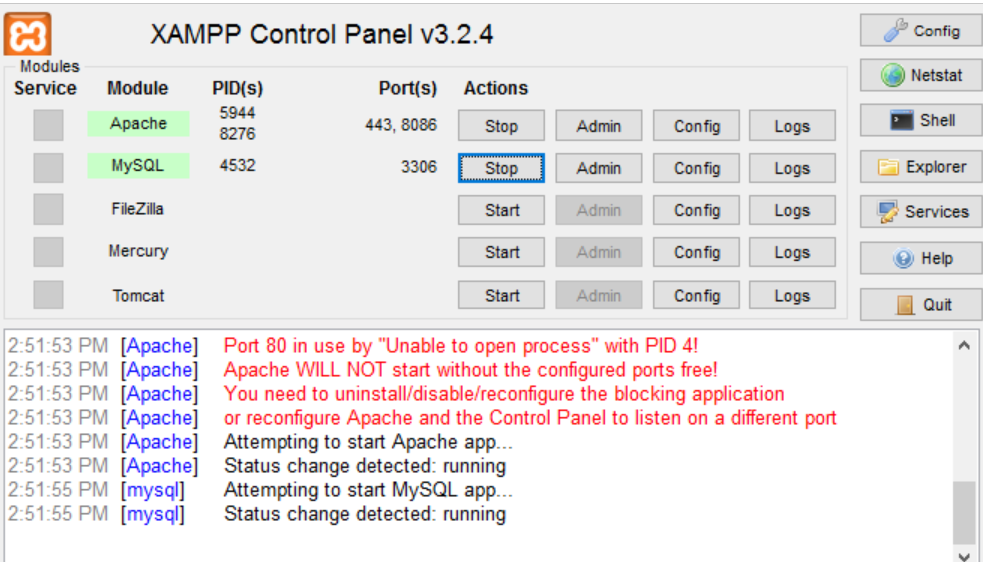
$row = $query->row\_array();

echo $row['title'];

echo $row['name'];

}

1. **Latihan.**
2. Pastikan database mysql sudah aktif di xampp. Lihat tampilan di bawah ini:



1. Buka aplikasi phpmyadmin di browser, selanjutnya buat database dan table-tabel sebagai berikut:

Nama Database: **latihanCI**

Table-tabel:

1. Nama table: kitab

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **field** | **Data type** | **size** | **key** |
| 1 | id | integer |  | PK |
| 2 | kategori | varchar | 20 | FK |
| 3 | judul | varchar | 200 |  |
| 4 | pengarang | varchar | 100 |  |
| 5 | harga | double |  |  |

1. Nama Tabel: Kategoris

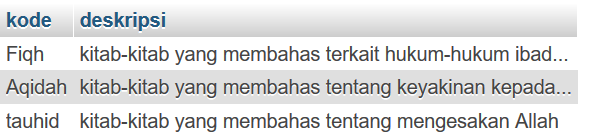
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **field** | **Data type** | **size** | **key** |
| 1 | kode | varchar | 20 | PK |
| 2 | deskripsi | varchar | 200 |  |

1. Selanjutnya masukkan data sebagai berikut:

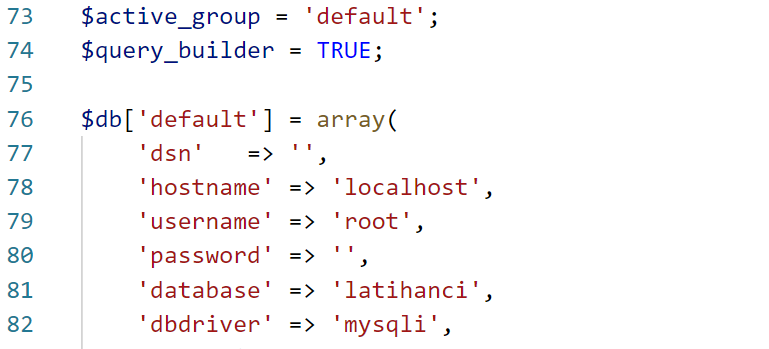
Data table kitab:



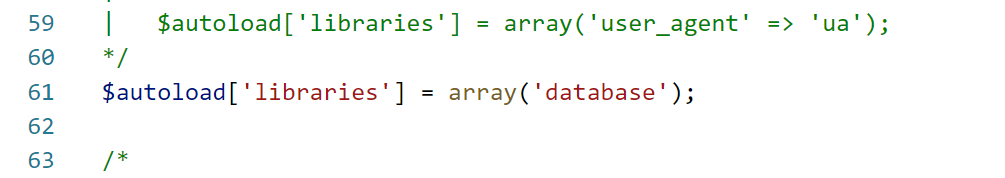
Data table kategoris



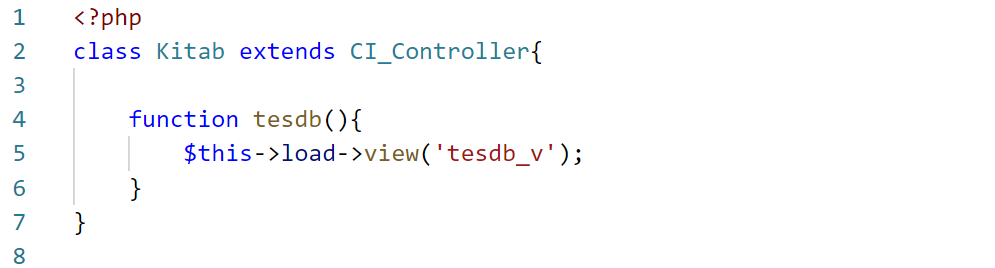
1. Kemudian bukalah file database.php di application/config/ ubah seperti gambar berikut ini:



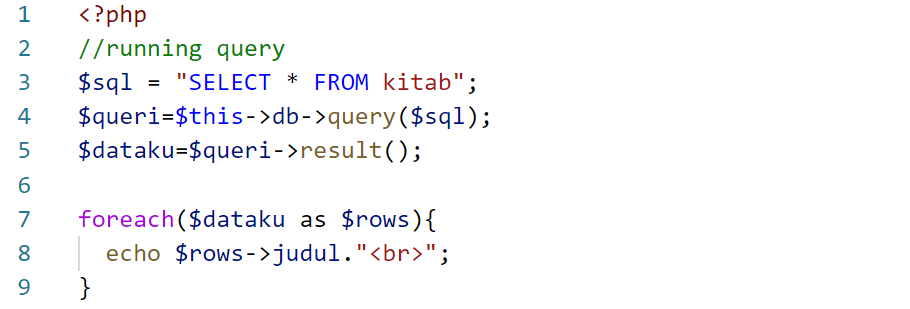
1. ubah file application/config/autoload.php dan tambahkan ‘database’ pada array autoload [libraries]



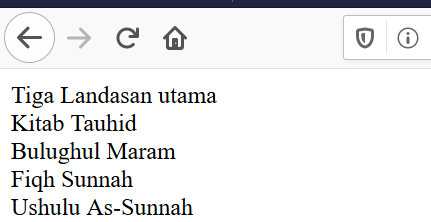
1. untuk memudahkan memahami perintah-perintah akses database maka perintagnya kita letakkan di views. Akan tetapi aplikasi sebenarnya perintah ini akan kita letakkan di models.   
   Selanjutnya buatlah file baru di controllers dengan nama **kitab.php**



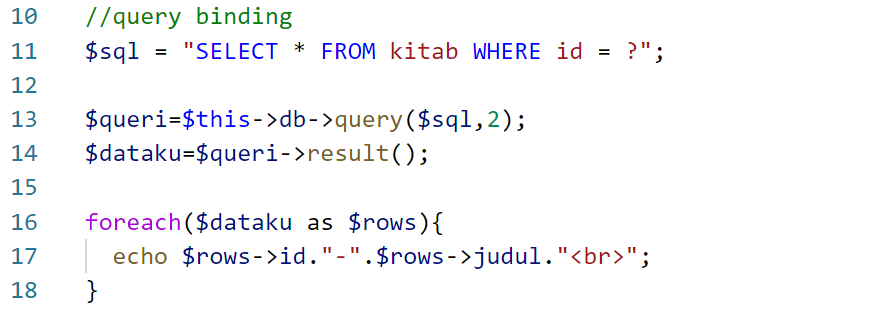
1. selanjutnya buatlah file baru di views dengan nama **tesdb\_v.php**



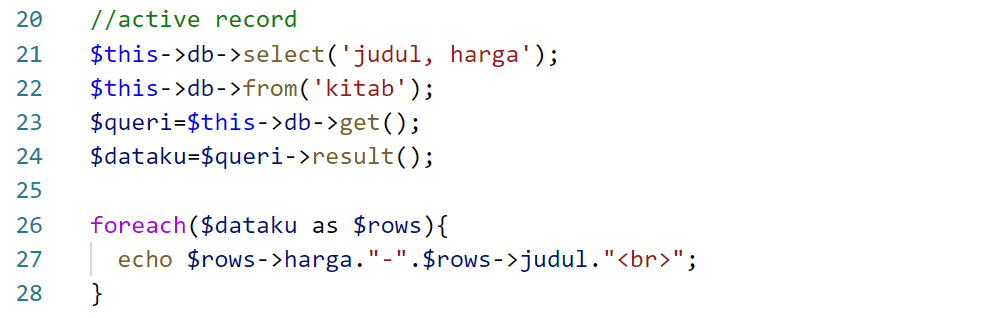
Hasilnya sebagai berikut:



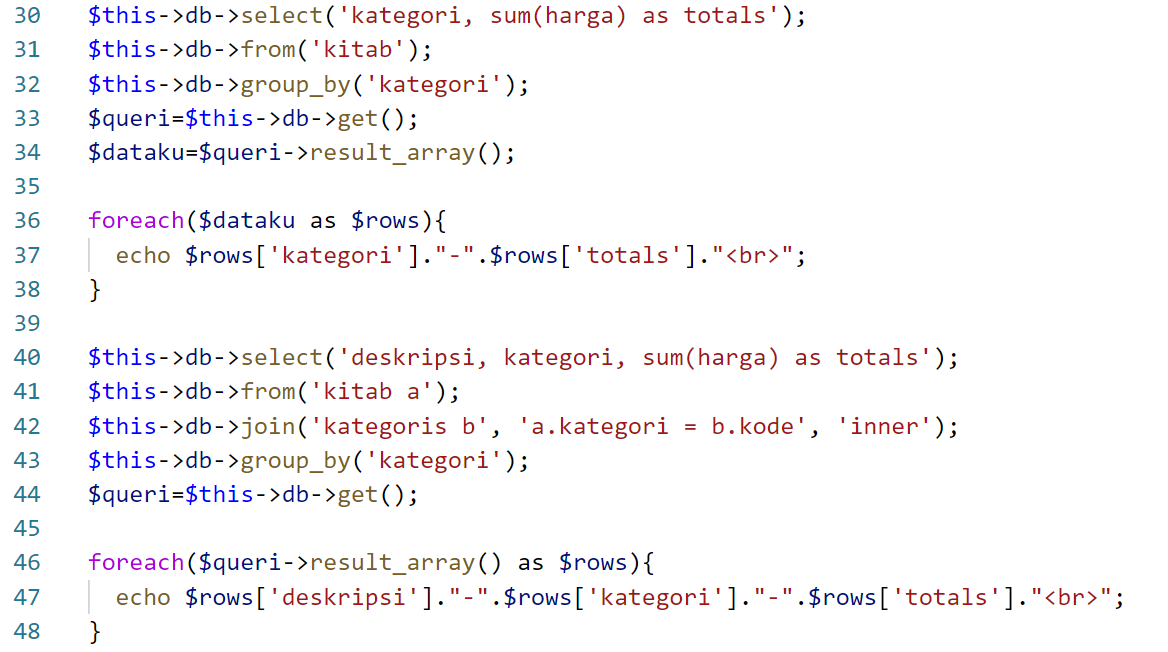
1. selanjutnya tambahkan script berikut untuk mencoba query binding



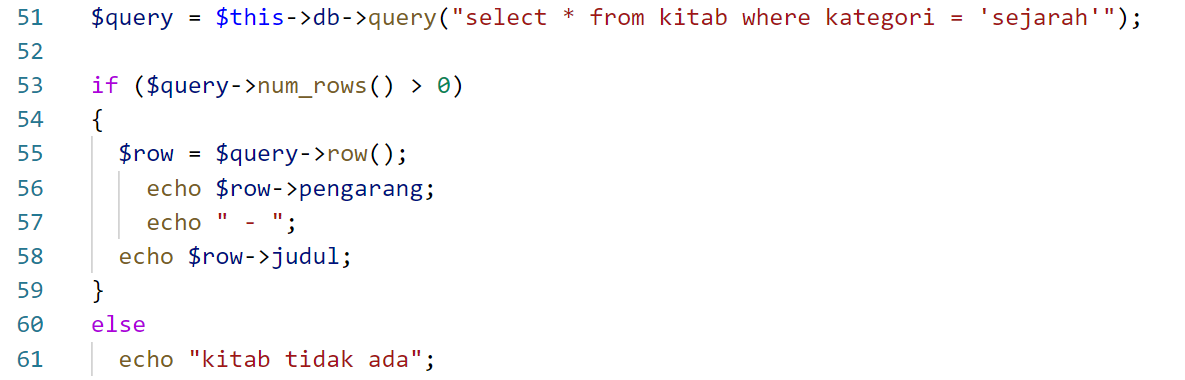
1. selanjutnya tambahkan script berikut untuk mencoba active record



1. selanjutnya tambahkan script berikut untuk mencoba active record grup\_by dengan result dalam bentuk array.



1. Selanjutnya tambahkan script dengan menggunakan rows()



1. **Soal Latihan**

Buatlah database dengan nama **Herbal,** tambahkan table-tabel berikut:

1. Nama table: produk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **field** | **Data type** | **size** | **key** |
| 1 | Kode\_produk | char | 6 | PK |
| 2 | id\_kat | integer |  | FK |
| 3 | Nama\_produk | varchar | 200 |  |
| 4 | deskripsi | varchar | 100 |  |
| 5 | harga | double |  |  |

1. Nama table: kategori

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **field** | **Data type** | **size** | **key** |
| 1 | Id\_kat | int |  | PK |
| 2 | Nama\_kat | varchar | 100 |  |

1. Masukkan data sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode\_produk** | **Id\_kat** | **Nama\_produk** | **deskripsi** | **harga** |
| MH001 | 1 | Madu Hutan Rayyan | Madu dengan khasiat mantaps | 70000 |
| MR001 | 1 | Madu Randu Ihsan | Madu dari pohon randu asli | 76000 |
| HS001 | 2 | Habba Khair | Habbatussaudah | 35000 |
| HM001 | 2 | HAbba Madu | Haba campur madu | 40000 |
| HM002 | 2 | Haba MAdu Anak | HAba plus madu buat anak | 45000 |

1. Masukkan data berikut:

|  |  |
| --- | --- |
| **Id\_kat** | **Nama kat** |
| 1 | Madu |
| 2 | Habbatusaudah |

1. Kemudian buat aplikasi dengan CI untuk menampilkan output sebagai berikut:

**Produk Herbal**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode\_produk** | **kategori** | **Nama\_produk** | **Harga** |
| MH001 | Madu | Madu Hutan Rayyan | 70000 |
| MR001 | Madu | Madu Randu Ihsan | 76000 |
| HS001 | Habbatusaudah | Habba Khair | 35000 |
| HM001 | Habbatusaudah | HAbba Madu | 40000 |
| HM002 | Habbatusaudah | Haba MAdu Anak | 45000 |
| **TOTAL** | | | **266000** |

1. Dan buat juga tampilan sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| **kategori** | **Jumlah harga** | **Rata-rata** | **Jumlah produk** |
| Madu | 146000 | 73000 | 2 |
| Habbatusaudah | 120000 | 40000 | 3 |