

|  |
| --- |
| **Web Programming II** |
|  |
| **Modul 4** |
| **Koneksi Database dengan CodeIgniter**  **#Studi Kasus: Pustaka-Booking** |

**Disusun oleh:**

**Dwi Intan Af’idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TAHUN AJARAN 2020/2021**

Daftar Isi

[Daftar Isi ii](#_Toc53735251)

[1 Persiapan Membuat Aplikasi Pustaka-Booking 3](#_Toc53735252)

[2 Setting Base URL 5](#_Toc53735253)

[3 Merancang dan Membuat Database 1](#_Toc53735254)

[4 Membuat Koneksi Database CI dengan MySQL 3](#_Toc53735255)

[5 Membuat Model untuk CRUD 5](#_Toc53735256)

1. Persiapan Membuat Aplikasi Pustaka-Booking

Pada pertemuan ini mahasiswa diharapkan dapat memahami persiapan dasar untuk memulai sebuah projek kasus. Mahasiswa juga mampu memahami desain dan pembuatan database.

Pada tahap ini akan dimulai membuat sebuah project aplikasi berbasis web yang dibuat dengan menggunakan CodeIgniter.

Pustaka-booking merupakan aplikasi reservasi atau booking peminjaman buku pada sebuah perpustakaan yang dilakukan secara online dengan melibatkan anggota, admin dan aplikasi pustaka-booking.

Aplikasi pustaka booking adalah aplikasi berbasis web yang dibuat untuk booking sebuah buku untuk dipinjam. Tujuan dibuatkannya aplikasi ini agar memudahkan para pengguna dalam mencari buku dan ingin meminjam buku tersebut. Jadi sebelum di datang ke perpustakaan dan meminjam sebuah buku. Pengguna bisa mencari terlebih dahulu buku yang akan dipinjam, untuk memastikan bahwa di perpustakaan yang akan dia kunjungi benar bahwa buku tersebut tersedia.

Alur logika sistem booking pustaka ini yaitu seseorang yang ingin melakukan booking diharuskan mendaftar menjadi anggota terlebih dahulu, selanjutnya ketika sudah menjadi anggota, baru dapat melakukan booking terhadap buku yang akan dipinjam. Setelah melakukan booking, anggota diharuskan mengambil buku yang telah dibooking dengan cara datang langsung ke perpustakaan dalam waktu 1x24 jam. Kemudian konfirmasi ke petugas atau admin untuk diambilkan buku yang telah dibooking berdasarkan bukti booking melalui aplikasi.

Kebutuhan user:

1. Admin
   1. Seorang admin dapat login ke dalam aplikasi pustaka-booking
   2. Seorang admin dapat menambah, melihat, mengubah, dan menghapus, data buku pada aplikasi pustaka-booking.
   3. Seorang admin dapat melihat data booking dan melanjutkan proses booking sampai buku dipinjam.
   4. Seorang admin dapat memproses pengembalian buku yang sudah selesai dipinjam.
2. Pengunjung
   1. Pengunjung dapat melihat-lihat data buku yang ada pada palikasi pustaka-booking
   2. Pengunjung dapat melakukan registrasi untuk menjadi anggota pustaka-booking
   3. Pengunjung dapat memberikan komentar melalui buku tamu
3. Anggota
   1. Anggota dapat login ke dalam sistem aplikasi pustaka-booking.
   2. Anggota dapat melakukan booking terhadap buku yang ada pada aplikasi pustaka-booking
   3. Anggota dapat mencetak bukti booking buku untuk dibawa ketika akan mengambil buku yang dibooking.

Kebutuhan sistem:

1. Sistem akan melakukan validasi username dan password ketika ada seorang user yang melakukan login
2. Sistem juga melakukan validasi data buku dan anggota yang akan diinput ke dalam database
3. Sistem akan memblokir dan memberikan notif apabila ada seorang pengunjung yang meng-klik tombol booking tanpa melakukan login terlebih dahulu.
4. Sistem akan menghapus secara otomatis data booking yang sudah lewat dari 1 x 24 jam
5. Sistem dapat mengkalkulasi denda yang harus dibayarkan ketika ada anggota yang terlambat mengembalikan buku.
6. Sistem dapat menampilkan buku yang dicari oleh user berdasarkan kriteria pencarian.

Persiapan untuk memulai membangun aplikasi pustaka-booking terdiri dari instalasi codeigniter, pemilihan tool pendukung, setting base\_url, load librari, helper, dan segala yang dibutuhkan. Tetapi disini tidak akan membahas instalasi codeigniter karena sudah sangat mudah dijelaskan di pertemuan awal. Disini hanya kan membahas mulai dari seting base\_url dan seterusnya.

1. Setting Base URL

Pengaturan base url adalah pengaturan URL dasar dalam sebuah project aplikasi yang dibuat dengan menggunakan codeigniter. Pengaturan base url di codeigniter terdapat di dalam file config.php di dalam folder project codeigniter.

**pustaka-booking/application/config/config.php**

Membuka beberapa library dan helper codeigniter yang kita butuhkan dalam pembuatan aplikasi pustaka-booking ini. Untuk menjalankan atau membuka library dan helper secara otomatis di codeiginter, kita bisa mengaturnya dalam file autoload.php dalam folder *application/config/autoload.php*.

**Pustaka-booking/application/config/autoload.php**

Library yang akan kita load secara otomatis adalah *database, session dan* *form\_validation*.

Cari sintak

$autoload['libraries'] = array();

Kemudian tambahkan database, session dan form\_validation. Sehingga menjadi seperti berikut:

$autoload['libraries'] = array('form\_validation', 'session', 'database'

Dan helper yang akan kita load secara otomatis adalah *url*:

$autoload['helper'] = array();

Kemudian tambahkan *“url” (karena kita hanya membutuhkan helper url disini).* Maka akan menjadi seperti berikut.

$autoload['helper'] = array('url');

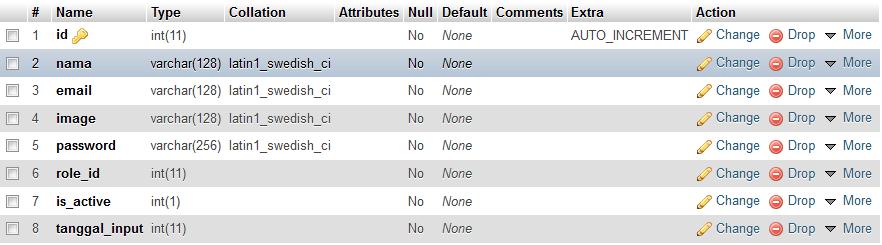
Proses autoload library dan helper selesai.

Selanjutnya kita akan membuat database untuk aplikasi pustaka-booking dan melakukan konfigurasi database pada codeigniter di bab selanjutnya.

1. Merancang dan Membuat Database

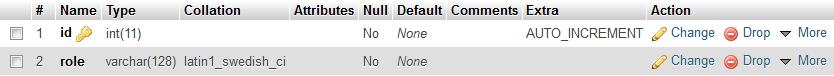
Buat sebuah database baru dengan nama “pustaka” kemudian buatlah table-table seperti berikut:

1. Tabel User (**user**)



1. Tabel Role (**role**)

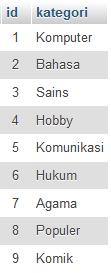
Isi tabel role



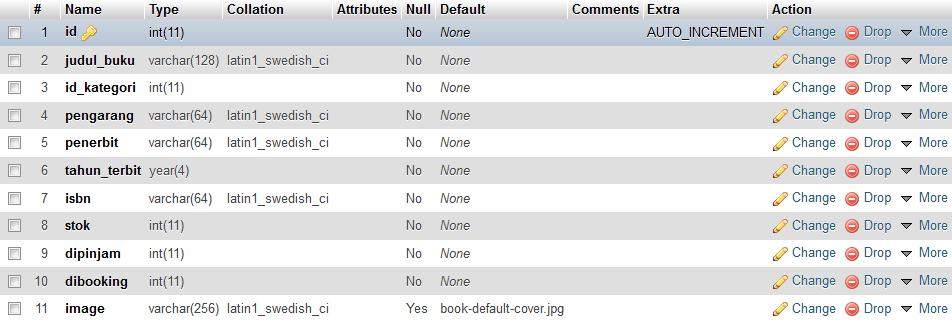
1. Tabel Kategori (**kategori**)

Isi tabel kategori





1. Tabel Buku (**buku**)



Isi tabel buku



1. Membuat Koneksi Database CI dengan MySQL

Selanjutnya untuk menghubungkan codeigniter dengan database buka file *config* *database* codeigniter yang terletak di**application/config/database.php**kemudian cari barissintak seperti berikut:

$active\_group = 'default';

|  |  |  |
| --- | --- | --- |
| $query\_builder | | = TRUE; |
| $db['default'] | | = array( |
| 'dsn' | => | '', |
| 'hostname' | | => 'localhost', |
| 'username' | | => '', |
| 'password' | | => '', |
| 'database' | | => '', |
| 'dbdriver' | | => 'mysqli', |
| 'dbprefix' | | => '', |
| 'pconnect' | | => FALSE, |
| 'db\_debug' | | => (ENVIRONMENT !== 'production'), |
| 'cache\_on' | | => FALSE, |
| 'cachedir' | | => '', |
| 'char\_set' | | => 'utf8', |
| 'dbcollat' | | => 'utf8\_general\_ci', |
| 'swap\_pre' | | => '', |
| 'encrypt' => FALSE, | | |
| 'compress' | | => FALSE, |
| 'stricton' | | => FALSE, |
| 'failover' | | => array(), |

'save\_queries' => TRUE

);

Kemudian masukkan username, password, dan nama database, untuk password jika tidak ada maka bisa dibiarkan kosong. Sehingga akan menjadi.

|  |  |  |
| --- | --- | --- |
| $db['default'] | | = array( |
| 'dsn' | => | '', |
| 'hostname' | | => 'localhost', |
| 'username' | | => 'root', |
| 'password' | | => '', |
| 'database' | | => 'pustaka', |
| 'dbdriver' | | => 'mysqli', |
| 'dbprefix' | | => '', |
| 'pconnect' | | => FALSE, |
| 'db\_debug' | | => (ENVIRONMENT !== 'production'), |
| 'cache\_on' | | => FALSE, |
| 'cachedir' | | => '', |
| 'char\_set' | | => 'utf8', |
| 'dbcollat' | | => 'utf8\_general\_ci', |
| 'swap\_pre' | | => '', |
| 'encrypt' => FALSE, | | |
| 'compress' | | => FALSE, |
| 'stricton' | | => FALSE, |
| 'failover' | | => array(), |

$active\_group = 'default';

$query\_builder = TRUE;

'save\_queries' => TRUE

);

1. Membuat Model untuk CRUD

Pada tahap ini akan dibuat fungsi-fungsi yang bisa gunakan berulang kali, tanpa harus repot banyak-banyak membuat fungsi pada model. contohnya untuk insert data ke database, hanya buat satu fungsi untuk insert data ke database. Dan bisa kita gunakan secara berulang-ulang kali. Begitu juga dengan fungsi edit, hapus dan update data pada database.

Buat 2 buah file model dengan nama **ModelUser.php** dan **ModelBuku.php**. **perlu diperhatikan untuk Penamaan Controller dan Model harus diawali dengan huruf besar di** codeigniter 3. Memang tidak masalah jika dibuat dengan nama kecil, tetapi akan menyalahi aturan penggunaan codeigniter yang benar dan akan terjadi trouble saat diupload dihosting.

**ModelUser.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelUser extends CI\_Model

{

public function simpanData($data = null)

{

$this->db->insert('user', $data);

}

public function cekData($where = null)

{

return $this->db->get\_where('user', $where);

}

public function getUserWhere($where = null)

{

return $this->db->get\_where('user', $where);

}

public function cekUserAccess($where = null)

{

$this->db->select('\*');

$this->db->from('access\_menu');

$this->db->where($where);

return $this->db->get();

}

public function getUserLimit()

{

$this->db->select('\*');

$this->db->from('user');

$this->db->limit(10, 0);

return $this->db->get();

}

}

**ModelBuku.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelBuku extends CI\_Model

{

//manajemen buku

public function getBuku()

{

return $this->db->get('buku');

}

public function bukuWhere($where)

{

return $this->db->get\_where('buku', $where);

}

public function simpanBuku($data = null)

{

$this->db->insert('buku',$data);

}

public function updateBuku($data = null, $where = null)

{

$this->db->update('buku', $data, $where);

}

public function hapusBuku($where = null)

{

$this->db->delete('buku', $where);

}

public function total($field, $where)

{

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

if(!empty($where) && count($where) > 0){ $this->db->where($where);

}

$this->db->from('buku');

return $this->db->get()->row($field);

}

//manajemen kategori

public function getKategori()

{

return $this->db->get('kategori');

}

public function kategoriWhere($where)

{

return $this->db->get\_where('kategori', $where);

}

public function simpanKategori($data = null)

{

$this->db->insert('kategori', $data);

}

public function hapusKategori($where = null)

{

$this->db->delete('kategori', $where);

}

public function updateKategori($where = null, $data = null)

$this->db->update('kategori', $data, $where);

}

//join

public function joinKategoriBuku($where)

{

$this->db->select('buku.id\_kategori,kategori.kategori'); $this->db->from('buku');

$this->db->join('kategori','kategori.id = buku.id\_kategori');

$this->db->where($where);

return $this->db->get();

}

}

Dengan model seperti di atas, Untuk menginput data ke database kita bisa menggunakan function *simpan(),* untuk menampilkan data atau mengambil data dari database kita bisa menggunakan function *tampil(),* untuk menghapus data dari database kita bisa menggunakan function *hapus(),* dan untuk mengupdate data pada database kita bisa menggunakan function *ubah().*

Kemudian kita harus mengatur autoload model di application/config/autoload.php. agar model-model yang sudah dibuat bisa digunakan dan di load secara otomatis. Jadi tidak perlu lagi kita load dari controller.

**Application/config/autoload.php**

Cari baris sintak berikut:

$autoload['model'] = array();

Kemudian ubah menjadi

$autoload['model'] = array('ModelUser', 'ModelBuku');