



# BELAJAR PHP

DENGAN

## TEKNIK OOP & MVC

Hirzi Widyan Putra

## Informasi

- Pada Modul ini hanya mengajarkan point – point utama atau point penting dari MySQL & PHP
- Yang diulas yaitu step - step bagaimana membuat sebuah web dari 0 sampai CRUD, Login & membuat sebuah report atau laporan
- Pada saat pertemuan praktikum, sistem yang akan digunakan yaitu 80% Praktik langsung atau live coding & 20% Teori, penjelasan lebih lengkap ada di modul / bisa ditanyakan langsung

## Daftar Isi

<b>BAB I Perkenalan.....</b>	<b>4</b>
1.1 Aplikasi Pendukung .....	4
1.2 Apa itu HTML, CSS, JavaScript, PHP, dan MySQL? .....	4
1.2.1 HTML .....	4
1.2.2 PHP .....	4
1.2.3 CSS .....	4
1.2.4 JS (JavaScript) .....	5
1.2.5 MySQL .....	5
1.3 OOP .....	6
1.4 MVC pada PHP .....	6
<b>BAB II Praktek.....</b>	<b>6</b>
2.1 Mendalami MySQL.....	6
2.1.1 Membuka phpMyAdmin.....	7
2.1.2 Membuat Database .....	8
2.1.3 Membuat Tabel .....	9
2.1.4 Membuat relasi antar tabel .....	10
2.1.5 Menambahkan Data .....	12
2.1.6 Menampilkan Data.....	12
2.1.7 Order By .....	15
2.1.8 Pencarian Data .....	16
2.1.9 Join .....	18
2.1.10 Mengubah Data .....	20
2.1.11 Menghapus Record / Data .....	22
2.1.12 Operasi Aritmatik.....	24
2.1.13 Operasi Logika .....	24
2.1.14 Operasi Perbandingan.....	24
2.2 Membuat Database untuk Aplikasi Buku Sederhana .....	25
2.3 Membuat tampilan dengan Template AdminLTE 3 .....	26
2.3.1 Menampilkan Extension file .....	26
2.3.2 Menyesuaikan tampilan halaman utama .....	26
2.4 Persiapan Aplikasi PHP OOP MVC.....	33
2.5 Memecah bagian header, konten, sidebar & footer dengan php.....	38
2.5.1 Bagian Header .....	38
2.5.2 Bagian Sidebar.....	38

2.5.3 Bagian Konten .....	38
2.5.4 Bagian Footer .....	38
2.6 Membuat CRUDS Kategori.....	40
2.6.1 Index Kategori .....	42
2.6.2 Tambah Kategori .....	46
2.6.3 Edit Kategori.....	51
2.6.4 Hapus Kategori.....	53
2.6.5 Cari Kategori .....	54
2.7 Membuat CRUDS Buku .....	59
2.7.1 Index Buku.....	59
2.7.2 Tambah Buku .....	63
2.7.3 Edit Buku.....	66
2.7.4 Hapus Buku .....	69
2.7.5 Cari Buku .....	70
2.7.6 Laporan Buku .....	71
2.8 CRUDS User .....	76
2.9 Membuat Halaman About .....	85
2.10 Membuat Login & Logout .....	87
2.10.1 Login .....	87
2.10.2 Logout.....	92
<b>DAFTAR PUSTAKA .....</b>	<b>94</b>

## BAB I Perkenalan

### 1.1 Aplikasi Pendukung

Sebelum kita membuat sebuah Aplikasi berbasis web, apa saja sih yang perlu digunakan?

Berikut Diantaranya :

- **Web Browser**
  - o Adalah Aplikasi atau perangkat yang digunakan untuk mencari atau menjelajahi informasi di dalam jaringan internet dari sebuah media seperti situs, blog, jejaring sosial, dan lain sebagainya yang tersimpan di dalam internet. (Chrome, Mozilla, dll)
  - o Nantinya Web Browser akan **menampilkan** hasil yang sudah kita buat.
- **Text Editor**
  - o Adalah Aplikasi yang biasa digunakan Programmer untuk membuat kode program.
  - o Contohnya Sublime Text, Atom, Visual Studio Code, Notepad++
- **Apache Web Server**
  - o Apache hadir dalam bentuk software atau sering disebut perangkat lunak server, software inilah yang nantinya menggantikan fungsi server sesungguhnya.
  - o Yaitu fungsinya adalah untuk membuat sebuah hubungan antara server dan browser (crome, opera, safari) milik pengunjung web sambil mengirimkan file bolak-balik (antara user-server).
  - o Apache sudah terdapat dalam Xampp, Wamp, dll

### 1.2 Apa itu HTML, CSS, JavaScript, PHP, dan MySQL?

#### 1.2.1 HTML

HTML yang merupakan singkatan dari Hypertext Markup Language-- merupakan kumpulan script yang bisa kita gunakan untuk membuat halaman web. Dari HTML ini, kita bisa menampilkan data baik berupa teks maupun gambar di situs web yang kita buat.

#### 1.2.2 PHP

PHP atau singkatan dari Hypertext Preprocessor, adalah script yang biasanya disisipkan di file HTML atau bisa juga berdiri sendiri. PHP biasanya dipake untuk situs yang bersifat dinamis.

#### 1.2.3 CSS

CSS, singkatan dari Cascading Style Sheet, merupakan sekumpulan kode yang bisa kita pakai untuk menetapkan beberapa aturan yang bisa diterapkan di file HTML maupun PHP. Jadi dengan begitu, tampilan / interface yang kita buat bisa jadi lebih teratur dengan menggunakan CSS. Dengan CSS, kita bisa mengatur beberapa hal di situs web yang kita buat, antara lain

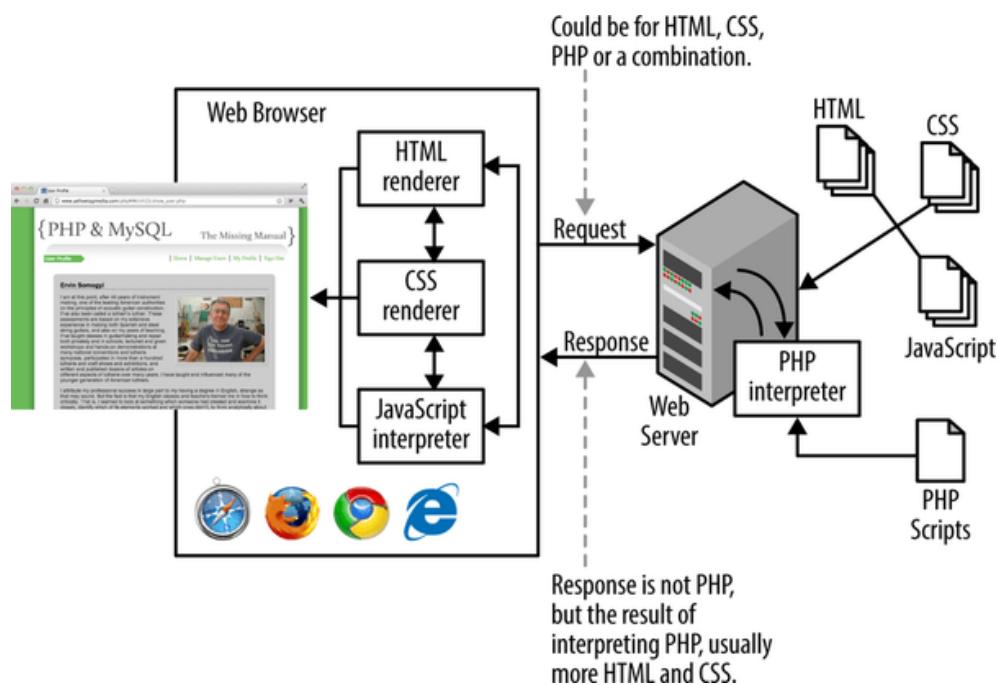
mengatur warna, ukuran, serta ketebalan font, kemudian kita juga bisa mengubah border tabel yang kita buat di situs web, dan masih banyak lagi.

#### 1.2.4 JS (JavaScript)

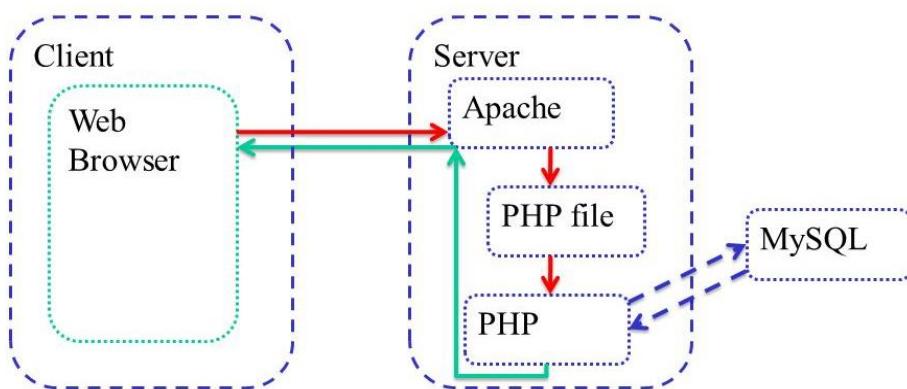
JavaScript adalah bahasa script populer yang bisa digunakan untuk membuat beberapa komponen pendukung situs web yang kita buat. Misalnya kita bisa membuat berbagai macam efek baik pada gambar maupun pada teks.

#### 1.2.5 MySQL

MySQL merupakan database engine atau server database yang mendukung bahasa database pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.



Gambar 1 Alur HTML, CSS, PHP & JS



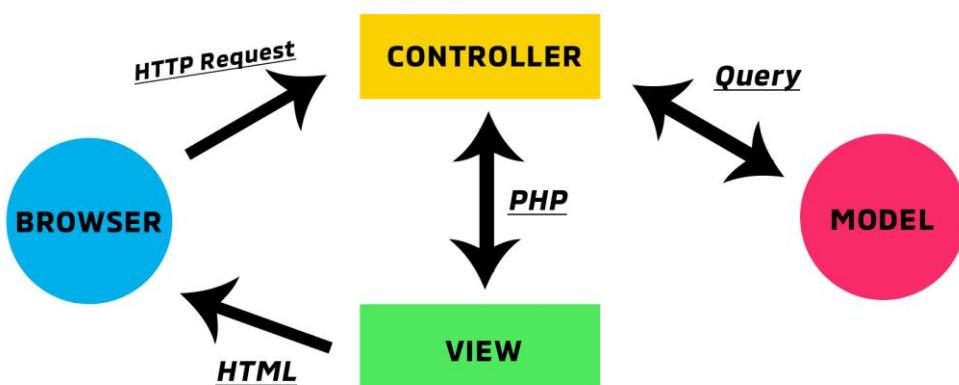
Gambar 2 Alur PHP & MySQL

### 1.3 OOP

OOP (Object Oriented Programming) atau yang dalam bahasa Indonesia berarti Pemrograman Berbasis Objek (PBO) adalah konsep dimana Property / Variable dan juga Method / Fungsi dibungkus dalam sebuah Class, yang kemudian akan di terapkan pada Objek - objek yang di deklarasikan.

### 1.4 MVC pada PHP

MVC atau (Model View Controller) adalah sebuah konsep pemrograman atau desain pattern yang dirancang untuk mempermudah programmer dalam pembuatan sebuah proyek dengan memisahkan antara model, view dan controller agar susunan direktori atau folder lebih teratur dan terorganisasi guna mempermudah tahap pengembangan selanjutnya.



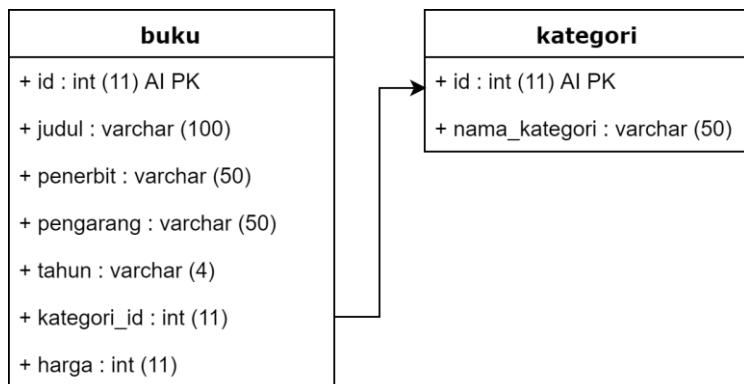
Gambar 3 Alur MVC

## BAB II Praktek

### 2.1 Mendalami MySQL

Dalam membuat sistem, kita pasti memerlukan sebuah database & tabel yang akan menampung data - data kita. Nah untuk membuatnya kita harus mempunyai rancangan database nya dulu. Misalnya kita akan membuat sebuah web yang menampung data **BUKU**

Maka sederhananya kita akan membuat rancangan seperti ini :



Gambar 4 Rancangan dasar tabel buku & kategori

Nah karena kita sudah membuat rancangannya, apa saja tabel dan kolom yang kita perlukan maka, kita akan langsung membuatnya diMySQL.

Sebenarnya ada 2 cara yang bisa kamu praktekkan, yaitu menggunakan **command prompt** dan langsung menggunakan di**phpmyadmin**. Mungkin masih ada yang bingung? Apa sih bedanya phpmyadmin dan yang langsung di commandprompt?

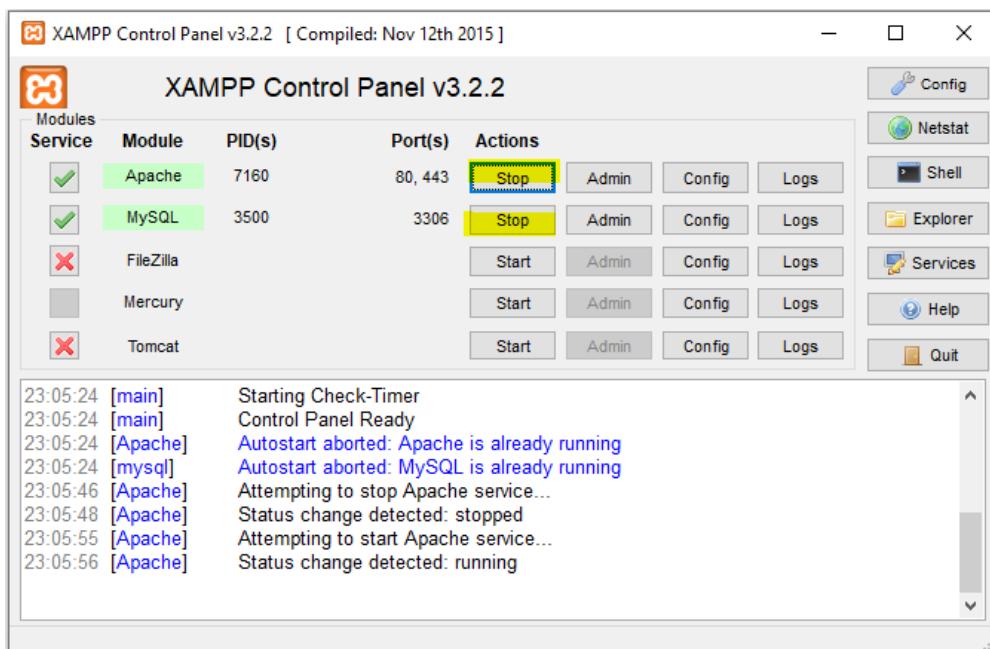
**phpMyAdmin** adalah sebuah alat yang dapat menangani administrasi dari MySQL dengan melalui web atau langsung ada interfacenya. phpMyAdmin disebut juga pembantu diberbagai operasi pada MySQL. phpMyAdmin mendukung MySQL dalam berbagai operasi, seperti mengolah basis data, indeks, user, permissions dan lain sebagainya tanpa harus mengetikkan query MySQL. Tapi meski begitu kita tetap wajib mengetahui bagaimana cara membuat, mengedit, hapus dan cari data di MySQL karena itu kita perlukan di file PHP kita nantinya.

Sedangkan yang langsung di cmd? Yup, disitu tidak ada interface jadi murni kita harus mengetikkan semua query MySQL satu persatu, tidak bisa langsung klik – klik seperti di phpmyadmin.

Nah walaupun kita akan praktek langsung di MySQL, saya akan tetap menuliskan sebenarnya query apa sih yang digunakan untuk membuat database, tabel dan operasi – operasi MySQL lainnya.

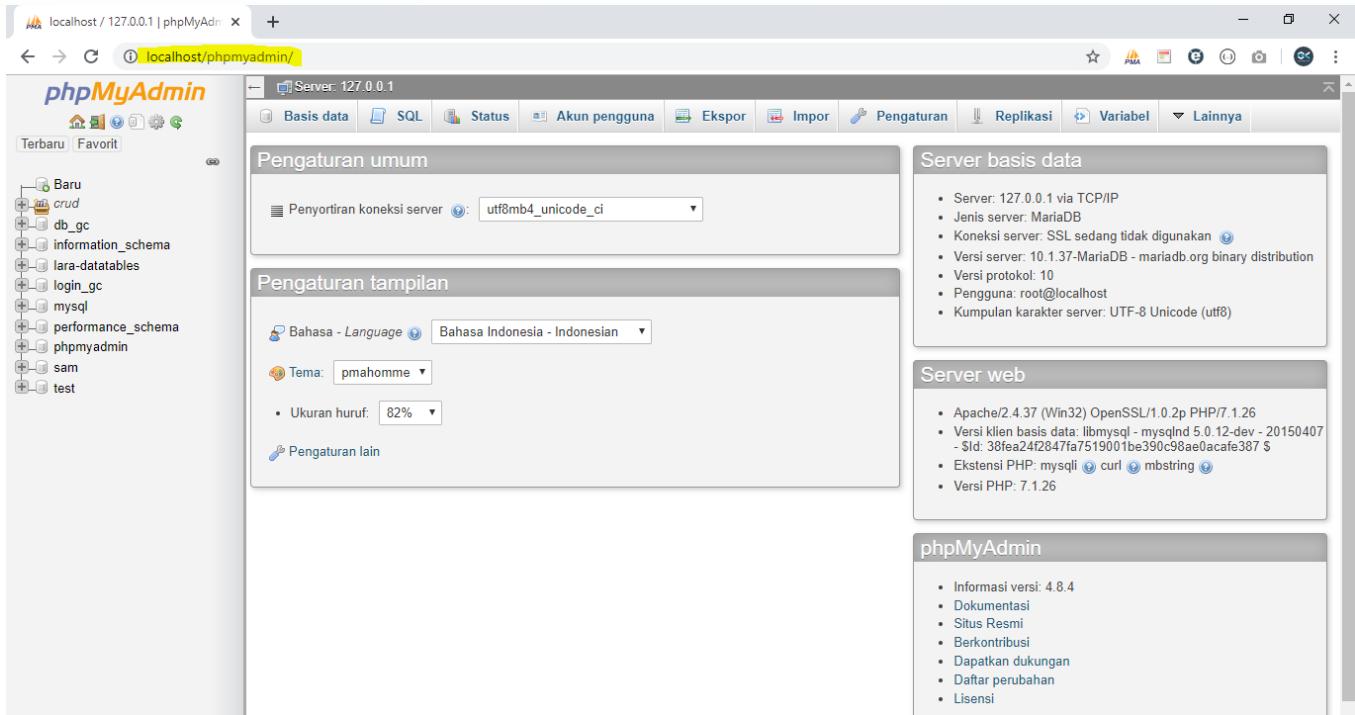
### 2.1.1 Membuka phpMyAdmin

1. Buka Xampp, *start Apache & MySQL* sampai bertuliskan menjadi *stop*



Gambar 5 Xampp

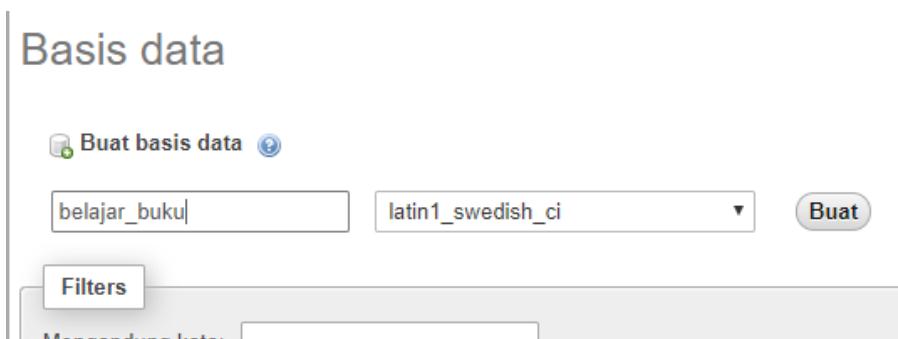
2. Buka browser kalian, tuliskan localhost/phpmyadmin



Gambar 6 phpMyAdmin

## 2.1.2 Membuat Database

1. Buat database, klik Link Baru yang ada di sebelah kiri atas
2. Tuliskan nama database yang akan kita buat yaitu belajar\_buku



Gambar 7 membuat database

3. Klik saja buat langsung maka kita sudah berhasil membuat database. Tanpa perlu ngoding query mysql tapi bisa membuat database? Yup benar banget, tapi dibalik itu semua ada query MySQL yang diproses oleh phpMyadmin.

```
create database belajar_buku;
```

Gambar 8 query mysql untuk membuat database

### 2.1.3 Membuat Tabel

- Kalau database sudah berhasil dibuat, kita langsung membuat tabel buku & tabel kategori. Tuliskan nama tabel dan jumlah kolomnya



Gambar 9 beri nama tabel buku

- Setelah kalian klik kirim, maka akan tampil text kolom & tipe datanya, isikan sesuai rancangan yang sudah kita bikin

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks	AJ
id	INT	11	Tidak ada			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
judul	VARCHAR	100	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
penerbit	VARCHAR	50	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
pengarang	VARCHAR	50	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
tahun	VARCHAR	4	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
kategori_id	INT	11	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
harga	INT	11	Tidak ada			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>

Gambar 10 struktur dari tabel buku

Penjelasan :

- Integer & Varchar adalah tipe data. Bedanya integer biasa digunakan untuk bilangan yang berupa angka bulat, misal 1, 2, 3, 100, 1000 tanpa ada koma. Kalau Varchar bisa menampung karakter misalnya (A s.d Z / 0 s.d 9999++ / simbol - simbol / spasi)
- Masih banyak tipe data lain yang bisa kalian gunakan beserta masing - masing fungsinya misalnya tipe data date untuk menampung tanggal (2019-01-14) atau tipe data enum, dll

- Id kenapa menggunakan Integer? Karena disitu Id saya buat A\_I (Auto Increment) maksudnya agar setiap kita menambahkan record / data, maka otomatis tergenerate jadi angka 1, 2, 3 dst...
  - PRIMARY KEY? Ya, PK dari tabel buku adalah id artinya kolom id harus UNIK nilainya, tidak boleh ada yang sama agar pada saat proses edit, delete dan cari itu mudah.
  - Tak Ternilai / Null : Yang dicentang berarti pada saat proses input, boleh nilainya kosong.
  - Panjang nilai ya bebas mau berapa saja sesuai keperluan hehe
- 3.** Langsung deh kalian klik kirim kalau udah diisi.

```
CREATE TABLE buku (id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, judul VARCHAR(100) NOT NULL, penerbit VARCHAR(50) NULL, pengarang VARCHAR(50) NULL, tahun VARCHAR(4) NULL, kategori_id INT(11) NULL, harga INT(11) NULL);
```

Gambar 11 query membuat tabel buku

#### 4. Membuat tabel kategori

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks	A.I	K
id	INT	11	Tidak ada			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
nama_kategori	VARCHAR	50	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 12 struktur tabel kategori

```
CREATE TABLE kategori ( id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, nama_kategori VARCHAR(50) NOT NULL);
```

Gambar 13 query membuat tabel kategori

#### 2.1.4 Membuat relasi antar tabel

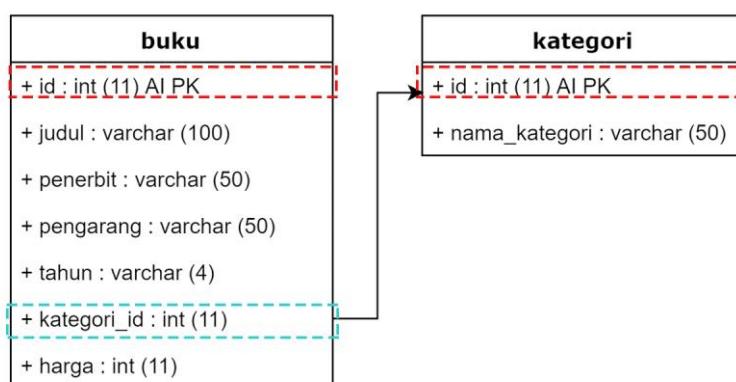
1. Kalau tabel yang kita rancang sudah terbuat, kita akan membuat relasi atau hubungan antar tabelnya dulu.

Pada database belajar\_buku ini terdapat 2 tabel yaitu antara buku dan kategori yang menghubungkannya adalah kategori\_id

Dimana pada tabel buku, ada id sebagai **Primary Key** dan kategori\_id sebagai **Foreign Key**.

**Primary Key** : Salah satu kolom unik untuk identifikasi baris dalam tabel yang tidak boleh ada sama nilainya, kolom ini tidak boleh berulang, dan tidak boleh kosong (null).

**Foreign key** : Suatu field dalam satu tabel yang digunakan untuk menghubungkan dua tabel. Dengan begitu, foreign key merujuk pada suatu kolom yang unik untuk mengidentifikasi baris dari satu tabel yang sama atau berbeda.



Gambar 14 Perbedaan PK & FK

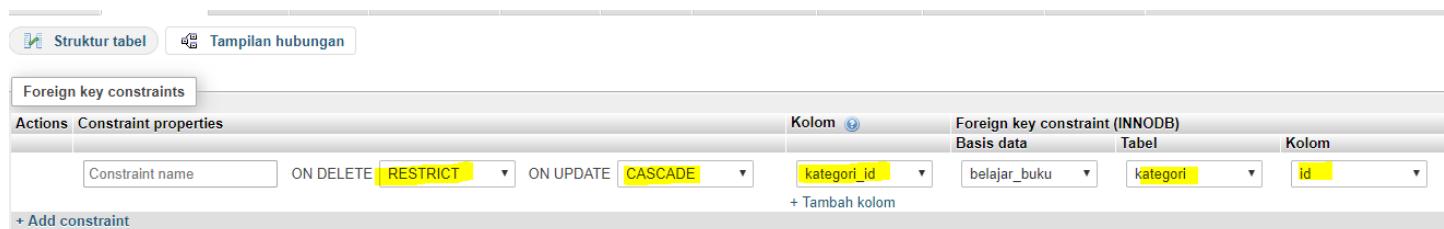
Yang warna **merah** adalah Primary Key, dan yang warna **biru** adalah Foreign Key.

## 2. Untuk membuat relasinya, pilih pada struktur lalu klik Tampilan hubungan / Relation View



Gambar 15 untuk relasikan tabel

## 3. Isi seperti berikut ini :



Gambar 16 Merelasikan tabel

Penjelasan :

- On Delete Restrict : Artinya jika di tabel kategori datanya dihapus, maka tidak diperbolehkan kalau di tabel buku terdapat kategori yang akan dihapus.

- On Update Cascade : Jika primary key di tabel kategori diubah, maka di tabel buku juga akan berubah.
- 4.** Kembali ke database belajar\_buku, pilih Desainer untuk melihat relasi tabel yang sudah dibuat



Gambar 17 relasi database buku & kategori

### 2.1.5 Menambahkan Data

- 1.** Tambahkan data di kategori dan buku dengan mengklik tab Insert / Tambahkan. Saya menambahkan kategori Novel, dll.

The screenshot shows the 'Tambahkan' (Add) form for the 'kategori' table. The 'Jenis' tab is selected. There are two rows: one for the 'id' column (type int(11)) and one for the 'nama\_kategori' column (type varchar(50)). In the 'nama\_kategori' row, the input field contains 'Novel'. At the bottom right is a 'Kirim' (Send) button.

Gambar 18 menambahkan data kategori

The screenshot shows a MySQL Workbench window with a query editor containing the following SQL code:

```
INSERT INTO kategori (id, nama_kategori) VALUES (NULL, 'Novel');
```

Gambar 19 query mysql untuk menambahkan data

Penjelasan :

- Id kenapa NULL? Karena ID sudah otomatis tergenerate menjadi angka yang berurutan oleh AUTO\_INCREMENT
- 2.** Untuk data buku, caranya sama jadi silahkan ditambahkan beberapa buku di phpMyAdmin langsung. Disini saya sudah menambahkan 3 buku.

### 2.1.6 Menampilkan Data

- 1.** Untuk menampilkan nya langsung klik saja tabel buku atau query MySQL nya seperti ini :

Gambar 20 menampilkan tabel buku

```
SELECT * FROM buku
```

Gambar 21 query mysql menampilkan semua data

Penjelasan :

- *SELECT \* FROM nama\_tabel*: Tanda bintang (\*) adalah tanda untuk menampilkan seluruh isi data/record ditabel buku.
- 2.** Menampilkan beberapa kolom saja? Misalnya kita menampilkan kolom judul & pengarang saja. Tinggal masukkan query dibawah ini pada tab SQL

```
SELECT judul, pengarang FROM buku;
```

Gambar 22 query menampilkan kolom tertentu

Hasilnya :

judul	pengarang
Danur	Risa Saraswati
Kisah Tanah Dayak	Aditya Harlan
Naruto Episode 2999	Kishimoto

Gambar 23 hasil query select kolom tertentu

- 3.** Menampilkan data dengan ketentuan tertentu, misalnya ingin menampilkan data dengan id 1

1 / 3

```
SELECT * FROM buku WHERE id = 1;
```

Gambar 24 query menampilkan berdasarkan id

Hasilnya :

		id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<a href="#"> Ubah</a> <a href="#"> Salin</a> <a href="#"> Hapus</a>	1	Danur	Gramedia	Risa Saraswati	2018	2	50000
<a href="#"></a>	<input type="checkbox"/> Pilih Semua	Dengan pilihan:		<a href="#"> Ubah</a>	<a href="#"> Salin</a>	<a href="#"> Hapus</a>	<a href="#"> Eksport</a>	

Gambar 25 hasil select where id 1

## Penjelasan :

- Karena buku Danur memiliki id 1, maka sesuai dengan select yang kita tentukan hanya id 1 yang ditampilkan

4. Menampilkan data yang harganya diatas 27000

1 / 1

```
SELECT * FROM buku where harga > 27000;
```

Gambar 26 query menampilkan harga lebih 27000

Hasilnya :

<input type="checkbox"/>	Ubah	Salin	Hapus	1	Danur	Gramedia	Risa Saraswati	2018	2
<input type="checkbox"/>	Ubah	Salin	Hapus	2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2

Gambar 27 hasil yang harga bukunya lebih dari 27000

## Penjelasan :

- 27000 hanya contoh ya.
  - Dari 3 data, hanya 2 yang harganya diatas dari 27000
  - Bisa juga menggunakan beberapa ketentuan misalnya :
    - o < Kurang dari 27000
    - o <= Kurang dari atau sama dengan 27000
    - o = sama dengan 27000

- o > lebih dari 27000
  - o  $\geq$  lebih dari atau sama dengan 27000

5. Menampilkan berdasarkan penerbit? Misalnya menampilkan penerbit oleh gramedia

Three colored circles are arranged horizontally. The first circle is red, the second is yellow, and the third is green.

```
SELECT * FROM buku WHERE penerbit = 'Gramedia';
```

Gambar 28 query menampilkan penerbit gramedia

Hasilnya :

				<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	1	Danur	Gramedia	Risa Saraswati	2018	2	50000
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000

Gambar 29 hasil query by penerbit

6. Menampilkan berdasarkan tahun & penerbit, misalnya mau menampilkan tahun terbit yang 2019 & penerbitnya gramedia

1 / 3

```
SELECT * FROM buku where penerbit = 'Gramedia' AND tahun = 2019
```

Gambar 30 query where penerbit dan tahun

Hasilnya :

	<input type="text"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000

Gambar 31 hasil query penerbit dan tahun

## Penjelasan :

- Karena memiliki 2 ketentuan yang wajib, maka harus menggunakan **AND**
  - Bisa juga menggunakan **OR** kalau yang wajib hanya salah satu, misalnya *Where penerbit = 'Gramedia' OR penerbit ='Loko Media'*;
  - Maka yang tampil juga berdasarkan penerbit dari gramedia atau loko media

## 2.1.7 Order By

1. Menampilkan data tapi dari yang terbaru dulu, bisa menggunakan ORDER BY kolom



```
SELECT * FROM buku ORDER BY id DESC
```

Gambar 32 query order by id desc

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	Danur	Gramedia	Risa Saraswati	2018	2	50000

Gambar 33 hasil query order by id

Penjelasan :

- Yang tampil id nya dari tertinggi ke terendah
- Maksud **DESC** : Tinggi ke rendah, kalau **ASC** : rendah ke tinggi
- Kamu juga bisa melakukan order by harga dari rendah ke tinggi dan sebaliknya.

## 2.1.8 Pencarian Data

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	Danur	Gramedia	Risa Saraswati	2018	2	50000
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	4	Naruto Episode 4000	Gramedia	Kishimoto	2020	3	28000

Gambar 34 data awal

1. Ingin mencari langsung data judul yang **sama persis** dengan WHERE misalnya mencari kata 'Danur'



```
SELECT * FROM buku WHERE judul = 'Danur';
```

Gambar 35 query pencarian sama persis

Opsi

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	Danur	Gramedia	Risa Saraswati	2018	2	50000

Gambar 36 hasil pencarian sama persis

2. Misalnya kita ingin mencari judul yang mengandung kata "an", ini bisa menggunakan WHERE **LIKE**

```
SELECT * FROM buku WHERE judul LIKE '%an%'
```

Gambar 37 query like

Hasilnya :

	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	<b>id</b>	<b>judul</b>	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>				1	Danur	Gramedia	Risa Saraswati	2018	2	50000
<input type="checkbox"/>				2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000

Pilih Semua    Dengan pilihan:  Ubah  Salin  Hapus  Eksport

Gambar 38 hasil query like an

Penjelasan :

- Tanda '%an%' akan mencari ke dalam kata yang mengandung an, misalnya disitu terdapat judul buku 'Danur' dengan 'Kisah Tanah Dayak'. Keduanya **mengandung** kata an
  - Jika tanda % hanya **didepan** saja, misalnya '%K' maka yang tampil hanya yang judulnya '**K**isah Tanah Dayak'.
  - Begitu juga sebaliknya, misal kita cuman memberi tanda % **dibelakang**, contohnya mencari 'h%' maka yang dicari hanya data yang memiliki huruf h dibelakang.
3. Bisa dikombinasikan juga dengan operator logika OR atau AND, contohnya kita akan mencari judul yang mengandung kata naruto lalu tahun terbitnya 2020.

```
SELECT * FROM buku WHERE judul LIKE '%Naruto%' AND tahun = '2020';
```

Gambar 39 query like dengan AND

Hasilnya :

	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	<b>id</b>	<b>judul</b>	penerbit	pengarang	<b>tahun</b>	kategori_id	harga
<input type="checkbox"/>				4	Naruto Episode 4000	Gramedia	Kishimoto	2020	3	28000

Pilih Semua    Dengan pilihan:  Ubah  Salin  Hapus  Eksport

Gambar 40 hasil query like dengan AND

## 2.1.9 Join

Tujuan Join adalah untuk menggabungkan tabel yang memiliki relasi misalnya tabel buku dengan tabel kategori, bisa dihubungkan karena ditabel buku terdapat kategori\_id, jadi kita akan menampilkan secara detail.

	+ Opsi	Ubah	Salin	Hapus	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>				1	Danur	Gramedia	Risa Saraswati	2018	2	50000	
<input type="checkbox"/>				2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000	
<input type="checkbox"/>				3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000	
<input type="checkbox"/>				4	Naruto Episode 4000	Gramedia	Kishimoto	2020	NULL	28000	

↑  Pilih Semua Dengan pilihan: Ubah Salin Hapus Eksport

Gambar 41 data awal tabel buku

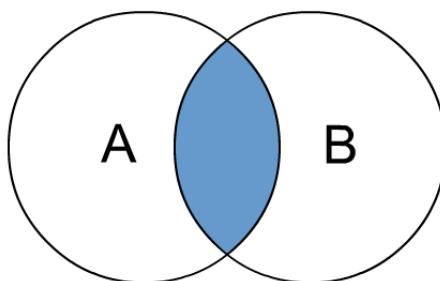
	+ Opsi	Ubah	Salin	Hapus	id	nama_kategori
<input type="checkbox"/>				1	Novel	
<input type="checkbox"/>				2	Horor	
<input type="checkbox"/>				3	Komik	
<input type="checkbox"/>				4	Drama	

Gambar 42 data awal tabel kategori

Keterangan :

- Tabel buku memiliki 4 data
- Data pada tabel buku dengan id 4 tidak memiliki kategori alias kategorinya NULL / kosong

### 1. Inner JOIN / JOIN biasa



Gambar 43 ilustrasi inner join



```
SELECT buku.*, kategori.* FROM buku INNER JOIN kategori ON buku.kategori_id = kategori.id;
```

Gambar 44 query inner join

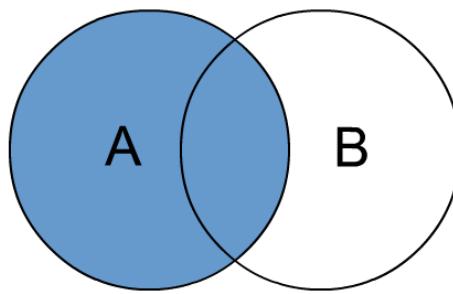
Hasilnya :

+ Opsi									
<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>	<b>id</b>	<b>nama_kategori</b>	
1	Danur	Gramedia	Risa Saraswati	2018	2	50000	2	Horor	
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000	2	Horor	
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000	3	Komik	

Gambar 45 hasil query inner join

Penjelasan :

- Tabel yang ditampilkan, masing – masing sesuai dengan kategorinya. Pada buku id dari judul Danur ada lah 2, nah id 2 itu tadi dimiliki oleh kategori Horor
  - Data buku dengan judul Naruto Episode 4000 tidak tampil karena tidak memiliki kategori
- 2. Left Join**



Gambar 46 ilustrasi left join



```
SELECT buku.*, kategori.* FROM buku LEFT JOIN kategori ON buku.kategori_id = kategori.id;
```

Gambar 47 query left join

Hasilnya :

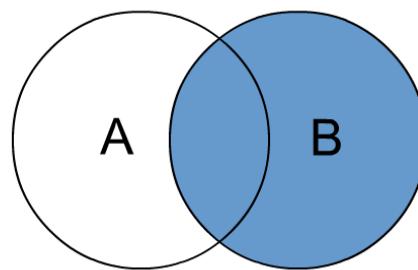
+ Opsi									
<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>	<b>id</b>	<b>nama_kategori</b>	
1	Danur	Gramedia	Risa Saraswati	2018	2	50000	2	Horor	
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000	2	Horor	
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000	3	Komik	
4	Naruto Episode 4000	Gramedia	Kishimoto	2020	NULL	28000	NULL	NULL	

Gambar 48 menampilkan hasil query left join

Penjelasan :

- Semua data pada tabel buku ditampilkan
- Pada buku dengan id 4 kan kategorinya NULL kok tetap ditampilkan? Karena left join menampilkan semua data buku walaupun kategorinya null

### 3. Right Join



Gambar 49 ilustrasi right join



```
SELECT buku.*, kategori.* FROM buku RIGHT JOIN kategori ON buku.kategori_id = kategori.id;
```

Gambar 50 query right join

Hasilnya :

+ Opsi	<u>id</u>	<u>judul</u>	<u>penerbit</u>	<u>pengarang</u>	<u>tahun</u>	<u>kategori_id</u>	<u>harga</u>	<u>id</u>	<u>nama_kategori</u>
	1	Danur	Gramedia	Risa Saraswati	2018	2	50000	2	Horor
	2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000	2	Horor
	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000	3	Komik
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	Novel
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	4	Drama

Gambar 51 hasil query right join

Penjelasan :

- Semua data pada tabel kategori tampil, walaupun tidak ada yang memiliki kategori novel maupun drama. Inilah yang dimaksud dengan right join.

#### 2.1.10 Mengubah Data

Pada operasi MySQL, ubah data merupakan salah satu penting dan query yang harus kamu hapal dan biasanya untuk mengubah data pasti dengan kondisi tertentu. Misal ingin merubah judul yang id nya 1 atau yang judulnya Danur.

Kalau diphpMyAdmin sih mau edit gampang, tinggal cari datanya lalu didouble klik pada kolom yang mau diubah. Tapi gimana querynya?

<u>id</u>	<u>judul</u>	<u>penerbit</u>	<u>pengarang</u>	<u>tahun</u>	<u>kategori_id</u>	<u>harga</u>
1	Danur	Gramedia	Risa Saraswati	2018	2	50000
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000
4	Naruto Episode 4000	Gramedia	Kishimoto	2020	NULL	28000

Gambar 52 data awal sebelum diubah

1. Ubah satu data cukup langsung menggunakan where, jadi kita mau ubah harga danur yang awalnya 50000 jadi 75000.



```
UPDATE buku set harga = 75000 where id = 1;
```

Gambar 53 query update data

Hasilnya :

<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>
1	Danur	Gramedia	Risa Saraswati	2018	2	75000
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000
4	Naruto Episode 4000	Gramedia	Kishimoto	2020	NULL	28000

Gambar 54 hasil setelah diupdate

Penjelasan :

- Karena judul buku danur memiliki id 1, maka pada kondisi update kita pakai where id = 1
- Bisa kok pakai where judul = 'Danur', tapi itu tidak direkomendasikan karena kolom judul itu tidak UNIK, jadi bisa saja judul Danur ada beberapa tapi hanya beda pengarang saja. Makanya id dibuat unik agar membedakan buku danur 1 dengan buku danur lainnya.
- Harga sudah berubah, awalnya 50000 jadi 75000

## 2. Ubah dengan beberapa kondisi

Misal kita ubah ternyata nih judul naruto tahun terbit 2020 keatas pengarang nya sudah beda, bukan kishimoto lagi terus kategorinya kita isi jadi 3 dan harganya naik jadi 33000. Jadi kondisinya where judul like naruto AND tahun terbit >= 2020



```
UPDATE buku SET pengarang = 'Udin', kategori_id = 3, harga = 33000 WHERE judul LIKE '%Naruto%' AND tahun >= 2020;
```

Gambar 55 query update multi kondisi

Hasilnya :

<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>
1	Danur	Gramedia	Risa Saraswati	2018	2	75000
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	30000
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	27000
4	Naruto Episode 4000	Gramedia	Udin	2020	3	33000

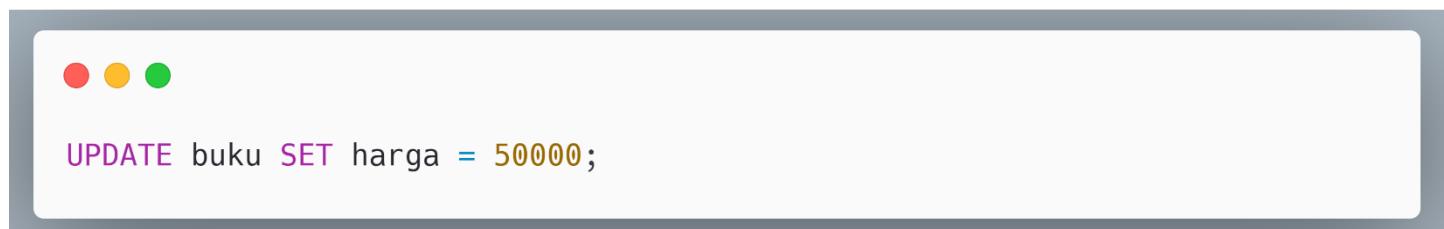
Gambar 56 hasil query update multi kondisi

Penjelasan :

- Yang berubah hanya naruto episode 4000, karena hanya dia yang memenuhi kondisi dari update tadi
- Yaitu yang nama nya naruto & tahun terbitnya sama dengan atau diatasnya 2020 hanya naruto episode 4000
- Disitu saya merubah beberapa kolom, diantaranya nama pengarang, kategori, dan harganya.

### 3. Ubah semua data langsung

Mungkin contohnya kita akan mengubah semua harga menjadi 50000. Ya langsung saja gini :



```
UPDATE buku SET harga = 50000;
```

Gambar 57 query ubah semua data

Hasilnya :

<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>
1	Danur	Gramedia	Risa Saraswati	2018	2	50000
2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	2	50000
3	Naruto Episode 2999	Gramedia	Kishimoto	2019	3	50000
4	Naruto Episode 4000	Gramedia	Udin	2020	3	50000

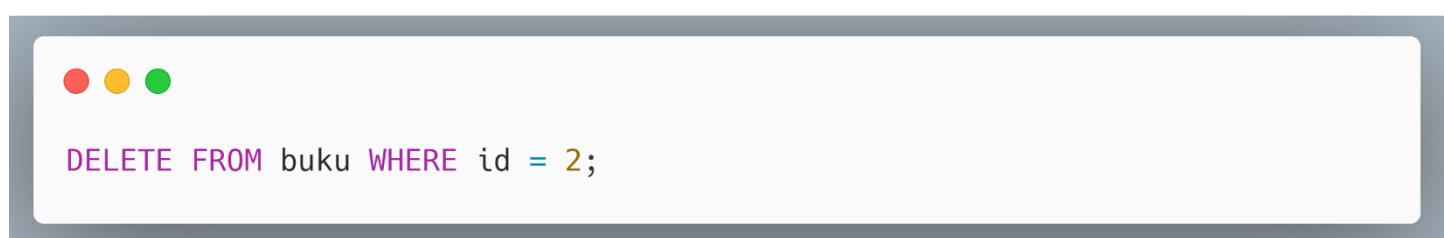
Gambar 58 hasil ubah semua

### 2.1.11 Menghapus Record / Data

			<b>id</b>	<b>judul</b>	<b>penerbit</b>	<b>pengarang</b>	<b>tahun</b>	<b>kategori_id</b>	<b>harga</b>
			Ubah	1	Danur	Gramedia	Risa Saraswati	2018	25000
			Ubah	2	Kisah Tanah Dayak	Loko Media	Aditya Harlan	2004	30000
			Ubah	3	Naruto Episode 2999	Gramedia	Kishimoto	2019	33000
			Ubah	4	Naruto Episode 4000	Gramedia	Udin	2020	70000

Gambar 59 data awal buku

1. Menghapus 1 data dengan kondisi jelas.misalnya kita akan menghapus buku dengan id 2



```
DELETE FROM buku WHERE id = 2;
```

Gambar 60 query delete

Hasilnya sudah jelas, buku dengan id 2 pasti akan terhapus.

## 2. Menghapus dengan 2 kondisi atau lebih

Misalnya menghapus yang penerbitnya gramedia dan harganya diatas 25000



```
DELETE FROM buku WHERE penerbit = 'Gramedia' AND harga > 25000;
```

Gambar 61 query delete 2 kondisi

Hasilnya yang tersisa dari semua tadi yaitu :

Opsi	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	judul	penerbit	pengarang	tahun	kategori_id	harga
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	Danur	Gramedia	Risa Saraswati	2018	2	25000

Gambar 62 sisa setelah query delete

## 3. Menghapus semua data di tabel tapi AUTO\_INCREMENT tidak ter-reset, jadi jika ditambahkan id nya akan tetap lanjut



```
DELETE FROM buku;
```

Gambar 63 query delete buku

## 4. Menghapus semua buku juga dan membuat AUTO\_INCREMENT kembali jadi 1



```
TRUNCATE TABLE buku;
```

Gambar 64 query truncate buku.

## 2.1.12 Operasi Aritmatik

Operator	Penjelasan
+	Penambahan
-	Pengurangan
*	Pengalian
/	Pembagian
%	Sisa hasil bagi (modulus)

## 2.1.13 Operasi Logika

Operator	Penjelasan
NOT atau !	Logika bukan
AND atau &&	Logika dan
OR atau	Logika atau
XOR	Logika bukan atau (XOR)

## 2.1.14 Operasi Perbandingan

Operator	Penjelasan
=	Sama dengan
<> atau !=	Tidak sama dengan
<=>	sama dengan (null safe)
<	kurang dari
<=	kurang dari atau sama dengan
>	lebih besar dari
>=	lebih besar atau sama dengan
BETWEEN	Berada pada batas tertentu
IN	Berada di dalam
IS NULL	Pengecekan apakah berisi NULL
IS NOT NULL	Pengecekan apakah bukan berisi NULL
LIKE	Pencarian menggunakan wildcard

Note :

- Yang tidak terdapat pada contoh – contoh diatas silahkan ditanyakan atau bisa googling sesuai dengan keperluan.

## 2.2 Membuat Database untuk Aplikasi Buku Sederhana

- Buat database dengan nama **buku\_kita**
- Buat tabel **buku**

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	<b>id</b>	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	<b>judul</b>	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada		
3	<b>penerbit</b>	varchar(50)	latin1_swedish_ci		Ya	NULL		
4	<b>pengarang</b>	varchar(50)	latin1_swedish_ci		Ya	NULL		
5	<b>tahun</b>	varchar(4)	latin1_swedish_ci		Ya	NULL		
6	<b>kategori_id</b>	int(11)			Ya	NULL		
7	<b>harga</b>	int(11)			Ya	NULL		

Gambar 65 struktur tabel buku

- Buat tabel **kategori**

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	<b>id</b>	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	<b>nama_kategori</b>	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada		

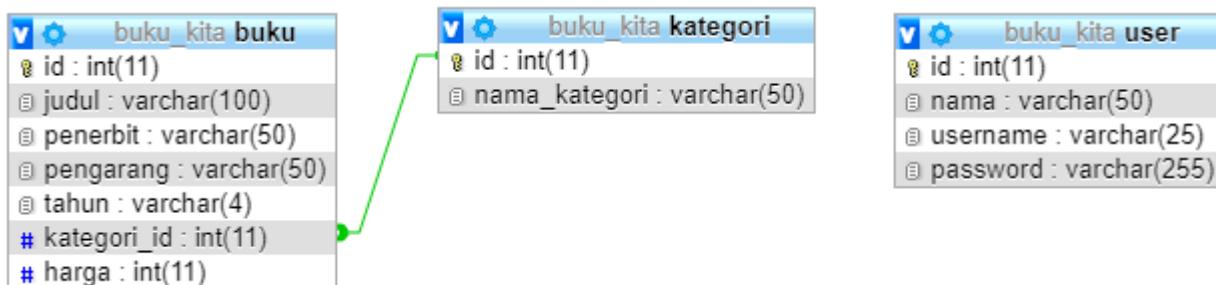
Gambar 66 struktur tabel kategori

- Buat tabel **user** (akan digunakan untuk login)

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	<b>id</b>	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	<b>nama</b>	varchar(50)	latin1_swedish_ci		Ya	NULL		
3	<b>username</b>	varchar(25)	latin1_swedish_ci		Tidak	Tidak ada		
4	<b>password</b>	varchar(255)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 67 struktur tabel user

- Bikin relasinya



Note :

- Silahkan buat tabel diatas dulu, jika ada perubahan akan kita rubah pada saat praktek PHP OOP sesuai dengan keperluan nanti.

## 2.3 Membuat tampilan dengan Template AdminLTE 3

Silahkan download templatanya dulu di : <http://bit.ly/adminlte3-gc>

### 2.3.1 Menampilkan Extension file

Untuk yang extension file nya tidak terlihat, silahkan klik logo windows di keyboard lalu ketikkan **folder options**, nah pilih tab View, pada bagian **Hide Extensions for known file types** hilangkan tanda centang, lalu Apply. Jadi tipe file misal index.html akan terlihat

### 2.3.2 Menyesuaikan tampilan halaman utama

Jadi nantinya kita akan membuat beberapa tampilan, diantaranya :

- Index utama
- Index buku
- Tambah buku
- Edit buku
- Index kategori
- Tambah kategori
- Edit kategori
- About me
- Login
- Register

Wah banyak juga ya hehe tapi tenang, kita hanya tinggal menyesuaikan aja kok

Kalau udah download, silahkan ikuti cara berikut :

1. Extract Folder Zip yang tadi di download ke htdocs
2. Buat folder baru didalam htdocs dengan nama **bukita**
3. Masuk ke dalam folder adminlte, lalu ke folder pages > examples, **blank.html** di copy dan paste di folder bukita
4. Lalu kembali ke folder adminlte, copy folder **dist & plugins**, yup paste difolder bukita

Name	Date modified	Type	Size
dist	25/12/2019 10:23	File folder	
plugins	25/12/2019 10:24	File folder	
blank.html	20/11/2019 16:41	HTML File	29 KB

Gambar 68 isi folder bukita

Nah, jadi sementara ada folder dist, plugins dan file blank.html di folder bukita.

5. Kita akan membuang bagian yang tidak perlu dibagian blank.html

6. Buka file blank.html,pada bagian head, sesuaikan link href nya & title menjadi seperti ini :

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Bukita</title>
  <!-- Tell the browser to be responsive to screen width -->
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Font Awesome -->
  <link rel="stylesheet" href="plugins/fontawesome-free/css/all.min.css">
  <!-- Ionicons -->
  <link rel="stylesheet"
    href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
  <!-- overlayScrollbars -->
  <link rel="stylesheet" href="dist/css/adminlte.min.css">
  <!-- Google Font: Source Sans Pro -->
  <link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700"
    rel="stylesheet">
</head>
```

Note : Menyesuaikan link css & plugins dengan menghilangkan .../

7. Bagian navbar sebagian kita hapus saja, dan buat seperti ini :

```
<!-- Navbar -->
<nav class="main-header navbar navbar-expand navbar-white navbar-light">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" data-widget="pushmenu" href="#"><i class="fas fa-bars"></i></a>
    </li>
  </ul>
  <!-- Right navbar links -->
  <ul class="navbar-nav ml-auto">
    <li class="nav-item d-none d-sm-inline-block">
      <a href="#" class="nav-link">Logout</a>
    </li>
  </ul>
</nav>
<!-- /.navbar -->
```

8. Sekarang pada bagian sidebar atau **<aside>** :

```
<!-- Main Sidebar Container -->
<aside class="main-sidebar sidebar-dark-primary elevation-4">
  <!-- Brand Logo -->
  <a href="#" class="brand-link">
    <span class="brand-text font-weight-light">Buku Kita</span>
  </a>

  <!-- Sidebar -->
```

```

<div class="sidebar">
  <!-- Sidebar user (optional) -->
  <div class="user-panel mt-3 pb-3 mb-3 d-flex">
    <div class="info">
      <a href="#" class="d-block">@herziwp</a>
    </div>
  </div>

  <!-- Sidebar Menu -->
  <nav class="mt-2">
    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu"
      data-accordion="false">
      <!-- Add icons to the links using the .nav-icon class
           with font-awesome or any other icon font library -->
      <li class="nav-item">
        <a href="index.php" class="nav-link">
          <i class="nav-icon fas fa-tachometer-alt"></i>
          <p>
            Dashboard
          </p>
        </a>
      </li>
      <li class="nav-item">
        <a href="index.php" class="nav-link">
          <i class="nav-icon fas fa-copy"></i>
          <p>
            Kategori
          </p>
        </a>
      </li>
      <li class="nav-item">
        <a href="index.php" class="nav-link">
          <i class="nav-icon fas fa-book"></i>
          <p>
            Buku
          </p>
        </a>
      </li>
      <li class="nav-item">
        <a href="index.php" class="nav-link">
          <i class="nav-icon fas fa-user"></i>
          <p>
            About Me
          </p>
        </a>
      </li>
    </ul>
  </nav>
  <!-- /.sidebar-menu -->
</div>
<!-- /.sidebar -->
</aside>

```

**9.** Bagian section content-header :

```
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-6">
        <h1>Halaman Utama</h1>
      </div>
    </div>
  </div><!-- /.container-fluid -->
</section>
```

**10.**Pada bagian <section class="content"> :

```
<section class="content">
  <!-- Default box -->
  <div class="card">
    <div class="card-header">
      <h3 class="card-title">Title</h3>
    </div>
    <div class="card-body">
      Start creating your amazing application!
    </div>
    <!-- /.card-body -->
    <div class="card-footer">
      Footer
    </div>
    <!-- /.card-footer-->
  </div>
  <!-- /.card -->
</section>
```

**11.**Bagian <footer> :

```
<footer class="main-footer">
  <b>Bukita 2019</b> All rights reserved.
</footer>
```

**Note : bagian <aside> </aside> dibawah footer, silahkan dihapus saja.**

**12.**Bagian javascript paling bawah sebelum tutup body :

```
<script src="plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="dist/js/demo.js"></script>
```

13.Nah itu tadi bagian - bagian nya, sekarang full dari blank.html nya sebagai berikut :

### Full coding blank.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Bukita</title>
<!-- Tell the browser to be responsive to screen width -->
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Font Awesome -->
<link rel="stylesheet" href="plugins/fontawesome-free/css/all.min.css">
<!-- Ionicons -->
<link href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" rel="stylesheet">
<!-- overlayScrollbars -->
<link rel="stylesheet" href="dist/css/adminlte.min.css">
<!-- Google Font: Source Sans Pro -->
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">
</head>
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
<!-- Navbar -->
<nav class="main-header navbar navbar-expand navbar-white navbar-light">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" data-widget="pushmenu" href="#"><i class="fas fa-bars"></i></a>
</li>
</ul>
<!-- Right navbar links -->
<ul class="navbar-nav ml-auto">
<li class="nav-item d-none d-sm-inline-block">
<a href="#" class="nav-link">Logout</a>
</li>
</ul>
</nav>
<!-- /.navbar -->

<!-- Main Sidebar Container -->
<!-- Main Sidebar Container -->
<aside class="main-sidebar sidebar-dark-primary elevation-4">
<!-- Brand Logo -->
<a href="#" class="brand-link">
<span class="brand-text font-weight-light">Buku Kita</span>
</a>

<!-- Sidebar -->
<div class="sidebar">
```

```

<!-- Sidebar user (optional) -->
<div class="user-panel mt-3 pb-3 mb-3 d-flex">
  <div class="info">
    <a href="#" class="d-block">@herziwp</a>
  </div>
</div>

<!-- Sidebar Menu -->
<nav class="mt-2">
  <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu"
data-accordion="false">
    <!-- Add icons to the links using the .nav-icon class
        with font-awesome or any other icon font library -->
    <li class="nav-item">
      <a href="index.php" class="nav-link">
        <i class="nav-icon fas fa-tachometer-alt"></i>
        <p>
          Dashboard
        </p>
      </a>
    </li>
    <li class="nav-item">
      <a href="index.php" class="nav-link">
        <i class="nav-icon fas fa-copy"></i>
        <p>
          Kategori
        </p>
      </a>
    </li>
    <li class="nav-item">
      <a href="index.php" class="nav-link">
        <i class="nav-icon fas fa-book"></i>
        <p>
          Buku
        </p>
      </a>
    </li>
    <li class="nav-item">
      <a href="index.php" class="nav-link">
        <i class="nav-icon fas fa-user"></i>
        <p>
          About Me
        </p>
      </a>
    </li>
  </ul>
</nav>
<!-- /.sidebar-menu -->
</div>
<!-- /.sidebar -->
</aside>

<!-- Content Wrapper. Contains page content -->

```

```

<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Halaman Utama</h1>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">Title</h3>
      </div>
      <div class="card-body">
        Start creating your amazing application!
      </div>
      <!-- /.card-body -->
      <div class="card-footer">
        Footer
      </div>
      <!-- /.card-footer-->
    </div>
    <!-- /.card -->

  </section>
  <!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
  <b>Bukita 2019</b> All rights reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<!-- jQuery -->
<script src="plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->

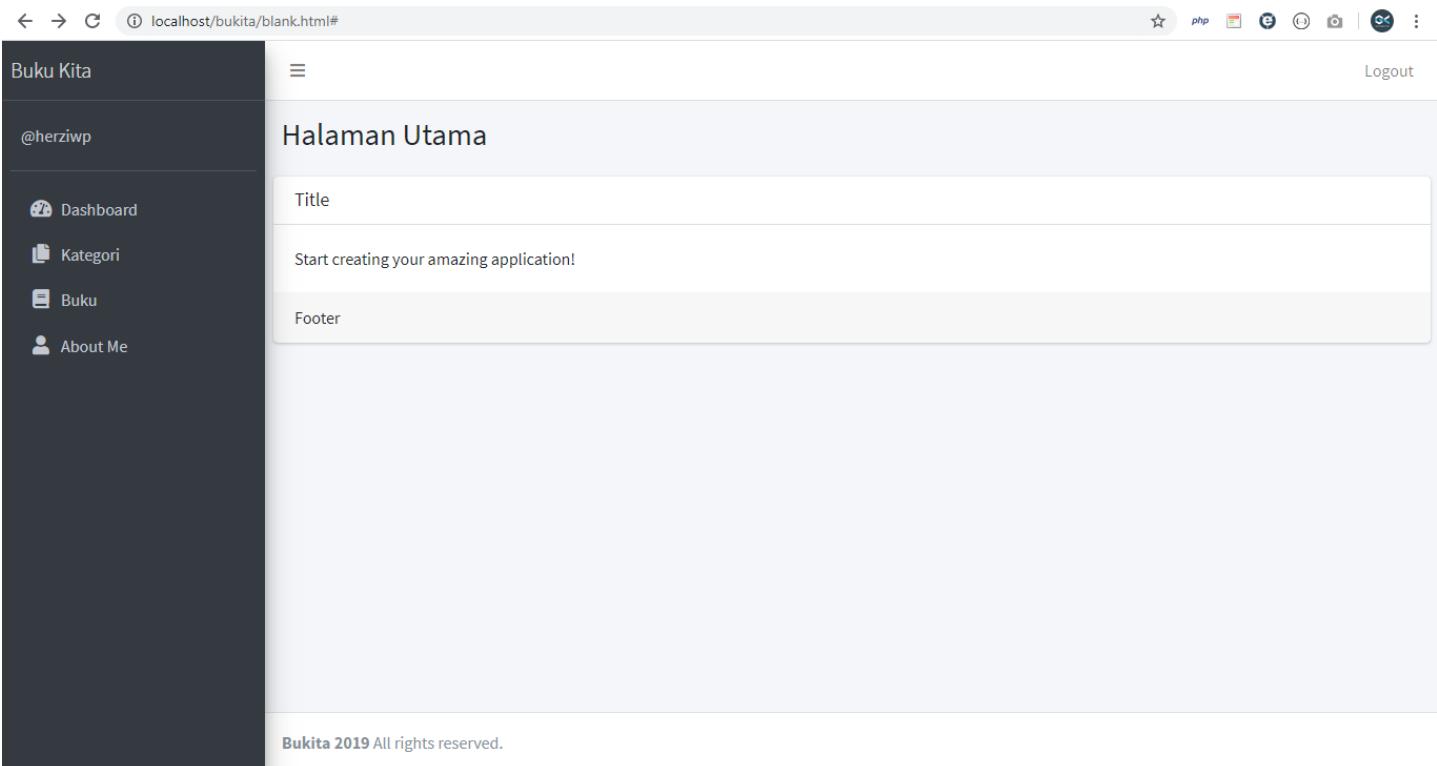
```

```
<script src="dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="dist/js/demo.js"></script>
</body>
</html>
```

**14.**Kita coba buka blank.html, pastikan xampp sudah nyala. Ketikkan di browser :

<http://localhost/bukita/blank.html>

**15.**Tampilan Halaman index dari buku kita



Gambar 69 tampilan awal buku kita

## 2.4 Persiapan Aplikasi PHP OOP MVC

1. Yang harus dilakukan terlebih dahulu yaitu membuat 2 folder utama yaitu folder **app** & **public**
2. Nah folder dist, plugins kita letakkan didalam folder public

his PC > Windows (C:) > xampp > htdocs > bukita > **public**

Name	Date modified	Type	Size
plugins	08/01/2020 20:15	File folder	
dist	08/01/2020 20:15	File folder	

Gambar 70 folder public

Penjelasan :

- Difolder ini adalah folder yang bisa diakses oleh user, misalnya file css, js, plugin & gambar

3. Didalam folder app, kita buat folder ini dulu :

Name	Date modified	Type	Size
config	08/01/2020 23:41	File folder	
controllers	08/01/2020 23:41	File folder	
core	08/01/2020 23:41	File folder	
models	08/01/2020 23:41	File folder	
views	08/01/2020 23:41	File folder	

Gambar 71 folder app

Penjelasan :

- Menyimpan folder & file – file penting php dalam menjalankan sistem ini, misalnya untuk controller, model, view, config database, dll

4. Pada Folder **public** kita buat sebuah file baru yang akan diakses oleh user yaitu **index.php**

Ketikkan sebagai berikut :

```
<?php
require_once '../app/init.php';
$app = new App;
```

Penjelasan :

- Nah isinya memanggil sebuah file init.php yang berada pada folder app, maka kita perlu membuat sebuah file init.php
- Lalu menjalankan class App, akan kita buat Class App nya.

5. Pada folder **app**, kita buat file lagi dengan nama **init.php**, isinya :

```
<?php
require_once 'core/App.php';
require_once 'core/Controller.php';
```

Penjelasan :

- File init.php ini fungsinya akan memanggil semua file yang kita butuhkan pada folder app
- Jadi si file index pada folder public, memanggil file init, nah pada file init akan memanggil file – file yang kita perlukan, semuanya jadi saling terhubung.
- Pada init.php memanggil lagi Class App dan Controller yang sebagai Class Utama Pembentuk Aplikasi MVC

6. Pada folder **core**, buat file **App.php**, isinya sementara gini dulu :

```
<?php

class App{
    public function __construct()
    {
        echo 'Praktikum WEB';
    }
}
```

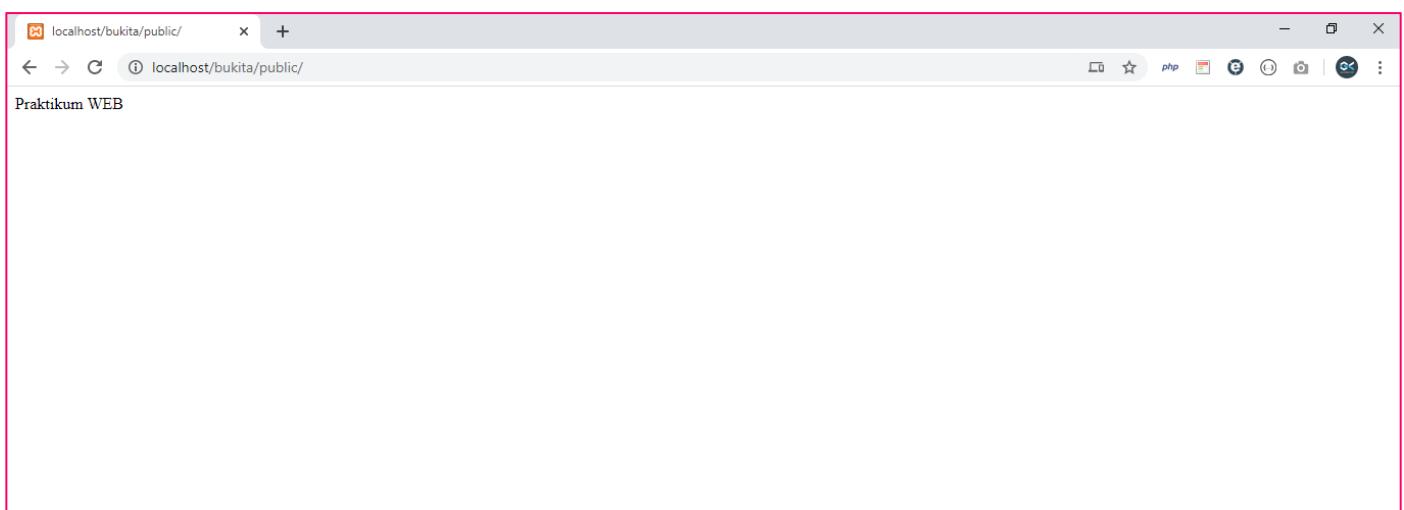
**7.** Masih di folder **core**, buat Class **Controller.php** :

```
<?php

class Controller{
}
```

**8.** Coba jalankan dilocalhost : <http://localhost/bukita/public/index.php>

Hasil sementara :



**9.** Kalau App sudah berhasil terpanggil seperti diatas, maka kembali ke class App.php, ketikkan sebagai berikut :

```
<?php

class App{

    protected $controller = 'Home';
    protected $method = 'index';
    protected $params = [];

    public function __construct(){
        $url = $this->parseURL();

        if(file_exists('../app/controllers/' . $url[0] . '.php')){
            $this->controller = $url[0];
            unset($url[0]);
        }
    }
}
```

```

    }

require_once '../app/controllers/' . $this->controller . '.php';
$this->controller = new $this->controller;

if(isset($url[1])){
    if(method_exists($this->controller, $url[1])){
        $this->method = $url[1];
        unset($url[1]);
    }
}

if(!empty($url)){
    $this->params = array_values($url);
}

call_user_func_array([$this->controller, $this->method], $this->params);
}

public function parseURL(){
    if( isset($_GET['url'])){
        $url = rtrim($_GET['url'], '/');
        $url = filter_var($url, FILTER_SANITIZE_URL);
        $url = explode('/', $url);
        return $url;
    }
}
}
}

```

Fungsi Class App :

- Membuat sistem routing URL
- Membersihkan URL atau sebutannya dengan Pretty URL
- Memanggil Controller sesuai dengan URL yang kita jalankan
- Disitu kita set controller Home, method defaultnya Index

**10.** Membuat file .htaccess pada folder App, simpan dengan nama .htaccess saja

Options -Indexes

Fungsinya :

- Agar user tidak bisa membuka folder App kita misalnya <http://localhost/bukita/app/> jika user membuka halaman tersebut, akan muncul pemberitahuan Access Forbidden!

**11.** Membuat file .htaccess pada folder Public,

Options -Multiviews

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php?url=$1 [L]
```

Fungsinya :

- Kalau ini fungsinya menuliskan ulang URL kita, misal memanggil <http://localhost/bukita/app/index.php/> maka akan di ReWrite oleh htaccess menjadi <http://localhost/bukita/app/index> saja tanpa file .php dibelakangnya

## 12. Mengisi data Class Controller.php pada folder core/Controller.php :

```
<?php

class Controller{
    public function view($view, $data = [])
    {
        require_once '../app/views/' . $view . '.php';
    }

    public function model($model)
    {
        require_once '../app/models/' . $model . '.php';
        return new $model;
    }
}
```

Fungsinya :

- Sebagai penghubung antar Controller ke Views dengan mengirimkan data yang ingin ditampilkan jika ada
- Sebagai penghubung juga ke Model untuk proses data MySQL sesuai permintaan Masing - masing controller (Misal data kategori meminta untuk ditampilkan recordsnya, maka kategori controller meminta request ke Class Controller method Model, lalu dari method model akan mengarahkan ke Masing - masing model)

## 13. Membuat Controller **Home.php** di folder **app/controllers**, berikut coding awal :

```
<?php

class Home extends Controller {
    public function index()
    {
        echo "Ini halaman home";
    }
}
```

Coba dijalankan dulu, apakah sudah berhasil?

## 2.5 Memecah bagian header, konten, sidebar & footer dengan php

Oke itu tadi kita sudah melakukan persiapannya, sekarang kita akan membuat bagian template nya terpecah antar konten, header, sidebar, footer

Buka file **blank.html** yang sebelumnya sudah kita sesuaikan

### 2.5.1 Bagian Header

1. Buat Folder baru di **app/views/** simpan dengan nama folder **templates**
2. Buat File **header.php** didalam **app/views/templates**
3. Buka dulu blank.html, pada coding :

```
<!DOCTYPE html>
.... sampai dengan ....
</nav>
```

4. Kalian CUT & PASTE didalam header.php

### 2.5.2 Bagian Sidebar

1. Buat file **sidebar.php** pada folder **app/views/templates**
2. Buka file blank.html tadi dibagian

```
<aside class="main-sidebar sidebar-dark-primary elevation-4">
.... sampai dengan ....
</aside>
```

3. Yup, sama CUT & PASTE ke sidebar.php

### 2.5.3 Bagian Konten

1. Buat folder baru didalam **app/views/** dengan nama **home**
2. Buat file **index.php** didalam **app/views/home**
3. Buka file blank.html

```
<div class="content-wrapper">
.... sampai dengan ....
</div>
<!-- /.content-wrapper -->
```

4. Pastikan calian CUT sampai **sebelum** `<footer class="main-footer">`

### 2.5.4 Bagian Footer

1. Buat file baru **footer.php** simpan di **app/views/templates**
2. Pada bagian blank.html

```
<footer class="main-footer">
```

.... sampai dengan ....

```
</html>
```

3. Cut dari bagian <footer> sampai habis kebawah, paste di footer.php

Sekarang pada folder app/controller/Home , ke Home.php method index tuliskan coding berikut ini :

```
<?php

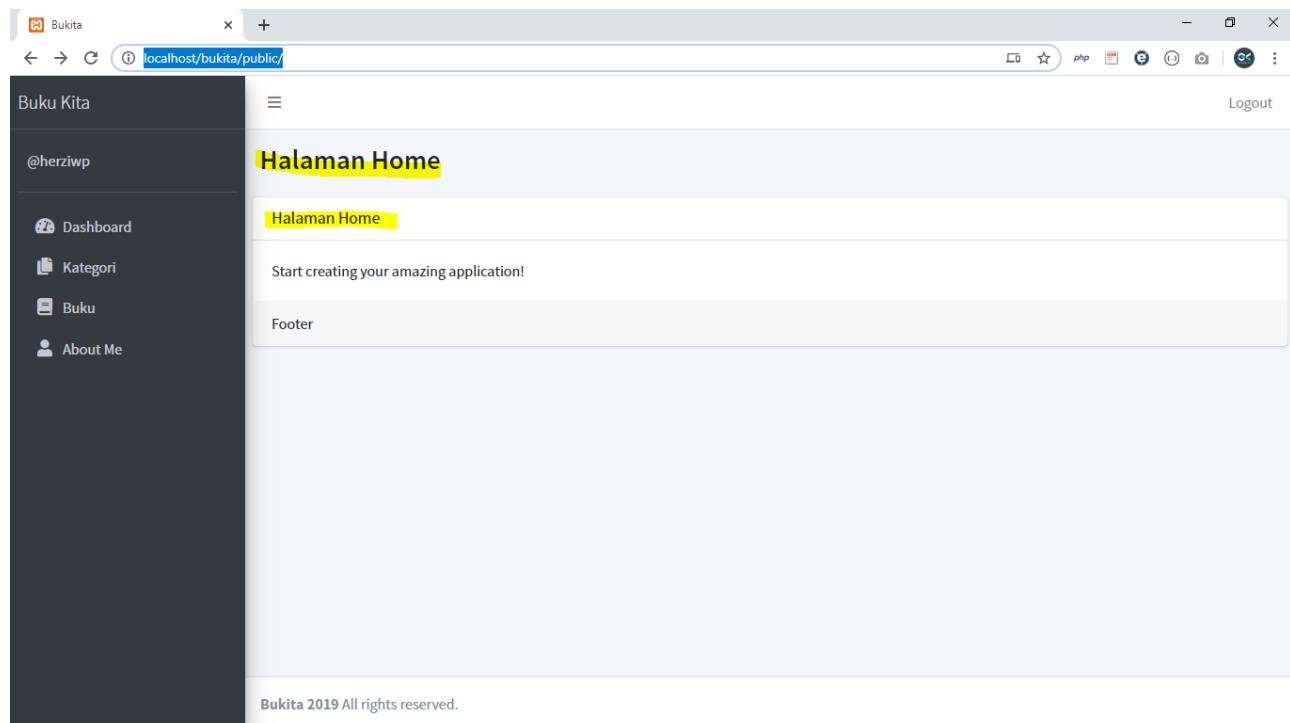
class Home extends Controller {
    public function index()
    {
        $data['title'] = 'Halaman Home';

        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('home/index', $data);
        $this->view('templates/footer');
    }
}
```

Pada File yang ada di app/views/home/index.php silahkan dibuka lalu edit bagian Tulisan Halaman Utama jadi seperti berikut ini : **<h1><?= \$data['title']; ?></h1>**

Lalu di h3 class card-title juga paste berikut ini : **<h3 class="card-title"><?= \$data['title']; ?></h3>**

Panggil di <http://localhost/bukita/public/> hasilnya :



Berarti kita sudah berhasil membuat halaman Home dengan Template yang sudah terpisah. Setelah ini kita akan membuat **CRUD Kategori**

## 2.6 Membuat CRUDS Kategori

Sebelumnya kita harus membuat konesinya dulu, maka pada folder **app/config** buat file baru dengan nama **config.php**

Isinya sebagai berikut :

```
<?php

define('base_url', 'http://localhost/bukita/public');

define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'buku_kita');
```

Penjelasan :

- Base\_url adalah url default kita
- Db\_name pastikan nama databasenya sama dengan yang ada di phpmyadmin kalian

Membuat Class **Database.php** simpan di **app/core**

```
<?php

class Database{
    private $host = DB_HOST;
    private $user = DB_USER;
    private $pass = DB_PASS;
    private $dbnm = DB_NAME;

    private $dbh;
    private $stmt;

    public function __construct()
    {
        $dsn = 'mysql:host=' . $this->host . ';dbname=' . $this->dbnm;

        $option = [
            PDO::ATTR_PERSISTENT => true,
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
        ];

        try{
            $this->dbh = new PDO($dsn,$this->user,$this->pass, $option);
        }catch(PDOException $e){
            die($e->getMessage());
        }
    }
}
```

```
public function query($query)
{
    $this->stmt = $this->dbh->prepare($query);
}

public function bind($param, $value, $type = null){
    if(is_null($type)){
        switch (true) {
            case is_int($value):
                $type = PDO::PARAM_INT;
                break;
            case is_bool($value):
                $type = PDO::PARAM_BOOL;
                break;
            case is_null($value):
                $type = PDO::PARAM_NULL;
                break;
            default:
                $type = PDO::PARAM_STR;
        }
    }

    $this->stmt->bindValue($param, $value, $type);
}

public function execute()
{
    $this->stmt->execute();
}

public function resultSet()
{
    $this->execute();
    return $this->stmt->fetchAll(PDO::FETCH_ASSOC);
}

public function single()
{
    $this->execute();
    return $this->stmt->fetch(PDO::FETCH_ASSOC);
}

public function rowCount()
{
    return $this->stmt->rowCount();
}
```

Filenya ada didalam folder Sesi 2, silahkan copy dari situ saja.

Setelah itu Ke folder app lalu file **init.php** panggil / require class Config & Database dibagian bawah :

```
require_once 'config/config.php';
require_once 'core/Database.php';
```

Pada **header.php** tambahkan `<?= base_url; ?>` pada link css contohnya seperti ini :

```
<!-- Font Awesome -->
<link rel="stylesheet" href="<?= base_url; ?>/plugins/fontawesome-free/css/all.min.css">
<!-- Ionicons -->
<link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
<!-- overlayScrollbars -->
<link rel="stylesheet" href="<?= base_url; ?>/dist/css/adminlte.min.css">
<!-- Google Font: Source Sans Pro -->
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">|
```

Lakukan hal yang sama di **footer.php** pada link javascript nya :

```
<!-- jQuery -->
<script src="<?= base_url; ?>/plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="<?= base_url; ?>/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="<?= base_url; ?>/dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="<?= base_url; ?>/dist/js/demo.js"></script>
```

## 2.6.1 Index Kategori

Silahkan Controller Home dicopy dulu lalu ganti nama dengan Kategori.php simpan di folder **app/controllers**

Isi dari **Kategori.php** pada folder controllers :

```

<?php

class Kategori extends Controller {
    public function index()
    {
        $data['title'] = 'Data Kategori';
        $data['kategori'] = $this->model('KategoriModel')->getAllKategori();
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('kategori/index', $data);
        $this->view('templates/footer');
    }

    public function cari()
    {
    }
}
```

```

public function edit($id){
}

public function tambah(){
}

public function simpanKategori(){
}

public function updateKategori(){
}

public function hapus($id){
}
}

```

Penjelasan :

- Berisi method - method dengan fungsi yang berbeda - beda
- Pada method index kita memanggil model **KategoriModel** lalu menuju method **getAllKategori()** didalam Class KategoriModel untuk mendapatkan data

Kalau sudah, buat Model dari kategori di **app/models** simpan dengan nama **KategoriModel.php** lsinya :

```

● ● ●

<?php

class KategoriModel {

    private $table = 'kategori';
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function getAllKategori()
    {
        $this->db->query('SELECT * FROM ' . $this->table);
        return $this->db->resultSet();
    }
}

```

```

public function getKategoriById($id)
{
}

public function tambahKategori($data)
{
}

public function updateDataKategori($data)
{
}

public function deleteKategori($id)
{
}

public function cariKategori()
{
}
}

```

Buat folder baru di views dengan nama kategori, lalu silahkan copy file **index.php** yang ada di **views/home** simpan ke **views/kategori**.

Pada sidebar silahkan edit link kategori dan home jadi seperti ini :

```

<li class="nav-item">
    <a href="= base_url(); ?/home" class="nav-link">
        <i class="nav-icon fas fa-tachometer-alt"></i>
        <p>
            Dashboard
        </p>
    </a>
</li>
<li class="nav-item">
    <a href="= base_url(); ?/kategori" class="nav-link">
        <i class="nav-icon fas fa-copy"></i>
        <p>
            Kategori
        </p>
    </a>
</li>

```

Sekarang coba refresh, dan klik kategori pada sidebar kalian.

Kita ubah tampilan index.php dari kategori, silahkan copy dari file saya bagikan

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="row">
            <div class="col-sm-12">

            </div>
        </div>
        <!-- Default box -->

        <div class="card">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title'] ?></h3> <a href=<?= base_url;
?>/kategori/tambah" class="btn float-right btn-xs btn btn-primary">Tambah Kategori</a>
            </div>
            <div class="card-body">

                <form action=<?= base_url; ?>/kategori/cari" method="post">
                    <div class="row mb-3">
                        <div class="col-lg-6">
                            <div class="input-group">
                                <input type="text" class="form-control" placeholder="" name="key" >
                            <div class="input-group-append">
                                <button class="btn btn-outline-secondary" type="submit">Cari Data</button>
                                <a class="btn btn-outline-danger" href=<?= base_url; ?>/kategori" >Reset</a>
                            </div>
                        </div>
                    </div>
                </div>
                </form>
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th style="width: 10px">#</th>
                            <th>Kategori</th>
                            <th style="width: 150px">Action</th>
                        </tr>
                    </thead>

```

```

<tbody>
<?php $no=1; ?>
<?php foreach ($data['kategori'] as $row) :?>
<tr>
<td><?= $no; ?></td>
<td><?= $row['nama_kategori'];?></td>
<td>
<a href=<?= base_url; ?>/kategori/edit/<?= $row['id'] ?>" class="badge badge-info">Edit</a> <a href=<?= base_url; ?>/kategori/hapus/<?= $row['id'] ?>" class="badge badge-danger" onclick="return confirm('Hapus data?');">Hapus</a>
</td>
</tr>
<?php $no++; endforeach; ?>
</tbody>
</table>
</div>
<!-- /.card-body -->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

Penjelasan :

- Disitu sudah saya masukkan codingan form untuk pencarian tapi masih belum berfungsi
- Pada tabel kita buat foreach atau perulangan yang datanya kita dapat dari **KategoriModel**

#	Kategori	Action
1	Horor	Edit Hapus
2	Rhome irama	Edit Hapus
3	lucuu	Edit Hapus
4	asiap	Edit Hapus

## 2.6.2 Tambah Kategori

Pada Index kategori tadi kita sudah menambahkan link untuk menuju ke Form Tambah tapi masih belum berfungsi karena pada kategori controller belum kita tambahkan isi method tambah()



```

public function tambah( )
{
    $data['title'] = 'Tambah Kategori';
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('kategori/create', $data);
    $this->view('templates/footer');
}

```

Kalau sudah, Buat File create.php simpan di folder app/views/kategori (isi nya kalian copy dibahan folder sesi 2)

```

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action=<?= base_url; ?>/kategori/simpanKategori" method="POST"
enctype="multipart/form-data">
                <div class="card-body">
                    <div class="form-group">
                        <label>Nama Kategori</label>
                        <input type="text" class="form-control" placeholder="masukkan kategori..." name="nama_kategori">
                    </div>
                </div>
                <!-- /.card-body -->

                <div class="card-footer">
                    <button type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>
    </section>

```

```

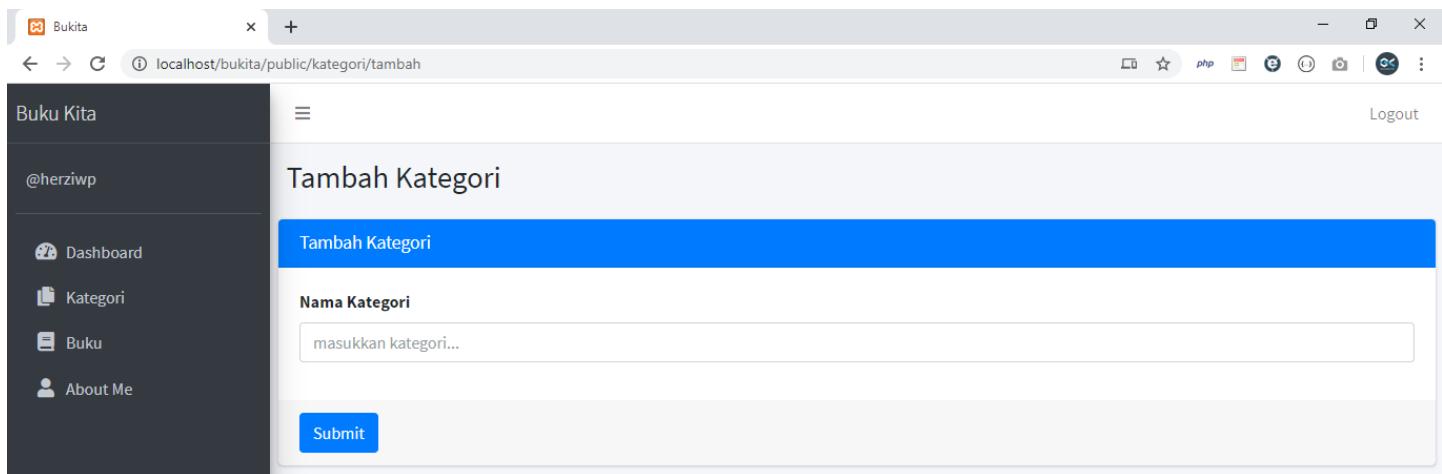
</div>

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
```

Penjelasan :

- Pastikan pada action form tujuannya sama dengan nama method yang ada di Kategori Controller
- Perhatikan name pada input type text jangan sampai salah

Coba kita jalankan dulu <http://localhost/bukita/public/kategori/tambah> :



Yup, untuk saat ini masih belum bisa jalan kita harus membuat method simpanKategori di Kategori Controller lalu harus membuat methode Simpan juga di Model Kategori.

Kita buat method simpanKategori di file **Kategori.php** Controller

```

public function simpanKategori()
{
    if( $this->model('KategoriModel')->tambahKategori($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','ditambahkan','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','ditambahkan','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}
```

Penjelasan :

- Disini proses pengiriman data dari form tambah tadi menuju model.

- Flasher adalah alert untuk mengirimkan pesan apakah kita berhasil atau tidak dalam proses simpan ini. Kita akan buat file flasher setelah ini.
- Header untuk redirect setelah kita menambahkan data akan dialihkan ke halaman tertentu seperti contoh diatas akan kembali diarahkan ke halaman kategori index

Kalau sudah, buat method tambahKategori(\$data) pada **KategoriModel.php**

```
public function tambahKategori($data)
{
    $query = "INSERT INTO kategori (nama_kategori) VALUES(:nama_kategori)";
    $this->db->query($query);
    $this->db->bind('nama_kategori', $data['nama_kategori']);
    $this->db->execute();

    return $this->db->rowCount();
}
```

Penjelasan :

- Proses Query Insert MySQL untuk menyimpan data ke tabel kategori

Membuat file **Flasher.php** untuk alert berhasil / tidak disetiap proses yang kita lakukan.

Silahkan buat file baru dengan nama Flasher.php simpan di folder **app/core**,

```
<?php

class Flasher{

    public static function setMessage($pesan, $aksi, $type)
    {

        $_SESSION['msg'] = [
            'pesan' => $pesan,
            'aksi' => $aksi,
            'type' => $type
        ];
    }

    public static function Message(){
        if( isset($_SESSION['msg']) )
        {

            echo '<div class="alert alert-'. $_SESSION['msg']['type'] . ' alert-dismissible fade show"
role="alert">
                Data <strong>'. $_SESSION['msg']['pesan'] .'</strong> '. $_SESSION['msg']['aksi'] .
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>';
        }
    }
}
```

```

        </div>';

        unset($_SESSION['msg']);
    }

}

```

Pada file **init.php**, kita require lagi tambahkan paling bawah :

```
require_once 'core/Flasher.php';
```

Lalu pada index.php didalam folder views/kategori, dibagian bawah section class="content" tambahkan

```
<?php Flasher::Message(); ?>
```

```

<section class="content">
<div class="row">
    <div class="col-sm-12">
        <?php
            Flasher::Message();
        ?>
        </div>
    </div>
    |   <!-- Default box -->

```

Karena Flasher menggunakan Session, maka pada index.php difolder **public/index.php** tambahkan :

```
if( !session_id() ) session_start();
```

```

<?php

if( !session_id() ) session_start();

require_once '../app/init.php';

$app = new App;

```

Proses tambah sudah selesai, coba tambahkan beberapa data dulu

The screenshot shows a web application interface. On the left is a sidebar with a dark background containing the text "Buku Kita" and a user profile "@herziwp". Below this are links for "Dashboard", "Kategori", "Buku", and "About Me". The main content area has a title "Data Kategori". A green banner at the top says "Data Berhasil ditambahkan". Below it is a table with one row, showing the category "Horor". The table has columns for "#", "Kategori", and "Action". The "Action" column contains "Edit" and "Hapus" buttons.

### 2.6.3 Edit Kategori

Tambahkan method edit(\$id) pada Kategori Controller

```
public function edit($id)
{
    $data['title'] = 'Detail Kategori';
    $data['kategori'] = $this->model('KategoriModel')->getKategoriById($id);
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('kategori/edit', $data);
    $this->view('templates/footer');
}
```

Penjelasan :

- Edit(\$id) , maksud \$id adalah id yang kita dapatkan dari URL untuk mengetahui data ID berapa yang akan kita edit.
- Alur sistemnya masih sama, yaitu kita memanggil lagi ke Kategori Model lalu ke method getKategoriById(\$id) untuk memanggil datanya.

Membuat method getKategoriById(\$id) pada KategoriModel.php

```
public function getKategoriById($id)
{
    $this->db->query('SELECT * FROM ' . $this->table . ' WHERE id=:id');
    $this->db->bind('id', $id);
    return $this->db->single();
}
```

Penjelasan :

- Query untuk menampilkan data tapi berdasarkan ID tertentu & hanya mengambil satu data saja karena ID Primary Key maka tidak mungkin lebih dari 1.

Membuat view **edit.php** untuk kategori, simpan di **views/kategori**

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Halaman Kategori</h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action="= base_url; ?&gt;/kategori/updateKategori" method="POST"
                  enctype="multipart/form-data"&gt;

                &lt;input type="hidden" name="id" value="<?= $data['kategori'][['id']]; ?&gt;"&gt;
                &lt;div class="card-body"&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label&gt;Nama Kategori&lt;/label&gt;
                        &lt;input type="text" class="form-control" placeholder="masukkan kategori..." name="nama_kategori" value="<?= $data['kategori'][['nama_kategori']]; ?&gt;"&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
                &lt;!-- /.card-body --&gt;

                &lt;div class="card-footer"&gt;
                    &lt;button type="submit" class="btn btn-primary"&gt;Submit&lt;/button&gt;
                &lt;/div&gt;
            &lt;/form&gt;
        &lt;/div&gt;
    &lt;/section&gt;
    &lt;!-- /.content --&gt;
&lt;/div&gt;
&lt;!-- /.content-wrapper --&gt;</pre

```

Mengirimkan data ke hasil edit ke model maka perlu menambahkan method updateKategori di Kategori Controller.

```
● ● ●

public function updateKategori(){
    if( $this->model('KategoriModel')->updateDataKategori($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','diupdate','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','diupdate','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}
```

Penjelasan :

- Mengirimkan data ke KategoriModel methode updateDataKategori

Maka kita perlu membuat method updateDataKategori didalam KategoriModel.php

```
● ● ●

public function updateDataKategori($data)
{
    $query = "UPDATE kategori SET nama_kategori=:nama_kategori WHERE id=:id";
    $this->db->query($query);
    $this->db->bind('id',$data['id']);
    $this->db->bind('nama_kategori',$data['nama_kategori']);
    $this->db->execute();

    return $this->db->rowCount();
}
```

Penjelasan :

- Query Proses Update yang diupdate hanya nama\_kategori.

Sampai sini Edit Kategori telah selesai.

## 2.6.4 Hapus Kategori

Untuk hapus kita tidak perlu membuat viewnya lagi, disini hanya melakukan proses hapus dari link yang dikirimkan oleh index kategori pada button hapus.

Maka kita tinggal membuat method hapus pada Kategori.php Controller



```
public function hapus($id){
    if( $this->model('KategoriModel')->deleteKategori($id) > 0 ) {
        Flasher::setMessage('Berhasil','dihapus','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','dihapus','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}
```

Penjelasan :

- Mengirim id yang ingin dihapus ke KategoriModel method deleteKategori(\$id)

Membuat method deleteKategori(\$id) pada **KategoriModel.php**



```
public function deleteKategori($id)
{
    $this->db->query('DELETE FROM ' . $this->table . ' WHERE id=:id');
    $this->db->bind('id',$id);
    $this->db->execute();

    return $this->db->rowCount();
}
```

Penjelasan :

- Query Delete by ID yang dikirimkan method hapus oleh Kategori Controller

Silahkan dicoba hehe

## 2.6.5 Cari Kategori

Yup, kita **tidak perlu** bikin view lagi tapi cukup pakai file index.php di folder view/kategori karena disitu tadi kita sudah menambahkan Form untuk Pencarian Data nya, tapi hanya sedikit penjelasan :

```
<form action="= base_url(); ?/kategori/cari" method="post">
    <div class="row mb-3">
        <div class="col-lg-6">
            <div class="input-group">
                <input type="text" class="form-control" placeholder="" name="key" >
                <div class="input-group-append">
```

```

<button class="btn btn-outline-secondary" type="submit">Cari Data</button>
<a class="btn btn-outline-danger" href="= base_url(); ?/kategori" >Reset</a>
</div>
</div>
</div>
</div>
</form>

```

Penjelasan :

- Pada action tujuannya ke Kategori Controller lalu method Cari berarti kita harus membuat method cari pada Kategori Controller
- Ada 1 buah Text Field dengan name='key' yang akan menampung keyword atau kata kunci yang akan kita cari pada nama kategori

Berikut isi method cari pada Kategori.php Controller :

```

public function cari()
{
    $data['title'] = 'Data Kategori';
    $data['kategori'] = $this->model('KategoriModel')->cariKategori();
    $data['key'] = $_POST['key'];
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('kategori/index', $data);
    $this->view('templates/footer');
}

```

Penjelasan :

- Mengirimkan keyword tadi ke KategoriModel Method cariKategori

Pada KategoriModel Method cariKategori() isinya sebagai berikut :

```

public function cariKategori()
{
    $key = $_POST['key'];
    $this->db->query("SELECT * FROM " . $this->table . " WHERE nama_kategori LIKE :key");
    $this->db->bind('key', "%$key%");
    return $this->db->resultSet();
}

```

Penjelasan :

- Kamu bisa menambahkan 2 kemungkinan disitu, misal ingin mencari berdasarkan id atau nama, kalau dicontoh ini hanya berdasarkan nama\_kategori yang mirip dengan keyword saja.

Ketika saya mengetik kata lucu maka yang tampil :

The screenshot shows a web browser window titled 'Bukita'. The URL is 'localhost/bukita/public/kategori/cari'. On the left, there's a sidebar with 'Buku Kita' at the top, followed by user info '@herziwp' and navigation links: 'Dashboard', 'Kategori', 'Buku', and 'About Me'. The main content area has a title 'Data Kategori' and a sub-section 'Data Kategori'. It features a search bar with 'Cari Data' and 'Reset' buttons. Below the search is a table with two columns: '# Kategori' and 'Action'. There is one row with data: '# 1 lucuu' and buttons 'Edit' and 'Hapus'.

Okay, berarti CRUDS pada Kategori ini sudah selesai. Silahkan ditest langsung aja nih siapa tau ada error hehe

Jadi berikut coding lengkap dari :

### Kategori.php (Controller)

```
<?php

class Kategori extends Controller {
    public function index()
    {
        $data['title'] = 'Data Kategori';
        $data['kategori'] = $this->model('KategoriModel')->getAllKategori();
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('kategori/index', $data);
        $this->view('templates/footer');
    }
    public function cari()
    {
        $data['title'] = 'Data Kategori';
        $data['kategori'] = $this->model('KategoriModel')->cariKategori();
        $data['key'] = $_POST['key'];
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('kategori/index', $data);
        $this->view('templates/footer');
    }
}
```

```

public function edit($id)
{
    $data['title'] = 'Detail Kategori';
    $data['kategori'] = $this->model('KategoriModel')->getKategoriById($id);
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('kategori/edit', $data);
    $this->view('templates/footer');
}

public function tambah()
{
    $data['title'] = 'Tambah Kategori';
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('kategori/create', $data);
    $this->view('templates/footer');
}

public function simpanKategori()
{
    if( $this->model('KategoriModel')->tambahKategori($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','ditambahkan','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','ditambahkan','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}

public function updateKategori(){
    if( $this->model('KategoriModel')->updateDataKategori($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','diupdate','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','diupdate','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}

public function hapus($id){
    if( $this->model('KategoriModel')->deleteKategori($id) > 0 ) {
        Flasher::setMessage('Berhasil','dihapus','success');
        header('location: '. base_url . '/kategori');
        exit;
    }else{
        Flasher::setMessage('Gagal','dihapus','danger');
        header('location: '. base_url . '/kategori');
        exit;
    }
}
}

```

## KategoriModel.php (Model)



```
<?php

class KategoriModel {

    private $table = 'kategori';
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function getAllKategori()
    {
        $this->db->query('SELECT * FROM ' . $this->table);
        return $this->db->resultSet();
    }

    public function getKategoriById($id)
    {
        $this->db->query('SELECT * FROM ' . $this->table . ' WHERE id=:id');
        $this->db->bind('id', $id);
        return $this->db->single();
    }

    public function tambahKategori($data)
    {
        $query = "INSERT INTO kategori (nama_kategori) VALUES(:nama_kategori)";
        $this->db->query($query);
        $this->db->bind('nama_kategori', $data['nama_kategori']);
        $this->db->execute();

        return $this->db->rowCount();
    }

    public function updateDataKategori($data)
    {
        $query = "UPDATE kategori SET nama_kategori=:nama_kategori WHERE id=:id";
        $this->db->query($query);
        $this->db->bind('id', $data['id']);
        $this->db->bind('nama_kategori', $data['nama_kategori']);
        $this->db->execute();

        return $this->db->rowCount();
    }

    public function deleteKategori($id)
    {
        $this->db->query('DELETE FROM ' . $this->table . ' WHERE id=:id');
        $this->db->bind('id', $id);
        $this->db->execute();

        return $this->db->rowCount();
    }

    public function cariKategori()
    {
        $key = $_POST['key'];
        $this->db->query("SELECT * FROM " . $this->table . " WHERE nama_kategori LIKE :key");
        $this->db->bind('key', "%$key%");
        return $this->db->resultSet();
    }
}
```

## Coding Lengkap **init.php**

```
<?php

require_once 'core/App.php';
require_once 'core/Controller.php';
require_once 'config/config.php';
require_once 'core/Database.php';
require_once 'core/Flasher.php';
```

## 2.7 Membuat CRUDS Buku

Karena pada saat membuat CRUDS Kategori kita sudah melakukan setting Config, app, database, dll maka di CRUDS Buku ini kita tidak perlu melakukannya lagi.

Kita hanya akan membuat Buku Controller, Buku Model dan Viewnya saja lagi.

### 2.7.1 Index Buku

Silahkan Copy Controllernya **Kategori.php** , lalu ganti jadi **Buku.php** pada folder **app\controllers**

Nah pada index tinggal ganti jadi Buku, contohnya seperti ini :

```
<?php

class Buku extends Controller {

    public function index()
    {
        $data['title'] = 'Data Buku';
        $data['buku'] = $this->model('BukuModel')->getAllBuku();
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('buku/index', $data);
        $this->view('templates/footer');
    }
}
```

Lalu coba disave, sekarang kita ke Model, caranya sama silahkan Copy KategoriModel.php didalam folder **app\models** ganti namanya jadi **BukuModel.php**

```
<?php

class BukuModel {

    private $table = 'buku';
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function getAllBuku()
    {
        $this->db->query("SELECT buku.*, kategori.nama_kategori FROM " . $this->table . " JOIN kategori ON
            kategori.id = buku.kategori_id");
        return $this->db->resultSet();
    }
}
```

Penjelasan :

- Perhatikan nama class dan selectnya pada method **getAllBuku()** silahkan diganti sesuai dengan di foto atas

Pada Views lagi, silahkan buat folder baru dengan nama folder buku. Berikut coding index buku : **(silahkan copy dari folder coding)**

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
<div class="row">
    <div class="col-sm-12">
        <?php
            Flasher::Message();
        ?>
    </div>
</div>
    <!-- Default box -->

    <div class="card">
        <div class="card-header">
```

```

<h3 class="card-title"><?= $data['title'] ?></h3> <div class="btn-group float-right"><a href="<?= base_url; ?>/buku/tambah" class="btn float-right btn-xs btn btn-primary">Tambah Buku</a><a href="<?= base_url; ?>/buku/laporan" class="btn float-right btn-xs btn btn-info">Laporan Buku</a></div>
</div>
<div class="card-body">

<form action="<?= base_url; ?>/buku/cari" method="post">
<div class="row mb-3">
<div class="col-lg-6">
<div class="input-group">
<input type="text" class="form-control" placeholder="" name="key" >
<div class="input-group-append">
<button class="btn btn-outline-secondary" type="submit">Cari Data</button>
<a class="btn btn-outline-danger" href="<?= base_url; ?>/buku" >Reset</a>
</div>
</div>
</div>

</div>
</div>
</form>
<table class="table table-bordered">
<thead>
<tr>
<th style="width: 10px">#</th>
<th>Judul</th>
<th>Penerbit</th>
<th>Pengarang</th>
<th>Tahun</th>
<th>Kategori</th>
<th>Harga</th>
<th style="width: 150px">Action</th>
</tr>
</thead>
<tbody>
<?php $no=1; ?>
<?php foreach ($data['buku'] as $row) :?>
<tr>
<td><?= $no; ?></td>
<td><?= $row['judul'];?></td>
<td><?= $row['penerbit'];?></td>
<td><?= $row['pengarang'];?></td>
<td><?= $row['tahun'];?></td>
<td><div class="badge badge-warning"><?= $row['nama_kategori'];?></div></td>
<td><?= $row['harga'];?></td>
<td>
<a href="<?= base_url; ?>/buku/edit/<?= $row['id'] ?>" class="badge badge-info">Edit</a> <a href="<?= base_url; ?>/buku/hapus/<?= $row['id'] ?>" class="badge badge-danger" onclick="return confirm('Hapus data?');">Hapus</a>
</td>
</tr>
<?php $no++; endforeach; ?>

```

```

        </tbody>
    </table>
</div>
<!-- /.card-body -->
<div class="card-footer">
    Footer
</div>
<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

Penjelasan :

- Yang berubah hanya dibagian tabel saja menyesuaikan dengan tabel buku jadi misalnya mau copy dari views kategori itu bisa saja tinggal tabel nya disesuaikan.
- Dan juga link - link nya

Sesuikan link yang ada pada sidebar.php, ganti href buku nya jadi seperti ini :

```

<li class="nav-item">
    <a href="= base_url(); ?/buku" class="nav-link">
        <i class="nav-icon fas fa-book"></i>
        <p>
            Buku
        </p>
    </a>
</li>

```

Nah sekarang coba jalankan dulu.

Karena masih belum ada datanya, sekarang kita membuat Tambah buku ya... lanjuutt

## 2.7.2 Tambah Buku

Pada Buku Controller, ketikkan method tambah seperti berikut :

```
public function tambah(){
    $data['title'] = 'Tambah Buku';
    $data['kategori'] = $this->model('KategoriModel')->getAllKategori();
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('buku/create', $data);
    $this->view('templates/footer');
}
```

Penjelasan :

- Kenapa ada Kategori Model lalu methode getAllKategori() disitu? Karena nanti pada saat menambah buku kita perlu memasukkan kategorinya jadi kita memanggil si Data Kategori Tersebut.

Lalu pada Views buku nya buat file baru pada folder **app\views\buku** beri nama **create.php**

**(coding bisa dicopy dari folder coding)**

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action=<?= base_url; ?>/buku/simpanbuku" method="POST"
                  enctype="multipart/form-data">
                <div class="card-body">
                    <div class="form-group">
                        <label>Judul</label>
                        <input type="text" class="form-control" placeholder="masukkan judul buku..." name="judul">
                    </div>
                </div>
            </form>
        </div>
    </section>
</div>
```

```

<div class="form-group">
    <label>Penerbit</label>
    <input type="text" class="form-control" placeholder="masukkan penerbit buku..." name="penerbit">
</div>
<div class="form-group">
    <label>Pengarang</label>
    <input type="text" class="form-control" placeholder="masukkan pengarang buku..." name="pengarang">
</div>
<div class="form-group">
    <label>Tahun</label>
    <input type="text" class="form-control" placeholder="masukkan tahun buku..." name="tahun">
</div>
<div class="form-group">
    <label>Kategori</label>
    <select class="form-control" name="kategori_id">
        <option value="">Pilih</option>
        <?php foreach ($data['kategori'] as $row) :?>
            <option value=<?= $row['id']; ?>><?= $row['nama_kategori']; ?></option>
        <?php endforeach; ?>
    </select>
</div>
<div class="form-group">
    <label>Harga</label>
    <input type="text" class="form-control" placeholder="masukkan harga buku..." name="harga">
</div>
</div>
<!-- /.card-body -->

<div class="card-footer">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

Kalau sudah coba dijalankan dulu deh kita cek apakah tampil create buku nya. Setelah itu kita lanjut ke Proses Simpan pada Controller dan juga Modelnya.

Silahkan sesuaikan lagi pada buku Controller tadi, jadi seperti ini :

```

public function simpanBuku(){
    if( $this->model('BukuModel')->tambahBuku($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','ditambahkan','success');
        header('location: '. base_url . '/buku');
        exit;
    }else{
        Flasher::setMessage('Gagal','ditambahkan','danger');
        header('location: '. base_url . '/buku');
        exit;
    }
}

```

Berikut coding untuk proses simpan pada BukuModel :

```

public function tambahBuku($data)
{
    $query = "INSERT INTO buku (judul, penerbit, pengarang, tahun, kategori_id, harga) VALUES(:judul, :penerbit,
        :pengarang, :tahun, :kategori_id, :harga)";
    $this->db->query($query);
    $this->db->bind('judul', $data['judul']);
    $this->db->bind('penerbit', $data['penerbit']);
    $this->db->bind('pengarang', $data['pengarang']);
    $this->db->bind('tahun', $data['tahun']);
    $this->db->bind('kategori_id', $data['kategori_id']);
    $this->db->bind('harga', $data['harga']);
    $this->db->execute();

    return $this->db->rowCount();
}

```

Penjelasan :

- Pastikan , (koma) jangan salah, Pastikan juga urutan kolom dan values nya sama
- Pada bind itu juga jangan sampai salah tujuan misal (tujuan untuk kolom harga, lalu data yang dikirimkan juga harus name nya harga)

Silahkan dicoba tambahkan :

### Data Buku

Data Berhasil ditambahkan

#	Judul	Penerbit	Pengarang	Tahun	Kategori	Harga	Action
1	Buku Belajar Memahami Wanita	Gramedia	Herzi	2020	<a href="#">Love Story</a>	99000	<a href="#">Edit</a> <a href="#">Hapus</a>

Footer

### 2.7.3 Edit Buku

Ke Buku.php pada controllers, silahkan sesuikan method **edit()** menjadi seperti ini :

```
public function edit($id){

    $data['title'] = 'Detail Buku';
    $data['kategori'] = $this->model('KategoriModel')->getAllKategori();
    $data['buku'] = $this->model('BukuModel')->getBukuById($id);
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('buku/edit', $data);
    $this->view('templates/footer');
}
```

Lalu bikin views nya , pada folder **app\views\buku** coba buat file baru dengan nama **edit.php (copy dari folder coding)**

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action=<?= base_url; ?>/buku/updateBuku" method="POST"
enctype="multipart/form-data">
                <input type="hidden" name="id" value=<?= $data['buku']['id']; ?>">
                <div class="card-body">
                    <div class="form-group">
                        <label>Judul</label>
                        <input type="text" class="form-control" placeholder="masukkan judul buku..." name="judul" value=<?= $data['buku']['judul']; ?>">
                    </div>
                    <div class="form-group">
                        <label>Penerbit</label>
```

```

<input type="text" class="form-control" placeholder="masukkan penerbit buku..." name="penerbit" value="<?= $data['buku']['penerbit'];?>">
</div>
<div class="form-group">
<label>Pengarang</label>
<input type="text" class="form-control" placeholder="masukkan pengarang buku..." name="pengarang" value="<?= $data['buku']['pengarang'];?>">
</div>
<div class="form-group">
<label>Tahun</label>
<input type="text" class="form-control" placeholder="masukkan tahun buku..." name="tahun" value="<?= $data['buku']['tahun'];?>">
</div>
<div class="form-group">
<label>Kategori</label>
<select class="form-control" name="kategori_id">
<option value="">Pilih</option>
<?php foreach ($data['kategori'] as $row) :?>
<option value="<?= $row['id']; ?>" <?php if($data['buku']['kategori_id'] == $row['id']) { echo "selected"; } ?>><?= $row['nama_kategori']; ?></option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Harga</label>
<input type="text" class="form-control" placeholder="masukkan harga buku..." name="harga" value="<?= $data['buku']['harga'];?>">
</div>
</div>
<div class="card-footer">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>
</section>
<!-- /.content -->
</div>

```

Penjelasan :

- Kita mengisi text value dengan masing – masing data. Contoh text tahun maka isi dengan value **<?= \$data['buku']['tahun'];?>**

Masih belum bisa dijalankan ya pasti error, coba kita ke BukuModel.php lalu pada method awalnya **getKategoriById** ganti jadi **getBukuByid**

```

public function getBukuById($id)
{
    $this->db->query('SELECT * FROM ' . $this->table . ' WHERE id=:id');
    $this->db->bind('id', $id);
    return $this->db->single();
}

```

Coba jalankan dulu deh :

Detail Buku

<b>Judul</b>	Buku Belajar Memahami Wanita
<b>Penerbit</b>	Gramedia
<b>Pengarang</b>	Herzi
<b>Tahun</b>	2020
<b>Kategori</b>	Love Story
<b>Harga</b>	99000

Datanya tampil tapi masih belum bisa kita proses edit. Untuk proses pengiriman datanya silahkan pada Buku.php controller method edit buku :

```
public function updateBuku(){
    if( $this->model('BukuModel')->updateDataBuku($_POST) > 0 ) {
        Flasher::setMessage('Berhasil','diupdate','success');
        header('location: '. base_url . '/buku');
        exit;
    }else{
        Flasher::setMessage('Gagal','diupdate','danger');
        header('location: '. base_url . '/buku');
        exit;
    }
}
```

Lalu pada BukuModel.php method update Data Buku :

```
public function updateDataBuku($data)
{
    $query = "UPDATE buku SET judul=:judul, penerbit=:penerbit, pengarang=:pengarang, pengarang=:pengarang, tahun
    :=tahun, kategori_id=:kategori_id, harga=:harga WHERE id=:id";
    $this->db->query($query);
    $this->db->bind('id',$data['id']);
    $this->db->bind('judul', $data['judul']);
    $this->db->bind('penerbit', $data['penerbit']);
    $this->db->bind('pengarang', $data['pengarang']);
    $this->db->bind('tahun', $data['tahun']);
    $this->db->bind('kategori_id', $data['kategori_id']);
    $this->db->bind('harga', $data['harga']);
    $this->db->execute();

    return $this->db->rowCount();
}
```

Penjelasan :

- **Harus teliti yaa..**

## 2.7.4 Hapus Buku

**Pastikan** pada index buku tadi sudah kita buat link untuk hapus bukunya

Kalau sudah, ke Buku.php Controller, lalu method hapus :

```
public function hapus($id){
    if( $this->model('BukuModel')->deleteBuku($id) > 0 ) {
        Flasher::setMessage('Berhasil','dihapus','success');
        header('location: '. base_url . '/buku');
        exit;
    }else{
        Flasher::setMessage('Gagal','dihapus','danger');
        header('location: '. base_url . '/buku');
        exit;
    }
}
```

Lalu di method deleteBuku pada **BukuModel.php**

```
public function deleteBuku($id)
{
    $this->db->query('DELETE FROM ' . $this->table . ' WHERE id=:id');
    $this->db->bind('id',$id);
    $this->db->execute();

    return $this->db->rowCount();
}
```

## 2.7.5 Cari Buku

Pastikan pada view index buku sudah terdapat form untuk mencari buku seperti ini :

```
<form action="= base_url(); ?/buku/cari" method="post">
    <div class="row mb-3">
        <div class="col-lg-6">
            <div class="input-group">
                <input type="text" class="form-control" placeholder="" name="key" >
            <div class="input-group-append">
                <button class="btn btn-outline-secondary" type="submit">Cari Data</button>
                <a class="btn btn-outline-danger" href="= base_url(); ?/buku" >Reset</a>
            </div>
        </div>
    </div>
</div>
</form>
```

Kalau ada, kita lanjut ke controller buku.php method **cari** :

```
public function cari()
{
    $data['title'] = 'Data Buku';
    $data['buku'] = $this->model('BukuModel')->cariBuku();
    $data['key'] = $_POST['key'];
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('buku/index', $data);
    $this->view('templates/footer');
}
```

Lalu di method cariBuku pada BukuModel.php

```
public function cariBuku()
{
    $key = $_POST['key'];
    $this->db->query("SELECT * FROM " . $this->table . " WHERE judul LIKE :key");
    $this->db->bind('key', "%$key%");
    return $this->db->resultSet();
}
```

## 2.7.6 Laporan Buku

Pada laporan buku ini saya bagi 2, yaitu PDF dan juga Preview Biasa yang sebenarnya kalian juga bisa export ke PDF melalui Print with Nitro / Print as PDF (bawaan windows 10)

Perbedaan nya :

- Laporan langsung export PDF :
  - o Susah Customnya
- Laporan dengan Preview tanpa export ke PDF langsung :
  - o Mudah Custom tapi harus bisa belajar CSS biar bisa custom sesuai kemauan kamu.

Sebelum membuat laporannya, kita perlu menggunakan beberapa css dan js tambahan.

Ikuti seperti berikut :

1. Folder **fpdf181**, copy dan paste pada folder **app**
2. Pada app/init.php tambahkan coding berikut untuk memanggil folder fpdf

```
<?php

require_once 'core/App.php';
require_once 'core/Controller.php';
require_once 'core/Database.php';
require_once 'core/Flasher.php';
require_once 'config/config.php';
require_once 'fpdf181/fpdf.php';
```

Kalau itu tadi package untuk FPDF atau Laporan yang langsung diexport ke PDF, jadi kalau kalian tidak perlu langsung ke PDF maka tidak perlu dilakukan.

Sekarang CSS tambahan untuk Lihat Laporan :

1. Copy file **normalize.min.css** dan **paper.css** ke dalam folder **public/dist/css**

Silahkan ke view Buku lalu index.php , tambahkan 1 buah link untuk Lihat Laporan Buku pada bagian class card-header :

```
<div class="card-header">
<h3 class="card-title"><?= $data['title'] ?></h3>
<div class="btn-group float-right">
  <a href=<?= base_url; ?>/buku/tambah" class="btn float-right btn-xs btn btn-primary">Tambah Buku</a>
  <a href=<?= base_url; ?>/buku/laporan" class="btn float-right btn-xs btn btn-info">Laporan Buku</a>
  <a href=<?= base_url; ?>/buku/lihatlaporan" class="btn float-right btn-xs btn btn-warning">Lihat
  Laporan Buku</a>
</div>
</div>
```

Pada Buku.php Controller, tambahkan 2 Method yaitu **lihatlaporan** dan **laporan. (coding bisa copy dari folder coding)**

```

public function lihatlaporan()
{
    $data['title'] = 'Data Laporan Buku';
    $data['buku'] = $this->model('BukuModel')->getAllBuku();
    $this->view('buku/lihatlaporan', $data);
}

public function laporan()
{
    $data['buku'] = $this->model('BukuModel')->getAllBuku();

    $pdf = new FPDF('p','mm','A4');
    // membuat halaman baru
    $pdf->AddPage();
    // setting jenis font yang akan digunakan
    $pdf->SetFont('Arial','B',14);
    // mencetak string
    $pdf->Cell(190,7,'LAPORAN BUKU',0,1,'C');

    // Memberikan space kebawah agar tidak terlalu rapat
    $pdf->Cell(10,7,"",0,1);

    $pdf->SetFont('Arial','B',10);
    $pdf->Cell(85,6,'JUDUL',1);
    $pdf->Cell(30,6,'PENERBIT',1);
    $pdf->Cell(30,6,'PENGARANG',1);
    $pdf->Cell(15,6,'TAHUN',1);
    $pdf->Cell(25,6,'KATEGORI',1);
    $pdf->Ln();
    $pdf->SetFont('Arial','',10);

    foreach($data['buku'] as $row) {
        $pdf->Cell(85,6,$row['judul'],1);
        $pdf->Cell(30,6,$row['penerbit'],1);
        $pdf->Cell(30,6,$row['pengarang'],1);
        $pdf->Cell(15,6,$row['tahun'],1);
        $pdf->Cell(25,6,$row['nama_kategori'],1);
        $pdf->Ln();
    }
    $pdf->Output('D', 'Laporan Buku.pdf', true);
}

```

Penjelasan :

- Pada kedua method diatas kita masih menggunakan method getAllBuku(); yang memanggil semua data buku. Kalian bisa mengganti dan membuat method baru misalnya getBukuByTanggal(); nah tinggal diquery deh maunya seperti apa.

Kalau udah controllernya, kita buat view untuk lihatlaporan dulu, simpan di **views/buku** dengan nama **lihatlaporan.php (coding bisa dicopy dari folder coding)**

```

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<title>Doc 1</title>

<!-- Normalize or reset CSS with your favorite library -->
<link rel="stylesheet" href="= base_url ?&gt;/dist/js/normalize.min.css"&gt;

&lt;!-- Load paper.css for happy printing --&gt;
&lt;link rel="stylesheet" href="<?= base_url ?&gt;/dist/css/paper.css"&gt;

&lt;!-- Set page size here: A5, A4 or A3 --&gt;
&lt;!-- Set also "landscape" if you need --&gt;
&lt;style type="text/css" media="print"&gt;
  @page
  {
    size: auto; /* auto is the initial value */
    margin: 0mm; /* this affects the margin in the printer settings */
  }
&lt;/style&gt;
&lt;style&gt;
.center {
  margin-right: auto;
  margin-left: auto;
  text-align: center;
}
h1 {
  font-size: 16px;
}
* {
  font-family: Calibri;
  font-size: 14px;
  -webkit-print-color-adjust:exact;
}
.table {
  border: solid 1px #DDEEEE;
  border-collapse: collapse;
  border-spacing: 0;
  font: normal 13px Arial, sans-serif;
  width: 100%;
}
.table thead th {
  background-color: #DDEFEF;
  border: solid 1px #DDEEEE;
  color: #336B6B;
  padding: 10px;
  text-align: left;
  text-shadow: 1px 1px 1px #fff;
}
.table tbody td {
  border: solid 1px #DDEEEE;
}
&lt;/style&gt;
</pre

```

```

color: #333;
padding: 10px;
text-shadow: 1px 1px 1px #fff;
}
</style>
</head>

<!-- Set "A5", "A4" or "A3" for class name -->
<!-- Set also "landscape" if you need -->
<body class="A4 potrait">

<!-- Each sheet element should have the class "sheet" -->
<!-- "padding-**mm" is optional: you can set 10, 15, 20 or 25 -->
<section class="sheet padding-15mm">
    <h1 class="center" style="text-decoration: underline; margin-top: 20px;"><?= $data['title'] ?></h1>
    <table class="table">
        <thead>
            <tr>
                <th style="width: 10px">#</th>
                <th>Judul</th>
                <th>Penerbit</th>
                <th>Pengarang</th>
                <th>Tahun</th>
                <th>Kategori</th>
                <th>Harga</th>
            </tr>
        </thead>
        <tbody>
            <?php $no=1; ?>
            <?php foreach ($data['buku'] as $row) :?>
            <tr>
                <td><?= $no; ?></td>
                <td><?= $row['judul']; ?></td>
                <td><?= $row['penerbit']; ?></td>
                <td><?= $row['pengarang']; ?></td>
                <td><?= $row['tahun']; ?></td>
                <td><div class="badge badge-warning"><?= $row['nama_kategori']; ?></div></td>
                <td><?= $row['harga']; ?></td>
            </tr>
            <?php $no++; endforeach; ?>
        </tbody>
    </table>
</section>
<script type="text/javascript">
// window.print();
// window.onafterprint = window.close;
</script>
</body>

</html>

```

Silahkan buka buku indexnya dulu dibrowser lalu bisa langsung dicoba deh :

Buku Kita

@herziwp

- Dashboard
- Kategori
- Buku
- About Me

Logout

## Halaman Buku

### Data Buku

#	Judul	Penerbit	Pengarang	Tahun	Kategori	Harga	Action
1	Buku Belajar Memahami Wanita	Gramedia	Herzi	2020	Love Story	99000	<a href="#">Edit</a> <a href="#">Hapus</a>

Footer

## LAPORAN BUKU

JUDUL	PENERBIT	PENGARANG	TAHUN	KATEGORI
Buku Belajar Memahami Wanita	Gramedia	Herzi	2020	Love Story

Doc 1 Doc 1 Downloads

localhost/bukita/public/buku/lihatlaporan

Data Laporan Buku

#	Judul	Penerbit	Pengarang	Tahun	Kategori	Harga
1	Buku Belajar Memahami Wanita	Gramedia	Herzi	2020	Love Story	99000

Bisa langsung print ke printer atau PDF juga lho tinggal **CTRL + P** pada browser

Doc 1 Doc 1 Downloads

localhost/bukita/public/buku/lihatlaporan

Data Laporan Buku

#	Judul	Penerbit	Pengarang	Tahun	Kategori	Harga
1	Buku Belajar Memahami Wanita	Gramedia	Herzi	2020	Love Story	99000

Print 1 page

Destination [Save as PDF](#)

Pages All

Layout Portrait

More settings

Save Cancel

## 2.8 CRUDS User

User ini fungsinya untuk data user yang akan login, jadi harus didaftarkan terlebih dahulu dimenu ini.

Kali ini saya anggap semua sudah bisa melakukan CRUDS maka Coding dari user ini akan saya bagikan langsung pada folder coding, silahkan kalian buat :

- User.php (Controller)
- UserModel.php (Model)
- Create, edit, Index (view user)

Silahkan Buat File, Copy & Paste sesuai foldernya masing – masing, jangan lupa buat sebuah menu baru untuk user pada sidebar.php

Pada user ini passwordnya menggunakan HASH MD5, agar password kita tidak terlihat sebuah string biasa, melainkan berupa angka dan kode yang diacak. Dan dicontroller proses simpan dan update juga berbeda karena melakukan pengecekan jika password sama atau tidak.

#	Nama	Username	Action
1	herr	123	<a href="#">Edit</a> <a href="#">Hapus</a>

### Halaman User

#### Tambah User

##### Nama

masukkan nama...

##### Username

masukkan username...

##### Password

masukkan password...

##### Ulangi Password

masukkan password...

[Submit](#)

## Halaman User

Edit User

Nama

herr

Username

123

Abaikan jika tidak ingin mengganti password.

Password

masukkan password...

Ulangi Password

Submit

## User.php (Controller)

```
<?php

class User extends Controller {
    public function index()
    {
        $data['title'] = 'Data User';
        $data['user'] = $this->model('UserModel')->getAllUser();
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('user/index', $data);
        $this->view('templates/footer');
    }
    public function cari()
    {
        $data['title'] = 'Data User';
        $data['user'] = $this->model('UserModel')->cariUser();
        $data['key'] = $_POST['key'];
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('user/index', $data);
        $this->view('templates/footer');
    }
    public function edit($id){

        $data['title'] = 'Edit User';
        $data['user'] = $this->model('UserModel')->getUserById($id);
        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('user/edit', $data);
        $this->view('templates/footer');
    }
}
```

```

}

public function tambah(){
    $data['title'] = 'Tambah User';
    $this->view('templates/header', $data);
    $this->view('templates/sidebar', $data);
    $this->view('user/create', $data);
    $this->view('templates/footer');
}

public function simpanUser(){
    if($_POST['password'] == $_POST['ulangi_password']) {
        $row = $this->model('UserModel')->cekUsername();
        if($row['username'] == $_POST['username']){
            Flasher::setMessage('Gagal','Username yang anda masukan sudah pernah digunakan!','danger');
            header('location: '. base_url . '/user/tambah');
            exit;
        } else {
            if( $this->model('UserModel')->tambahUser($_POST) > 0 ) {
                Flasher::setMessage('Berhasil','ditambahkan','success');
                header('location: '. base_url . '/user');
                exit;
            } else {
                Flasher::setMessage('Gagal','ditambahkan','danger');
                header('location: '. base_url . '/user');
                exit;
            }
        }
    } else {
        Flasher::setMessage('Gagal','password tidak sama.','danger');
        header('location: '. base_url . '/user/tambah');
        exit;
    }
}

public function updateUser(){
    if(empty($_POST['password'])) {
        if( $this->model('UserModel')->updateDataUser($_POST) > 0 ) {
            Flasher::setMessage('Berhasil','diupdate','success');
            header('location: '. base_url . '/user');
            exit;
        }else{
            Flasher::setMessage('Gagal','diupdate','danger');
            header('location: '. base_url . '/user');
            exit;
        }
    } else {
        if($_POST['password'] == $_POST['ulangi_password']) {
            if( $this->model('UserModel')->updateDataUser($_POST) > 0 ) {
                Flasher::setMessage('Berhasil','diupdate','success');
                header('location: '. base_url . '/user');
            }
        }
    }
}

```

```

        exit;
    }else{
        Flasher::setMessage('Gagal','diupdate','danger');
        header('location: '.base_url . '/user');
        exit;
    }
} else {
    Flasher::setMessage('Gagal','password tidak sama.','danger');
    header('location: '.base_url . '/user/tambah');
    exit;
}
}

public function hapus($id){
    if( $this->model('UserModel')->deleteUser($id) > 0 ) {
        Flasher::setMessage('Berhasil','dihapus','success');
        header('location: '.base_url . '/user');
        exit;
    }else{
        Flasher::setMessage('Gagal','dihapus','danger');
        header('location: '.base_url . '/user');
        exit;
    }
}
}

```

### UserModel.php (Model)

```

<?php

class UserModel {

    private $table = 'user';
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function getAllUser()
    {
        $this->db->query('SELECT * FROM ' . $this->table);
        return $this->db->resultSet();
    }

    public function getUserById($id)
    {
        $this->db->query('SELECT * FROM ' . $this->table . ' WHERE id=:id');
        $this->db->bind('id',$id);
        return $this->db->single();
    }
}

```

```

public function tambahUser($data)
{
    $query      =      "INSERT      INTO      user      (nama,username,password)
VALUES(:nama,:username,:password)";
    $this->db->query($query);
    $this->db->bind('nama',$data['nama']);
    $this->db->bind('username',$data['username']);
    $this->db->bind('password', md5($data['password']));
    $this->db->execute();

    return $this->db->rowCount();
}

public function cekUsername(){
    $username = $_POST['username'];
    $this->db->query("SELECT * FROM user WHERE username = :username");
    $this->db->bind('username', $username);
    return $this->db->single();
}

public function updateDataUser($data)
{
    if(empty($_POST['password'])) {
        $query = "UPDATE user SET nama=:nama WHERE id=:id";
        $this->db->query($query);
        $this->db->bind('id',$data['id']);
        $this->db->bind('nama',$data['nama']);
    } else {
        $query = "UPDATE user SET nama=:nama, password=:password WHERE
id=:id";
        $this->db->query($query);
        $this->db->bind('id',$data['id']);
        $this->db->bind('nama',$data['nama']);
        $this->db->bind('password',md5($data['password']));
    }

    $this->db->execute();

    return $this->db->rowCount();
}

public function deleteUser($id)
{
    $this->db->query('DELETE FROM ' . $this->table . ' WHERE id=:id');
    $this->db->bind('id',$id);
    $this->db->execute();

    return $this->db->rowCount();
}

public function cariUser()
{

```

```

    $key = $_POST['key'];
    $this->db->query("SELECT * FROM " . $this->table . " WHERE nama LIKE :key");
    $this->db->bind('key', "%$key%");
    return $this->db->resultSet();
}
}

```

## Index User (View)

```

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Halaman User</h1>
                </div>
            </div><!-- /.container-fluid -->
        </section>

    <!-- Main content -->
    <section class="content">
        <div class="row">
            <div class="col-sm-12">
                <?php
                    Flasher::Message();
                ?>
            </div>
        </div>
        <!-- Default box -->

        <div class="card">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title'] ?></h3> <a href="<?= base_url; ?>/user/tambah" class="btn float-right btn-xs btn btn-primary">Tambah User</a>
            </div>
            <div class="card-body">

                <form action="<?= base_url; ?>/user/cari" method="post">
                    <div class="row mb-3">
                        <div class="col-lg-6">
                            <div class="input-group">
                                <input type="text" class="form-control" placeholder="" name="key" >
                            <div class="input-group-append">
                                <button class="btn btn-outline-secondary" type="submit">Cari Data</button>
                                <a class="btn btn-outline-danger" href="<?= base_url; ?>/user" >Reset</a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>

```

```

</form>
<table class="table table-bordered">
  <thead>
    <tr>
      <th style="width: 10px">#</th>
      <th>Nama</th>
      <th>Username</th>
      <th style="width: 150px">Action</th>
    </tr>
  </thead>
  <tbody>
    <?php $no=1; ?>
    <?php foreach ($data['user'] as $row) :?>
    <tr>
      <td><?= $no; ?></td>
      <td><?= $row['nama']; ?></td>
      <td><?= $row['username']; ?></td>
      <td>
        <a href=<?= base_url; ?>/user/edit/<?= $row['id'] ?>" class="badge badge-info">Edit</a> <a href=<?= base_url; ?>/user/hapus/<?= $row['id'] ?>" class="badge badge-danger" onclick="return confirm('Hapus data?');">Hapus</a>
      </td>
    </tr>
    <?php $no++; endforeach; ?>
  </tbody>
</table>
</div>
<!-- /.card-body -->
<div class="card-footer">
  Footer
</div>
<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

## Create User (View)

```

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Halaman User</h1>
        </div>
      </div>
    </div><!-- /.container-fluid -->

```

```

    </section>

    <!-- Main content -->
    <section class="content">
        <div class="row">
            <div class="col-sm-12">
                <?php
                    Flasher::Message();
                ?>
            </div>
        </div>
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action=<?= base_url; ?>/user/simpanuser" method="POST"
enctype="multipart/form-data">
                <div class="card-body">
                    <div class="form-group">
                        <label >Nama</label>
                        <input type="text" class="form-control" placeholder="masukkan nama..." name="nama">
                    </div>
                    <div class="form-group">
                        <label >Username</label>
                        <input type="text" class="form-control" placeholder="masukkan username..." name="username">
                    </div>
                    <div class="form-group">
                        <label >Password</label>
                        <input type="password" class="form-control" placeholder="masukkan password..." name="password">
                    </div>
                    <div class="form-group">
                        <label >Ulangi Password</label>
                        <input type="password" class="form-control" name="ulangi_password">
                    </div>
                </div>
                <!-- /.card-body -->

                <div class="card-footer">
                    <button type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>

    </section>
    <!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

## Edit User (Views)

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Halaman User</h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <form role="form" action=<?= base_url; ?>/user/updateUser" method="POST"
enctype="multipart/form-data">

                <input type="hidden" name="id" value=<?= $data['user']['id']; ?>>
                <div class="card-body">
                    <div class="form-group">
                        <label>Nama</label>
                        <input type="text" class="form-control" placeholder="masukkan nama..." name="nama" value=<?= $data['user']['nama']; ?>" autofocus>
                    </div>
                    <div class="form-group">
                        <label>Username</label>
                        <input type="text" class="form-control" placeholder="masukkan username..." name="username" value=<?= $data['user']['username']; ?>" readonly>
                    </div>
                    <blockquote class="quote-warning">Abaikan jika tidak ingin mengganti password.</blockquote>
                    <div class="form-group">
                        <label>Password</label>
                        <input type="password" class="form-control" placeholder="masukkan password..." name="password">
                    </div>
                    <div class="form-group">
                        <label>Ulangi Password</label>
                        <input type="password" class="form-control" name="ulangi_password">
                    </div>
                </div>
            <!-- /.card-body -->
        </div>
    </section>
```

```

<div class="card-footer">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
```

## 2.9 Membuat Halaman About

Halaman about hanya menampilkan data profil singkat tentang anda, jadi tidak ada CRUDS disini. Yang pertama dilakukan yaitu silahkan copy **controller** dari **Home.php**, lalu paste dan ganti nama filenya jadi **About.php**

Jadi pada folder controllers saat ini ada 5 file :

PC > Windows (C:) > xampp > htdocs > bukita > app > controllers				
Name		Date modified	Type	Size
About.php		01/01/2020 17:35	PHP File	1 KB
Buku.php		19/01/2020 14:39	PHP File	4 KB
Home.php		12/01/2020 17:01	PHP File	1 KB
Kategori.php		19/01/2020 13:27	PHP File	3 KB
User.php		19/01/2020 16:02	PHP File	4 KB

Nah silahkan sesuaikan coding pada about.php jadi seperti berikut ini :

```

<?php

class About extends Controller {
    public function index()
    {
        $data['title'] = 'Halaman AboutMe';
        $data['nama'] = 'Herzi';
        $data['alamat'] = 'Alamat Rumah kalian';
        $data['no_hp'] = '082148867999';

        $this->view('templates/header', $data);
        $this->view('templates/sidebar', $data);
        $this->view('about/index', $data);
        $this->view('templates/footer');
    }
}
```

Lalu kita buat tampilan aboutnya, silahkan buat folder baru dengan nama **about** didalam folder **app > views**, nah didalam folder about, bikin file nama nya index.php

Isi dari index.php :

```
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1><?= $data['title']; ?></h1>
                </div>
            </div>
        </div><!-- /.container-fluid -->
    </section>

    <!-- Main content -->
    <section class="content">

        <!-- Default box -->
        <div class="card">
            <div class="card-header">
                <h3 class="card-title"><?= $data['title']; ?></h3>
            </div>
            <div class="card-body">
                <div class="row">
                    <div class="col-md-2 text-right">
                        Nama :
                    </div>
                    <div class="col-md-10 text-left">
                        <?= $data['nama']; ?>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-2 text-right">
                        No. Hp :
                    </div>
                    <div class="col-md-10 text-left">
                        <?= $data['no_hp']; ?>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-2 text-right">
                        Alamat :
                    </div>
                    <div class="col-md-10 text-left">
                        <?= $data['alamat']; ?>
                    </div>
                </div>
            </div>
        <!-- /.card-body -->
```

```
<div class="card-footer">
  Footer
</div>
<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
```

Terakhir silahkan link pada **sidebar.php** tambahkan link yang mengarah pada halaman about.

```
<li class="nav-item">
  <a href="= base_url(); ?/about" class="nav-link">
    <i class="nav-icon fas fa-user"></i>
    <p>
      About Me
    </p>
  </a>
</li>
```

Kalau udah tinggal jalanin deh <http://localhost/bukita/public/about> :

## 2.10 Membuat Login & Logout

Login dan logout ini sangat penting dalam sebuah sistem agar tidak sembarang orang bisa masuk, tentunya nanti bisa dikembangkan untuk pencatatan record siapa yang terakhir update dan siapa yang menambahkan data disuatu table.

### 2.10.1 Login

Buat file baru dengan nama **Login.php** simpan didalam folder **app > controllers**, lalu isi dengan coding berikut :

```
<?php

class Login extends Controller {
    public function index()
    {
        $data['title'] = 'Halaman Login';

        $this->view('login/login', $data);
    }

    public function prosesLogin() {
        if($row = $this->model('LoginModel')->checkLogin($_POST) > 0 ) {
            $_SESSION['username'] = $row['username'];
            $_SESSION['nama'] = $row['nama'];
            $_SESSION['session_login'] = 'sudah_login';

            header('location: '. base_url . '/home');
        } else {
            Flasher::setMessage('Username / Password','salah','danger');
            header('location: '. base_url . '/login');
            exit;
        }
    }
}
```

Penjelasan :

- Terdapat method prosesLogin(), dimana proses pengiriman data username dan password ke LoginModel.php jadi jika data ditemukan maka data username, nama, session\_login akan disimpan pada SESSION untuk dilanjutkan ke halaman home.
- Tapi jika gagal atau data tidak ditemukan, maka akan kembali ke halaman login

Kalau controllersnya sudah, kita lanjut bikin Model yang menghubungkan form login kita ke database pada tabel user, pada folder app > models > tambahkan file baru LoginModel.php

```
<?php

class LoginModel {

    private $table = 'user';
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function checkLogin($data)
    {
        $query = "SELECT * FROM user WHERE username = :username AND password = :password";
        $this->db->query($query);
        $this->db->bind('username', $data['username']);
```

```

        $this->db->bind('password', md5($data['password']));
        $data = $this->db->single();
        return $data;
    }

}

```

Penjelasan :

- Nah disini mengambil data dari prosesLogin tadi, untuk cocokin langsung ke tabel user lalu mengirimkan data kembali ke prosesLogin

Controllers udah, model login udah, berarti tinggal tampilan loginnya dong hehe nah sudah saya siapkan silahkan dicopy dari folder coding, buat folder baru nama login didalam folder views, lalu buat file didalam folder login dengan nama login.php :

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Bukita | Login</title>
<!-- Tell the browser to be responsive to screen width -->
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Font Awesome -->
<link rel="stylesheet" href=<?= base_url; ?>/plugins/fontawesome-free/css/all.min.css">
<!-- Ionicons -->
<link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
<!-- iCheck bootstrap -->
<link rel="stylesheet" href=<?= base_url; ?>/plugins/iCheck-bootstrap/icheck-
bootstrap.min.css">
<!-- Theme style -->
<link rel="stylesheet" href=<?= base_url; ?>/dist/css/adminlte.min.css">
<!-- Google Font: Source Sans Pro -->
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700"
rel="stylesheet">
</head>
<body class="hold-transition login-page">
<div class="login-box">
<div class="login-logo">
<b>Bu</b>Kita
</div>
<!-- /.login-logo -->
<div class="card">
<div class="card-body login-card-body">
<p class="login-box-msg">Silahkan login terlebih dahulu.</p>

<form action=<?= base_url; ?>/login/prosesLogin" method="post">
<div class="input-group mb-3">

```

```

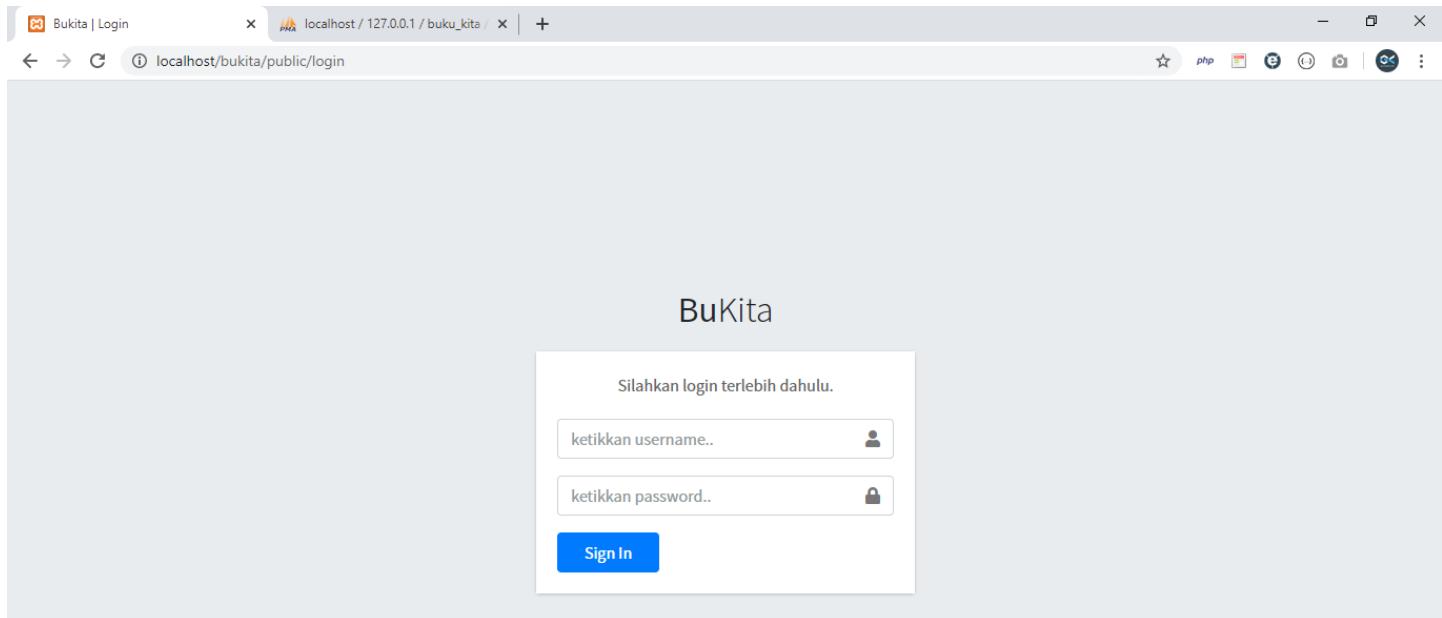
<input type="text" class="form-control" placeholder="ketikkan username.." name="username">
<div class="input-group-append">
  <div class="input-group-text">
    <span class="fas fa-user"></span>
  </div>
</div>
<div class="input-group mb-3">
  <input type="password" class="form-control" placeholder="ketikkan password.." name="password">
<div class="input-group-append">
  <div class="input-group-text">
    <span class="fas fa-lock"></span>
  </div>
</div>
</div>
<div class="row">
  <!-- /.col -->
  <div class="col-4">
    <button type="submit" class="btn btn-primary btn-block">Sign In</button>
  </div>
  <!-- /.col -->
</div>
</form>
</div>
<!-- /.login-card-body -->
</div>
<div class="row">
  <div class="col-sm-12">
    <?php
      Flasher::Message();
    ?>
  </div>
</div>
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="= base_url; ?&gt;/plugins/jquery/jquery.min.js"&gt;&lt;/script&gt;
&lt;!-- Bootstrap 4 --&gt;
&lt;script src="<?= base_url; ?&gt;/plugins/bootstrap/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
&lt;!-- AdminLTE App --&gt;
&lt;script src="<?= base_url; ?&gt;/dist/js/adminlte.min.js"&gt;&lt;/script&gt;

&lt;/body&gt;
&lt;/html&gt;
</pre

```

Coba panggil : <http://localhost/bukita/public/login>



Lakukan login dengan user yang kalian daftarkan tadi, kalau berhasil akan menuju halaman Home, kalau gagal yaa kembali ke halaman login ini.

Sekarang kita atur gimana biar halaman home, buku, kategori, dll agar harus login dulu baru bisa diakses, contohnya kalian sekarang belum login tapi buka halaman buku pasti masih bisa? Nah harusnya ketika membuka halaman buku, harus login dulu baru bisa terbuka. Caranya?

Tambahkan `__construct()` disetiap controller

```
public function __construct()
{
    if($_SESSION['session_login'] != 'sudah_login') {
        Flasher::setMessage('Login','Tidak ditemukan. ','danger');
        header('location: '. base_url . '/login');
        exit;
    }
}
```

Contohnya pada Buku.php :

```
<?php

class Buku extends Controller {
    public function __construct()
    {
        if($_SESSION['session_login'] != 'sudah_login') {
            Flasher::setMessage('Login','Tidak ditemukan. ','danger');
            header('location: '. base_url . '/login');
            exit;
        }
    }
    public function index()
    {
        $data['title'] = 'Data Buku';
    }
}
```

Nah lakukan atau buat coding diatas pada semua controller **kecuali** Login.php

## 2.10.2 Logout

Login dan Logout ini berpasangan, jadi ketika login kita harus membuat logoutnya juga, jadi buat file Logout.php didalam controllers

Codingnya :

```
<?php

class Logout {
    public function Logout(){
        session_start();
        session_destroy();
        header('location: '. base_url . '/login');
    }
}
```

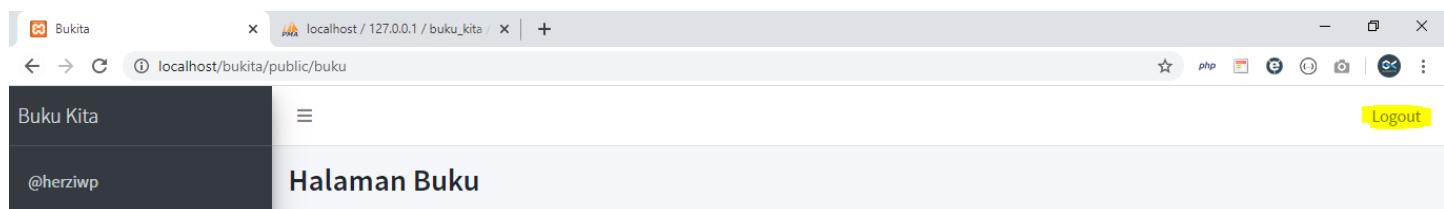
Penjelasan :

- Menghapus SESSION login, dan mengembalikan ke halaman login

Untuk link logoutnya, pada header.php didalam folder views > templates > header.php paling bawah tambahkan link untuk logoutnya seperti berikut :

```
<ul class="navbar-nav ml-auto">
    <li class="nav-item d-none d-sm-inline-block">
        <a href="= base_url(); ?/logout" class="nav-link">Logout</a>
    </li>
</ul>
</nav>
<!-- /.navbar -->
```

Coba refresh lalu klik logout pada header :



Kalau saya sih udah berhasil nih hehe Tambahan , silahkan buat file .htaccess simpan pada folder bukita

```
<IfModule mod_rewrite.c>
    Options -Multiviews
    RewriteEngine On
    RewriteRule ^$ public/ [L]
    RewriteRule (.*)$ public/$1 [L]
</IfModule>
```

 app	01/01/2020 20:07	File folder
 public	01/01/2020 18:11	File folder
 .htaccess	25/12/2019 17:41	HTACCESS File 1 KB

Penjelasan :

- Fungsinya agar kita tidak perlu lagi mengakses ke public untuk membuka program kita, jadi kita hanya ketik localhost/bukita saja dia otomatis masuk ke dalam folder public.

Selesai deh.

*Semoga bermanfaat untuk kita semua ya! Saya @herziwp mohon maaf jika terdapat kesalahan kata maupun penjelasan*

*Terimakasih.*

Pesan :

- Silahkan kreasikan program ini agar logika pemrograman mu semakin terasah
- Usahalah untuk memperbaiki dan menyelesaikan masalah program mu sendiri dengan google kalau sudah tidak bisa, baru bertanya dengan teman / orang yang bisa
- Cari keyword dengan singat, jelas dan padat, contoh "cara mencari data dengan tanggal php"
- Selalu tambahkan bahasa pemrograman yang kamu gunakan pada keyword pencarian seperti dipoint atas
- Belajar untuk memanajemen waktu agar kerja dan kuliahmu seimbang sehingga tidak tergesa - gesa untuk mengerjakan program.