

# **Pemrograman Mobile Dengan WebView**

**Arik Sofan Tohir, M.Kom**

-  
-  
-

# **Pemrograman Mobile Dengan Web View**

Penulis: Arik Sofan Tohir, M.Kom

Editor: Arik Sofan Tohir, M.Kom

Tata Bahasa: Arik Sofan Tohir, M.Kom an

Tata Letak: Arik Sofan Tohir, M.Kom

Sampul: Arik Sofan Tohir, M.Kom

Diterbitkan melalui:

-

-

Telp: -

email: -

-

Versi 1, April 2022

Kediri, 2022

-

ISBN: -

Hak Cipta dilindungi Undang-undang

*All right reserved*

-

# Kata Pengantar

Dear, para mahasiswa

Mengawali kata pengantar ini penulis mengucapkan syukur alhamdulillah karena dengan izin Allah yang Maha pengasih lagi penyayang, penulis bisa memberikan yang selalu memberikan limpahan rahmat, taufiq, hidayah serta inayahnya kepada seluruh umat manusia pada umumnya dan kepada penulis khususnya.

## Daftar Isi

Bab I Pendahuluan .....	5
CodeIgniter.....	5
Konsep MVC .....	5
<b>Model</b> .....	5
<b>View</b> .....	5
<b>Controller</b> .....	5
Webview .....	5
RDBMS.....	6
Bab II Desain Sistem.....	6
Desain Database .....	6
<b>tb_siswa</b> .....	6
<b>tb_jenis_bayar</b> .....	6
<b>tb_bayar</b> .....	7
Integrasi Template.....	7
Bab III Setting Aplikasi .....	7
Implementasi Pada CodeIgniter .....	7
<b>Import Data</b> .....	7
<b>Setting Koneksi</b> .....	10
<b>Jenis Pembayaran</b> .....	11

## **Bab I Pendahuluan**

### **CodeIgniter**

Codeigniter merupakan salah satu framework atau kerangka kerja yang populer dikalangan para developer khususnya developer PHP. Didalam framework (kerangka kerja) ini berisi kumpulan function, procedure, class dan library untuk mempermudah para developer dalam membuat aplikasi. Dengan adanya framework (kerangka kerja) secara otomatis penulisan code program akan lebih rapi yang pada akhirnya akan mempermudah ketika proses debugging.

### **Konsep MVC**

Konsep MVC (Model, View, Controller) merupakan salah satu teknik untuk memecahkan atau membagi kode program kedalam beberapa modul terpisah, sehingga akan lebih mempermudah untuk melakukan manajemen aplikasi.

#### **Model**

Class model merupakan komponen yang berhubungan langsung dengan database, umumnya digunakan untuk melakukan proses insert, update, delete dan select. Model ini tidak berhubungan langsung dengan view akan tetapi akan berhubungan dengan controller.

#### **View**

Komponen view merupakan bagian yang berhubungan dengan user interface, komponen ini dengan file-file html,css, JS dan file-file yang berkaitan dengan tampilan. Komponen ini berhubungan dengan controller dimana bagian controller yang akan mengatur kapan sebuah view akan dipanggil.

#### **Controller**

Class controller merupakan komponen yang berhubungan langsung dengan view dan model. Class inilah yang bertugas menerima request user dan memberikan hasil dari hasil request.

#### **Webview**

Webview merupakan salah satu fasilitas yang bisa kita gunakan untuk membangun sebuah aplikasi berbasis android dengan memanfaatkan script HTML,CSS dan Javascript.

## RDBMS

Relational Database Management System (RDBMS) adalah merupakan suatu engine yang digunakan untuk menyimpan dan mengolah data. Salah satu contoh RDBMS adalah MySQL, dalam pembahasan kali ini penulis menggunakan Mysql Server Community Edition Versi 5.7.

## Bab II Desain Sistem

### Desain Database

Pada pembahasan ini penulis akan mencoba menyajikan implementasi pemrograman mobile dengan webview dengan studi kasus sistem informasi pembayaran sekolah, pada sistem ini akan dibuat dua jenis sistem pembayaran sekolah sederhana dan yang bersifat kompleks. Untuk desain database seperti terlihat dibawah ini.

#### tb\_siswa

Adalah tabel yang digunakan untuk menampung data siswa, berikut desain dari tabel jenis barang.

Field	Type	Length	Key	Keterangan
Kode	int	11	*	Auto Increment
Nama	Varchar	255		
Alamat	Varchar	255		

*Tabel 1 tb\_siswa*

#### tb\_jenis\_bayar

Adalah tabel yang digunakan untuk menampung atau mencatat data jenis pembayaran dengan desain tabel sebagai berikut.

Field	Type	Length	Key	Keterangan
Kode	Int	11	*	Auto increment
Jenis_bayar	Varchar	255		
Nominal	Int	11		

*Tabel 2 tb\_jenis\_bayar*

### tb\_bayar

Adalah tabel yang digunakan untuk mencatat data pembayaran siswa dimana tabel ini merupakan tabel yang berelasi dengan beberapa tabel, berikut desain dari tabel tb\_bayar .

Field	Type	Length	Key	Keterangan
Kode	Int	11	*	Auto increment
Kode_siswa	Int	11		Relasi tabel tb_siswa
Kode_jenis	Int	11		Relasi tabel tb_jenis_bayar
Qty	Int	11		
Nominal	Int	11		
Tangga	Date	-		

*Tabel 3 tb\_bayar*

### Integrasi Template

Agar hasil tampilan lebih maksimal dan user friendly maka akan digunakan template yang memanfaatkan bootstrap. Untuk mendownload template yang digunakan dalam pembahasan ini silahkan kunjungi github penulis di alamat berikut <https://github.com/akara19/Pemrograman-mobile/tree/main/Bahan>, silahkan download file bernama **ci\_mobile.zip**.

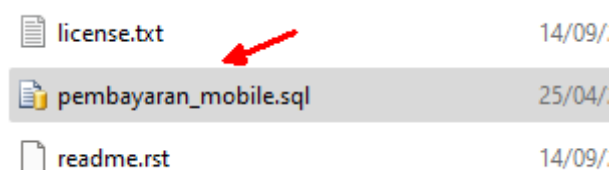
## Bab III Setting Aplikasi

### Implementasi Pada CodeIgniter

Pada sub bab ini akan dibagi menjadi beberapa sub bahasan lagi untuk lebih mempermudah dalam melakukan proses debugging.

### Import Data

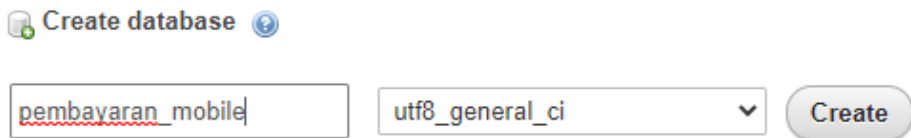
Untuk lebih mempermudah maka dalam bahasan ini penulis menyediakan bahan data yang digunakan, silahkan extract file yang sudah didownload sebelumnya (sub bab **Integrasi Template**), setelah file diextract silahkan cari file bernama pembayaran\_mobile.sql



*Gambar 1 File pembayaran\_mobile.sql*

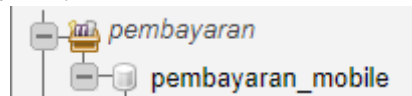
File diatas berisi desain sistem pembayaran sekolah yang akan digunakan untuk pembahasan berikutnya. Untuk mengimport data silahkan buat database pada mysql atau mariaDb terlebih dahulu, untuk editornya silahkan menggunakan editor yang familier, disini penulis menggunakan editor phpMyAdmin.

1. Buka phpMyAdmin
2. Pilih Tab Databases
3. Buat database dengan nama pembayaran\_mobile
4. Lalu klik tombol create



*Gambar 2 Membuat Database*

Jika langkah ini berhasil maka pada menu sebelah kanan akan terlihat sebuah database baru dengan nama pembayaran\_mobile, seperti terlihat pada gambar dibawah ini.

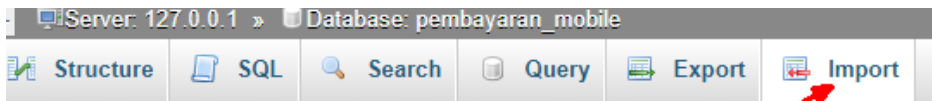


*Gambar 3 Database*

Setelah database berhasil dibuat selanjutnya adalah proses untuk mengimpor data kedalam database yan baru dibuat.

1. Pada menu sebelah kiri klik pembayaran\_mobile
2. Selanjutnya pilih tab import
3. Klik tombol Choose File





tab Import

## Importing into the database "pembayaran\_m

### File to import:

Tombol Choose File

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer:

Choose File

No file chosen

(Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file:

utf-8

Gambar 4 Import data dummy

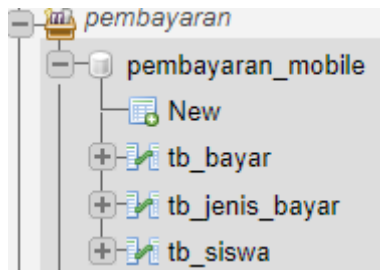
**Catatan :** Sebelum mengimport data silahkan lihat terlebih dahulu batasan maksimal data yang bisa diimport, pada phpMyAdmin yang penulis gunakan batasan maximal import datanya adalah 40MB, jika batasan maximal phpMyAdmin yang anda gunakan lebih kecil dari file yang akan diimport disarankan untuk melakukan compress data terlebih dahulu kedalam format .zip

4. Jika gambar sudah terpilih silahkan klik tombol Go, jika tidak terlihat silahkan scrool halaman kebawah



Gambar 5 Tombol Go

Jika import data berhasil maka akan terdapat daftar tabel pada database pembayaran\_mobile, seperti terlihat pada gambar dibawah ini.

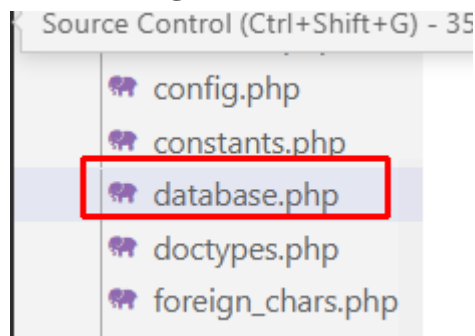


Gambar 6 Daftar Tabel Hasil Import

## Setting Koneksi

Setting koneksi merupakan langkah untuk menghubungkan CodeIgniter dengan database yang sudah kita buat, untuk setting koneksi database pada Codeignter silahkan ikuti langkah-langkah berikut, sebagai catatan dalam penulisan ini penulis menggunakan editro visual studio code untuk mebuat aplikasi dengan framwork CodeIgniter.

1. Extarct File yang sudah didownload dari github
2. Copy file ke **C:\xampp\htdocs** (untuk pengguna windows), silahkan sesuaikan dengan lokasi anda masing-masing
3. Jika sudah dicopy silahkan buka file bernama database yang terletak pada folder **application/config**



4. Silahkan rubah setting koneksi database sesua dengan setting database anda

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => '127.0.0.1',
    'username' => 'root',
    'password' => '1986',
    'database' => 'pembayaran_mobile',
    'dbdriver' => 'mysqli',
    'dsnprefix' => ''
);
```

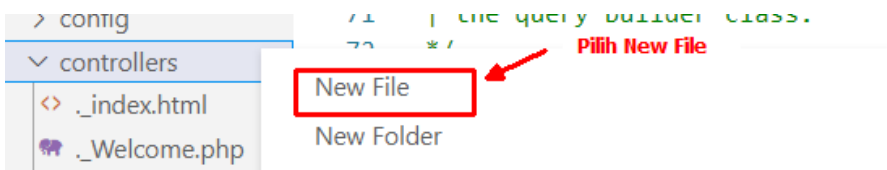
- hostname bisa diisi **localhost** atau **127.0.0.1**
- username : **root** (default) atau sesuaikan dengan pengaturan database anda
- password : sesuaikan dengan password database anda, jika database anda tidak ada passwordnya silahkan kosongi
- database

## Jenis Pembayaran

Pada sub bab ini akan dibahas cara pembuatan class-class yang diperlukan untuk untuk proses penyimpanan data jenis pembayaran, langkah-langkahnya sebagai berikut

### Controller

Pada folder controller klik kanan lalu pilih **New File** lalu beri nama file **JenisBayar**



Ketikkan script seperti dibawah ini

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

class JenisBayar extends CI_Controller {
    public function index()
    {
```

```

        $this->template-
>load('Dashboard', 'JenisBayar');
    }

    public function Simpan() {
        $Jenis = $this->input->post('Jenis');
        $Nominal = $this->input->post('Nominal');
        $data = array('Jenis_bayar'=>$Jenis,
                      'Nominal'=>$Nominal);
        $this->db->insert('tb_jenis_bayar', $data);

        echo "<meta http-equiv='refresh' content='0;
url=".base_url().".index.php/JenisBayar'>";
    }

    public function Lihat() {
        $data['jenis'] = $this->db-
>get('tb_jenis_bayar')->result();
        $this->template-
>load('Dashboard', 'LihatJenisBayar', $data);
    }

    public function Edit() {
        $Kode=$this->uri->segment(3);
        $data['jenis'] = $this->db-
>get_where('tb_jenis_bayar',
        array('Kode'=>$Kode), 0, 0)->result();
        $this->template-
>load('Dashboard', 'JenisBayarEdit', $data);
    }

    public function Update() {
        $Kode = $this->input->post('Kode');
        $Jenis = $this->input->post('Jenis');
        $Nominal = $this->input->post('Nominal');

        $data = array('Jenis_bayar'=>$Jenis,
                      'Nominal'=>$Nominal);
        $this->db->where('Kode', $Kode);
        $this->db->update('tb_jenis_bayar', $data);
    }

```

```

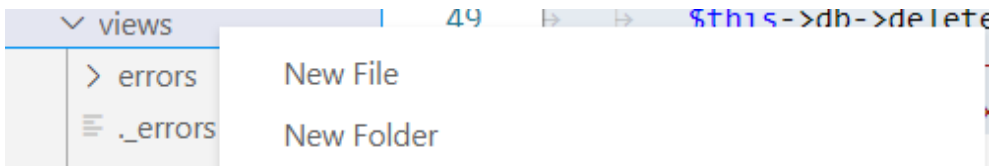
        echo "<meta http-equiv='refresh' content='0;
url=".
        base_url()."index.php/JenisBayar/Lihat'>";
    }

    public function Delete(){
        $Kode = $this->uri->segment(3);
        $this->db->where('Kode', $Kode);
        $this->db->delete('tb_jenis_bayar');
        echo "<meta http-equiv='refresh' content='0;
url=".
        base_url()."index.php/JenisBayar/Lihat'>";
    }
}

```

## View

Setelah controller untuk class jenis bayar selesai, selanjutnya adalah pembuatan view, dimana view ini akan digunakan sebagai file yang akan menampilkan data jenis bayar.



Untuk pembuatan view silahkan klik kanan pada folder View lalu beri nama file **JenisBayar**, selanjutnya ketik script seperti dibawah ini.

```

<div class="content">
    <div class="container-fluid">
        <div class="row">
            <?php echo
form_open_multipart('JenisBayar/Simpan');?>
                <div class="form-group">
                    <label>Jenis Bayar</label>
                    <input type="text" class="form-control"
name="Jenis">
                </div>

```

```

        <div class="form-group">
            <label>Nominal</label>
            <input type="text" class="form-control"
name="Nominal">
        </div>

        <div class="form-group">
            <input type="submit" value="Simpan"
class="btn btn-success">
            <a href="<?php echo
base_url('index.php/JenisBayar/Lihat'); ?>">
                <input type="button" value="Lihat
Data"

                class="btn btn-danger">
            </a>
        </div>
    </form>

</div>
</div>
</div>

```

Script diatas digunakan untuk menampilkan form input data jenis pembayaran, berikut tampilannya.

Jenis Bayar

Nominal

Simpan

Lihat Data

Selanjutnya adalah pembuatan view untuk menampilkan data, untuk membuatnya silahkan klik kanan lagi pada folder view dan beri nama lihat, lalu beri nama **LihatJenisBayar** lalu ketikkan script seperti dibawah ini.

```

<div class="content">
    <div class="container">
        <div class="row">
            <table class="table">
                <th>Jenis Bayar</th>
                <th>Nominal</th>
                <th>Aksi</th>
                <?php
                    foreach($jenis as $_jenis){
                        echo "<tr>".
                            "<td>".$_jenis->Jenis_bayar."</td>".
                            "<td>".$_jenis->Nominal."</td>".
                            "<td>".
                                anchor('JenisBayar/Edit/'.$_jenis->Kode,'Edit',
                                    array('Title'=>'Edit Data','class'=>'btn btn-
info'))<td>".
                                anchor('JenisBayar/Delete/'.$_jenis->Kode,'Delete',
                                    array('Title'=>'Delete Data','class'=>'btn btn-danger'))<td>".
                                "</td>".
                            "</tr>";
                    };
                ?>
            </table>
        </div>
    </div>
</div>

```

Berikut tampilan dari script diatas yang akan menampilkan halaman data yang sudah dimasukkan.

Jenis Bayar	Nominal	Aksi
SPP	10000	<div>EditDelete</div>

Selanjutnya adalah pembuatan view untuk proses form update data, dimana form ini akan digunakan sebagai halaman untuk merubah data, untuk membuat view form ubah data langkah ada dengan membuat file pada folder view dengan cara klik kanan pada folder view lalu pilih New File dan beri nama file dengan **JenisBayarEdit** dan masukkan script seperti dibawah ini.

```

<div class="content">
    <div class="container-fluid">
        <div class="row">

```

```

        <?php
            echo
            form_open_multipart('JenisBayar/Update');
                foreach($jenis as $_jenis);
            ?>
                <div class="form-group">
                    <label>Jenis Bayar</label>
                    <input type="hidden" name="Kode"
                        value="<?php echo $_jenis->Kode; ?>">

                    <input type="text" name="Jenis"
class="form-control"
                        value="<?php echo $_jenis->Jenis_bayar;
?>">

                </div>
                <div class="form-group">
                    <label>Nominal</label>
                    <input type="text" name="Nominal"
class="form-control"
                        value="<?php echo $_jenis->Nominal;
?>">

                </div>

                <div class="form-group">
                    <input type="submit" value="Update"
class="btn btn-success">
                    <a href="<?php echo
base_url('index.php/JenisBayar/Lihat'); ?>">
                        <input type="button" value="Lihat
Data" class="btn btn-danger">
                    </a>
                </div>
            </form>

        </div>
    </div>
</div>

```

Script diatas akan menghasilkan sebuah tampilan atau halaman sebagai berikut



Jenis Bayar

SPP

Nominal

10000

Update

Lihat Data

### Link

Setelah semua view selesai langkah selanjutnya adalah menghubungkan dengan halaman utama atau halaman dashboard, untuk menghubungkan dengan halaman dashboard silahkan buka file **Home.php** yang berada pada folder view, silahkan ubah pada baris ke-7 dan 12 seperti terlihat pada gambar dibawah ini

```
6 <div class="box">
7   <a href="{?php echo base_url('index.php/JenisBayar'); ?}">
8     
10  </div>
11 <div class="row">
12   <p id="title1">Jenis Bayar</p>
13 ..
```

Diagram illustrating the code changes in **Home.php** (lines 7 and 12) for linking to the main page or dashboard:

- Line 7: The `href` attribute is updated to `{?php echo base_url('index.php/JenisBayar'); ?}`. This is labeled as **Nama Menu** (Menu Name).
- Line 12: The `id` attribute is updated to `title1`. This is labeled as **Nama Controller** (Controller Name).

Jika sudah silahkan buka halaman utama atau halaman dashboard dan klik tombol menu **Jenis Bayar**



SISTEM INFORMASI  
PEMBAYARAN SEKOLAH



**Tombol Jenis Bayar**



Ikuti Kami Di Media Sosial



