

Dokumentasi dalam API(Application Program Interface) dalam menampilkan semua, detail, menambah, mengubah, dan menghapus data

Pratikto Arya Wicaksana

6 Juli 2020

## **Bab I**

### **Pendahuluan**

Dokumen ini merupakan dokumen teknis perangkat lunak yang berisi penjelasan mengenai web-jamaah.

#### **1.1 Tujuan**

Tujuan dari pembuatan dokumen teknis perangkat lunak ini diantaranya sebagai berikut :

Dokumen ini digunakan untuk menjelaskan fitur API(*Application Program Interface*) web-jamaah serta fungsi-fungsi yang terdapat didalamnya.

Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

#### **1.2 Batasan Masalah**

Web-Jamaah adalah perangkat lunak berbasis web yang dirancang dengan Framework Laravel dan menggunakan sistem basis data Mysql. Fitur yang tersedia pada Web-Jamaah adalah memasukan data, mengubah data, dan menampilkan data identitas jamaah. Web-jamaah dijalankan menggunakan peramban Mozilla Firefox dan Mysql pada aplikasi XAMPP.

#### **1.3 Deskripsi Umum Dokumen**

Dokumen teknis ini dibagi menjadi tiga bagian utama. Bagian utama berisi penjelasan tentang dokumen teknis yang mencakup tujuan pembuatan dokumen ini, lingkup masalah diselesaikan oleh Web-Jamaah yang dikembangkan, definisi, referensi dan deskripsi umum.

Bagian kedua berisi penjelasan secara umum mengenai Web-Jamaah yang dikembangkan meliputi struktur basis data, . API untuk menampilkan semua data, API untuk menampilkan detail data, API untuk menambah data, API untuk mengubah data, dan API untuk menghapus data

Bagian ketiga berisi uraian secara lebih rinci cara menggunakan Web-Jamaah.

## Bab II

### 2.1 Struktur basis data

Dalam pembuatan web-jamaah spesifikasi tipe data pada basis data yang akan dipakai sebagai berikut:

Nama Field	Tipe data	Primary Key	Null
idjamaah	(VARCHAR 15)	Ya	Tidak
namajamaah	(VARCHAR 50)	Tidak	Tidak
gender	(ENUM L/P)	Tidak	Tidak
telpjamaah	(VARCHAR 15)	Tidak	Tidak

### 2.2 API(Application Program Interface) untuk menampilkan semua data

Tahap pertama yaitu membuat model data yang basis datanya akan dijadikan model dan menentukan primary key untuk pemanggilan record di basis data seperti kode dibawah ini:

```
class Modeldatas extends Model
{
    protected $table = 'data_jamaah';
    protected $primaryKey = 'idjamaah';
    public $timestamps = false;
}
```

Setelah itu membuat controller untuk menampilkan semua data menggunakan perintah all() dari model Modeldatas yang dideklarasikan pada file Modeldatas:

```
public function index () {
    $data = array(
        'data_jamaah' => Modeldatas::all()
    );
    return view ('home', $data);
}
```

Untuk Routenya ditambah dan Agar index menjadi halaman utama maka string address diketik hanya garis miring.maka kode ditulis seperti dibawah:

```
Route::get('/p', function () { return view('welcome'); });

Route::get('/', 'HomeController@index');
```

Untuk menampilkan datanya menggunakan form maka formnya dibuat dalam [home.blade.php](#) pada file view yang isinya daftar dalam bentuk table dan memiliki tombol aksi pada setiap record untuk melakukan operasi data seperti detail, ubah dan hapus. Kode program ditulis dibawah ini ini :

```
<head>
  <h1 style="text-align:center;">Kumpulan Data Member Jamaah</h1>
  <link rel="stylesheet" type="text/css" href="/css/bootstrap.css">
</head>
<body>
  <br>
  <h3><a href="{{ url('tambah')}}">+ Tambah Data</a></h3>
  <br>
  <table class="table table-bordered table-striped">
    <tr>
      <th>nama jamaah</th>
      <th>gender</th>
      <th>no. telepon</th>
      <th>Aksi</th>
    </tr>
    @foreach($data_jamaah as $row)
      <tr>
        <td width="30%">{{ $row->namajamaah }}</td>
        <td width="10%">{{ $row->gender }}</td>
        <td width="20%">{{ $row->telpjamaah }}</td>
        <td width="40%">
          <form action="{{ url( 'detail/' . $row->idjamaah) }}">
            <button type="submit">Detail</button>
          </form>
          <form action="{{ url( 'ubah/' . $row->idjamaah) }}">
            <button type="submit">Ubah</button>
          </form>
          <form action="{{ url( 'hapus/' . $row->idjamaah) }}">
            <button type="submit">Hapus</button>
          </form>
        </td>
      </tr>
    @endforeach
  </table>
</body>
```

## 2.3 API(Application Program Interface) untuk menampilkan detail data

Untuk menampilkan detail data dengan fungsi `find()` mencari data berdasarkan idjamaah. pada HomeController.php ditambah dengan kode ini:

```

public function detail ($idjamaah) {
    $data = array(
        'data_jamaah' => Modeldatas::find($idjamaah)
    );
    return view ('detail', $data);
}

```

Kemudian agar bisa view detail.blade.php yaitu menggunakan tambahan route seperti dibawah:

```

Route::get('/detail/{idjamaah}', 'HomeController@detail');

```

Untuk menampilkan view di detail.blade.php dengan kode dibawah.

```

<link rel="stylesheet" type="text/css" href="/css/bootstrap.css">

<h1 style="text-align:center;">Data Jamaah</h1>
<br>
<table style="position:relative;" class="table table-bordered table-striped">
    <tr>
        <td style="width:150px">Id Jamaah </td>
        <td>{{ $data_jamaah->idjamaah }}</td>
    </tr>
    <tr>
        <td>Nama </td>
        <td>{{ $data_jamaah->namajamaah }}</td>
    </tr>
    <tr>
        <td>Gender </td>
        <td>{{ $data_jamaah->gender }}</td>
    </tr>
    <tr>
        <td>no. telepon </td>
        <td>{{ $data_jamaah->telpjamaah }}</td>
    </tr>
</table>

<br>
<form action="{{ url('/') }}">
    <button type="submit">Kembali</button>
</form>
<br>

```

## 2.4 API(Application Program Interface) untuk tambah data

Untuk menampilkan detail data pada HomeController.php ditambah dengan kode ini: Pada awal terdapat \$message berisi peringatan Ketika form ada yang kosong. Pada selanjut terdapat fungsi validasi agar form yang diisi sesuai Dengan kriteria kebutuhan pada Batasan tipe data yang di deklarasikan. Kemudian input atribut pada objek data yang di post dari view

dari parameter request.setelah melalui itu semua, untuk memberi peringatan bahwa data berhasil di kirim maka diberi peringatan dari percabangan if-else dan di variable tambah memasukan data ke basis data melalui fungsi insert.

```
public function tambah (Request $request) {

    $messages = [
        'required' => ':attribute wajib diisi !',
        'min' => ':attribute harus diisi minimal :min karakter',
        'max' => ':attribute harus diisi maksimal :max karakter',
        'unique' => ':attribute pada form ini telah dimiliki'
    ];

    if($request->isMethod('post')) {

        $request->validate([
            'idjamaah' => 'required|unique:data_jamaah|max:15',
            'namajamaah' => 'required|max:50',
            'gender' => 'required',
            'telpjamaah' => 'required|max:15'
        ], $messages);

        $data = array(
            'idjamaah' => $request->input('idjamaah'),
            'namajamaah' => $request->input('namajamaah'),
            'gender' => $request->input('gender'),
            'telpjamaah' => $request->input('telpjamaah')
        );

        $tambah = Modeldatas::insert($data);
        if($tambah) {
            echo 'tambah data berhasil';
        } else {
            echo 'tambah data gagal';
        }
    }
    return view('form');
}
```

Kemudian agar bisa membukanya yaitu menggunakan tambahan route seperti dibawah:

```
Route::match(['get', 'post'], '/tambah', 'HomeController@tambah');
```

Untuk menampilkan view di form.blade.php.pada form tersebut terdapat perintah if untuk mendeteksi form ini digunakan menambah atau mengubah data, maka form disini mendeteksi ada atau tidaknya data jamaah yang terkirim dari controller. Setiap form sudah dipasang \$error untuk menunjukan value yang diisi oleh pengguna tersebut salah atau kosong. Kode terteras seperti dibawah dibawah.

```

<link rel="stylesheet" type="text/css" href="/css/bootstrap.css">

@if($data_jamaah ?? "")
    <h1 style="text-align:center;">Ubah data jamaah Dengan ID Jamaah = {{ $data_jamaah-
>idjamaah }}</h1>
@else
    <h1 style="text-align:center;">Tambah Entry Data Jamaah</h1>
@endif

<form method="post" style="font-size: 20px;position: relative; left:30px;">
    {{ csrf_field() }}

    @if($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <div>
        <label for="idjamaah">Id Jamaah :</label><br>
        <input type="text" id="idjamaah" @if($data_jamaah ?? "") value="{{ $data_jamaah-
>idjamaah }}" @endif name="idjamaah">
    </div>

    <br>

    <div>
        <label for="namajamaah">Nama Jamaah :</label><br>
        <input type="text" id="namajamaah" @if($data_jamaah ?? "") value="{{ {
$data_jamaah->namajamaah }}" @endif name="namajamaah">
    </div>

```

```

<br>
<div>
    <label for="gender">Gender :</label><br>
    <input type="radio" id="gender" name="gender" value="L">
    <label for="male">Pria</label>
    <input type="radio" id="gender" name="gender" value="P">
    <label for="female">Wanita</label>
</div>
<br>
<div>
    <label for="telpjamaah">Telepon Jamaah :</label><br>
    <input type="text" id="telpjamaah" @if($data_jamaah ?? ") value="{{ $data_jamaah-
>telpjamaah }}" @endif name="telpjamaah">
</div>
<br>
<button type="submit">Submit</button>
<button type="reset">Reset</button>
</form>

<a style="font-size: 20px;" href="{{ url('/') }}"> Kembali </a>

```

## 2.4 API(Application Program Interface) untuk ubah data

Untuk mengubah data dengan request id pada basis data. Struktur hampir sama dengan fungsi tambah bedanya pada deklarasi parameter terdapat \$idjamaah untuk menentukan record yang akan di ubah dan Ketika di simpan menggunakan fungsi save() maka HomeController.php ditambah dengan kode ini:

```

public function ubah ($idjamaah, Request $request) {
    $data_jamaah = Modeldatas::find($idjamaah);

    $messages = [
        'required' => 'attribute wajib diisi !',
        'min' => 'attribute harus diisi minimal :min karakter',
        'max' => 'attribute harus diisi maksimal :max karakter',
    ];
}

```



```

        'unique' => ':attribute pada form ini telah dimiliki'
    ];

    if($request->isMethod('post')) {

        $request->validate([
            'idjamaah'    => 'required|max:15',
            'namajamaah'  => 'required|max:50',
            'gender'       => 'required',
            'telpjamaah'  => 'required|max:15'
        ], $messages);

        $data_jamaah->idjamaah    = $request->input('idjamaah');
        $data_jamaah->namajamaah  = $request->input('namajamaah');
        $data_jamaah->gender       = $request->input('gender');
        $data_jamaah->telpjamaah  = $request->input('telpjamaah');
        $ubah = $data_jamaah->save();
        if($ubah) {
            echo 'Ubah Artikel Berhasil';
        }else {
            echo 'Ubah Artikel Gagal';
        }
    }

    return view('form', array('data_jamaah' => $data_jamaah));
}

```

Kemudian agar bisa membukanya yaitu menggunakan tambahan route dengan method match untuk menampung isian idjamaah ke view dan sebaliknya ke controller. seperti dibawah:

```
Route::match(['get', 'post'], '/ubah/{idjamaah}', 'HomeController@ubah');
```

Untuk View ubah data sudah sama seperti pada 2.4 API(Application Program Interface) untuk tambah data. karena form yang digunakan mirip dengan tambah data.

## 2.5 API(Application Program Interface) untuk hapus data

Hapus data hanya perlu ditulis di controller dan menampilkan sebuah alert. Fungsi hapus ini sama seperti object pemanggilan find pada ubah dan detail tetapi data yang dipilih dihapus dengan fungsi delete.

```
public function hapus ($idjamaah) {  
    $data = Modeldatas::find($idjamaah);  
    $hapus = $data->delete();  
    if($hapus) {  
        echo "<script>alert('Hapus Data Berhasil !')</script>";  
        echo '<meta http-equiv="refresh" content="0;url='. url('/') ."'>';  
    }  
}
```

Dibawah merupakan route penghubung url dari view ke controller untuk menghapus data yang dipilih dari primary key idjamaah.

```
Route::get('/hapus/{idjamaah}', 'HomeController@hapus');
```

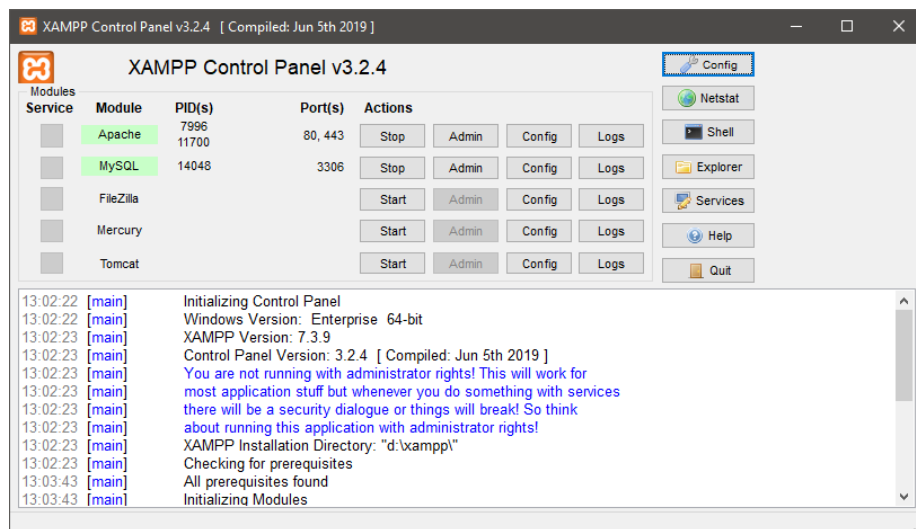
## BAB III

### Cara menggunakan Aplikasi web-jamaah

Dalam menggunakan aplikasi web-jamaah ada tahap-tahap untuk mengeksekusi webnya.

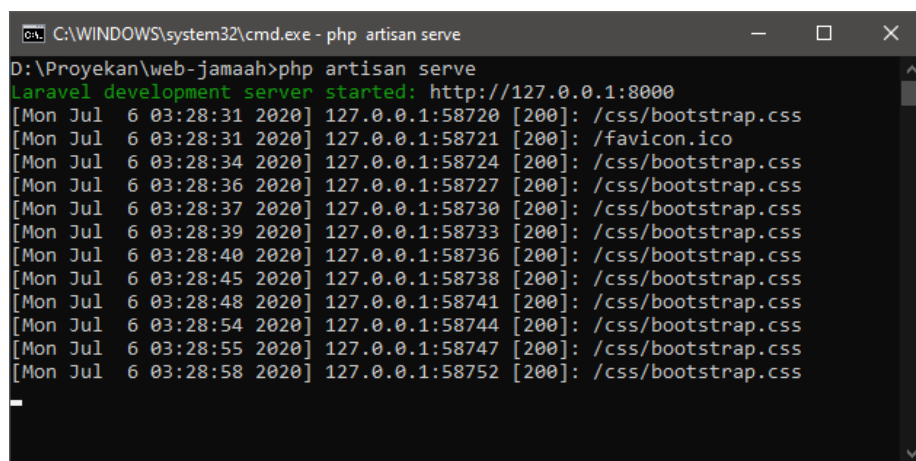
#### 3.1 Mengaktifkan XAMPP

Cara mengaktifkan Mysql menggunakan paket aplikasi XAMPP dengan menekan button action kemudian start module apache dan Mysql.



#### 3.2 Mengaktifkan PHP Artisan Serve

Cara mengaktifkan web-jamaah menggunakan perintah php artisan serve pada direktori web yang tersimpan di drive D:\Proyekan\web-jamaah



#### 3.3 Menggunakan fitur pada Web-Jamaah

Tampilan awal pada web-jamaah



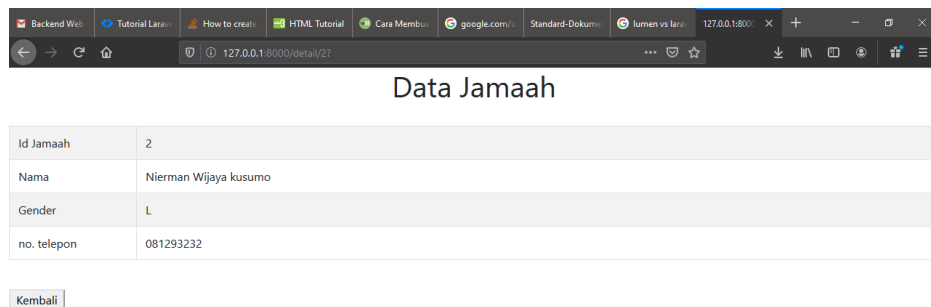
Kumpulan Data Member Jamaah

[+ Tambah Data](#)

nama jamaah	gender	no. telepon	Aksi
Nierman Wijaya kusumo	L	081293232	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>

### 3.3.1 Menampilkan Detail Data

Berisi detail data jamaah.



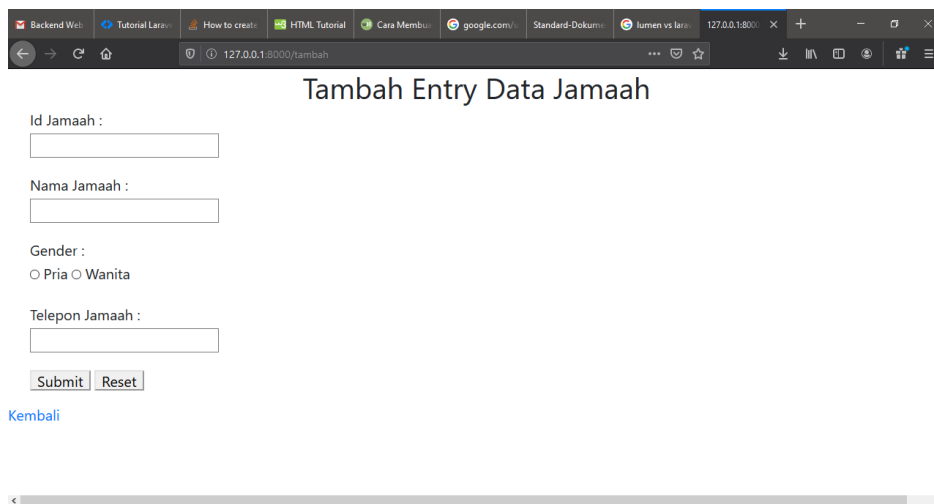
Data Jamaah

Id Jamaah	2
Nama	Nierman Wijaya kusumo
Gender	L
no. telepon	081293232

[Kembali](#)

### 3.3.2 Menambah Data

Tampilan bentuk form untuk menambah entry data jamaah.



Tambah Entry Data Jamaah

Id Jamaah :

Nama Jamaah :

Gender :   
☐ Pria ☐ Wanita

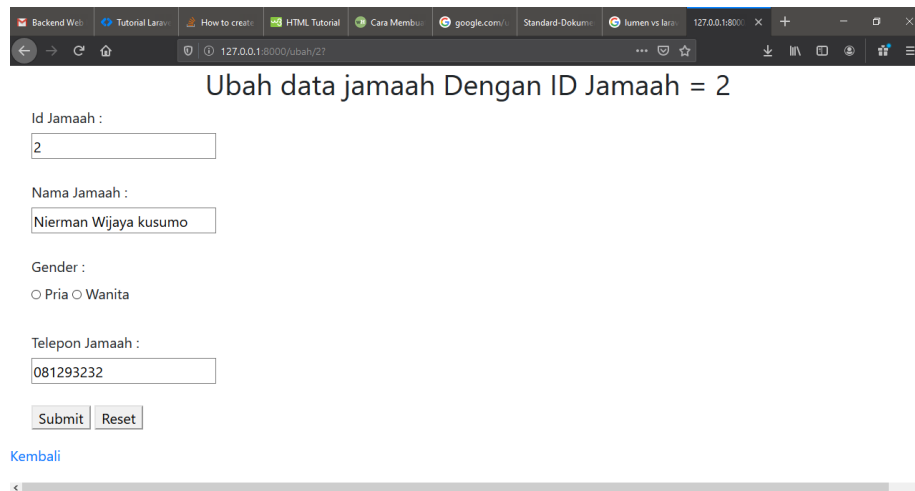
Telepon Jamaah :

[Submit](#) [Reset](#)

[Kembali](#)

### 3.3.3 Mengubah Data

Sama seperti form menambah data. Pada mengubah data terdapat id jamaah yang bisa dilihat di judul dan mengubah datanya.



Ubah data jamaah Dengan ID Jamaah = 2

Id Jamaah :

Nama Jamaah :

Gender :  
☐ Pria ☐ Wanita

Telepon Jamaah :

[Kembali](#)

### 3.3.4 Menghapus Data

Saat tabel aksi menekan hapus maka record pada table akan terhapus dan menampilkan alert seperti tampilan dibawah.

