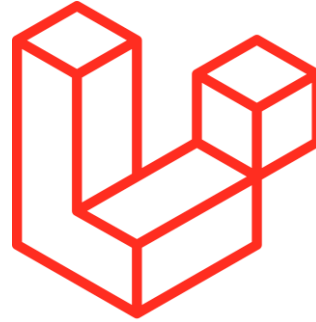


Pemrograman Backend



Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur



7. Form Processing dan Validasi

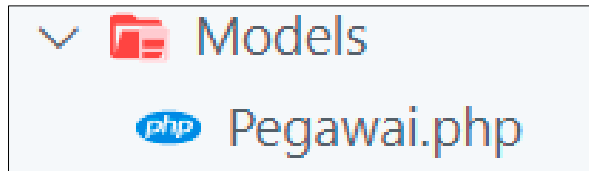


Mendefinisikan Model Eloquent

Untuk membuat model dari tabel pegawai yang sudah tersedia, buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\example-app>php artisan make:model Pegawai  
Model created successfully.
```

Dengan perintah di atas, maka di dalam folder models akan ada file baru Pegawai.php.





Konfigurasi Model Eloquent

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pegawai extends Model
{
    use HasFactory;
    //tambahkan kode untuk mapping ke tabel pegawai
    protected $table = 'pegawai';
}
```





Fungsi Insert pada Model Eloquent

Untuk membuat record baru dalam database dari sebuah model, cukup buat instance model baru dan panggil fungsi `save()` pada model eloquent. Berikut ini contoh membuat fungsi insert pada model eloquent:

```
$user = new User;  
$user->name = 'Fawwaz';  
$user->save();
```



Fungsi Update pada Model Eloquent

Untuk memperbarui data pada model, Anda dapat mengambil ID pada tabel, lalu mengubah sebuah kolom, dan gunakan fungsi `save()`. Berikut ini contoh membuat fungsi update pada model eloquent:

```
$user = User::find(1);  
$user->email = 'fawwaz@gmail.com';  
$user->save();
```



Fungsi Delete pada Model Eloquent

Untuk menghapus data pada model, Anda dapat mengambil ID pada suatu tabel, untuk menghapus data gunakan fungsi delete(). Berikut ini contoh membuat fungsi update pada model eloquent:

```
$user = User::find(1);  
$user->delete();
```





Proses Form Processing dan Validasi

Koneksi Database

Membuat Model Eloquent

Membuat Controller

Menambahkan Route

Membuat View





Koneksi Database

Bagi Anda untuk yang menggunakan XAMPP silahkan buka file **.env** yang berada di root folder aplikasi Anda. Lalu isikan seperti kode di dalam slide dengan penjelasan kode program sebagai berikut:

```
DB_CONNECTION=mysql //koneksi databasenya ke database mysql
DB_HOST=localhost //hostnya berada di lokal
DB_PORT=3306 //nomor port database mysql
DB_DATABASE=dblaravel //nama basis data
DB_USERNAME=root //user basis data
DB_PASSWORD=rahasia //password user basis data
```



Membuat Controller

Sebelum kita membuat form, memproses dan memvalidasinya, terlebih dahulu kita buat controller baru, buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\example-app>php artisan make:controller PegawaiController --resource  
Controller created successfully.
```

Dengan perintah di atas, maka di dalam folder app/Http/controller akan ada file baru PegawaiController.php. Di dalam file PegawaiController.php yang berada di folder folder app/Http/controller, secara otomatis akan terbentuk fungsi-fungsi CRUD (Create, Read, Update, Delete) tetapi masih kosong. Tugas kitalah yang akan mengisi body fungsi-fungsi yang berada di dalam controller tersebut.



Menambahkan Routing Baru

Selanjutnya tambahkanlah routing baru untuk mengakses fungsi-fungsi yang ada di PegawaiController. Silahkan buka file **routes/web.php** lalu tambahkan kode di bawah ini:

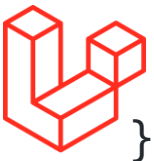
```
//Tambahkan route baru di routes/web.php  
use App\Http\Controllers\PegawaiController;  
  
Route::resource('pegawai', PegawaiController::class);
```



Fungsi Index untuk Menampilkan Data

<?php

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
//tambahan kode dibawah ini
use DB;
use App\Models\Pegawai;
class PegawaiController extends Controller
{
    public function index() //fungsi untuk menampilkan data pegawai
    {
        //dapatkan seluruh data pegawai dengan query builder
        $ar_pegawai = DB::table('pegawai')->get();
        //arahkan ke halaman baru dengan menyertakan data pegawai(compact)
        //di resources/views/pegawai/index.blade.php
        return view('pegawai.index',compact('ar_pegawai'));
    }
}
```





Fungsi Create untuk Menampilkan Form

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
//tambahan kode dibawah ini
use DB;
use App\Models\Pegawai;
class PegawaiController extends Controller
{
    public function create()
    {
        //mengarahkan ke hal form input
        return view('pegawai.form');
    }
}
```



Membuat Form di views/pegawai/form.blade.php

```
resources > views > pegawai >  form.blade.php > ...  
1  @extends('adminlte::page')  
2  @section('title', 'Data Pegawai')  
3  @section('content_header')  
4      <h1>Form Pegawai</h1>  
5  @stop  
6  @section('content')  
7      {{-- Isi Konten Form Pegawai --}}  
8  @stop  
9  @section('css')  
10     <link rel="stylesheet" href="/css/admin_custom.css">  
11 @stop  
12 @section('js')  
13     <script> console.log('Hi!'); </script>  
14 @stop
```

Validasi Form di views/pegawai/form.blade.php

Tambahkan validasi form, di atas form pegawai, tambahkan kode di bawah ini:

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



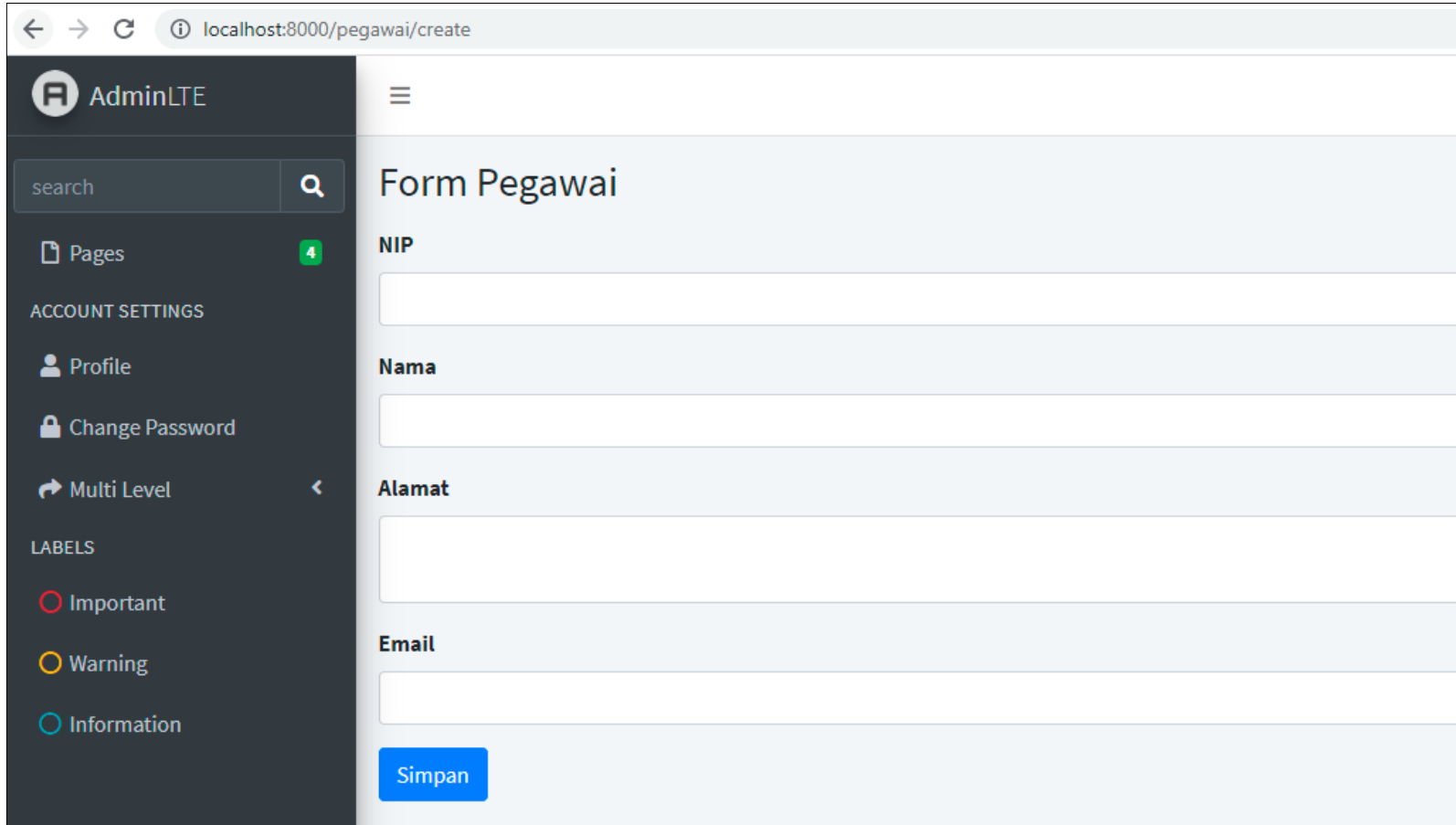


Isi Konten Form Pegawai

resources > views > pegawai >  form.blade.php > ...

```
6  @section('content')
7  <form method="POST" action="{{ route('pengarang.store') }}">
8      @csrf
9      {{-- cross-site request forgery (CSRF) = pencegahan serangan yang dilakukan
10         oleh pengguna yang tidak terautentikasi --}}
11      <div class="form-group">
12          <label>NIP</label><input type="text" name="nip" class="form-control"/>
13      </div>
14      <div class="form-group">
15          <label>Nama</label><input type="text" name="nama" class="form-control"/>
16      </div>
17      <div class="form-group">
18          <label>Alamat</label><textarea name="alamat" class="form-control"></textarea>
19      </div>
20      <div class="form-group">
21          <label>Email</label><input type="email" name="email" class="form-control"/>
22      </div>
23      <button type="submit" class="btn btn-primary">Simpan</button>
24  </form>
25  @stop
```


Tampilan Form Pegawai



The screenshot shows a web application interface for creating a new employee. The browser address bar indicates the URL is `localhost:8000/pegawai/create`. The interface features a dark sidebar on the left with the 'AdminLTE' logo and a search bar. The sidebar menu includes sections for 'ACCOUNT SETTINGS' (Profile, Change Password, Multi Level) and 'LABELS' (Important, Warning, Information). The main content area is titled 'Form Pegawai' and contains four input fields for 'NIP', 'Nama', 'Alamat', and 'Email'. A blue 'Simpan' (Save) button is located at the bottom of the form.

← → ↻ ⓘ localhost:8000/pegawai/create

AdminLTE

search 🔍

📁 Pages 4

ACCOUNT SETTINGS

👤 Profile

🔒 Change Password

🔗 Multi Level <

LABELS

🔴 Important

🟡 Warning

🟢 Information

☰

Form Pegawai

NIP

Nama

Alamat

Email

Simpan



Validasi Data di Fungsi Store(1)

```
public function store(Request $request)
{
    //1. proses validasi data
    $validasi = $request->validate(
    [
        'nip'=>'required|unique:pegawai|numeric',
        'nama'=>'required|max:50',
        'alamat'=>'required',
        'email'=>'required|max:50|regex:/(.+)(.+)@(.+)\.(.+)/i',
    ],
    //2. menampilkan pesan kesalahan
    //(di slide selanjutnya)
```





Validasi Data di Fungsi Store(2)

```
//pesan kesalahan saat invalid data (kelanjutan slide sebelumnya)
[
  'nip.required'=>'NIP Wajib di Isi',
  'nip.unique'=>'NIP Tidak Boleh Sama',
  'nip.numeric'=>'Harus Berupa Angka',
  'nama.required'=>'Nama Wajib di Isi',
  'alamat.required'=>'Alamat Wajib di Isi',
  'email.required'=>'Email Wajib di Isi',
  'email.regex'=>'Harus berformat email',
],
);
//2. processing form (di slide selanjutnya)
```





Processing Form

```
//lanjutan slide sebelumnya
//2. proses input data tangkap request dari form input
DB::table('pegawai')->insert(
    [
        'nip'=>$request->nip,
        'nama'=>$request->nama,
        'alamat'=>$request->alamat,
        'email'=>$request->email,
    ]
);
//2.landing page
return redirect('/pegawai');
}
```





Tampilan Validasi Data

localhost:8000/pegawai/create

AdminLTE

search

Pages 4

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Form Pegawai

- NIP Tidak Boleh Sama
- Nama Wajib di Isi
- Alamat Wajib di Isi
- Email Wajib di Isi

NIP

Nama

Alamat

Email

Simpan



Membuat View Data Pegawai

```
resources > views > pegawai > index.blade.php > ...
1  @extends('adminlte::page')
2  @section('title', 'Data Pegawai')
3  @section('content_header')
4      <h1>Data Pegawai</h1>
5  @stop
6  @section('content') {{-- Isi Konten Data Pegawai --}}
7  @php
8      $ar_judul = ['No', 'NIP', 'Nama', 'Alamat', 'Email'];
9      $no = 1;
10 @endphp
11 > <table class="table table-striped"> ...
30 </table>
31 @stop
32 @section('css')
33     <link rel="stylesheet" href="/css/admin_custom.css">
34 @stop
35 @section('js')
36     <script> console.log('Hi!'); </script>
37 @stop
```



Isi Konten Pegawai(1)

```
@extends('adminlte::page')
@section('title', 'Data Pegawai')
@section('content_header') <h1>Data Pegawai</h1> @stop
@section('content') {{-- Isi Konten Data Pegawai --}}
@php
$ar_judul = ['No', 'NIP', 'Nama', 'Alamat', 'Email']; $no = 1;
@endphp
<a class="btn btn-primary btn-md"
  href="{{ route('pegawai.create') }}" role="button">Tambah Pegawai</a><br/><br/>
<table class="table table-striped">
  <thead>
    <tr>
      @foreach($ar_judul as $jdl)
        <th>{{ $jdl }}</th>
      @endforeach
    </tr>
  </thead>
```





Isi Konten Pegawai(2)

```
{{-- Lanjutan Isi Konten Data Pegawai --}}
```

```
<tbody>
```

```
  @foreach($ar_pegawai as $peg)
```

```
    <tr>
```

```
      <td>{{ $no++ }}</td>
```

```
      <td>{{ $peg->nip }}</td>
```

```
      <td>{{ $peg->nama }}</td>
```

```
      <td>{{ $peg->alamat }}</td>
```

```
      <td>{{ $peg->email }}</td>
```

```
    </tr>
```

```
  @endforeach
```

```
</tbody>
```

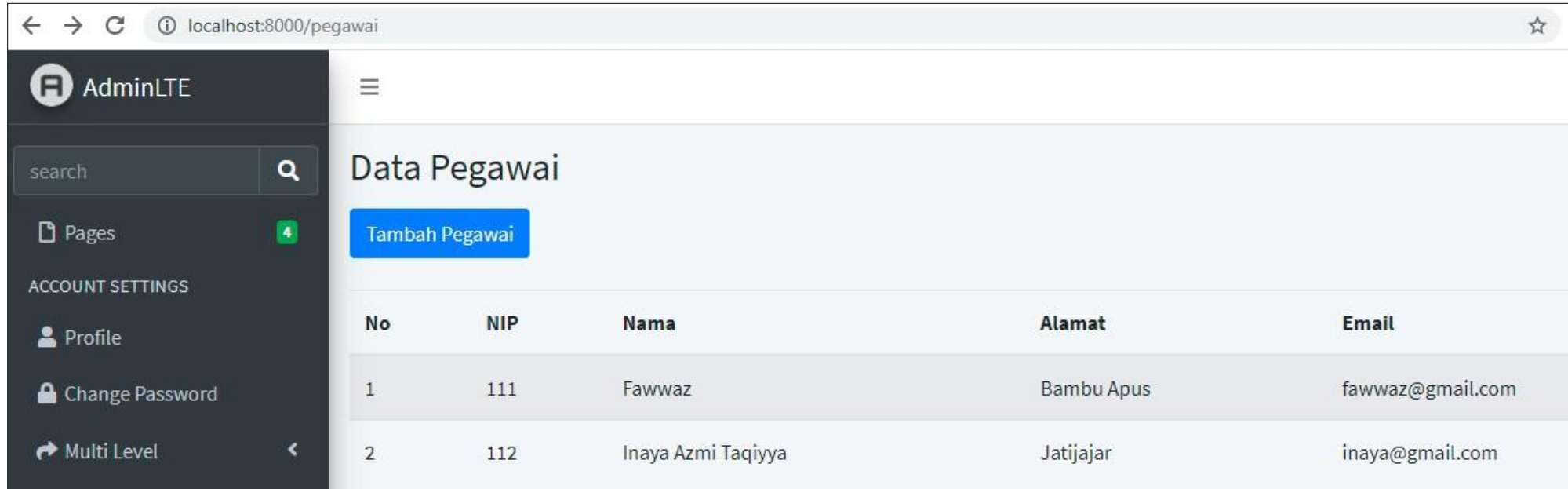
```
</table>
```

```
@stop
```



Tampilan Data Pegawai Setelah Berhasil Diinput

Setelah data berhasil diinput, maka akan diarahkan ke halaman `views/pegawai/index.blade.php`, tampilan halaman data pegawai seperti gambar di bawah ini:



The screenshot shows a web browser window with the URL `localhost:8000/pegawai`. The page features a dark sidebar on the left with the 'AdminLTE' logo and a search bar. The main content area is titled 'Data Pegawai' and includes a blue button labeled 'Tambah Pegawai'. Below this is a table with five columns: 'No', 'NIP', 'Nama', 'Alamat', and 'Email'. The table contains two rows of data.

| No | NIP | Nama | Alamat | Email |
|----|-----|--------------------|------------|------------------|
| 1 | 111 | Fawwaz | Bambu Apus | fawwaz@gmail.com |
| 2 | 112 | Inaya Azmi Taqiyya | Jatijajar | inaya@gmail.com |

**TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...**

