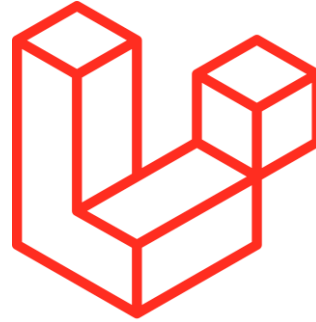


Pemrograman Backend



Pesantren PeTIK II YBM PLN

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



8. Model Eloquent dan Relationship



Implementasi Desain ER Multi Table dengan Relationship

Desain ER Multi Table

Membuat Model Eloquent

Membuat Controller

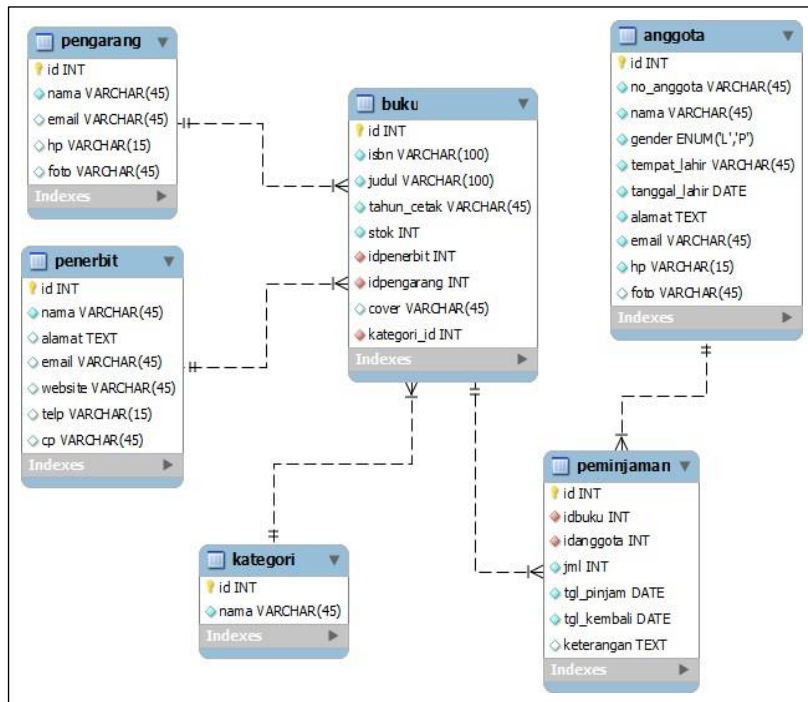
Menambahkan Route

Membuat View



Desain ER Multi Table dengan Relationship

Untuk membuat desain ER multi table dengan relationship, Anda bisa gunakan software MySQLWorkbench. Berikut ini contoh rancangan ERD perpustakaan:



Migration dan Seeding

Setelah selesai mendesain ER multi table dengan relationship dengan software MySQLWorkbench, selanjutnya forwardlah desain databasenya ke database MySQL/MariaDB dengan tools database → forward engineer.

Karena bekerja database dengan metode database first, maka kita tidak perlu memigrasi database, cukup melakukan seeding untuk data dummy. Silahkan lakukan seeding pada tabel-tabel yang sudah direlasikan, misalnya:

- Seorang pengarang mengarang banyak buku (one to many)
- Sebuah penerbit menerbitkan banyak buku (one to many)
- Sebuah kategori buku memiliki banyak buku (one to many)
- Dan seterusnya



Membuat Sedding

Untuk membuat sedding dari relasi tabel yang sudah ada, buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\nasrul>e:

E:\>cd xampp\htdocs\example-app

E:\xampp\htdocs\example-app>php artisan make:seeder PengarangSeeder
Seeder created successfully.

E:\xampp\htdocs\example-app>php artisan make:seeder PenerbitSeeder
Seeder created successfully.

E:\xampp\htdocs\example-app>php artisan make:seeder KategoriSeeder
Seeder created successfully.

E:\xampp\htdocs\example-app>php artisan make:seeder BukuSeeder
Seeder created successfully.
```

Mengisi Data Sedding Pengarang

Untuk mengisi data di tabel pengarang, caranya adalah buka file PengarangSeeder.php yang berada di folder seeders, lalu isi data seperti kode program di bawah ini:

use DB; //tambahkan kode ini di atas class PengarangSeeder extends Seeder

```
public function run()
{
    DB::table('pengarang')->insert(
        [
            [
                'nama' => 'Deden Hamdani', 'email' => 'deden@gmail.com', 'hp' => '0857123456',
            ],
            [
                'nama' => 'Dedi Iskandar', 'email' => 'dedi@gmail.com', 'hp' => '0857789456',
            ],
            [
                'nama' => 'Siti Aminah', 'email' => 'siti@gmail.com', 'hp' => '0813789456',
            ]
        ]
    );
}
```





Mengisi Data Sedding Penerbit

Untuk mengenerate data 10 penerbit secara acak, buka file PenerbitSeeder.php yang berada di folder seeders, lalu isi data seperti kode program di bawah ini:

use DB; //tambahkan kode ini di atas class PenerbitSeeder extends Seeder

```
public function run()
{
    for ($i=0; $i < 10; $i++) {
        DB::table('penerbit')->insert([
            'nama' => uniqid('nama_'),
            'alamat' => uniqid('alamat_'),
            'email' => uniqid().'@gmail.com',
            'website' => uniqid('www.').'com' ,
            'telp' => '021-'.rand(),
            'cp'=> uniqid('cp_')
        ]);
    }
}
```





Mengisi Data Seeding Kategori

Untuk mengisi data di tabel kategori, buka file KategoriSeeder.php yang berada di folder seeders, lalu isi data seperti kode program di bawah ini:

`use DB; //tambahkan kode ini di atas class KategoriSeeder extends Seeder`

```
public function run()
{
    DB::table('kategori')->insert(
        [
            [
                'nama' => 'Sejarah'
            ],
            [
                'nama' => 'Sosial & Politik'
            ],
            [
                'nama' => 'Pertanian'
            ]
        ]
    );
}
```





Mengisi Data Sedding Buku

Untuk mengisi data di tabel buku, caranya adalah buka file BukuSeeder.php yang berada di folder seeders, lalu isi data seperti kode program di bawah ini:

use DB; //tambahkan kode ini di atas class BukuSeeder extends Seeder

```
public function run()
{
    DB::table('buku')->insert(
        [
            [
                'isbn'=>'111', 'judul'=>'Sejarah Kemerdekaan RI', 'tahun_cetak'=>2021,
                'stok'=>10, 'idpengarang'=>1, 'idpenerbit'=>2, 'idkategori'=>1
            ],
            [
                'isbn'=>'112', 'judul'=>'Menanam Jagung Unggul', 'tahun_cetak'=>2021,
                'stok'=>15, 'idpengarang'=>2, 'idpenerbit'=>1, 'idkategori'=>3
            ]
        ]
    );
}
```



Eksekusi Data Seeding

Setelah mengisi data pegawai sesuai dengan kolom-kolomnya, langkah selanjutnya adalah mengeksekusi data seedingnya. Buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

```
E:\xampp\htdocs\example-app>php artisan db:seed --class=PengarangSeeder  
Database seeding completed successfully.
```

```
E:\xampp\htdocs\example-app>php artisan db:seed --class=PenerbitSeeder  
Database seeding completed successfully.
```

```
E:\xampp\htdocs\example-app>php artisan db:seed --class=KategoriSeeder  
Database seeding completed successfully.
```

```
E:\xampp\htdocs\example-app>php artisan db:seed --class=BukuSeeder
```



Mendefinisikan Model Eloquent

Untuk studi kasus ini, Kita menggunakan metode Database First. Setelah membuat ER multi tabel, selanjutnya kita definisikan model eloquent, buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

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

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

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

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



Konfigurasi Model Eloquent Pengarang

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

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





Konfigurasi Model Eloquent Penerbit

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

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





Konfigurasi Model Eloquent Kategori

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

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





Konfigurasi Model Eloquent Buku

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

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



Membuat Controller

Setelah membuat model-model eloquent, langkah selanjutnya adalah membuat controller di masing-masing tabel, buka terminal dan masuk ke root path aplikasi Laravel kita, jalankan perintah di bawah ini:

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

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

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

E:\xampp\htdocs\example-app>php artisan make:controller BukuController --resource
```



Menambahkan Routing Baru

Selanjutnya tambahkanlah routing baru untuk setiap controller yang sudah dibuat, buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
use App\Http\Controllers\PengarangController;  
use App\Http\Controllers\PenerbitController;  
use App\Http\Controllers\KategoriController;  
use App\Http\Controllers\BukuController;
```

```
Route::resource('pengarang', PengarangController::class);  
Route::resource('penerbit', PenerbitController::class);  
Route::resource('kategori', KategoriController::class);  
Route::resource('buku', BukuController::class);
```



Fungsi Index untuk Menampilkan Data Relasi Buku

```
<?php
//file BukuController.php yang berada di app/Http/Controllers
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use DB; //tambahan
class BukuController extends Controller
{
    public function index()
    {
        $ar_buku = DB::table('buku') //join tabel dengan Query Builder Laravel
            ->join('pengarang', 'pengarang.id', '=', 'buku.idpengarang')
            ->join('penerbit', 'penerbit.id', '=', 'buku.idpenerbit')
            ->join('kategori', 'kategori.id', '=', 'buku.idkategori')
            ->select('buku.*', 'pengarang.nama', 'penerbit.nama AS pen',
                'kategori.nama AS kat')->get();
        return view('buku.index',compact('ar_buku'));
    }
}
```



Membuat View Data Buku

Buatlah file `index.blade.php` untuk menampilkan data buku, di folder `resources/views/buku`, lalu tambahkan kode di bawah ini:

```
resources > views > buku >  index.blade.php > ...  
1  @extends('adminlte::page')  
2  @section('title', 'Data Buku')  
3  @section('content_header')  
4      <h1>Data Buku</h1>  
5  @stop  
6  @section('content')  
7      {{-- Isi Konten Data Buku --}}  
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
```



Isi Konten Data Buku(1)

```
@section('content')
@php
$ar_judul = ['No', 'ISBN', 'Judul', 'Stok', 'Pengarang', 'Penerbit', 'Kategori'];
$no = 1;
@endphp
<h3>Daftar Buku</h3>
<a class="btn btn-primary" href="{{ route('buku.create') }}"
  role="button">Tambah</a><br/><br/>
<table class="table table-striped">
  <thead>
    <tr>
      @foreach($ar_judul as $jdl)
        <th>{{ $jdl }}</th>
      @endforeach
    </tr>
  </thead>
  {{-- <tbody> lanjut ke slide berikutnya --}}
```



Isi Konten Data Buku(2)

{{-- Lanjutan Isi Konten Data Buku dari slide sebelumnya --}}

<tbody>

@foreach(\$ar_buku as \$b)

<tr>

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

<td>{{ \$b->isbn }}</td>

<td>{{ \$b->judul }}</td>

<td>{{ \$b->stok }}</td>

<td>{{ \$b->nama }}</td>

<td>{{ \$b->pen }}</td>

<td>{{ \$b->kat }}</td>

</tr>

@endforeach

</tbody>

</table>

@stop



Tampilan Data Relasi Buku

Setelah membuat halaman view relasi buku, selanjutnya buka halaman data relasi buku, di web browser, pada url ketikkan: `http://localhost:8000/buku`. Maka tampillah halaman relasi buku, seperti gambar di bawah ini:



No	ISBN	Judul	Stok	Pengarang	Penerbit	Kategori
1	111	Sejarah Kemerdekaan RI	10	Deden Hamdani	nama_60cc0ca9aec71	Sejarah
2	112	Menanam Jagung Unggul	15	Dedi Iskandar	nama_60cc0ca9a1dd8	Pertanian

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

