

# Pemrograman Web 2

Pertemuan 13

Laravel Eloquent: Relationship

# Eloquent Relationship

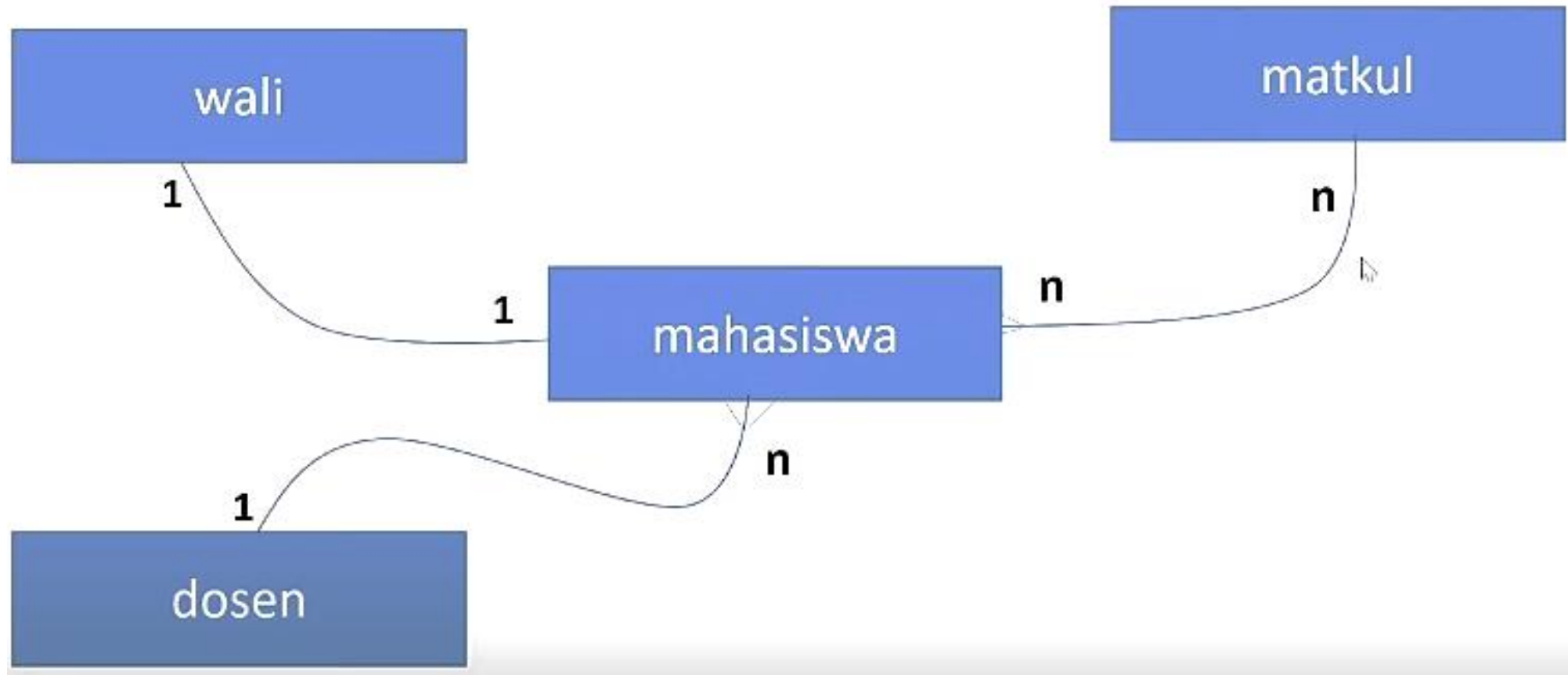
Eloquent Relationships adalah fitur di Laravel yang memungkinkan Kita untuk mendefinisikan hubungan antara model database dengan mudah.

Eloquent Relationships membantu dalam mengelola data yang terkait dengan models, seperti hubungan one-to-one, one-to-many, many-to-one, dan many-to-many.

Dengan menggunakan Eloquent Relationships, kita bisa mengakses data terkait dengan lebih mudah dan efisien, tanpa perlu menulis query SQL yang kompleks.

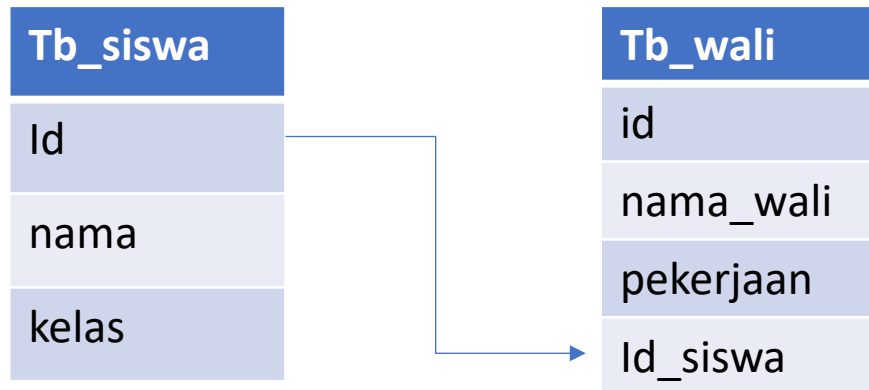
# Eloquent Relationship

Contoh Gambaran relationship:



# Eloquent Relationship

Contoh kita akan membuat tabel relasi pada tabel siswa dengan tabel wali murid



# Eloquent Relationship

- Pertama adalah kita harus membuat model dan migrate untuk tabel wali:

```
C:\xampp\htdocs\lat-laravel>php artisan make:model wali -m
```

```
INFO Model [C:\xampp\htdocs\lat-laravel\app\Models\wali.php] created successfully.
```

```
INFO Model [C:\xampp\htdocs\lat-laravel\app\Models\wali.php] created successfully.
```

```
INFO Model [C:\xampp\htdocs\lat-laravel\app\Models\wali.php] created successfully.
```

```
INFO Migration [C:\xampp\htdocs\lat-laravel\database\migrations/2025_06_05_083330_create_walis_table.php] created successfully.
```

- Dengan menjalankan perintah tersebut, kita berhasil membuat file model “wali” dan table “walis”

# Eloquent Relationship

- Selanjutnya isikan kode berikut pada file

“database\migrations\.....\_create\_walis\_table.php,”:

```
2025_06_05_083330_create_walis_table.php database\migrations\2025_06_05_083330_create_walis














1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('walis', function (Blueprint $table) {
15               $table->id();
16               $table->string('nama_wali');
17               $table->string('pekerjaan');
18               $table->string('id_siswa');
19               $table->timestamps();
20           });
21       }
22
23       /**
24        * Reverse the migrations.
25        */
26       public function down(): void
```

Selanjutnya lakukan migrate database dengan perintah “php artisan migrate”

- Lalu cek di phpMyAdmin apakah tabel wali sudah ada?
- Bila sudah ada silakan isi tabel wali dengan data!

# Eloquent Relationship

- Contoh tabel wali

 				id	nama_wali	pekerjaan	id_siswa	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Darman	PNS	1	2025-06-05 16:20:48	2025-06-05 16:20:48
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Didin	Swasta	2	2025-06-05 16:21:47	2025-06-05 16:21:47
	<input type="checkbox"/> Check all	With selected:		 Edit	 Copy		 Delete	 Export	

# Eloquent Relationship

- Buat route ke halaman “wali” dengan menambahkan kode berikut ke file routes/web.php:

```
15 Route::get('/siswa/wali', [SiswaController::class, 'wali']);
```

- Tambahkan function relationship pada file app/Http/controllers/SiswaController:

```
54  
55 // one to one relationship function  
56 v public function wali()  
57 v {  
58 v     $siswa = SiswaModel::all();  
59     return view('siswa.wali', ['siswas'=> $siswa]);  
60 }  
61  
62 }
```



# Eloquent Relationship

- Karena kita akan mengakses data wali dari halaman “data siswa”, maka kita perlu membuat link dari file `resources/views/siswa/list.blade.php`
- Kita akan menempatkan link tersebut di bawah tabel data siswa

```
list.blade.php resources\views\siswa\list.blade.php
12 <table class="table table-bordered">
20 <tr>
21 <td>{{ $siswa->nama }}</td>
22 <td>{{ $siswa->email }}</td>
23 <td>{{ $siswa->kelas }}</td>
24 <td>
25 <a href="{{ url('/siswa/update/' . $siswa->id) }}">Ubah</a> |
26 <a href="{{ url('/siswa/delete/' . $siswa->id) }}">Hapus</a>
27 </td>
28 </tr>
29 @endforeach
30 </table>
31 <a href="{{ url('/siswa/wali') }}">Data Wali</a>
32 @endsection
```

Link data wali yang  
ditambahkan posisi  
di bawah table data  
siswa

# Eloquent Relationship

- Selanjutnya buat file baru Bernama “wali.blade.php” di folder resources/views/siswa, lalu ketikan kode berikut ini:

```
wali.blade.php resources\views\siswa\wali.blade.php\table.table-bordered\tr\td
1  @extends('layouts.app')
2  @section('title', 'Data Wali')
3  @section('content')
4      <h1>Data Wali</h1>
5
6      <table class="table table-bordered">
7          <tr>
8              <th>Nama</th>
9              <th>Email</th>
10             <th>Kelas</th>
11             <th>Nama Wali</th>
12             <th>Pekerjaan Wali</th>
13         </tr>
14         @foreach($siswas as $siswa)
15             <tr>
16                 <td>{{ $siswa->nama }}</td>
17                 <td>{{ $siswa->email }}</td>
18                 <td>{{ $siswa->kelas }}</td>
19                 <td>{{ $siswa->wali->nama_wali }}</td>
20                 <td>{{ $siswa->wali->pekerjaan }}</td>
21             </tr>
22         @endforeach
23     </table>
24 @endsection
25
```

# Eloquent Relationship

- Selanjutnya isikan kode berikut pada file  
“app/Http/Models/SiswaModel.php”

```
SiswaModel.php app\Models\SiswaModel.php\...  
1  <?php  
2  namespace App\Models;  
3  use Illuminate\Database\Eloquent\Model;  
4  use Illuminate\Database\Eloquent\Factories\HasFactory;  
5  
6  class SiswaModel extends Model  
7  {  
8      use HasFactory;  
9      protected $table = 'siswa';  
10  
11     public function wali()  
12     {  
13         return $this->hasOne(Wali::class, 'id_siswa', 'id');  
14     }  
15 }  
16
```

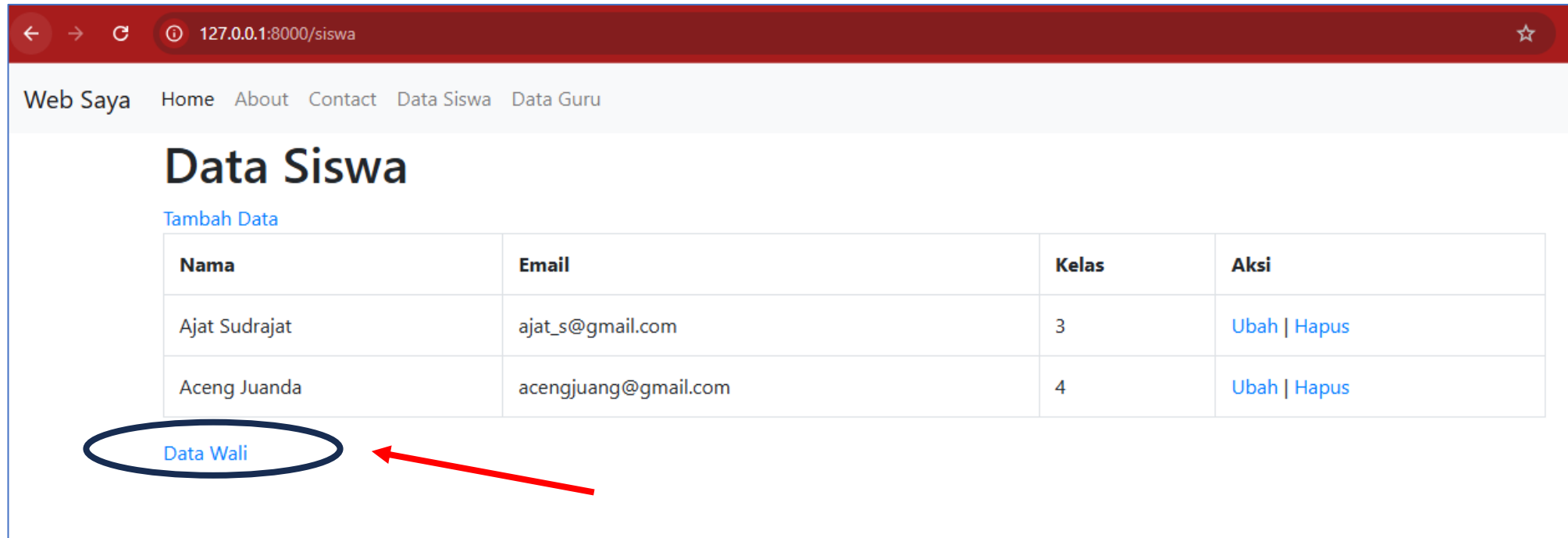
# Eloquent Relationship

- Selanjutnya isikan kode berikut pada file  
“app/Http/Models/wali.php”

```
wali.php app\Models\wali.php wali siswa
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class wali extends Model
9  {
10     use HasFactory;
11     protected $table = 'walis';
12     protected $fillable = [
13         'nama_wali',
14         'pekerjaan',
15         'id_siswa'
16     ];
17     public function siswa()
18     {
19         return $this->belongsTo(SiswaModel::class );
20     }
21 }
```

# Eloquent Relationship

- Langkah terakhir adalah jalankan server, lalu akses data wali dari halaman data siswa:



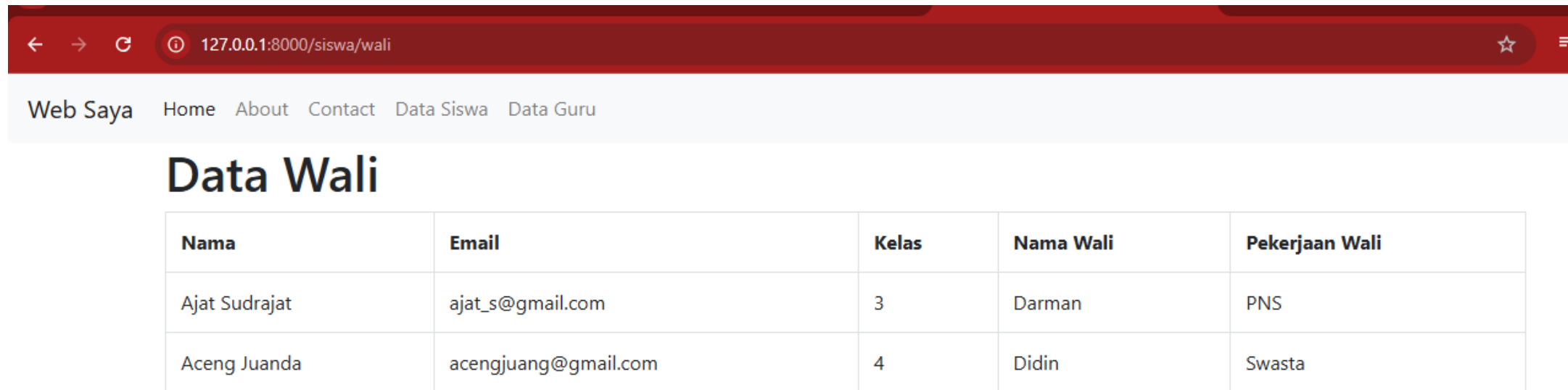
The screenshot shows a web browser at the address 127.0.0.1:8000/siswa. The page has a navigation bar with links: Web Saya, Home, About, Contact, Data Siswa, and Data Guru. The main heading is "Data Siswa". Below it is a link "Tambah Data". A table displays student information with columns: Nama, Email, Kelas, and Aksi. The table contains two rows of data. Below the table, the link "Data Wali" is circled in blue, and a red arrow points to it from the right.

Nama	Email	Kelas	Aksi
Ajat Sudrajat	ajat_s@gmail.com	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
Aceng Juanda	acengjuang@gmail.com	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>

[Data Wali](#)

# Eloquent Relationship

- Maka akan muncul halaman Data Wali



Nama	Email	Kelas	Nama Wali	Pekerjaan Wali
Ajat Sudrajat	ajat_s@gmail.com	3	Darman	PNS
Aceng Juanda	acengjuang@gmail.com	4	Didin	Swasta

# Latihan

- Buat relationship antara table guru dan tabel mata Pelajaran dengan skema many to many

