

Relationship

LARAVEL 8

Muhammad Nugraha, M.Eng

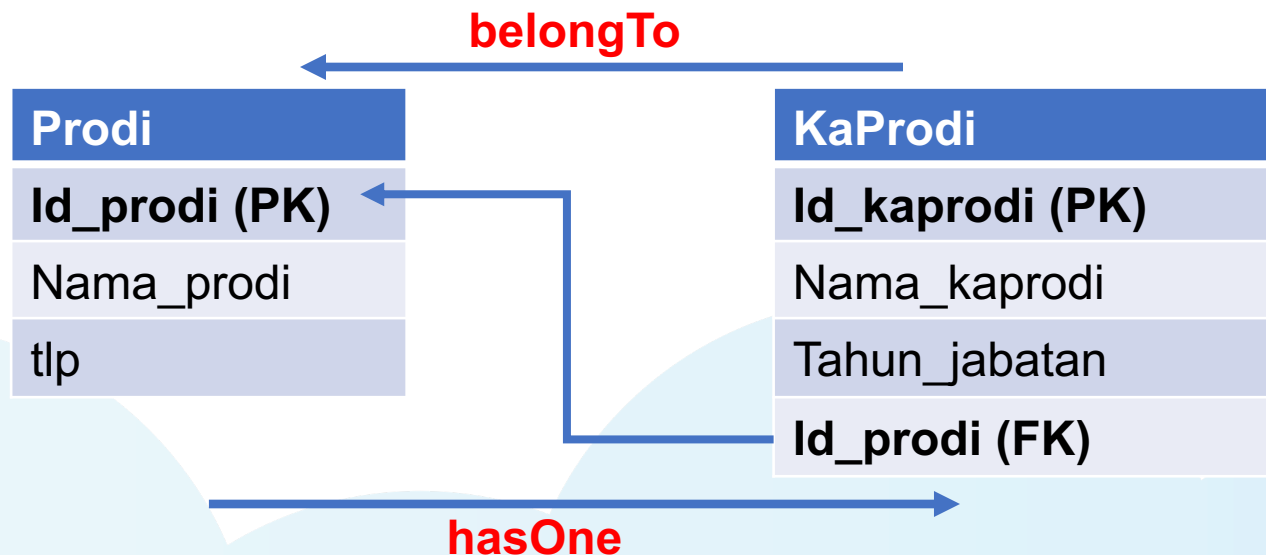


Relationship di Laravel

- ❑ Relationship merupakan salah satu feature yang ada di Laravel untuk menyambungkan antar tabel di database.
- ❑ Relationship ini bekerja seperti pada SQL untuk menghubungkan antar tabel dengan PK dan FK.
- ❑ Jenis relationship yang bisa kita gunakan ada 3, antaralain:
 - One to One
 - One to Many
 - Many to Many

One to One

- ❑ One to one relationship terjadi jika data A hanya boleh memiliki satu data B (A has one B) dan data B hanya boleh dimiliki oleh data A (B belongs to A).
- ❑ Dalam database, kita cukup memberikan foreign key di tabel B yang akan diisi dengan ID dari data di tabel A.
- ❑ ID tersebut merepresentasikan data A yang memiliki data di tabel B.



Id_prodi	Nama_prodi	tlp
1	Teknologi Mesin	0262 - 8301041
2	Mekatronika	0262 - 8301041
3	Teknologi Listrik	0262 - 8301041
4	Teknologi RPL	0262 - 8301041

hasOne

belongsTo

Id_kaprodi	Nama_kaprodi	Tahun_jabatan	Id_prodi
012	Ade Irvan	2018 - 2023	1
023	Emanuel Agung	2018 - 2023	2
033	Andi Tulus	2018 - 2023	3
042	Muhammad Nugraha	2018 - 2023	4

Mendefinisikan Model

- ❑ Buat model prodi dan model kaprodi (catatan: sebelumnya tabel prodi dan kaprodi sudah di buat di DB).

```
class Prodi extends Model
{
    use HasFactory;

    protected $table = "prodi";
    protected $primaryKey = "Id_prodi";

    public function kaprodi()
    {
        return $this->hasOne('App\Models\KaProdi');
    }
}
```

```
class KaProdi extends Model
{
    use HasFactory;

    protected $table = "kaprodi";
    protected $primaryKey = "Id_kaprodi";

    public function prodi()
    {
        return $this->belongsTo('App\Models\Prodi');
    }
}
```

○ Membuat Controller Prodi

```
class ProdiController extends Controller
{
    public function index()
    {
        $prodi = Prodi::all();
        return view('prodi', ['dataProdi' => $prodi]);
    }
}
```

Membuat View Prodi

```
<table class="table table-bordered table-striped">
  <thead>
    <tr>
      <th>ID Prodi</th>
      <th>Nama Prodi</th>
      <th>Nama Kaprodi</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($prodi as $p)
      <tr>
        <td>{{ $p->Id_prodi }}</td>
        <td>{{ $p->Nama_prodi }}</td>
        <td>{{ $p->kaprodi->Nama_kaprodi }}</td>
      </tr>
    @endforeach
  </tbody>
</table>
```

pada model prodi
memanggil function kaprodi
dan kolom Nama_kaprodi



○ One to Many

- ❑ Cara kerja relasi one to many di Laravel konsepnya saja saja seperti one to one, hanya saja beda perintah di modelnya.
- ❑ Jika di relasi one to one nama relasinya **hasOne** dan **belongsTo**, maka di one to many nama relasinya **hasMany** dan **belongsTo**.

```
class Prodi extends Model
{
    use HasFactory;

    protected $table = "prodi";
    protected $primaryKey = "Id_prodi";

    public function kaprodi()
    {
        return $this->hasMany('App\Models\KaProdi');
    }
}
```

```
class KaProdi extends Model
{
    use HasFactory;
    protected $table = "kaprodi";
    protected $primaryKey = "Id_kaprodi";

    public function prodi()
    {
        return $this->belongsTo('App\Models\Prodi');
    }
}
```

Ket: contoh model di atas hanya menunjukkan penggunaan syntax **hasMany** saja, bukan relasi sebenarnya

○ Many to Many

- ❑ Cara kerja relasi many to many juga sama dengan yang sebelumnya, hanya saja beda perintah di modelnya.
- ❑ Jika di relasi one to many nama relasinya **hasMany** dan **belongsTo**, maka di many to many nama relasinya **belongsToMany** dan **BelongsToMany**.

```
class Prodi extends Model
{
    use HasFactory;

    protected $table = "prodi";
    protected $primaryKey = "Id_prodi";

    public function kaprodi()
    {
        return $this->belongsToMany('App\Models\KaProdi');
    }
}
```

```
class KaProdi extends Model
{
    use HasFactory;
    protected $table = "kaprodi";
    protected $primaryKey = "Id_kaprodi";

    public function prodi()
    {
        return $this->belongsToMany('App\Models\Prodi');
    }
}
```

Ket: contoh model di atas hanya menunjukkan penggunaan syntax **belongsToMany** saja, bukan relasi sebenarnya

Thanks !