Eloquent ORM Quick Reference

Tanggal: 27 November 2025

1. Model Dasar

Eloquent ORM adalah Active Record pattern di Laravel. Setiap tabel database memiliki Model untuk berinteraksi dengannya.

Membuat Model

```
# Dengan migration
php artisan make:model Nasabah -m

# Dengan factory dan seeder
php artisan make:model Transaksi -mfs

# Dengan semua files
php artisan make:model Rekening -a
```

Struktur Model

```
class Nasabah extends Model
{
    use HasFactory;

    protected $table = 'nasabah';

    protected $fillable = [
        'nama', 'email', 'no_ktp',
        'no_rekening', 'saldo',
    ];

    protected $hidden = ['no_ktp', 'pin_hash'];

    protected function casts(): array
    {
        return [
            'saldo' => 'decimal:2',
            'is_active' => 'boolean',
        ];
    }
}
```

2. Query Dasar

Retrieve Data

```
// Semua records
$nasabah = Nasabah::all();

// Find by ID
$nasabah = Nasabah::find(1);
$nasabah = Nasabah::findOrFail(1);

// Where clauses
$aktif = Nasabah::where('status', 'aktif')->get();

$priority = Nasabah::where('saldo', '>=', 100000000)
    ->where('tier', 'priority')
    ->get();

// whereIn, whereBetween
$nasabah = Nasabah::whereIn('id', [1, 2, 3])->get();
$trx = Transaksi::whereBetween('tanggal', [
    '2024-01-01', '2024-01-31'
])->get();
```

Ordering & Pagination

```php
$nasabah = Nasabah::orderBy('nama', 'asc')->get();
$terbaru = Nasabah::latest()->get();
$nasabah = Nasabah::take(10)->skip(10)->get();
$nasabah = Nasabah::paginate(15);
```

3. Relationships

One to One

```
class Nasabah extends Model
{
    public function akun(): HasOne
    {
        return $this->hasOne(Akun::class);
    }
}

class Akun extends Model
{
    public function nasabah(): BelongsTo
    {
        return $this->belongsTo(Nasabah::class);
    }
}
```

One to Many

```
class Nasabah extends Model
{
    public function transaksi(): HasMany
    {
        return $this->hasMany(Transaksi::class);
    }
}

class Transaksi extends Model
{
    public function nasabah(): BelongsTo
    {
        return $this->belongsTo(Nasabah::class);
    }
}
```

Many to Many

```
class Nasabah extends Model
{
    public function produk(): BelongsToMany
    {
        return $this->belongsToMany(Produk::class)
            ->withPivot('tanggal_buka', 'status')
            ->withTimestamps();
    }
}

// Usage
$nasabah->produk()->attach($produkId);
$nasabah->produk()->detach($produkId);
$nasabah->produk()->sync([1, 2, 3]);
```

4. Eager Loading

Mengatasi N+1 query problem.

Basic Eager Loading

```
// Tanpa eager loading (N+1 problem)
$nasabah = Nasabah::all();
foreach ($nasabah as $n) {
    echo $n->akun->no_rekening; // Query tiap iterasi!
}

// Dengan eager loading
$nasabah = Nasabah::with('akun')->get();

// Multiple relationships
$nasabah = Nasabah::with(['akun', 'transaksi'])->get();

// Nested
$nasabah = Nasabah::with('transaksi.kategori')->get();
```

Constraining Eager Loads

```php
$nasabah = Nasabah::with([
    'transaksi' => function ($query) {
        $query->where('status', 'sukses')
              ->orderBy('tanggal', 'desc')
              ->take(10);
    }
])->get();

// With count
$nasabah = Nasabah::withCount('transaksi')->get();
echo $nasabah->first()->transaksi_count;
```

5. Query Scopes

Local Scopes

```
class Nasabah extends Model
{
    public function scopeAktif($query)
    {
        return $query->where('status', 'aktif');
    }

    public function scopeTier($query, $tier)
    {
        return $query->where('tier', $tier);
    }

    public function scopeMinSaldo($query, $amount)
    {
        return $query->where('saldo', '>=', $amount);
    }
}

// Usage
$priority = Nasabah::aktif()
    ->tier('priority')
    ->minSaldo(100000000)
    ->get();
```

Mutators & Accessors

```
class Nasabah extends Model
{
    protected function saldo(): Attribute
    {
        return Attribute::make(
            get: fn ($v) => number_format($v, 2, ',', '.')
        );
    }

    protected function pin(): Attribute
    {
        return Attribute::make(
            set: fn ($v) => bcrypt($v)
        );
    }
}
```