

MODUL 7

DATABASE QUERY BUILDER



CAPAIAN PEMBELAJARAN

1. Bisa membuat memanipulasi Rekaman Database: Query Builder
2. Mengimplementasikan method DB table(), raw(), select(), where(), get(), first(), sum(), count(), min(), max(), avg(), distinct(), selectRaw(), groupBy(), join().



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Sistem Operasi Linux/Window 10
2. PHP 5.x, diutamakan versi 7.x keatas
3. Composer
4. XAMPP (PHP, Apache, MySQL)
5. Text Editor Notepad++ atau Visual Studio Code



DASAR TEORI

Laravel telah menyediakan kelas untuk mengakses ke tabel lainnya yaitu menggunakan kelas `Illuminate\Support\Facades\DB`;

Query Builder adalah perintah SQL untuk mengakses rekaman dalam tabel yang dituliskan dengan menggunakan method. Nama method sama dengan perintah di SQL, misal: SELECT menggunakan `select(..)`, table menggunakan `table(..)`, WHERE menggunakan method `where(..)`.

Rangkaian perintah menggunakan method berantai sebagai contoh:

```
DB::table(...)->select(...)->where(...)->get().
```

Beberapa perintah SQL yang tidak didukung method menggunakan raw SQL, misal:

```
select(..)->sum(DB::raw('stok*harga')).
```

Query builder dibangun menggunakan pustaka *PHP Data Object (PDO)*, untuk melindungi aplikasi terhadap serangan injeksi SQL, perlu menambahkan parameter sebagai mengikat.

Saat ini Laravel mendukung empat jenis database antara lain:

- MySQL
- PostgreSQL
- SQLite
- SQL Server



PRAKTIK

1. Buatlah controller dengan perintah:

```
php artisan make:controller DbController
```

2. Membaca rekaman tabel kota, buka DbController.php dan tambahkan seperti pada kode program berikut:

```
public function bacaDb1()
{
    $kota = DB::table('kota')->get();
    return view('db.bacaDb1', ['kota' => $kota]);
}
```

3. Tambahkan views bacaDb1.blade.php simpan ke dalam folder views/db/.., seperti program berikut:

```
@extends('layouts.app')
<h3>TABEL KOTA</h3>
<table class="table table-striped table-bordered">
<tr><td>ID</td><td>KOTA</td><td>ID PROP</td></tr>
@foreach($kota as $k)
    <tr>
        <td>{{ $k->id }}</td>
        <td>{{ $k->nama_kota }}</td>
        <td>{{ $k->propinsi_id }}</td>
    </tr>
@endforeach
</table>
```

3. Untuk mengakses tambahan file `routes/web.php`, seperti pada kode program berikut:

```
Route::get('db/bacaDb1', 'DbController@bacaDb1');
```

4. Uji dengan memanggil url <http://localhost:8000/db/bacaDb1>

4. Jika hanya ingin membaca satu baris rekaman dari tabel dalam database, bisa menggunakan method `first()`. Buka `DbController.php` dan tambahkan seperti pada kode program berikut:

```
public function bacaDb2()
{
    $kota = DB::table('kota')->where('nama_kota', 'Bantul')
        ->first();

    echo $kota->id."->";
    echo $kota->nama_kota;
}
```

5. Menggunakan tabel `barang` dengan isi rekaman seperti pada tabel berikut:

id	jenis_id	nama_barang	satuan	harga	stok	user_id
.0001	1	Air Mineral	Botol	5000	7	1
.0002	2	Minyak Goreng	Pcs	20000	16	1
.0003	1	Penggaris	Batang	5000	100	1
.0004	1	Penghapus	Pcs	4000	29	1
.0005	1	Buku Bergaris	Lb	5000	8	1
.0007	1	Spidol	Batang	5000	200	1
.0008	2	Gula Putih	Kg	10000	10	1
.0009	1	Buku Gambar	Pcs	10000	10	1
.0010	1	Buku Bacaan	Pcs	10000	10	1
.0011	1	Kertas A 4	Rem	30000	2	1
.0012	1	Buku Gambar Mewarn	Pcs	10000	10	1
.0013	2	Kopi Mix	Saset	2500	100	1
.0020	1	Buku Tulis	Pcs	10000	10	1

```
$hargaMaks = DB::table('barang')->max('harga');

//harga rata3 fungsi aggregate avg(), dengan jenis_id=1
$hargaRata2 = DB::table('barang')
    ->where('jenis_id', '1')
    ->avg('harga');

//harga minimal/fungsi aggregate min(), dengan jenis_id=1
$hargaMin = DB::table('barang')
    ->where('jenis_id', '1')
    ->min('harga');

//harga maksimal/fungsi aggregate max(), dengan jenis_id=1
$hargaMax = DB::table('barang')
    ->where('jenis_id', '1')
    ->max('harga');

// harga x stok
$jumHarga = DB::table('barang')
    ->where('jenis_id', '1')
    ->sum(DB::raw('stok*harga'));

//membaca rekaman memilih kolom memilih id, nama_barang,
```

```

//satuan, stok//
$barang = DB::table('barang')
    ->select('id','nama_barang','satuan','stok')
    ->get();

//membaca semua rekaman semua kolom
$barang = DB::table('barang')->select('*')->get();

//membaca rekaman menggunakan DISTINCT
$barang = DB::table('barang')
    ->select('jenis_id')
    ->distinct('jenis_id')
    ->get();

//menjumlahkan per kelompok jenis barang
$barang = DB::table('barang')
    ->select('jenis_id',DB::raw('count(*) as jumlah'))
    ->groupBy('jenis_id')
    ->get();

//membaca jenis_id, dan harga * stok
$jumlahHarga=DB::table('barang')
    ->where('jenis_id','1')
    ->sum(DB::raw('stok*harga'));

//membaca rekaman memilih kolom memilih id, nama_barang,
//satuan, stok//
//dan harga * stok jenis barang=1
$barang = DB::table('barang')
    ->select('id','nama_barang','stok','satuan','harga')
    ->selectRaw('stok * harga as total')
    ->where('jenis_id',1)
    ->get();

```

5. Menambah Rekaman barang seperti pada kode program berikut:

```

public function insertData()
{
    $br= DB::table('barang')->insert(
        [
            'id'=>10021,
            'jenis_id'=>2,
            'nama_barang'=>'Tepung Trigu',
            'satuan'=>'Kg',
            'harga'=>5000,
            'stok'=>10,
            'user_id'=>1]);
    echo "Berhasil menyimpan ". $br." rekaman";
}

```

6. Mengubah rekaman barang dengan id=10021

```
public function updateData()
{
    $stok=100;
    $br=DB::table('barang')->where('id',10021)
        ->update(['stok'=>$stok]);

    echo "Berhasil mengubah ". $br." rekaman";
}
```

7. Menghapus 1 rekaman barang

```
public function deleteData()
{
    $br=DB::table('barang')->where('id',10021)->delete();

    echo "Berhasil menghapus ". $br." rekaman";
}
```



LATIHAN

1. Buatlah program untuk merekan katalog buku menggunakan tabel pada pertemuan sebelumnya menggunakan query builder, insert(), update(), delete()
2. Buatlah program untuk menampilkan daftar buku, dan pengarang,dengan menggunakan relasi join(),



TUGAS

1. Suatu kasus penjualan barang transaksi akan dilakukan melalui online, berbasis web, terdiri dari tabel barang, jual, dan detail_jual.
2. Pada saat barang dijual, akan mengurangi stok barang



REFERENSI

1. Database: Query Builder, <https://laravel.com/docs/5.8/queries>