

QUERY BUILDER

1.1 Kompetensi Utama

- Diharapkan mahasiswa dapat memahami Query Builder pada laravel.

1.2 Kompetensi Penunjang

- Mengenalkan kepada mahasiswa tentang Query Builder
- Membuat query SQL menggunakan Query Builder pada laravel

1.3 Dasar Teori

1. Pengertian Query Builder

Query builder adalah interface khusus yang disediakan Laravel untuk mengakses database. Berbeda dengan raw query dimana kita menulis langsung perintah query SQL, di dalam query builder perintah SQL ini diakses menggunakan method. Artinya, kita tidak menulis langsung perintah SQL, tapi hanya memanggil method-method saja.

Keunggulan dari cara ini adalah kode program kita menjadi lebih rapi dan lebih mudah dibaca. Karena tidak menulis query secara langsung, program kita tidak terikat ke satu jenis database saja. Artinya jika ingin beralih dari database MySQL ke SQLite, tidak akan banyak kendala. Namun kelemahan dari query builder, kita terpaksa harus mempelajari nama-nama method yang ada.

Sebagai bahan praktek, buat file migrasi untuk tabel dosens dan controller DosenController.

```
php artisan make:migration create_dosens_table --create=dosens
```

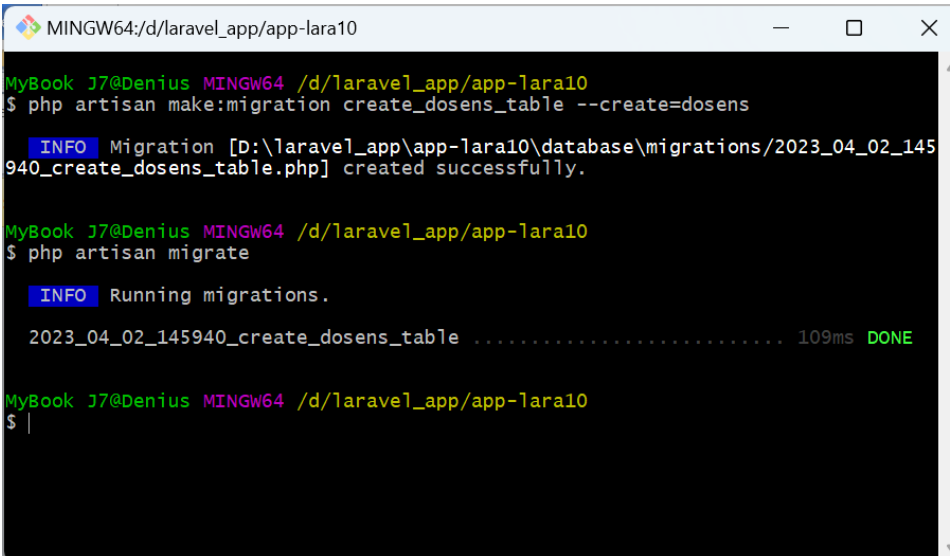


```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan make:migration create_dosens_table --create=dosens
INFO Migration [D:\laravel_app\app-lara10\database\Migrations\2023_04_02_145940_create_dosens_table.php] created successfully.
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ |
```

Buka file migrasi dosen dan buat struktur tabel dosen seperti di bawah ini :

```
9 {
10     /**
11      * Run the migrations.
12      */
13     public function up(): void
14     {
15         Schema::create('dosens', function (Blueprint $table) {
16             $table->id();
17             $table->char('nik',18);
18             $table->string('nama');
19             $table->string('email')->unique();
20             $table->char('no_telp',15);
21             $table->string('prodi');
22             $table->string('alamat');
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
```

Save file migrate, lalu jalankan kembali php artisan migrate:



```
MINGW64:/d/laravel_app/app-lara10
MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan make:migration create_dosens_table --create=dosens

[INFO] Migration [D:\laravel_app\app-lara10\database\migrations\2023_04_02_145940_create_dosens_table.php] created successfully.

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ php artisan migrate

[INFO] Running migrations.

2023_04_02_145940_create_dosens_table ..... 109ms DONE

MyBook J7@Denius MINGW64 /d/laravel_app/app-lara10
$ |
```

Pastikan tabel dosen sudah berhasil di buat atau lihat di phpmyadmin

Server: 127.0.0.1 » Database: app_lara10 » Table: dosen

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nik	char(18)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 nama	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 no_telp	char(15)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 prodi	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 alamat	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	8 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	9 updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all
 With selected: ☐ Browse ☐ Change ☐ Drop ☐ Primary ☐ Unique ☐ Index ☐ Spatial ☐ Full

Print Propose table structure Move columns Normalize

Add 1 column(s) after updated_at Go

Buat DosenController dengan menjalankan :

php artisan make:controller DosenController

Kemudian buka file **route/web.php** dan tambahkan beberapa route di bawah ini :

```

10
11
12
13
14
15
16
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
22 Route::get('/insert-dosen', [DosenController::class, 'insertDosen']);
23 Route::get('/insert-banyak-dosen', [DosenController::class, 'insertBanyakDosen']);
24 Route::get('/update-dosen', [DosenController::class, 'updateDosen']);
25 Route::get('/update-where-dosen', [DosenController::class, 'updateWhereDosen']);
26 Route::get('/update-or-insert', [DosenController::class, 'updateOrInsert']);
27 Route::get('/delete-dosen', [DosenController::class, 'deleteDosen']);
28 Route::get('/get', [DosenController::class, 'get']);
29 Route::get('/get-tampil', [DosenController::class, 'getTampil']);
30 Route::get('/get-view', [DosenController::class, 'getView']);
31 Route::get('/get-where', [DosenController::class, 'getWhere']);
32 Route::get('/select-dosen', [DosenController::class, 'selectDosen']);
33 Route::get('/take', [DosenController::class, 'take']);
34 Route::get('/first', [DosenController::class, 'first']);
35 Route::get('/find', [DosenController::class, 'find']);
36 Route::get('/raw', [DosenController::class, 'raw']);
37

```

2. Input Data (DB::insert)

Query builder menyediakan method insert() untuk proses input data ke dalam tabel. Method ini butuh sebuah argument dalam bentuk associative array dengan key berupa nama kolom tabel.

Untuk mempraktekkannya, tambahkan method insertDosen pada DosenCotroller:

```

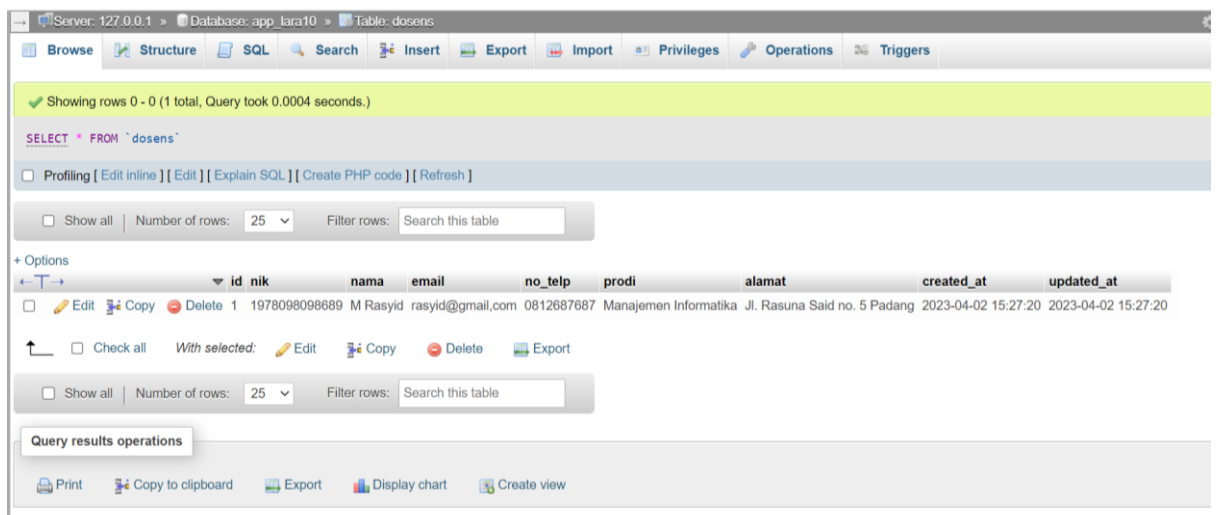
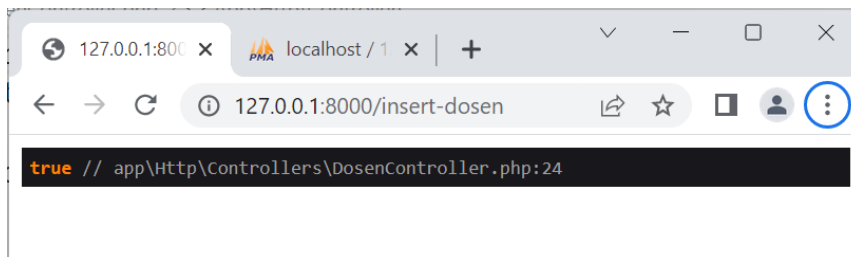
app > Http > Controllers > DosenController.php > {} App\Http\Controllers
2 namespace App\Http\Controllers;
3 use Illuminate\Support\Facades\DB;
4
5 class DosenController extends Controller
6 {
7     public function insertDosen()
8     {
9         $query=DB::table('dosens')->insert(
10             [
11                 'nik'=>'1978098098689',
12                 'nama'=>'M Rasyid',
13                 'email'=>'rasyid@gmail.com',
14                 'no_telp'=>'0812687687',
15                 'prodi'=>'Manajemen Informatika',
16                 'alamat'=>'Jl. Rasuna Said no. 5 Padang',
17                 'created_at'=>now(),
18                 'updated_at'=>now(),
19             ]
20         );
21         dd($query);
22     }
23 }

```

Jangan lupa, Dalam controller ini kita memerlukan **DB facade**, sehingga butuh perintah untuk mengimport-nya, yakni kode pada baris 3.

Use `Illuminate\Support\Facades\DB`.

Jalankan **localhost:8000/insert-dosen** pada browser, untuk memastikan data berhasil tersimpan, silahkan buka phpmyadmin dan lihat isi tabel dosens



3. Input Banyak Data

Method insert() juga bisa melakukan banyak input sekaligus. Caranya, isi argument dengan nested array atau array 2 dimensi, seperti contoh berikut:

```
app > Http > Controllers > DosenController.php > DosenController > insertBanyakDosen
23
24 public function insertBanyakDosen()
25 {
26     $query=DB::table('dosens')->insert(
27     [
28         [
29             'nik'=>'19800089212',
30             'nama'=>'M Yazid',
31             'email'=>'yazid@gmail.com',
32             'no_telp'=>'081378979',
33             'prodi'=>'TRPL',
34             'alamat'=>'Jl. Sutomo no. 1 Padang',
35             'created_at'=>now(),
36             'updated_at'=>now(),
37         ],
38         [
39             'nik'=>'19828887678',
40             'nama'=>'Deni',
41             'email'=>'deni@gmail.com',
42             'no_telp'=>'0812887888',
43             'prodi'=>'TRPL',
44             'alamat'=>'Jl. M Hatta no. 15 Padang',
45             'created_at'=>now(),
46             'updated_at'=>now(),
47         ],
48     ],
49     );
50     dd($query);
51 }
52 }
```

Jalankan method insertBanyakDosen() dengan mengakses alamat:

localhost:8000/insert-banyak-dosen, kemudian lihat hasilnya di phpmyadmin

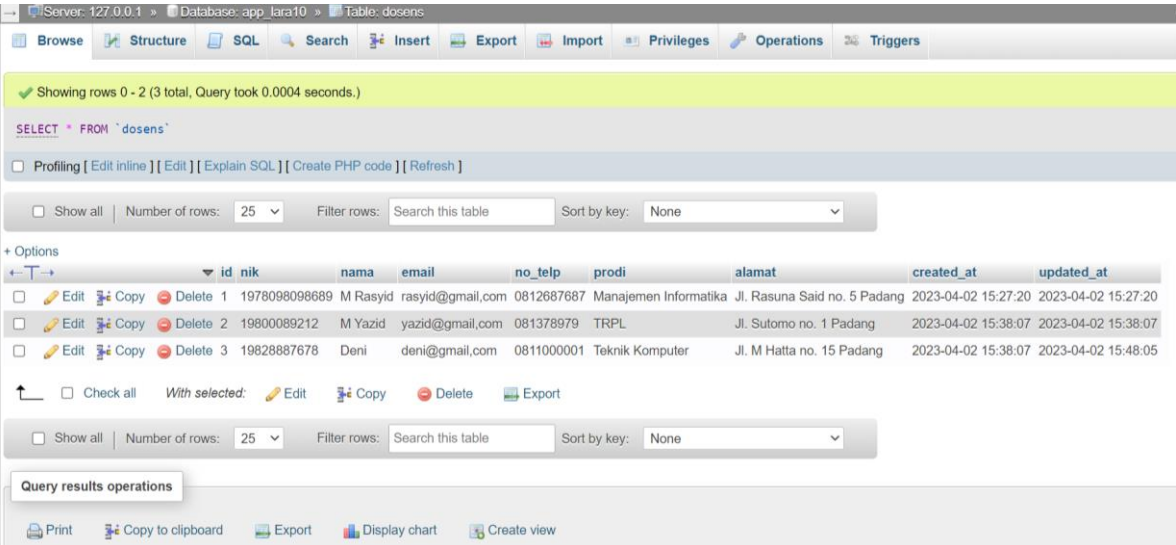
Server: 127.0.0.1 » Database: app_lara10 » Table: dosens

4. Mengupdate Data

Untuk proses update data tabel, query builder menyediakan method update. Method ini butuh sebuah argument berbentuk array yang berisi data yang akan di update. Namun untuk proses update, kita butuh bantuan dari method where() untuk menentukan kondisi yang dicari Sebagai contoh, mengubah data prodi dan no_telp dari dosen bernama 'Deni', berikut kode yang dibutuhkan:

```
app > Http > Controllers > DosenController.php > DosenController
50     dd($query);
51 }
52
53 public function updateDosen()
54 {
55     $query=DB::table('dosens')
56         ->where('nama','Deni')
57         ->update(
58         [
59             'no_telp'=>'0811000001',
60             'prodi'=>'Teknik Komputer',
61             'updated_at'=>now(),
62         ]
63     );
64     dd($query);
65 }
66 }
67
```

Argument dari method update() ini berbentuk associative array dengan data yang ingin di update. Jalankan method updateDosen() dengan mengakses alamat: localhost:8000/update-dosen, kemudian lihat hasilnya di phpmyadmin



Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

SELECT * FROM `dosens`

Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	nik	nama	email	no_telp	prodi	alamat	created_at	updated_at
<input type="checkbox"/>	1	1978098098689	M Rasyid	rasyid@gmail.com	0812687687	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang	2023-04-02 15:27:20	2023-04-02 15:27:20
<input type="checkbox"/>	2	19800089212	M Yazid	yazid@gmail.com	081378979	TRPL	Jl. Sutomo no. 1 Padang	2023-04-02 15:38:07	2023-04-02 15:38:07
<input type="checkbox"/>	3	19828887678	Deni	deni@gmail.com	0811000001	Teknik Komputer	Jl. M Hatta no. 15 Padang	2023-04-02 15:38:07	2023-04-02 15:48:05

Query results operations: Print | Copy to clipboard | Export | Display chart | Create view

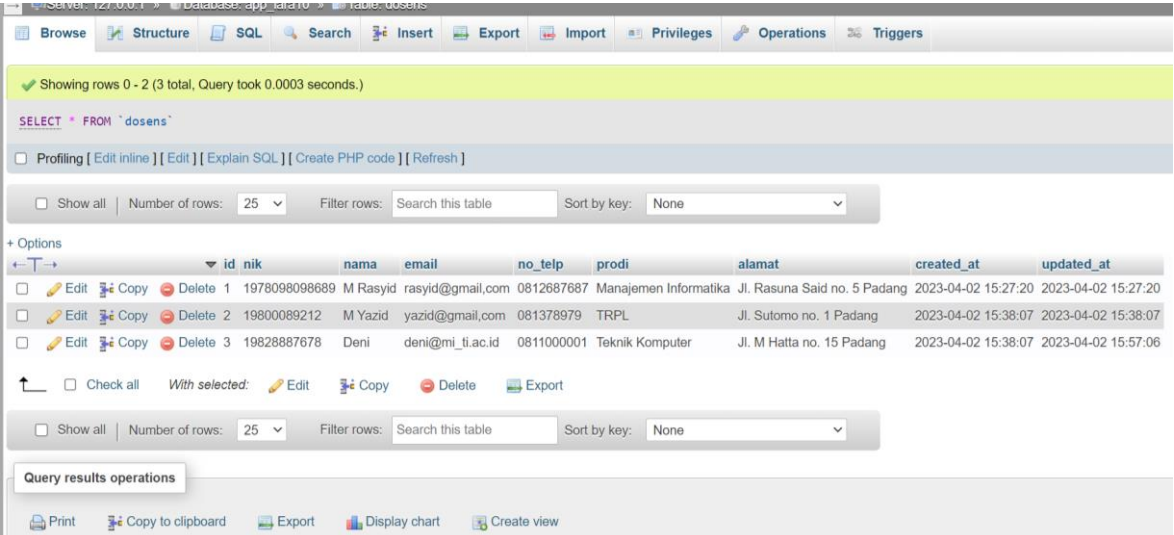
Prodi dan no_telp sudah berubah sesuai dengan perintah update yang dibuat.

5. Update Where

Jika kita butuh membuat 2 atau lebih kondisi update, misalnya untuk dosen dengan nama 'deni' dan prodi selain 'TRPL', maka bisa men-chaining method where() seperti contoh berikut:

```
65     }
66
67     public function updateWhereDosen()
68     {
69         $query=DB::table('dosens')
70             ->where('nama','Deni')
71             ->where('prodi','<>','TRPL')
72             ->update(
73             [
74                 'email'=>'deni@mi_ti.ac.id',
75                 'updated_at'=>now(),
76             ]
77             );
78         dd($query);
79     }
80 }
```

Jalankan method updateWhereDosen()



Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

SELECT * FROM 'dosens'

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	nik	nama	email	no_telp	prodi	alamat	created_at	updated_at
<input type="checkbox"/>	1	1978098098689	M Rasyid	rasyid@gmail.com	0812687687	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang	2023-04-02 15:27:20	2023-04-02 15:27:20
<input type="checkbox"/>	2	19800089212	M Yazid	yazid@gmail.com	081378979	TRPL	Jl. Sutomo no. 1 Padang	2023-04-02 15:38:07	2023-04-02 15:38:07
<input type="checkbox"/>	3	1982887678	Deni	deni@mi_ti.ac.id	0811000001	Teknik Komputer	Jl. M Hatta no. 15 Padang	2023-04-02 15:38:07	2023-04-02 15:57:06

Check all | With selected: Edit | Copy | Delete | Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

6. Update or Insert

Query builder juga menyediakan method lain bernama updateOrCreate(), yang merupakan gabungan dari proses **update** dan **insert**. Jika data yang ingin di update belum ada di tabel, maka jalankan proses insert. Berikut contoh penggunaannya:


```

80
81 public function updateOrInsert()
82 {
83     $query=DB::table('dosens')->updateOrInsert(
84         [
85             'nik'=>'1978092800001',
86         ],
87         [
88             'nama'=>'Steve Job',
89             'email'=>'steve@gmail.com',
90             'no_telp'=>'08128874444',
91             'prodi'=>'TRPL',
92             'alamat'=>'Jl. M Hatta no. 1 Padang',
93             'created_at'=>now(),
94             'updated_at'=>now(),
95         ]
96     );
97     dd($query);
98 }

```

Method `updateOrInsert()` butuh 2 buah argument. Argument pertama berupa associative array yang berisi kondisi yang dicari. Dalam contoh di atas, kita ingin mencari dosen dengan nik 1978092800001. Kemudian sebagai argument kedua adalah data yang akan di update, yang juga ditulis dalam bentuk associative array.

Jika dalam tabel terdapat dosen dengan nik 1978092800001, maka data dosen tersebut akan di update dengan isi dari argument kedua. Namun jika tidak ditemukan dosen dengan nik tersebut, maka lakukan proses insert.

Jalankan method `updateOrInsert()` ini dengan mengakses halaman `localhost:8000/update-or-insert`. Lihat hasilnya:

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

SELECT * FROM `dosens`

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	nik	nama	email	no_telp	prodi	alamat	created_at	updated_at
<input type="checkbox"/>	1	1978098098689	M Rasyid	rasyid@gmail.com	0812687687	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang	2023-04-02 15:27:20	2023-04-02 15:27:20
<input type="checkbox"/>	2	19800089212	M Yazid	yazid@gmail.com	081378979	TRPL	Jl. Sutomo no. 1 Padang	2023-04-02 15:38:07	2023-04-02 15:38:07
<input type="checkbox"/>	3	19828887678	Deni	deni@mi_ti.ac.id	0811000001	Teknik Komputer	Jl. M Hatta no. 15 Padang	2023-04-02 15:38:07	2023-04-02 15:57:06
<input type="checkbox"/>	4	1978092800001	Steve Job	steve@gmail.com	08128874444	TRPL	Jl. M Hatta no. 1 Padang	2023-04-02 16:04:29	2023-04-02 16:04:29

Number of rows: 25 Filter rows: Search this table Sort by key: None

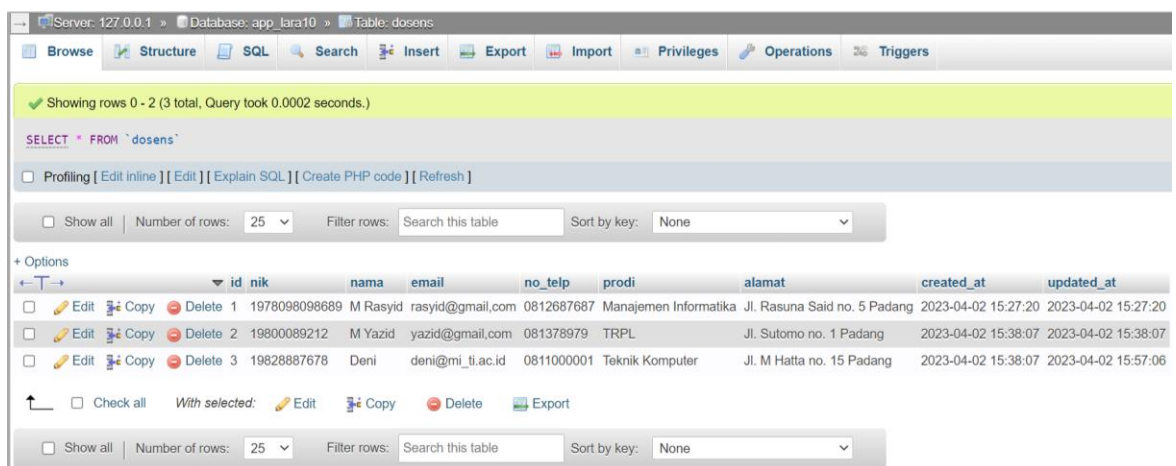
Karena dosen dengan nik 1978092800001 tidak ditemukan, maka data tersebut akan ditambah ke dalam tabel dosens.

7. Menghapus Data

Query builder menyediakan method `delete()` untuk menghapus data. Method ini tidak butuh argument apapun, dimana untuk menentukan kolom mana yang akan dihapus kita butuh method `where()`.

```
99 |
100 | public function deleteDosen()
101 | {
102 |     $query=DB::table('dosens')
103 |         ->where ('nik','1978092800001')
104 |         ->delete();
105 |     dd($query);
106 | }
107 |
108 |
```

Setelah dijalankan, data sekarant tinggal menjadi 3



The screenshot shows a database management interface for a table named 'dosens'. The table contains 3 rows of data. The columns are: id, nik, nama, email, no_telp, prodi, alamat, created_at, and updated_at. The data is as follows:

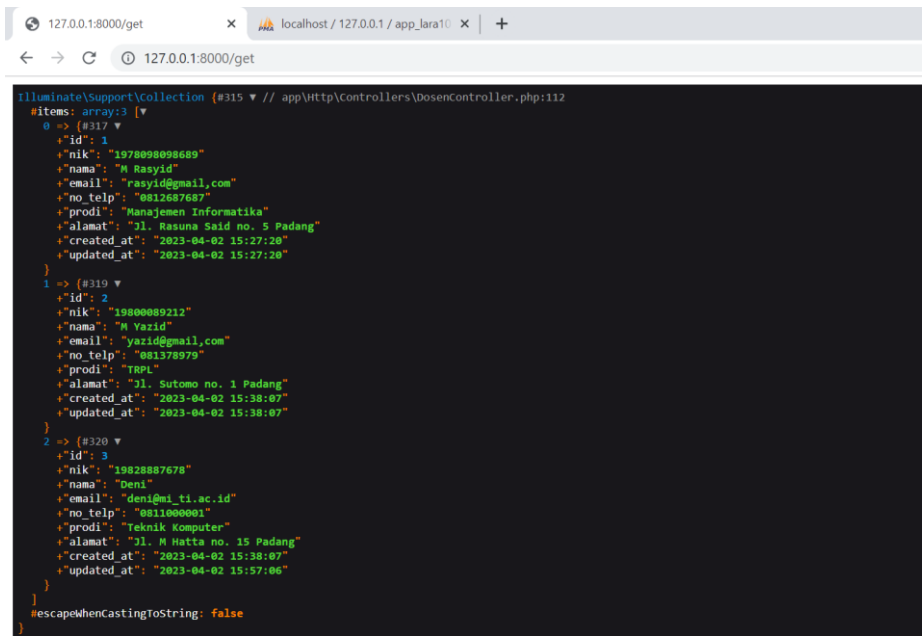
	id	nik	nama	email	no_telp	prodi	alamat	created_at	updated_at
<input type="checkbox"/>	1	1978098098689	M Rasyid	rasyid@gmail.com	0812687687	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang	2023-04-02 15:27:20	2023-04-02 15:27:20
<input type="checkbox"/>	2	19800089212	M Yazid	yazid@gmail.com	081378979	TRPL	Jl. Sutomo no. 1 Padang	2023-04-02 15:38:07	2023-04-02 15:38:07
<input type="checkbox"/>	3	19828887678	Deni	deni@mi_ti.ac.id	0811000001	Teknik Komputer	Jl. M Hatta no. 15 Padang	2023-04-02 15:38:07	2023-04-02 15:57:06

8. Menampilkan Data dengan get()

Pertama, untuk menampilkan semua data dari tabel dosen, bisa menggunakan method `get()` yang langsung diakses dari `DB::table('dosens')`:

```
102 |     $query=DB::table('dosens')
103 |         ->where ('nik','1978092800001')
104 |         ->delete();
105 |     dd($query);
106 | }
107 |
108 | public function get()
109 | {
110 |     $query=DB::table('dosens')->get();
111 |
112 |     dd($query);
113 | }
114 |
```

Hasilnya :

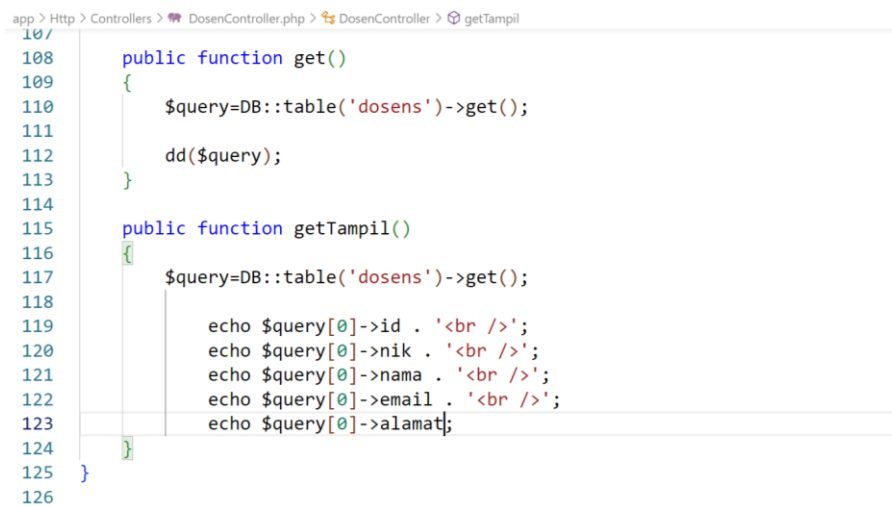


```
Illuminate\Support\Collection (#315 // app\Http\Controllers\DosenController.php:112)
#items: array:3 [▼]
  0 => (#317 ▼)
    +id: 1
    +nik: "19780900098689"
    +nama: "M Rasyid"
    +email: "rasyid@gmail.com"
    +no_telp: "0812687687"
    +prodi: "Manajemen Informatika"
    +alamat: "Jl. Rasuna Said no. 5 Padang"
    +created_at: "2023-04-02 15:27:20"
    +updated_at: "2023-04-02 15:27:20"
  1 => (#319 ▼)
    +id: 2
    +nik: "19800009212"
    +nama: "M Yazid"
    +email: "yazid@gmail.com"
    +no_telp: "081378979"
    +prodi: "TRPL"
    +alamat: "Jl. Sutomo no. 1 Padang"
    +created_at: "2023-04-02 15:38:07"
    +updated_at: "2023-04-02 15:38:07"
  2 => (#320 ▼)
    +id: 3
    +nik: "19828887678"
    +nama: "Deni"
    +email: "deni@mi_ti.ac.id"
    +no_telp: "0811000001"
    +prodi: "Teknik Komputer"
    +alamat: "Jl. M Hatta no. 15 Padang"
    +created_at: "2023-04-02 15:38:07"
    +updated_at: "2023-04-02 15:37:06"
  #escapeWhenCastingToString: false
}
```

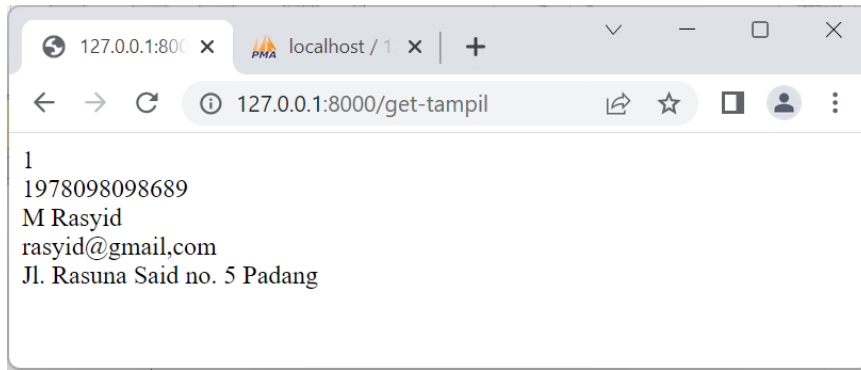
9. Menampilkan Data dari Object

Hasil dari method `get()` adalah sebuah **collection**. Setiap baris tabel dosens menjadi object dari array yang tersimpan dalam bentuk collection.

Untuk menampilkan isi baris pertama menggunakan perintah `echo` :



```
app > Http > Controllers > DosenController.php > DosenController > getTampil
107 /
108     public function get()
109     {
110         $query=DB::table('dosens')->get();
111
112         dd($query);
113     }
114
115     public function getTampil()
116     {
117         $query=DB::table('dosens')->get();
118
119         echo $query[0]->id . '<br />';
120         echo $query[0]->nik . '<br />';
121         echo $query[0]->nama . '<br />';
122         echo $query[0]->email . '<br />';
123         echo $query[0]->alamat;
124     }
125 }
126
```



10. Menampilkan Data dengan View

Untuk menampilkan data ke view, buat method getView() seperti di bawah ini:

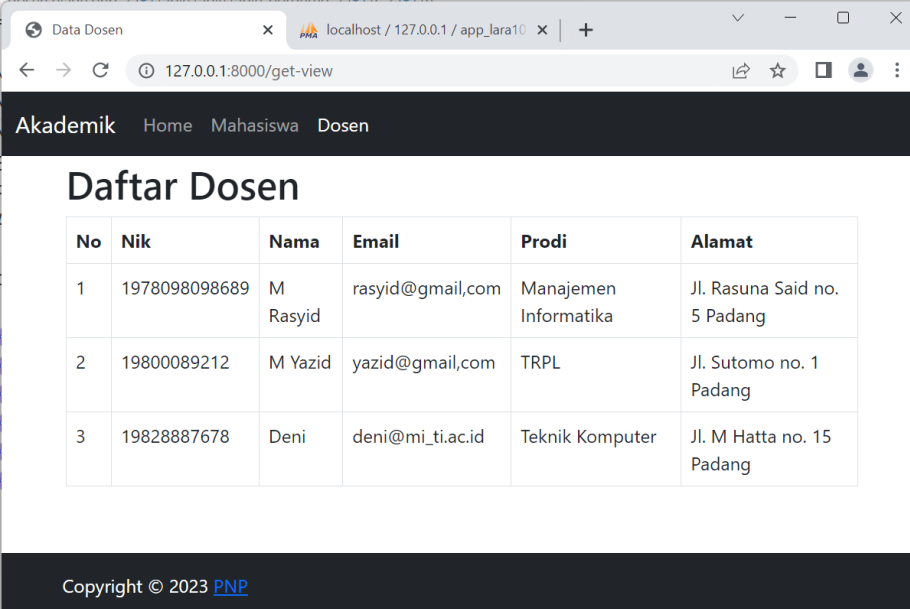
```
121 |         echo $query[0]->nama . '<br />';  
122 |         echo $query[0]->email . '<br />';  
123 |         echo $query[0]->alamat;  
124 |     }  
125 |  
126 |     public function getView()  
127 |     {  
128 |         $query=DB::table('dosens')->get();  
129 |         return view('akademik.dosen',['dosens'=>$query]);  
130 |     }  
131 | }  
132 |
```

Berikutnya, silahkan buka file dosen.blade.php di folder views\akademik.

Kemudian ubah dengan kode berikut:

```
resources > views > akademik > dosen.blade.php > table.table-bordered > tr > td
1 @extends('layouts.main')
2 @section('title','Data Dosen')
3 @section('navDosen','active')
4
5
6 @section('content')
7 <h1>Daftar Dosen </h1>
8 <table class="table table-bordered">
9     <tr>
10         <th>No</th>
11         <th>Nik</th>
12         <th>Nama</th>
13         <th>Email</th>
14         <th>Prodi</th>
15         <th>Alamat</th>
16     </tr>
17 @foreach ($dosens as $dosen)
18     <tr>
19         <td>{{ $dosen->id }}</td>
20         <td>{{ $dosen->nik }}</td>
21         <td>{{ $dosen->nama }}</td>
22         <td>{{ $dosen->email }}</td>
23         <td>{{ $dosen->prodi }}</td>
24         <td>{{ $dosen->alamat }}</td>
25     </tr>
26 @endforeach
27 </table>
28 @endsection
29
```

Hasilnya :



No	Nik	Nama	Email	Prodi	Alamat
1	1978098098689	M Rasyid	rasyid@gmail.com	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang
2	19800089212	M Yazid	yazid@gmail.com	TRPL	Jl. Sutomo no. 1 Padang
3	19828887678	Deni	deni@mi_ti.ac.id	Teknik Komputer	Jl. M Hatta no. 15 Padang

Copyright © 2023 [PNP](#)

11. Menampilkan Data dengan where () dan orderBy()

Query builder juga menyediakan berbagai method lain untuk proses pencarian yang lebih detail, misalnya digabung dengan method where() dan orderBy() seperti contoh berikut:

```

app > Http > Controllers > DosenController.php > DosenController > getView
124     }
125
126     public function getView()
127     {
128         $query=DB::table('dosens')->get();
129         return view('akademik.dosen',['dosens'=>$query]);
130     }
131
132     public function getWhere()
133     {
134         $query=DB::table('dosens')
135             ->where('prodi','TRPL')
136             ->orderBy('nama','desc')
137             ->get();
138         return view('akademik.dosen',['dosens'=>$query]);
139     }
140 }
141

```

Hasilnya :

No	Nik	Nama	Email	Prodi	Alamat
2	19800089212	M Yazid	yazid@gmail.com	TRPL	Jl. Sutomo no. 1 Padang

Copyright © 2023 [PNP](#)

12. Menampilkan Data dengan Method select()

Method lain yang tersedia adalah select(), yang bisa dipakai untuk membatasi nama kolom. Berikut contoh penggunaannya:

```

140
141     public function selectDosen()
142     {
143         $query=DB::table('dosens')
144             ->select('nik','nama as nama_dosen')
145             ->get();
146         dd($query);
147     }
148 }
149

```

```
127.0.0.1:8000/select-dosen x localhost / 127.0.0.1 / app_lara10 x +
127.0.0.1:8000/select-dosen
Illuminate\Support\Collection {#319 // app\Http\Controllers\DosenController.php:146
  #items: array:3 [▼
    0 => {#320 ▼
      +"nik": "1978098098689"
      +"nama_dosen": "M Rasyid"
    }
    1 => {#323 ▼
      +"nik": "19800089212"
      +"nama_dosen": "M Yazid"
    }
    2 => {#324 ▼
      +"nik": "19828887678"
      +"nama_dosen": "Deni"
    }
  ]
  #escapeWhenCastingToString: false
}
```

13. Menampilkan data dengan method skip() dan take()

Method pembatasan lain yang bisa kita pakai adalah skip() dan take(). Keduanya mirip seperti fungsi LIMIT di query MySQL, yakni untuk membatasi jumlah baris yang akan diambil.

Kedua method butuh sebuah argument berupa angka yang berisi jumlah baris. Method skip() dipakai untuk melompati baris, sedangkan take() untuk mengambil baris. Berikut contoh penggunaannya:

```
148
149     public function take()
150     {
151         $query=DB::table('dosens')
152             ->orderBy('nama','asc')
153             ->skip(1)
154             ->take(2)
155             ->get();
156         return view('akademik.dosen',['dosens'=>$query]);
157     }
158 }
159
```

Hasilnya :

Data Dosen x localhost / 127.0.0.1 / app_lara10 x +

127.0.0.1:8000/take

Akademik Home Mahasiswa Dosen

Daftar Dosen

No	Nik	Nama	Email	Prodi	Alamat
1	1978098098689	M Rasyid	rasyid@gmail.com	Manajemen Informatika	Jl. Rasuna Said no. 5 Padang
2	19800089212	M Yazid	yazid@gmail.com	TRPL	Jl. Sutomo no. 1 Padang

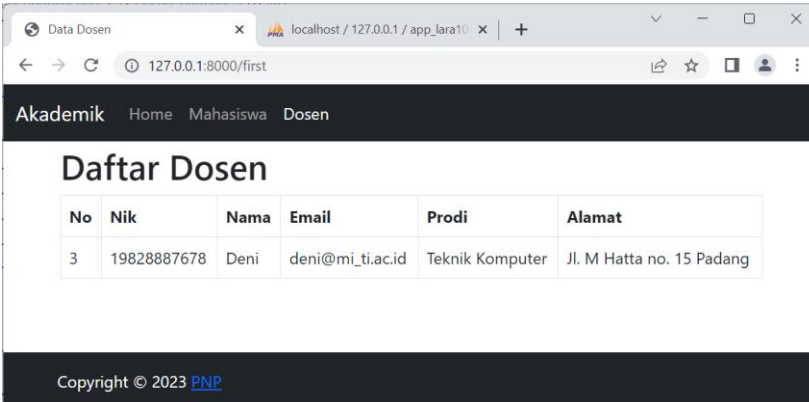
Copyright © 2023 [PNP](#)

14. Menampilkan data dengan method first()

Selanjutnya terdapat juga method first() yang bisa dipakai untuk mengambil 1 baris data pertama saja. Ini biasa digabung dengan kondisi where() seperti contoh berikut:

```
153         ->skip(1)
154         ->take(2)
155         ->get();
156         return view('akademik.dosen', ['dosens'=>$query]);
157     }
158
159     public function first()
160     {
161         $query=DB::table('dosens')
162             ->where('nama', 'Deni')->first();
163         return view('akademik.dosen', ['dosens'=>[$query]]);
164     }
165 }
```

Hasilnya :



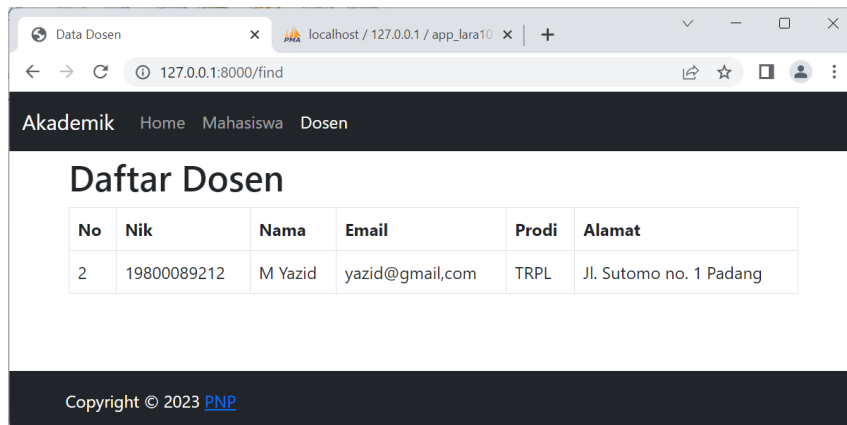
No	Nik	Nama	Email	Prodi	Alamat
3	19828887678	Deni	deni@mi.ti.ac.id	Teknik Komputer	Jl. M Hatta no. 15 Padang

15. Menampilkan Data menggunakan method find()

Method find() bisa dipakai secara cara singkat untuk mengambil 1 data tabel yang memiliki id tertentu. Id tersebut diinput sebagai argument ke dalam method find(). Berikut contoh penggunaannya:

```
165
166     public function find()
167     {
168         $query=DB::table('dosens')
169             ->find(2);
170         return view('akademik.dosen', ['dosens'=>[$query]]);
171     }
172 }
173
```

Hasilnya :



16. Menampilkan Data menggunakan method raw()

Method `selectRaw()` bisa dipakai untuk menjalankan query select dalam bentuk perintah raw SQL. Berikut contoh penggunaannya:

```
172 |  
173 |  
174 | public function raw()  
175 | {  
176 |     $query=DB::table('dosens')  
177 |     ->selectRaw('count(*) as total_dosen')  
178 |     ->get();  
179 |  
180 |     echo $query[0]->total_dosen;  
181 | }
```

Di sini kita menjalankan query `count(*) as total_dosen` untuk menghitung jumlah baris yang terdapat di dalam tabel `dosens`

