

LANJUTAN LARAVEL

Muhammad Hasbi Habibi
Catur Satria Adhi Fauzi

Materi!

Model (Eloquent)

Model adalah salah satu struktur dari MVC yang menghubungkan Controller dengan database

Command Create Model

```
Hasbi@DESKTOP-RDVG3LN MINGW64 /c:/laragon/www/test-app  
$ php artisan make:model Item  
Model created successfully.
```

Migration

Migration adalah fitur yang tersedia di Laravel untuk mengontrol database dan memungkinkan sesama developer dapat berbagi database yang sama. Tidak hanya saling berbagi database migration memungkinkan developer dapat mengupdate atau bahkan menghapus field, table atau element lain dari database dan developer lain dapat update terbaru dari database

Command Generate Migration

```
php artisan make:migration create_users_table
```

```
php artisan make:migration create_users_table --create=users
```

```
php artisan make:migration add_votes_to_users_table --table=users
```

Command Run Migration

```
php artisan migrate
```

Seeder

Seeder adalah salah satu fitur di Laravel yang dapat membantu developer untuk mengisi database dengan data dummy ketika developer tidak memiliki data yang real.

Command Generate Seeder

```
php artisan make:seeder UsersTableSeeder
```

Command Run Seeder

```
php artisan db:seed
```

```
php artisan db:seed --class=UsersTableSeeder
```

Validation Request

Laravel memiliki beberapa cara untuk melakukan validasi data yang masuk. Secara default controller menggunakan class `ValidatesRequests` yang menyediakan validasi request dan aturan validasi lainnya.

```
public function store(Request $request)
{
    $validatedData = $request->validate([
        'title' => 'required|unique:posts|max:255',
        'body' => 'required',
    ]);

    // The blog post is valid...
}
```

```
public function store(StoreBlogPost $request)
{
    // The incoming request is valid...

    // Retrieve the validated input data...
    $validated = $request->validated();
}
```

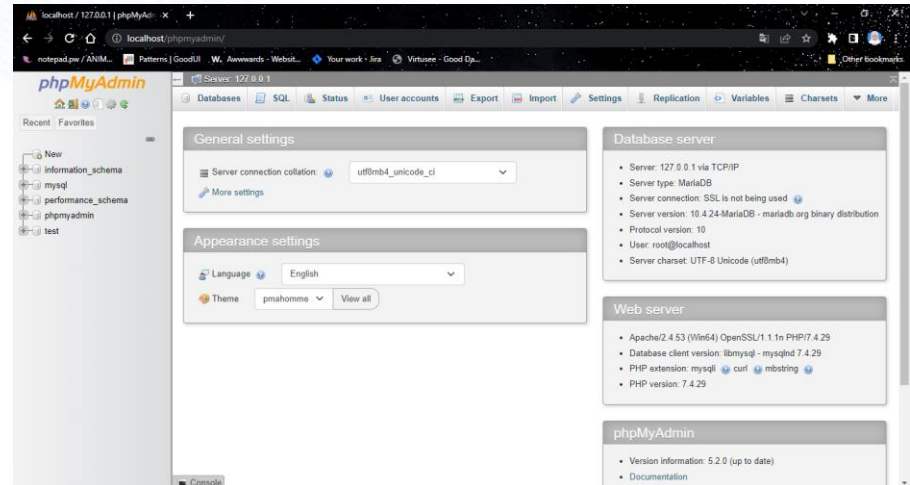
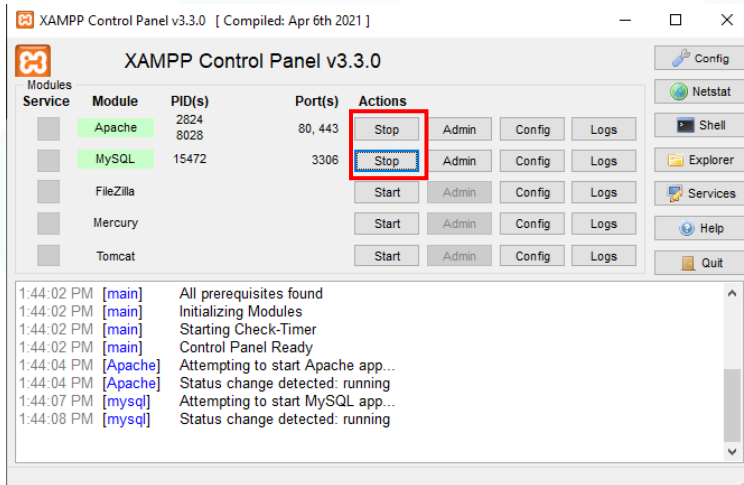
Command Generate Request

```
php artisan make:request StoreBlogPost
```

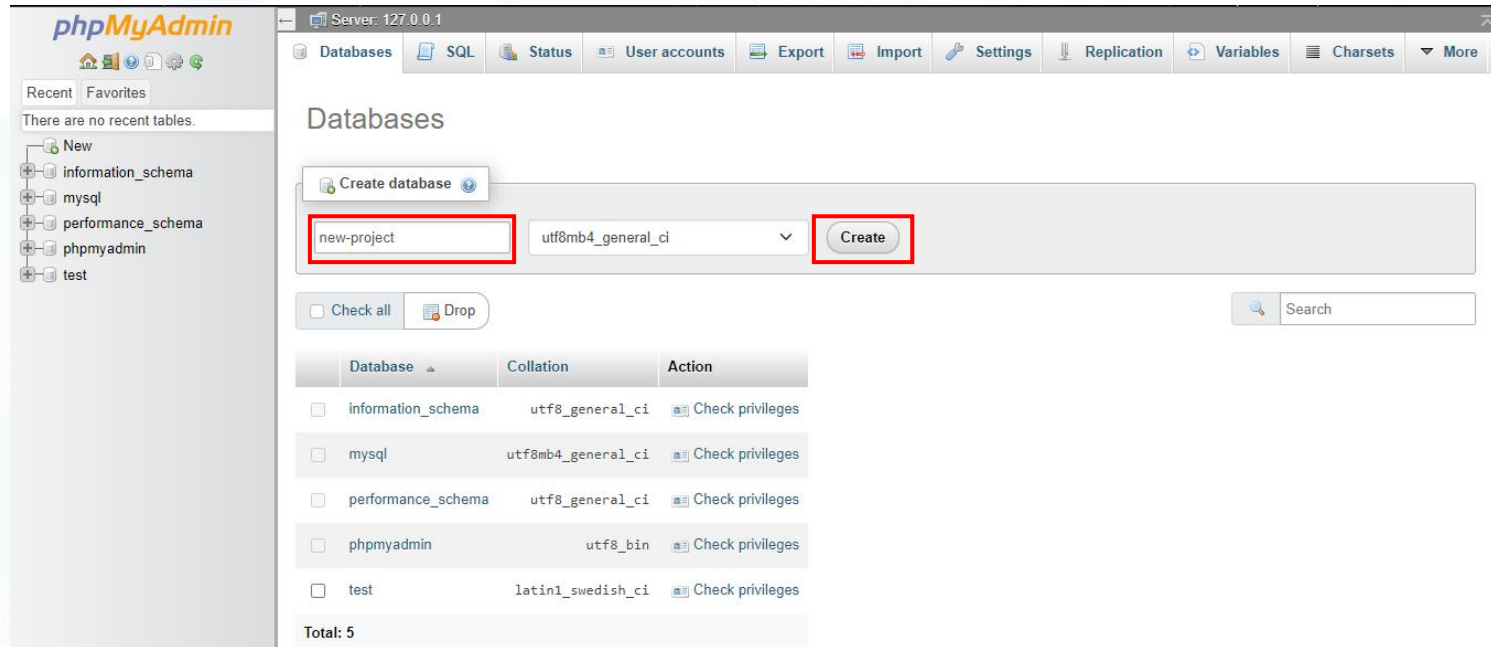
Praktek **Kuy!**

Model

Membuka XAMPP lalu tekan tombol **Start** pada **Apache** dan **MySQL**.
Kemudian membuka URL *localhost/phpmyadmin* untuk mulai membuat basis data melalui localhost.



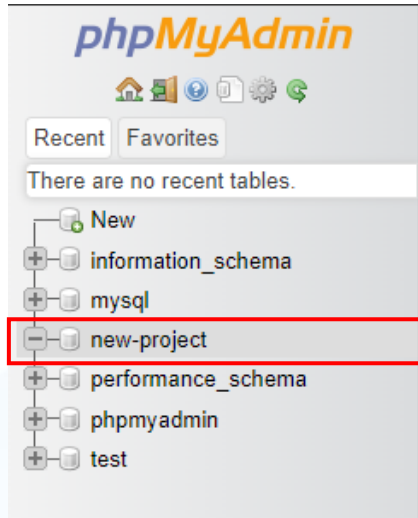
Klik tombol **Database** pada menubar. Halaman akan menampilkan form untuk *create database*. Kemudian masukkan nama database yang akan dibuat misalnya **new-project**, klik tombol **create**.



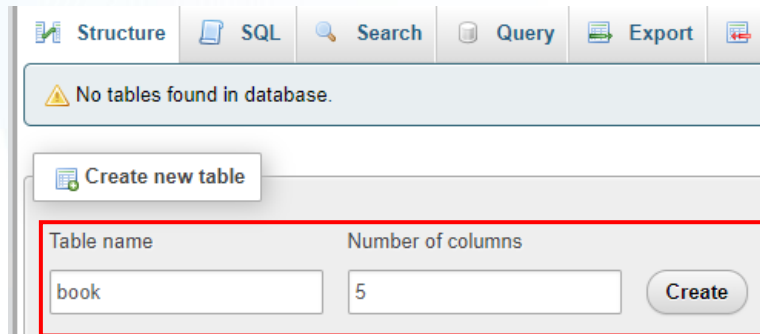
The screenshot shows the phpMyAdmin interface. The top navigation bar includes links for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, Charsets, and More. The left sidebar shows a tree view of databases: information_schema, mysql, performance_schema, phpmyadmin, and test. The main area displays the 'Databases' page. A 'Create database' form is highlighted with a red box, showing 'new-project' in the name field and 'utf8mb4_general_ci' in the collation dropdown. The 'Create' button is also highlighted with a red box. Below the form, there are checkboxes for 'Check all' and 'Drop', and a search bar. A table lists existing databases with their collations and actions.

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> mysql	utf8mb4_general_ci	Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	Check privileges
<input type="checkbox"/> test	latin1_swedish_ci	Check privileges

Total: 5



Database yang telah dibuat akan muncul pada sidebar. Klik database yang telah dibuat pada sidebar.



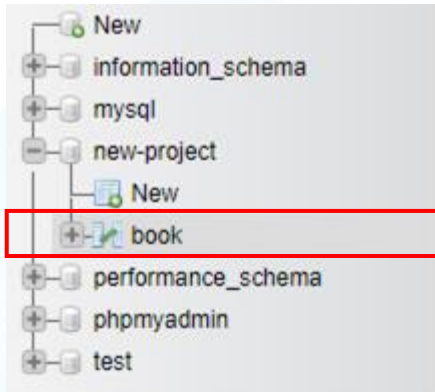
Pada menu **Structure** database tersebut, isikan form pembuatan tabel seperti pada gambar kemudian klik tombol **Create**.

Isikan form untuk membuat struktur tabel tadi seperti pada gambar. Kemudian klik tombol **save**. Untuk selengkapnya mengenai database nanti akan dijelaskan pada pertemuan materi **Basis Data**.

Table name: Add column(s)

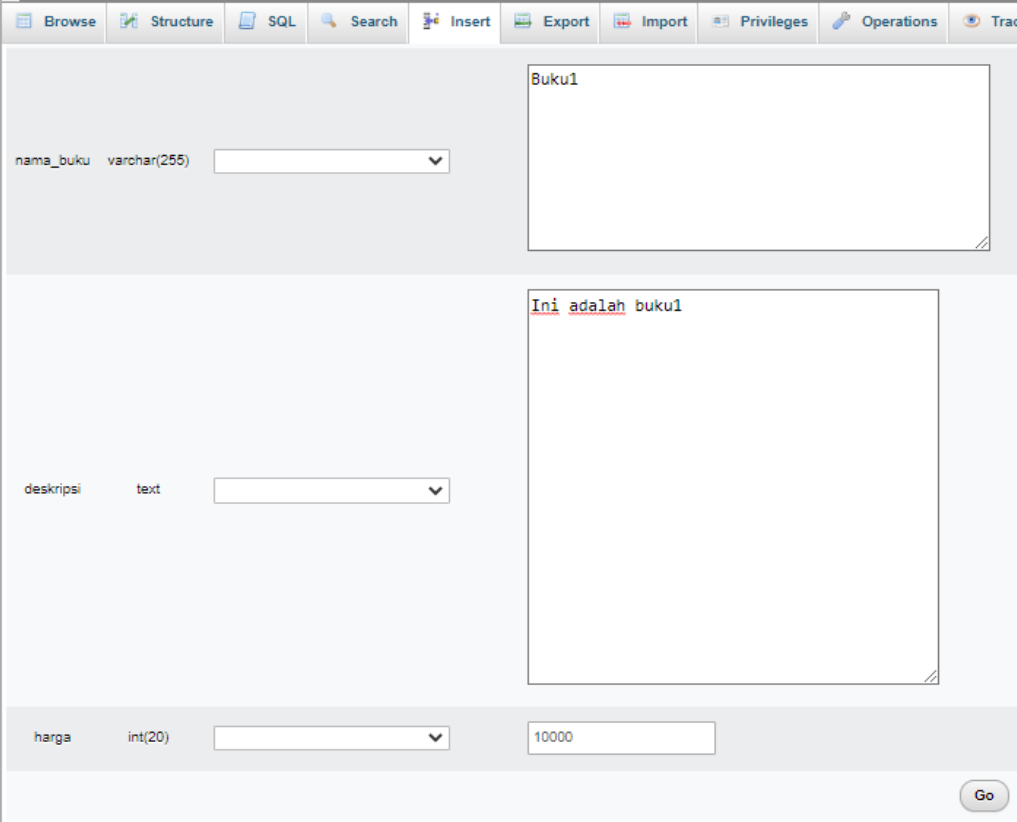
Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	
<input type="text" value="id"/> <small>Pick from Central Columns</small>	INT	10	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
<input type="text" value="nama_buku"/> <small>Pick from Central Columns</small>	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
<input type="text" value="deskripsi"/> <small>Pick from Central Columns</small>	TEXT		None			<input type="checkbox"/>	---	<input type="checkbox"/>
<input type="text" value="harga"/> <small>Pick from Central Columns</small>	INT	20	None			<input type="checkbox"/>	---	<input type="checkbox"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>



Tabel yang telah dibuat akan muncul pada sidebar.

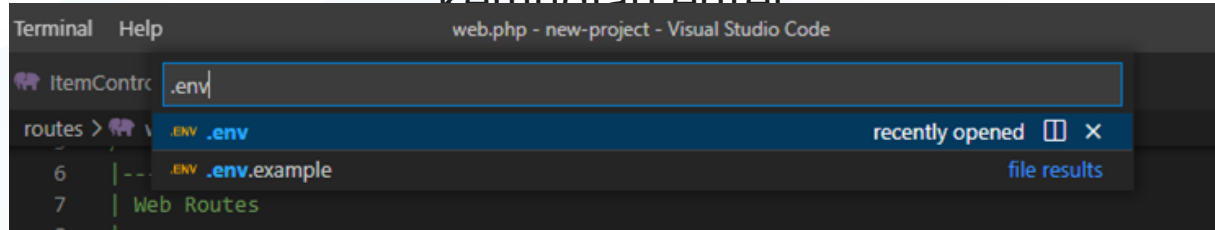
Menambahkan satu data pada tabel **book** database yang telah dibuat dengan mengisi form pada menu **insert**. Sebagai contoh pada gambar. Kemudian



The screenshot shows the MySQL Insert form for the 'book' table. The form has three main sections: 'nama_buku' (varchar(255)), 'deskripsi' (text), and 'harga' (int(20)). The 'nama_buku' field is currently empty. The 'deskripsi' field contains the text 'Ini adalah buku1'. The 'harga' field is currently empty. The 'Go' button is visible at the bottom right.

Field	Type	Value
nama_buku	varchar(255)	
deskripsi	text	Ini adalah buku1
harga	int(20)	10000

Untuk menghubungkan antara database dengan Laravel. Pada VS Code project Laravel tadi buka file **.env** dengan Ctrl + v ketikkan **.env** kemudian enter



Mengubah **DB_DATABASE** dengan nama database yang telah dibuat sebelumnya yaitu **new-project**

```
.env .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:CAU3aqFfbXZpgW6Mion0n0Bsyix6v5G0buN7gKcas40=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=new-project
15 DB_USERNAME=root
16 DB_PASSWORD=
```

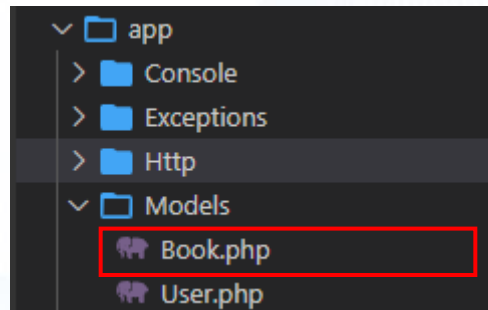
Untuk memberhentikan perintah yang masih jalan sebelumnya menggunakan **Ctrl + c** pada terminal, kemudian untuk membuat model melalui terminal dengan memasukkan perintah

php artisan make:model *nama-model*

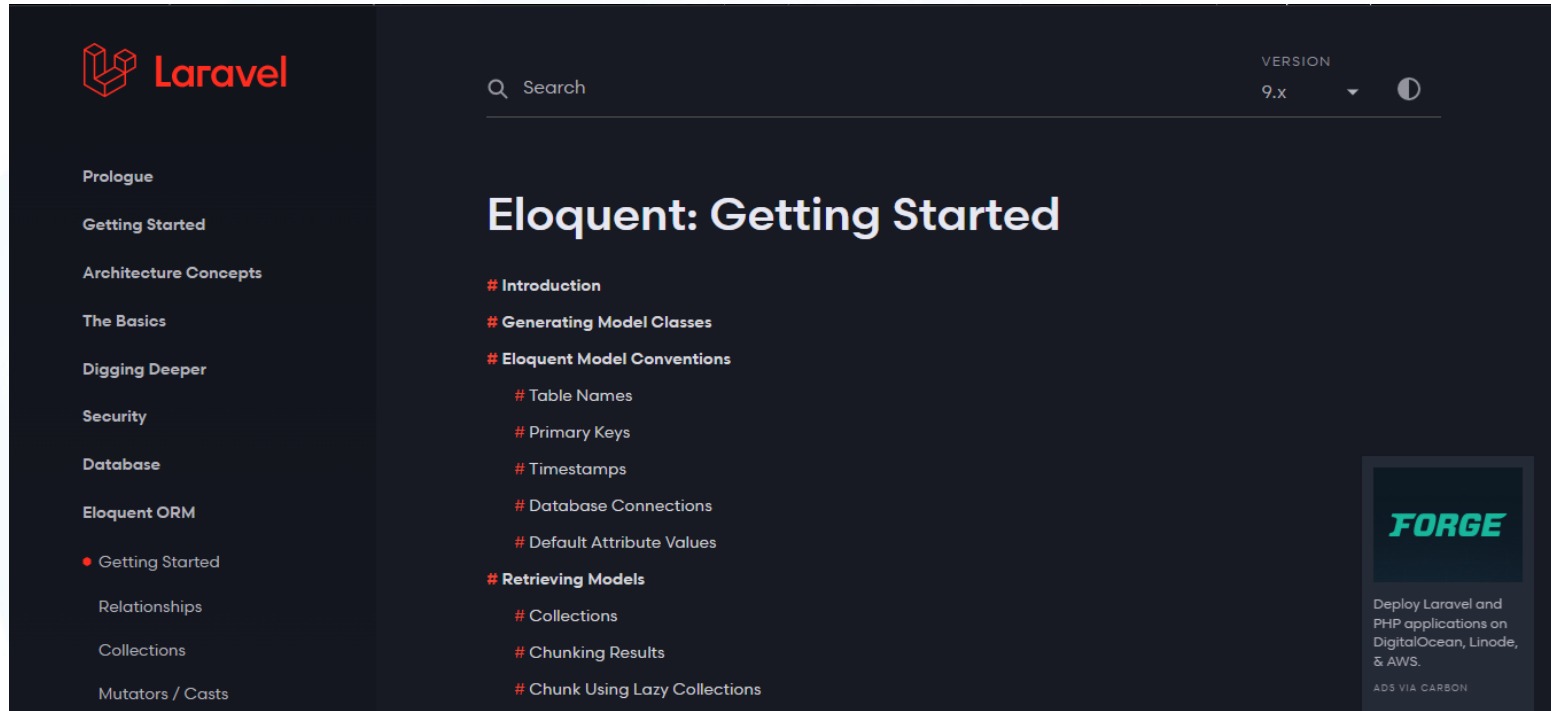
sebagai contoh untuk membuat model dengan nama **Book**

```
PS C:\xampp\htdocs\new-project> php artisan make:model Book
Model created successfully.
```

Model yang telah dibuat akan membuat file baru pada direktori **app/Http/Models** dengan nama file **Books.php**



Untuk melihat fungsi apa saja yang bisa dilakukan pada model dapat dilihat pada <https://laravel.com/docs/9.x/eloquent>



The screenshot shows the Laravel documentation website. The top navigation bar includes the Laravel logo, a search bar, and a version selector set to '9.x'. The left sidebar contains a table of contents with 'Getting Started' highlighted. The main content area is titled 'Eloquent: Getting Started' and lists several topics: Introduction, Generating Model Classes, Eloquent Model Conventions (with sub-points: Table Names, Primary Keys, Timestamps, Database Connections, Default Attribute Values), Retrieving Models (with sub-points: Collections, Chunking Results, Chunk Using Lazy Collections), and a Forge advertisement for deploying applications on DigitalOcean, Linode, and AWS.

Laravel

VERSION 9.x

Prologue

Getting Started

Architecture Concepts

The Basics

Digging Deeper

Security

Database

Eloquent ORM

- Getting Started
- Relationships
- Collections
- Mutators / Casts

Eloquent: Getting Started

- # Introduction
- # Generating Model Classes
- # Eloquent Model Conventions
 - # Table Names
 - # Primary Keys
 - # Timestamps
 - # Database Connections
 - # Default Attribute Values
- # Retrieving Models
 - # Collections
 - # Chunking Results
 - # Chunk Using Lazy Collections

FORGE

Deploy Laravel and PHP applications on DigitalOcean, Linode, & AWS.

ADS VIA CARBON

Menambahkan function model sesuai dengan tabel pada database yang telah dibuat seperti pada gambar berikut.

```
8 class Book extends Model
9 {
10     use HasFactory;
11
12     protected $table = "book";
13
14     protected $primaryKey = "id";
15
16     public $timestamps = false;
17
18     protected $fillable = [
19         'nama_buku',
20         'deskripsi',
21         'harga',
22     ];
23 }
24
```

Nama

tabel

Primary

key

<https://laravel.com/docs/9.x/eloquent#timestamps>

Kolom
tabel

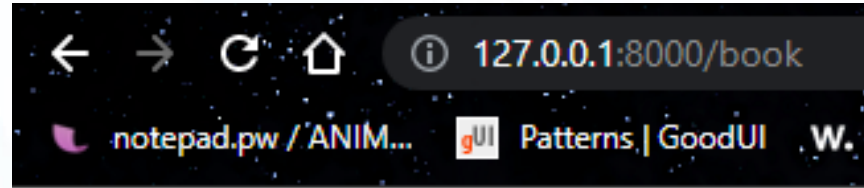
Mengambil setiap data dari database yang telah dan memunculkan pada halaman URL dibuat menggunakan controller. Menambahkan sintaks pada fungsi **index** seperti pada gambar.

```
public function index()
{
    $bookModel = new Book();

    $books = $bookModel->get();

    foreach($books as $book){
        echo "nama buku : " . $book->nama_buku . '<br>';
        echo "deskripsi buku : " . $book->deskripsi . '<br>';
        echo "harga buku : " . $book->harga . '<br>';
    }
}
```

Setelah perubahan disimpan, muat ulang pada browser. Browser akan menampilkan halaman seperti berikut.



nama buku : Buku1
deskripsi buku : Ini adalah buku1
harga buku : 10000

Migration

Menghapus tabel **Item** telah ada sebelumnya karena selanjutnya akan mencoba membuat tabel Item menggunakan migration. Pilih tabel Item pada database **new-project** di <http://localhost/phpmyadmin/>. Kemudian klik **Drop** untuk menghapus tabel **Item**. Kemudian klik **OK** saat konfirmasi penghapusan tabel.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> book	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> item	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
2 tables	Sum	6	InnoDB	utf8mb4_general_ci	32.0 KiB	0 B

Confirm

You are about to DESTROY a complete table! Do you really want to execute "DROP TABLE `item`"?

☒ Enable foreign key checks

OK

Cancel

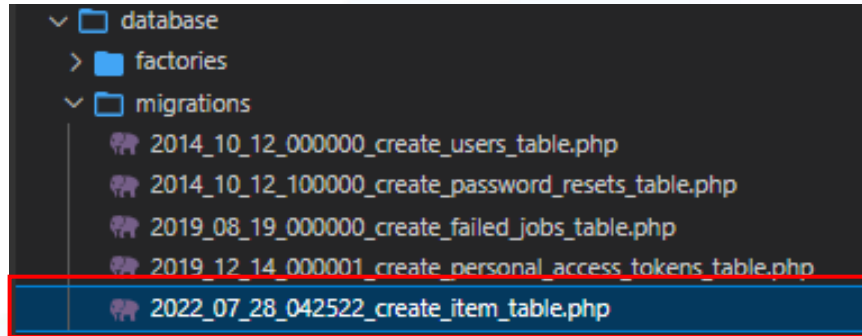
Untuk membuat migration dengan memasukkan perintah

php artisan make:migration *nama_migrasi*

Sebagai contoh untuk membuat migrasi pembuatan tabel **Item**,
masukkan perintah pada terminal

php artisan make:migration *create_item_table*

```
PS C:\xampp\htdocs\new-project> php artisan make:migration create_item_table  
Created Migration: 2022_07_28_042522_create_item_table
```



Setelah migration berhasil terbentuk, pada direktori **database/migration** akan terbentuk file migration baru.

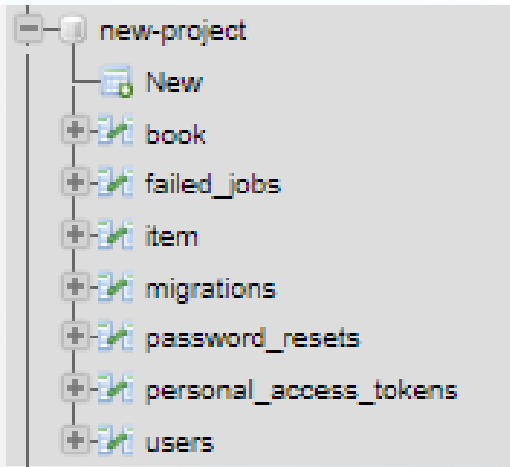
```
14 public function up()
15 {
16     Schema::create('item', function (Blueprint $table) {
17         $table->id();
18         $table->string('item_name');
19         $table->string('item_type');
20         $table->integer('item_price');
21         $table->longText('item_desc');
22         $table->timestamps();
23     });
24 }
```

Pada file tersebut, menambahkan sintaks untuk membentuk kolom-kolom pada tabel **Item**.


```
Migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table (86.59ms)  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table (49.74ms)  
Migrating: 2019_08_19_000000_create_failed_jobs_table  
Migrated: 2019_08_19_000000_create_failed_jobs_table (38.65ms)  
Migrating: 2019_12_14_000001_create_personal_access_tokens_table  
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (104.88ms)  
Migrating: 2022_07_28_042522_create_item_table  
Migrated: 2022_07_28_042522_create_item_table (32.13ms)
```

Untuk membuat menjalankan migration dengan memasukkan perintah

php artisan migrate



Pada database yang ada di localhost akan terbentuk tabel-tabel baru sesuai dengan file migrasi yang ada pada project.

Seeder

Untuk membuat seeder dengan memasukkan perintah

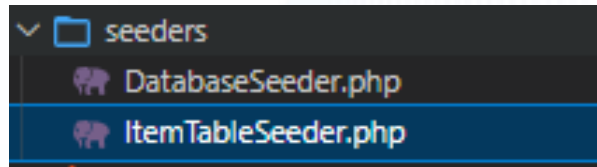
php artisan make:seeder *nama_seeder*

Sebagai contoh untuk membuat seeder untuk tabel **Item**, masukkan perintah pada terminal

php artisan make:seeder *ItemTableSeeder*

```
PS C:\xampp\htdocs\new-project> php artisan make:seeder ItemTableSeeder  
Seeder created successfully.
```

Setelah itu, pada direktori **database/seeder** akan terbentuk file php baru dengan nama **ItemTableSeeder.php**



```

1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Seeder;
6  use Carbon\Carbon;
7  use Illuminate\Support\Facades\DB;
8  use Illuminate\Support\Str;
9
10 class ItemTableSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      *
15      * @return void
16      */
17     public function run()
18     {
19         $jml = 100;
20         for($a=0; $a<$jml; $a++){
21             DB::table('item')->insert([
22                 'item_name'=>Str::random(25),
23                 'item_type'=>Str::random(10),
24                 'item_price'=>rand(10000,100000),
25                 'item_desc'=>Str::random(50),
26                 'created_at'=>Carbon::now(),
27                 'updated_at'=>Carbon::now(),
28             ]);
29         }
30     }
31 }

```

Pada file tersebut, menambahkan sintaks untuk menentukan tipe data seeder yang akan dibentuk pada kolom-kolom tabel **Item**.

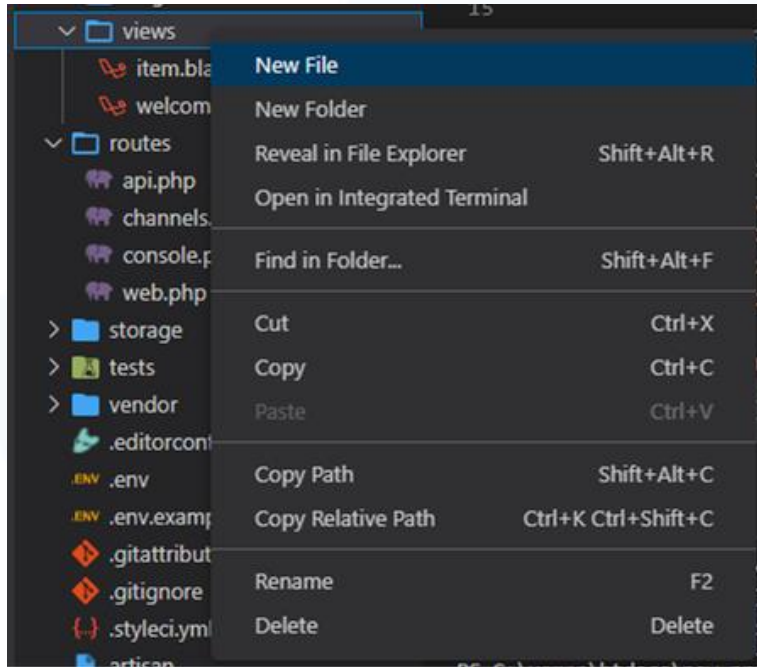
Untuk menjalankan pembuatan seeder dengan memasukkan perintah

```
PS C:\xampp\htdocs\new-project> php artisan db:seed --class=ItemTableSeeder
Database seeding completed successfully.
```

Setelah itu, lihat data tabel **Item** pada **localhost/phpMyAdmin**. Tabel tersebut sudah terdapat data dummy yang dibuat melalui

		id	item_name	item_type	item_price	item_desc	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	opiTv97e9AI03enN8BSj7MnMT	WNSt61Zz5n	55290	WpEhU1F7ZzcmRkdwJFP45804ZKbQN8mihZg2QtFHX0ug7s78J	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	2	enTJl3fqCjlyOKDCJYajYGsRn	TVnVBAJn3d	84579	7wQjPb6TH7ivLFXG1l2q5I2uNFFLJZQHLojbxzDeBBvLoJ6H	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	3	RcY16CYmlolbXcKsDtsTEYtQ	ik1hbZZVWq	62055	srFePU8aOUxi5sjLcWZu4dAwUbaFXTFoge18y7TD2OvdOvEspb	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	4	mD5xDORu74g6q9Xr4LluCYR	OMEP4x5Osl	69117	lqX4dlxG4X64wtVXsVXi5SYW0tvixo0DzdTEBZia440wFWk18	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	5	hLkGTKYEQGoSDAz3RjDTBpCI	uWfYXUxuj	71846	82yXkSWLnIE329NPEdijPgQnQvVWxb95yoIWoweCRb7U9wRpk	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	6	i101v8QpPVdixn2NoRi4mpY06	frCJqwc0V	58928	wGDPpMCqXk8wF56c49PoCbmMMvrXZ841KEXJnsWTTqU1jW1L	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	7	4Z2UpQbfbTvRs9NP8Zu9jQk	OXqFwPwiea	69153	qvPmA3MBCqj41MbtOkMv08PjKxCNPQIRngJf2XZ9teGRStEO7m	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	8	O2XgeDMFGdCE9eTJ8DqKq1Xy	o4uqJk0eNX	58884	2CZhOylfRqTLwDoGrcx1qoVhRV9jGBHH8blbJNO0dwaLDEzk4A	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	9	OkgybKmlMmAs5AkYCC7ep3VR1	ppIJ15ELg	74764	e0qeSPoEFYs0sAP4BIHy16PZgLKant7geK76qTfvv1UpHeatro	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	10	OHkRI9ot7H0nWum8rz43hGhow	IF0Vp7Qg29	63932	Q2gM1FInozrtqAOhgH7BcLtwfIRgbwVvjsvvhvPNhF8siecUbl	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	11	3phgqjJGZrJdEg5PDpk0m6C	nKytMthy2S	55090	MheGebLKtU5x7aODzHCX5hOdOLRbGd6avenqyJEMA3IXFyTJj	2022-07-28 06:44:06	2022-07-28 06:44:06
<input type="checkbox"/>	Edit Copy Delete	12	KxuuZOdBRa5ebwEubpfO1Frb	8BQtgwcA3k	23757	phW4JHBPfVRwzvfYRpAHLyR31Draa1HT1PwgSndBp5bQzsV	2022-07-28 06:44:06	2022-07-28 06:44:06

Validation Request



Klik kanan pada folder **resource\view** kemudian pilih **New File** untuk menambahkan file view & blade pada folder **resource\view** dengan nama file **insertItem.blade.php**

Menambahkan sintaks pada **insertItem.blade.php** untuk membuat form menambahkan data ke tabel **Item** pada

```
1 <!DOCTYPE html>
2 <html lang="{{str_replace('_', '-', app()->getLocale())}}">
3     <head>
4         <meta charset="utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6
7         <title>Laravel</title>
8         <style>
9             body {
10                 font-family: 'Nunito', sans-serif;
11             }
12         </style>
13     </head>
14     <body class="antialiased">
15         <form action="{{url('item/post')}}" method="post">
16             @csrf
17             <table style="border: solid 1px; border-color: #000">
18                 <tr>
19                     <td>Nama Item</td>
20                     <td><input type="text" name="item_name"></td>
21                 </tr>
22                 <tr>
23                     <td>Tipe Item</td>
24                     <td><input type="text" name="item_type"></td>
25                 </tr>
```


Menambahkan sintaks pada **insertItem.blade.php** untuk membuat form menambahkan data ke tabel **Item** pada

```
24         <td><input type="text" name="item_type"></td>
25     </tr>
26     <tr>
27         <td>Harga Item</td>
28         <td><input type="text" name="item_price"></td>
29     </tr><tr>
30         <td>Deskripsi Item</td>
31         <td><input type="text" name="item_desc"></td>
32     </tr>
33     <tr>
34         <td><button type="submit"> Submit </button></td>
35     </tr>
36 </table>
37 @if ($errors->any())
38 <div class="alert alert-danger">
39     <ul>
40         @foreach ($errors->all() as $error)
41             <li>{{$error}}</li>
42         @endforeach
43     </ul>
44 </div>
45 @endif
46 </form>
47
48 </body>
49 </html>
```

Untuk menjalankan pembuatan request dengan memasukkan perintah

php artisan make:request *nama-request*

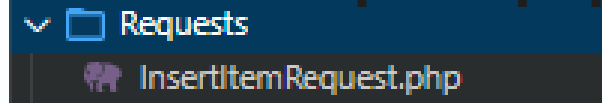
Sebagai contoh membuat request untuk menambahkan data baru

php artisan make:request InsertItemRequest

```
PS C:\xampp\htdocs\new-project> php artisan make:request InsertItemRequest  
Request created successfully.
```

Setelah request berhasil terbentuk, pada direktori **Requests** akan terbentuk file request baru bernama

InsertItemRequest.php



```

1  <?php
2
3  namespace App\Http\Requests;
4
5  use Illuminate\Foundation\Http\FormRequest;
6
7  class InsertItemRequest extends FormRequest
8  {
9
10     /**
11      * Get the validation rules that apply to the request.
12      *
13      * @return array
14      */
15     public function rules()
16     {
17         return [
18             'item_name'=>'required|max:15',
19             'item_type'=>'required|max:10',
20             'item_price'=>'required|numeric',
21             'item_desc'=>'required',
22         ];
23     }
24 }
25
26

```

Pada file tersebut, menambahkan sintaks untuk membentuk field yang sesuai dengan kolom tabel database yang akan ditambahkan datanya.

Menambahkan route menuju form penambahan item dan route menuju tampilan table setelah data baru ditambahkan pada file **web.php**

```
Route::get('/item/create', 'App\Http\Controllers\ItemController@insertPage');  
Route::post('/item/post', 'App\Http\Controllers\ItemController@insert');
```

Kemudian tambahkan fungsi pada **ItemController.php** untuk menambahkan data pada table **Item**.

```
18  public function insertPage(){  
19      return view('insertItem');  
20  }  
21  
22  public function insert(InsertItemRequest $request){  
23      $post = $request->all();  
24      $insertItem = [  
25          'item_name'=>$post['item_name'],  
26          'item_type'=>$post['item_type'],  
27          'item_price'=>$post['item_price'],  
28          'item_desc'=>$post['item_desc'],  
29      ];  
30  
31      $item = Item::create($insertItem);  
32      $item->save();  
33  
34      return redirect()->route('item');  
35  }
```

Menjalankan Laravel dengan memasukkan perintah

php artisan serve

Buka browser kemudian masukan URL

http://127.0.0.1:8000/item/create , maka browser akan menampilkan tampilan form untuk menambahkan data baru seperti sebagai berikut




Nama Item	<input type="text"/>
Tipe Item	<input type="text"/>
Harga Item	<input type="text"/>
Deskripsi Item	<input type="text"/>
<input type="submit" value="Submit"/>	

Isikan form kemudian klik tombol **Submit**.

Nama Item	Buku anak SD
Tipe Item	Buku
Harga Item	10000
Deskripsi Item	Ini adalah buku sekolah anak
<input type="button" value="Submit"/>	

Setelah menambahkan data maka system akan menampilkan halaman table dan data yang baru saja ditambahkan telah ada pada table database serta pada halaman URL.

100	7HAWRLhYp9o9LBg4hUSiEsfUt	DfWcQF8oQn	23996	8CLeGr9ZBQfkU4ZjWesFq7hyHI4f700U9wCmjIVCidvceYS5Vc
101	Buku anak SD	Buku	10000	Ini adalah buku sekolah anak sd

<input type="checkbox"/>	 Edit	 Copy	 Delete	101	Buku anak SD	Buku	10000	Ini adalah buku sekolah anak sd	2022-07-28 09:14:19	2022-07-28 09:14:19
--------------------------	--	--	--	-----	--------------	------	-------	---------------------------------	---------------------	---------------------

Terima **Kasih**