

# 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();
}
```

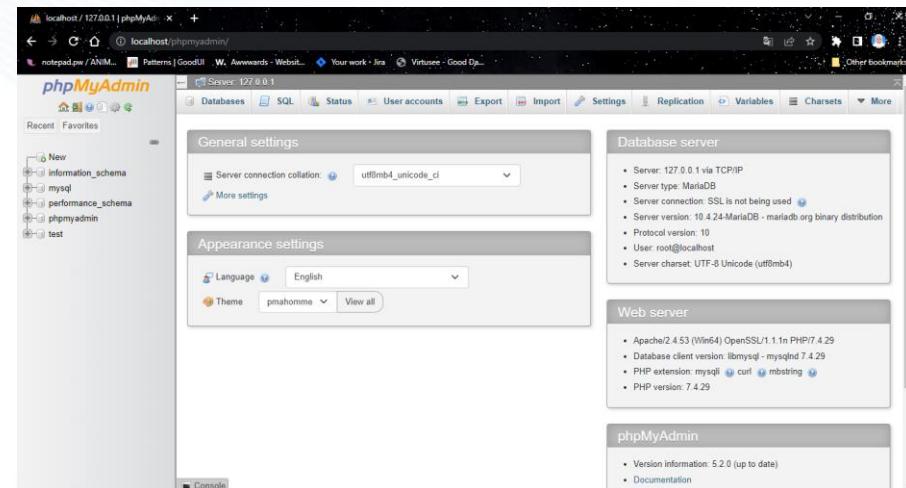
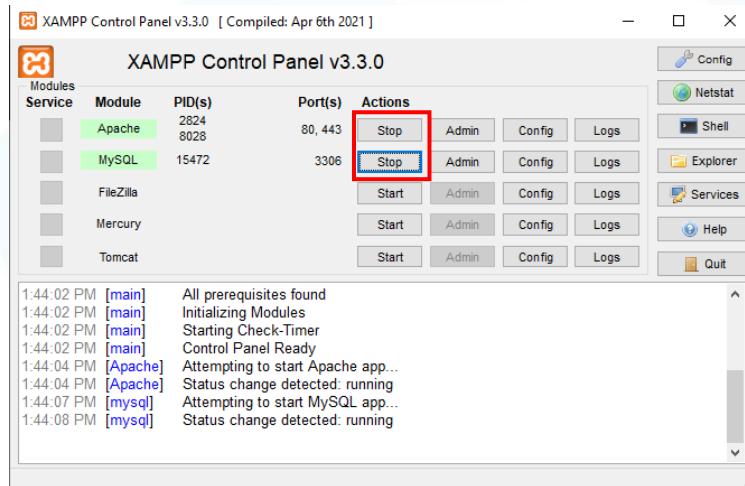
## Command Generate Request

```
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 for managing databases. The top navigation bar includes links for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables,Charsets, and More. The main area is titled 'Databases' and displays a list of existing databases: information\_schema, mysql, performance\_schema, phpmyadmin, and test. A modal window titled 'Create database' is open, prompting the user to enter a database name ('new-project') and select a collation ('utf8mb4\_general\_ci'). The 'Create' button is highlighted with a red box. Below the modal, there are 'Check all' and 'Drop' buttons, and a search bar. The table below lists the current databases with their collations and actions like 'Check privileges'. The total number of databases is indicated as 5.

Database	Collation	Action
information_schema	utf8_general_ci	<a href="#">Check privileges</a>
mysql	utf8mb4_general_ci	<a href="#">Check privileges</a>
performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
phpmyadmin	utf8_bin	<a href="#">Check privileges</a>
test	latin1_swedish_ci	<a href="#">Check privileges</a>

Total: 5

The screenshot shows the phpMyAdmin interface. On the left, there's a sidebar with a tree view of databases: New, information\_schema, mysql, new-project (which is highlighted with a red box), performance\_schema, phpmyadmin, and test. Below the sidebar is a navigation bar with tabs: Structure, SQL, Search, Query, Export, and another tab that is partially visible. Under the Structure tab, it says 'No tables found in database.' A 'Create new table' button is visible. In the main area, there's a form to create a new table. The 'Table name' field contains 'book' and the 'Number of columns' field contains '5'. A 'Create' button is at the bottom right of the form. The entire 'Table name' and 'Number of columns' input area is also highlighted with a red box.

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

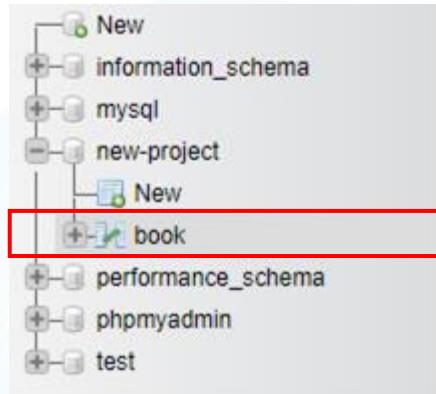
Pada menu **Structure** database tersebut, isikan form pembuatan tabel seperti pada gambar kemudian klik tombol **Create**.

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: book    Add 1 column(s)    Go

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I
id	INT	10	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/> PRIMARY
nama_buku	VARCHAR	255	None			<input type="checkbox"/>	--	<input type="checkbox"/>
deskripsi	TEXT		None			<input type="checkbox"/>	--	<input type="checkbox"/>
harga	INT	20	None			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		None			<input type="checkbox"/>	--	<input type="checkbox"/>



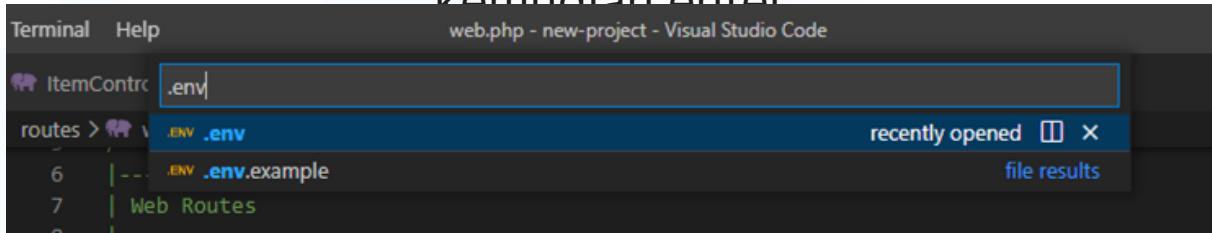
Tabel yang telah dibuat akan muncul pada sidebar.

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

The screenshot shows the MySQL Workbench interface with the 'Insert' tab selected. A table named 'Buku1' is displayed with three columns: 'nama\_buku' (varchar(255)), 'deskripsi' (text), and 'harga' (int(20)). The 'deskripsi' column contains the value 'Ini adalah buku1'. The 'harga' column has a value of '10000' entered. A 'Go' button is visible at the bottom right.

nama_buku	deskripsi	harga
varchar(255)	Ini adalah buku1	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=
```

The line "DB\_DATABASE=new-project" is highlighted with a red rectangular box.

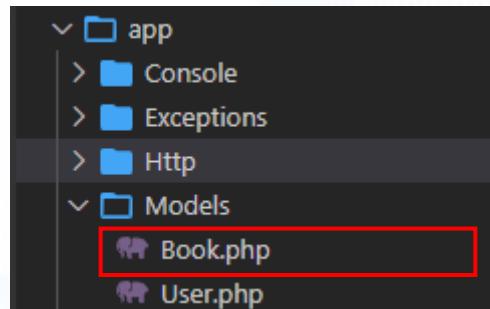
Untuk memberhentikan perintah yang masih jalan sebelumnya menggunakan **Ctrl + c** pada terminal, kemudian untuk membuat model melalui terminal dengan memasukkan perintah

**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 with a dark theme. On the left, there's a sidebar with navigation links: Prologue, Getting Started, Architecture Concepts, The Basics, Digging Deeper, Security, Database, Eloquent ORM, Getting Started (which is highlighted with a red dot), Relationships, Collections, and Mutators / Casts. The main content area has a title "Eloquent: Getting Started" and a list of sections under it, each preceded by a "#": Introduction, Generating Model Classes, Eloquent Model Conventions, Table Names, Primary Keys, Timestamps, Database Connections, Default Attribute Values, Retrieving Models, Collections, Chunking Results, and Chunk Using Lazy Collections. In the top right corner, there's a "VERSION" dropdown set to "9.x" and a small circular icon. In the bottom right corner, there's a "FORGE" logo with the text "Deploy Laravel and PHP applications on DigitalOcean, Linode, & AWS." Below that, it says "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"; → Nama
13
14     protected $primaryKey = "id"; → Primary
15
16     public $timestamps = false; → key
17
18     protected $fillable = [
19         'nama_buku',
20         'deskripsi',
21         'harga',
22     ];
23 }
24
```

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

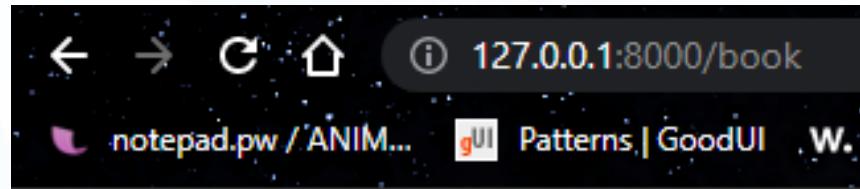
Mengambil setiap data dari database yang telah dan memunculkan pada halaman URL dibuat menggunakan controller. Menambahkan sintaks pada fungsi **index** seperti pada gambar.

```
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.

The screenshot shows the 'Tables' section of the PHPMyAdmin interface. There are two tables listed: 'book' and 'item'. The 'item' table has a red box around its 'Drop' button. Below the table list, there is a 'Sum' link. A confirmation dialog box is overlaid on the page. The dialog title is 'Confirm'. The message inside says: 'You are about to DESTROY a complete table! Do you really want to execute "DROP TABLE `item`"?'. There is a checked checkbox labeled 'Enable foreign key checks' with a red box around it. At the bottom of the dialog are 'OK' and 'Cancel' buttons, with 'OK' also having a red box around it.

Table	Action	Rows	Type	Collation	Size	Overhead
book	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
item	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
2 tables	Sum				32.0 KiB	0 B

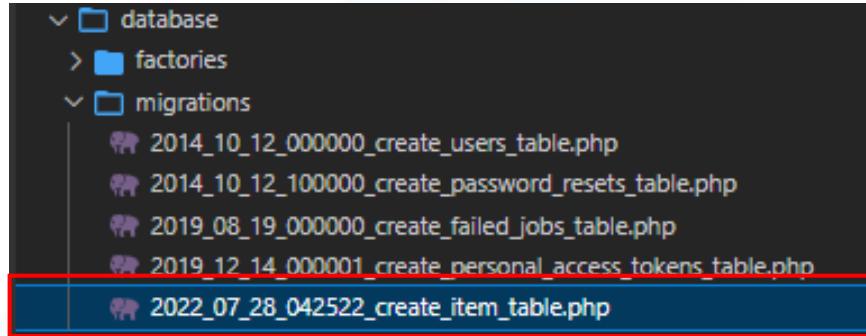
Untuk membuat migration dengan memasukkan perintah

**`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
```

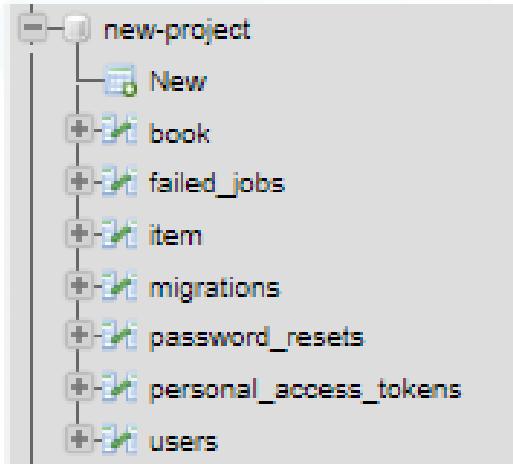


Setelah migration berhasil terbentuk, pada direktori **database/migration** akan terbentuk file migration baru.

```
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 seusai dengan file migrasi yang ada pada project.

# Seeder

Untuk membuat seeder dengan memasukkan perintah

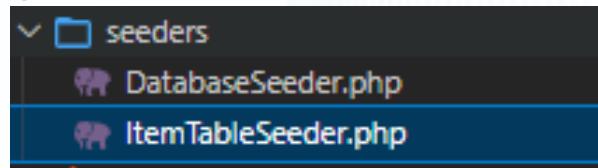
**php artisan make:seeder *nama\_seeder***

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**.

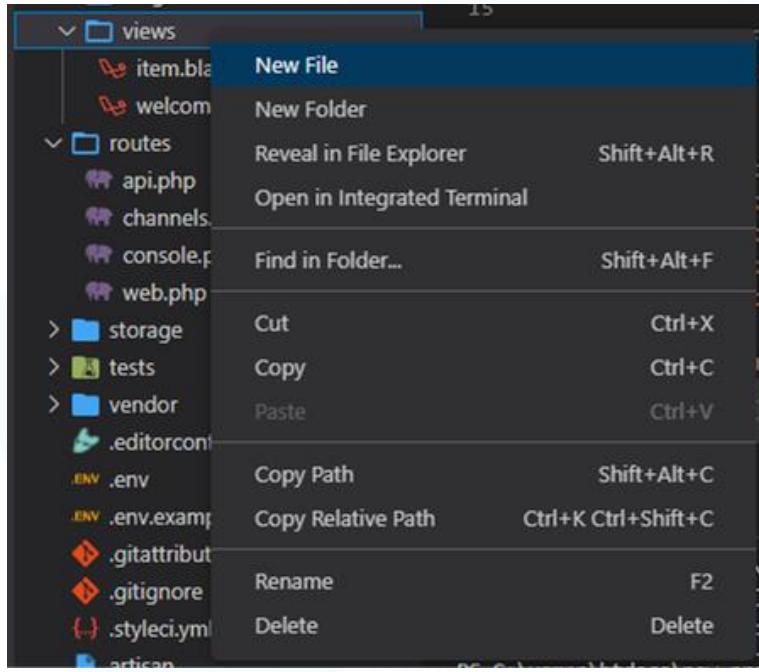
Untuk menjalankan pembuatan seeder dengan memasukkan perintah

```
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

	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	id	item_name	item_type	item_price	item_desc	created_at	updated_at
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1	op1Tvg7eAI03enN8Bsj7MnMT	WNSt61zz5n	5920	WENLJHtqzLmPtwJFP45804ZKbQN8mhZg2QtFHX0ug7s78J		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2	onTJI3fqCJlyOKDCJYajYgsRn	TVnVBAJn3d	84579	7wQjPB6Th7vLfXG112q5U2uNFFLJZQHLojbxzDcBbVLoJ6H		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3	Rc1Y8CYtmlobLxKsQdsTeYtQy	ik1hBZZVWq	62055	srFePUaQuXu5sjLcWzU4dAwUbaXTFfoge18y17D2OvdOvEspb		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4	mD5xDOrUT74g6kq0Xr4LuCYR	OMEPx45OsI	69117	lqX4dxLG4X64wtVsXbVh15SYW0tvxa0DzdTEBzA440wFwK18		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	5	hLGTKTyEQGosSDAz3RjDTbpCl	uWFYNXNUxoj	71848	82yxKSwLniE329NPedjjPgQnDtWxB9yelWvwceCrB7U9wRpk		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	6	i101v8QpPVdixncRoR4mpY06	frCJqwC0V	58928	wGDPPkMcQxk8wF56c49PobimMMvrXZ841KExJnsWfTqU1jW1L		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	7	4Z2UjpQfbTvr8nP8Zu0jqLk	OxQfpwPwiea	69153	qPMa3MBcJq41MbtOkMv08PjKxCPNQIRngJf2XZ0teGR3IE07m		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	8	O2XgeDMFKGdCE6tTJ8qdQk1Xy	o4uqjk0eNk	58884	2CZHoylRqTlwDgGrx1qoVhRv9Q/GBHb6gJbNQ0dwbaLDEzk4A		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	9	0kgybKmlMmAs5AkYCC7ep3vR1	ppIIJ15ELg	74764	e0qeSPoSPEfyosApaP4BhY16PZgLkant7geK78qTfv1UpHeatro		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	10	OHkrR9ot7HnWUm8z43hGhov	IF0Vp7QzQ9	63932	Q2gM1FtNczrtQAhgHTBldLtwfIRgbwVjsvhvPNhF0siebUbL		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	11	3phqjgtJGZJrdgg5PDpkXkm8C	nKyTMthy2S	55090	MheGebLKtu5x7aOdZCxH5bOdDOLRbGd6avenqyJEMA3IXFytJ		2022-07-28 06:44:06	2022-07-28 06:44:06
<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	12	KxuuZObDrBa5ebwlEubpf01Fr	8BQtgwca3k	23757	phW4JHBPvflRwzvfyRpAHlyR31dDrc1HT1PwgSndBp5bQzsV		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 #000">
18              <tr>
19                  <td>Nama Item</td>
20                  <td><input type="text" name="item_name"></td>
21              </tr>
22              <tr>
23                  <td>Type 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***

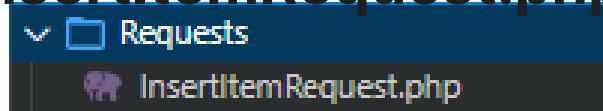
Sebagai contoh membuat request untuk menambahkan data baru

**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 |     }
36 | }
```

Menjalankan Laravel dengan memasukan 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

seperti berikut

Nama Item	<input type="text"/>
Tipe Item	<input type="text"/>
Harga Item	<input type="text"/>
Deskripsi Item	<input type="text"/>
<input type="button" 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.

ID	Kode Item	ID Kategori	Harga	Deskripsi
100	7HAwRLhYp9o9LBg4hUSiEsfUt	DfWcQF8oQn	23996	8CLeGr9ZBQfkU4ZjWesFq7hyHI4f700U9wCmjIVCidvceYS5Vc
101	Buku anak SD	Buku	10000	Ini adalah buku sekolah anak sd

	Edit	Copy	Delete	ID	Nama Item	Tipe Item	Harga	Deskripsi	Created At	Updated At
				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