



**INSTITUT TEKNOLOGI DEL**  
**MATERI PRAKTIKUM**  
**1131204– Pengembangan Situs Web II**  
**SEMESTER II - TAHUN AJARAN 2023/2024**

<b>Session Date</b>	:	21 Februari 2024
<b>Semester</b>	:	2
<b>Courses</b>	:	Pengembangan Situs Web II
<b>Week/Session</b>	:	11/3
<b>Key Topics</b>	:	Konsep MVC dan Pengenalan Framework Laravel
<b>Activity</b>	:	Praktikum
<b>Duration</b>	:	110 menit
<b>Delivery</b>	:	PHP application
<b>Deadline of delivery</b>	:	Akhir Sesi Praktikum
<b>Place of delivery</b>	:	Ecourse
<b>Goal</b>	:	Mahasiswa mampu menerapkan konsep-konsep MVC dengan menggunakan Framework Laravel

**PENUGASAN:**

Sebelum bekerja, setiap mahasiswa harus membaca instruksi di bawah ini.

**Sangat disarankan bagi anda untuk:**

1. Melakukan observasi pada setiap kode yang ada.
2. Menggunakan editor seperti VSCode atau JetBrains PhpStorm untuk membantu Anda belajar.
3. Mencari sumber-sumber lain seperti buku, artikel, bahkan video untuk memperkaya wawasan dan meningkatkan pemahaman Anda.
4. Jika Anda merasa ada hal yang belum dipahami, silakan untuk berkonsultasi pada TA.
5. Setiap file yang Anda buat **harus** ditambahkan keterangan seperti di bawah ini:

```
<!--
    Nama : Nama Anda
    NIM : XXX
    Kelas : XXX
-->
```

# Introduction

Pada praktikum ini anda akan mempelajari bagaimana:

1. Memahami konsep Model View Controller (MVC)
2. Menerapkan MVC pada PHP ketika mengembangkan sebuah web
3. Mengenali dan mengimplementasikan salah satu framework PHP yaitu **Laravel**
4. Cara untuk instalasi Framework Laravel menggunakan Composer
5. Membuat CRUD (*Create, Read, Update, Delete*) menggunakan Framework Laravel

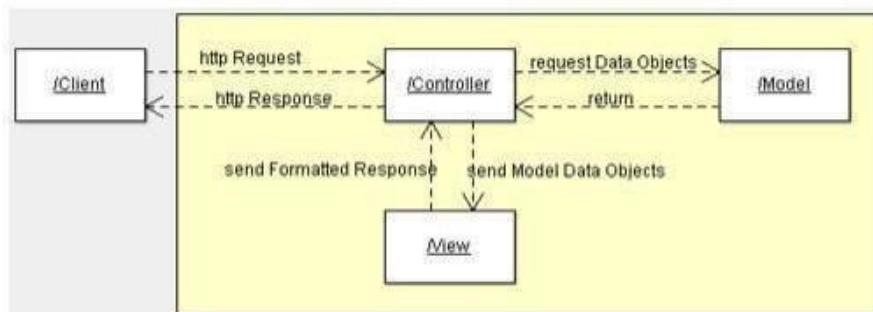
## MVC Observation

Pada sesi ini anda akan mempelajari mengenai konsep MVC (Model View Controller), Anda diharapkan mampu mengetahui mvc concept dan bagaimana mvc tersebut dikonstruksi menjadi serangkaian framework.

## MVC Observation

MVC atau *Model View Controller* merupakan salah satu implementasi dari prinsip SoC untuk aplikasi yang Anda kembangkan. SoC atau *Separation of Concern* adalah suatu teknik di mana Anda akan membagi - bagikan kode ke dalam beberapa bagian.

Masing - masing bagian dari kode ini mempunyai tanggung jawabnya masing – masing. Misalnya saja ada bagian kode yang khusus hanya menangani basis data. Lalu, ada bagian kode yang hanya menangani tampilan. Dan ada juga bagian kode yang menghubungkan antara tampilan dengan basis data. Pada bagian ini ditunjukkan bagaimana konsep MVC dapat dicapai.



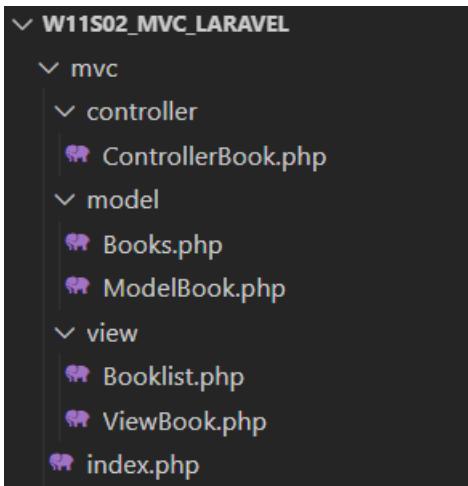
Anda diasumsikan bahwa telah mengerti definisi controller, view, dan model. Secara garis besar, diagram di atas hanya menjelaskan mengenai bagian apa, dan kepada bagian apa data, request, dan modularisasi yang terjadi.

## Penerapan Konsep MVC

Konsep mvc yang akan dibuat pada sesi ini sifatnya sangat sederhana, namun jika anda mengikutinya dengan baik, maka anda dapat dipastikan sudah mengerti secara garis besar, bagaimana mvc diterapkan.

### a. Struktur Folder

Berikut adalah struktur folder dari MVC pada direktori Anda.



Jika anda perhatikan, terdapat folder utama, yakni **mvc** yang hanya berisi file **index.php**, kemudian di dalam folder utama itu, terdapat tiga folder yakni **controller**, **model**, dan **view**, masing-masing di dalam folder tersebut adalah file kode program yang merepresentasikan sesuai dengan nama folder tersebut.

### b. Kode Program

Silahkan buat kode program berikut dan simpan sesuai dengan diagram folder yang diberikan di **bagian a.**

### File: mvc/controller/ControllerBook.php

```
<?php
include_once("model/ModelBook.php");
class ControllerBook{
    public $model;

    public function __construct()
    {
        $this->model = new ModelBook();
    }

    public function invoke(){
        if(isset($_GET['book'])){
            //Pada bagian ini ditunjukkan semua data buku tersedia
            $books = $this->model->getBookList();
            include 'view/Booklist.php';
        } else{
            //Pada bagian ini ditunjukkan data buku yang direquest
            $book = $this->model->getBook($_GET['book']);
            include 'view/ViewBook.php';
        }
    }
}
?>
```

### File: mvc/model/Book.php

```
<?php
//Class ini mengkonstruksikan bagaimana struktur data buku
class Book{
    public $isbn;
    public $title;
    public $author;
    public $description;
    public function __construct($isbn,$title, $author, $description)
    {
        $this->isbn = $isbn;
        $this->title = $title;
        $this->author = $author;
        $this->description = $description;
    }
}
?>
```

### File: mvc/model/ModelBook.php

```
<?php
include_once("model/Book.php");
class ModelBook{
    public function getBookList(){
        //bagian ini adalah bentuk hardcode data buku untuk menggambarkan database
        return array(
            "ISBN01" => new Book( isbn: "ISBN01", title: "Secret Life of Broqila", author: "Prada Siahaan", description: "Humorous Book"),
            "ISBN02" => new Book( isbn: "ISBN02", title: "Moonwalker", author: "J.Walker", description: "Science Book"),
            "ISBN03" => new Book( isbn: "ISBN03", title: "Lucifer Bapa Peradaban Gelap", author: "Lucifer", description: "Not at All"),
            "ISBN04" => new Book( isbn: "ISBN04", title: "PHP for Dummies", author: "Albert", description: "Just Practice"),
            "ISBN05" => new Book( isbn: "ISBN05", title: "Longman TOEFL Preparation ", author: "Debora Phillips", description: "Do Your Best")
        );
    }

    public function getBook($isbn){
        $allBooks = $this->getBookList();
        return $allBooks[$isbn];
    }
}
?>
```

### File: mvc/view/Booklist.php

```
<html>
<head>
    <title>List of Books</title>
</head>
<body>
    <table>
        <tr>
            <th>Title</th>
            <th>Author</th>
            <th>Description</th>
        </tr>
        <?php
        foreach ($books as $key => $book){
            echo '<tr>
                    <td><a href="index.php?book='.$book->isbn.'">'. $book->title.'</a></td>
                    <td>'.$book->author.'</td>
                    <td>'.$book->description.'</td>
                </tr>';
        }
        ?>
    </table>
</body>
</html>
```

### File: mvc/view/ViewBook.php

```
<html>
<head></head>
<body>
<?php
    echo 'Title: '.$book->title. <br>;
    echo 'Author: '.$book->author. <br>;
    echo 'Description: '.$book->description. <br>;
?>
</body>
</html>
```

File: **mvc/index.php**

```
<?php
    include_once('controller/ControllerBook.php');
    $controller = new ControllerBook();
    $controller->invoke();
?>
```

Jalankan program di atas dan pelajari setiap baris code program, sehingga Anda mengerti alur dari konsep MVC.

## Framework Laravel

Laravel merupakan salah satu framework Laravel yang bersifat *opensource* dan sangat popular dari hasil Google Trends 5 tahun terakhir (2015-2020). Laravel dirancang agar proses pengembangan aplikasi web menjadi lebih mudah dan cepat karena berbagai fitur bawaan didalamnya. Fitur-fitur tersebut merupakan salah satu alasan mengapa banyak pengembangan memanfaatkan kerangka kerja Laravel yang menggunakan konsep Model- View-Controller (MVC). Laravel berada di bawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode. Pada saat ini pengembangan Laravel berada di versi 8.0 dan baru dirilis pada 8 September 2020 yang lalu dan hanya mendukung versi php 7.3.0 ke atas.

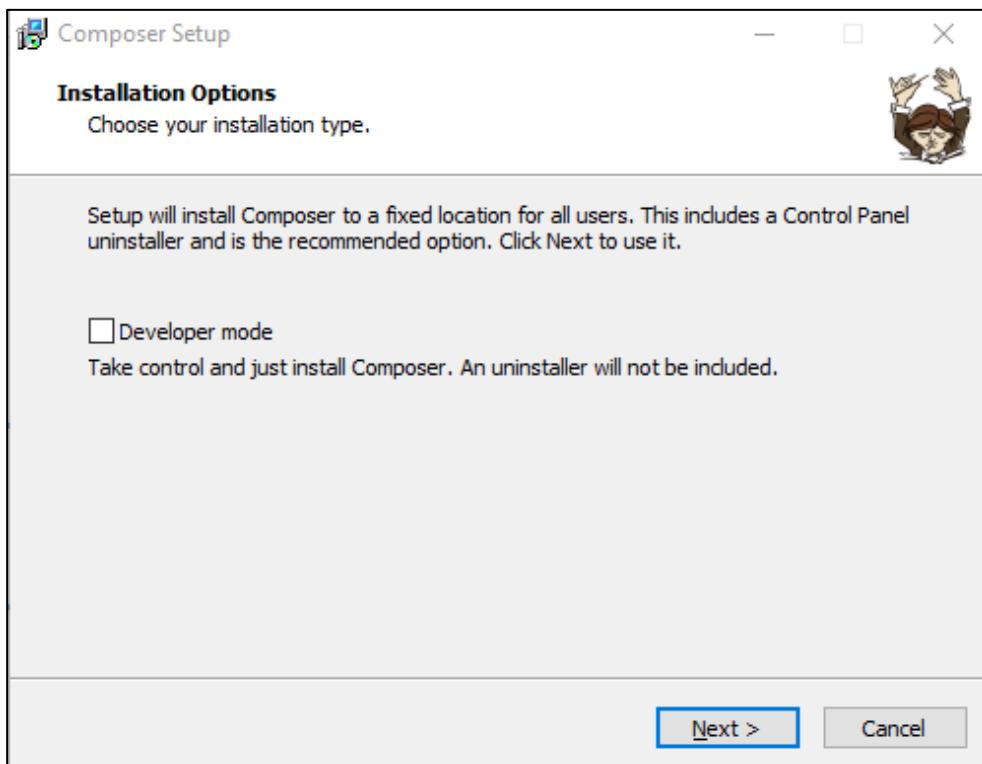
Pada laravel 8, laravel menggunakan jetstream untuk *scaffolding*-nya, tidak lagi bootstrap, vue/react, perubahan ini yang paling terasa karena di versi 8 ini terdapat beberapa tambahan fitur built-in pada login register seperti login, registration, email verification, two-factor authentication, session management, API support via Laravel Sanctum, and optional team management. Pada laravel 8, terdapat perubahan pada direktori app/Models, jika pada laravel 7, tidak terdapat folder khusus untuk model, yang artinya model di taruh pada luar folder.

Pada praktikum minggu ke-11, Anda diharapkan sudah melakukan instalasi framework laravel, jikalau belum ikuti instruksi instalasi yang ada dan bagi yang sudah melakukan instalasi segera ikuti instruksi persiapan lingkungan kerja. Pada praktikum ini kita menggunakan laravel versi 8, jadi Anda dapat melakukan eksplorasi lebih terhadap fitur-fitur tambahan sehingga nantinya akan mempermudah Anda untuk mengimplementasikan pada proyek Anda.

## Instalasi Laravel

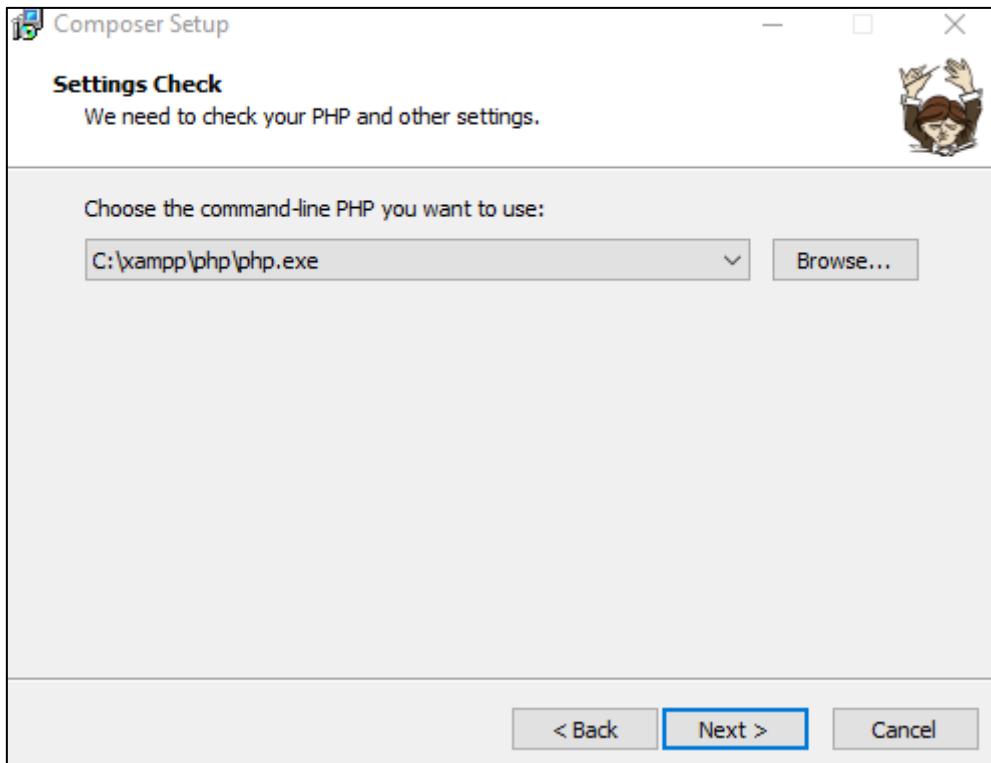
Untuk melakukan instalasi Laravel, ada beberapa cara yang dapat dilakukan, namun pada praktikum kali ini, instalasi akan dilakukan menggunakan composer. Sebelum melakukan instalasi Laravel, kita harus install composer terlebih dahulu, ikuti cara berikut untuk install composer:

1. Kunjungi laman <https://getcomposer.org/>
2. Klik tombol download
3. Temukan kata “Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.” Dan klik [Composer-Setup.exe](#)
4. Setelah selesai di download, klik file composer yang sudah didownload.
5. Akan muncul seperti gambar berikut, klik Next.

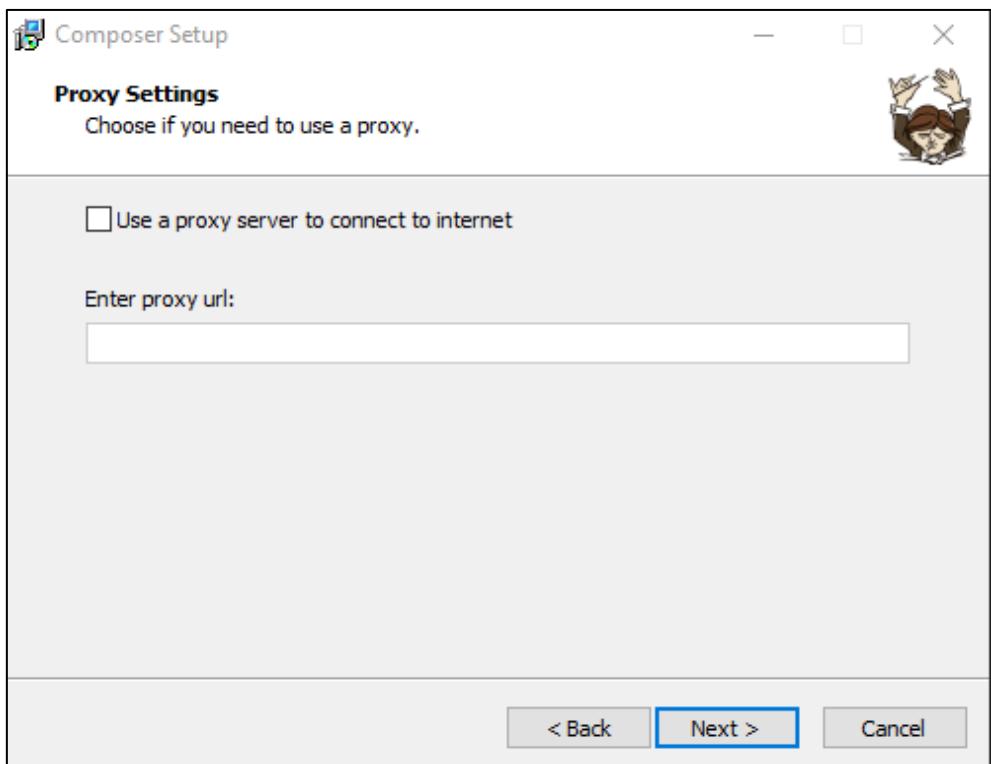


6. Setelah itu, akan muncul seperti gambar berikut. Selanjutnya Anda akan diminta untuk menentukan lokasi file php (php.exe) yang akan digunakan pada command line (PHPCLI-Command Line Interface). File tersebut biasanya berada di dalam folder php. Jika Anda menggunakan XAMPP, lokasi file ini akan berada di C:\xampp\php. Klik Next
7. Pastikan pilih lokasi instalasi Composer berjalan di C:\xampp\php\php.exe. Klik Next

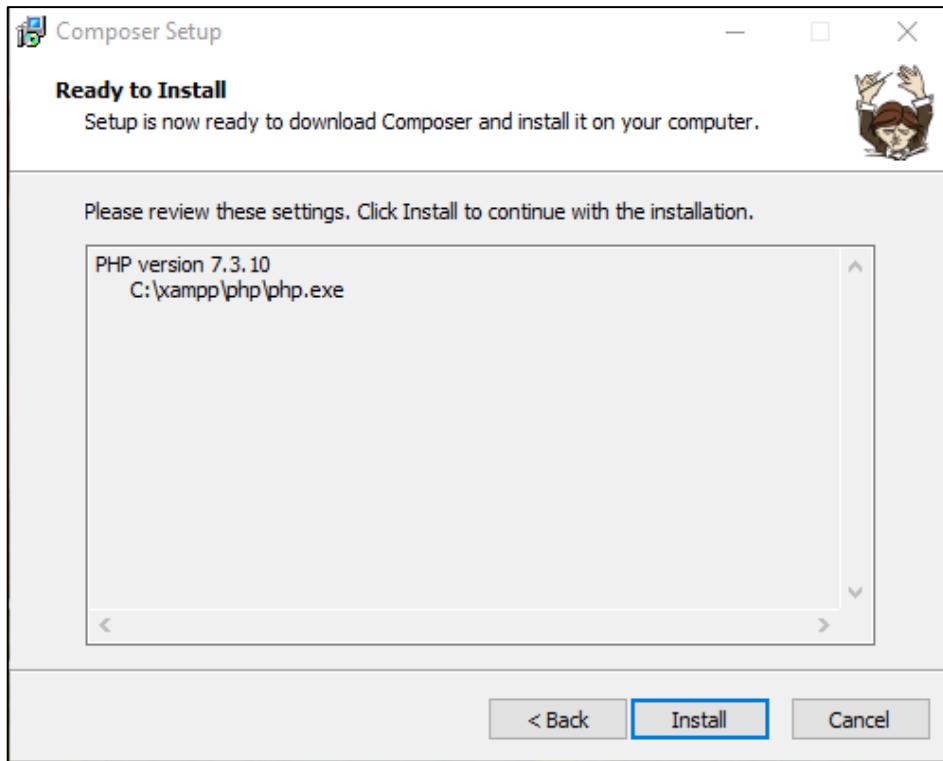
apabila lokasi file php sudah benar.



8. Kemudian akan muncul seperti gambar berikut, klik next.



9. Setelah itu, akan muncul dialog review lokasi instalasi seperti gambar berikut, jika sudah benar klik install.



10. Tunggu hingga proses instalasi selesai hingga akan muncul dialog, klik next.

11. Proses installasi composer sudah selesai, dan akan muncul dialog seperti gambar berikut, klik finish.



12. Setelah itu cek apakah composer memang sudah benar telah terinstall di computer anda, dengan cara buka CMD dan ketikkan **composer –version** dan tekan enter, jika sudah terinstall, akan muncul seperti gambar berikut.

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the standard Microsoft Windows header with icons for minimize, maximize, and close. The text area displays the following command and its output:

```
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>composer --version
composer version 2.0.10 2021-02-23 16:11:37

C:\Users\Kevin>
```

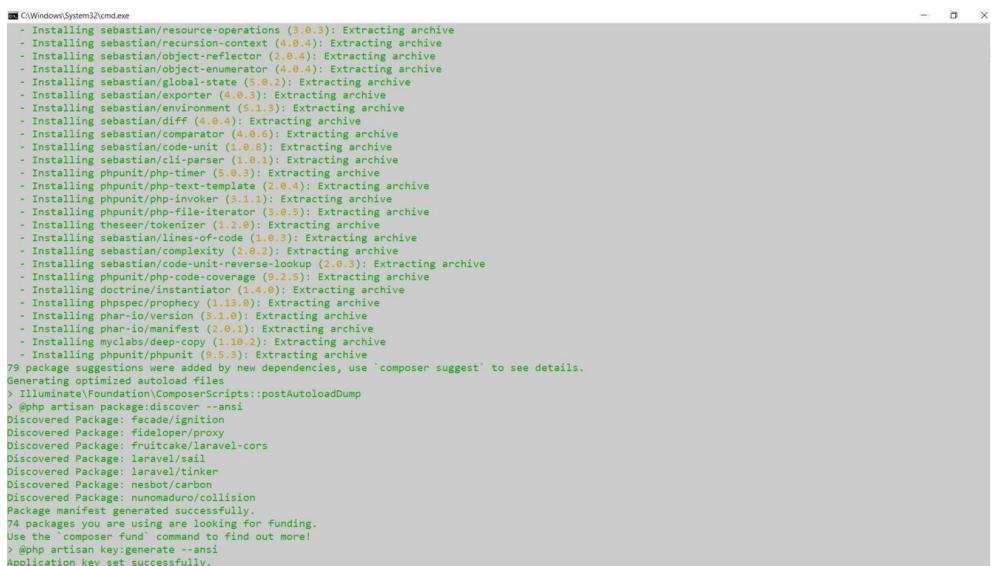
The text "Finish" is visible at the bottom of the screen, indicating the user has completed the previous step.

## Persiapan Lingkungan Kerja

1. Hal yang pertama dilakukan adalah menentukan lokasi project, tidak sama seperti PHP yang penyimpanan project nya di folder ..\xampp\htdocs, project Laravel bebas disimpan dimana saja. Pada praktikum kali ini, kita akan install Laravel di drive C:/.
2. Pada drive C:/ buat folder baru bernama, **Instalasi Laravel**, kemudian masuk ke dalam folder tersebut. Pada path folder yang ada diatas, klik path tersebut, kemudian ketik cmd dan tekan enter, sehingga akan terbuka Command Prompt seperti gambar berikut:



3. Pada command prompt, ketikkan **composer create-project laravel/laravel Template** Kemudian tekan enter, dan tunggu proses instalasi hingga selesai dan akan muncul seperti gambar berikut:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1448]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Instalasi Laravel>

C:\Instalasi Laravel> composer create-project laravel/laravel Template
- Installing sebastian/resource-operations (3.0.3): Extracting archive
- Installing sebastian/recursion-context (4.0.4): Extracting archive
- Installing sebastian/object-reflector (2.0.4): Extracting archive
- Installing sebastian/object-enumerator (4.0.4): Extracting archive
- Installing sebastian/global-state (5.0.2): Extracting archive
- Installing sebastian/exporter (4.0.3): Extracting archive
- Installing sebastian/environment (5.1.3): Extracting archive
- Installing sebastian/diff (1.0.0): Extracting archive
- Installing sebastian/complexity (2.0.2): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (2.0.3): Extracting archive
- Installing sebastian/code-unit (1.0.0): Extracting archive
Installing sebastian/cli-parser (1.0.1): Extracting archive
Installing phpunit/php-timer (5.0.3): Extracting archive
Installing phpunit/php-text-template (2.0.4): Extracting archive
Installing phpunit/php-invoker (3.1.1): Extracting archive
Installing phpunit/php-file-iterator (3.0.3): Extracting archive
Installing thessein/tokenizer (1.2.0): Extracting archive
Installing sebastian/lines-of-code (1.0.3): Extracting archive
- Installing sebastian/complexity (2.0.2): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (2.0.3): Extracting archive
- Installing sebastian/php-code-coverage (9.2.5): Extracting archive
- Installing sebastian/phpunit (11.0.0): Extracting archive
- Installing phar-io/version (3.1.0): Extracting archive
- Installing phar-io/manifest (2.0.1): Extracting archive
- Installing myclabs深深拷贝 (1.0.3): Extracting archive
- Installing phpunit/phpunit (9.5.3): Extracting archive
79 package suggestions were added by new dependencies, use 'composer suggest' to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
74 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan key:generate --ansi
Application key set successfully.
```

4. Setelah itu ketikkan **cd Dealer-KitaAja** pada command prompt.
5. Kemudian untuk menjalankan project, ketikkan **php artisan serve** pada command prompt dan klik link yang di blok seperti di gambar, atau pada browser anda, ketikkan <http://127.0.0.1:8000>
6. Instalasi project Laravel telah selesai.

# CRUD Menggunakan Laravel

Dalam pengembangan aplikasi menggunakan Laravel, konsep yang digunakan adalah konsep MVC(Model- View-Controller). Konsep ini yang akan kita gunakan untuk melakukan CRUD dengan Laravel. CRUD(Create- Read-Update-Delete) merupakan fungsi yang generik atau umum untuk semua pengembangan aplikasi karena dengan adanya CRUD dapat mengelola data yang terdapat di basis data. Anda hanya menyesuaikan table yang akan melakukan fungsi CRUD tersebut.

## A. Proses Create dan Read

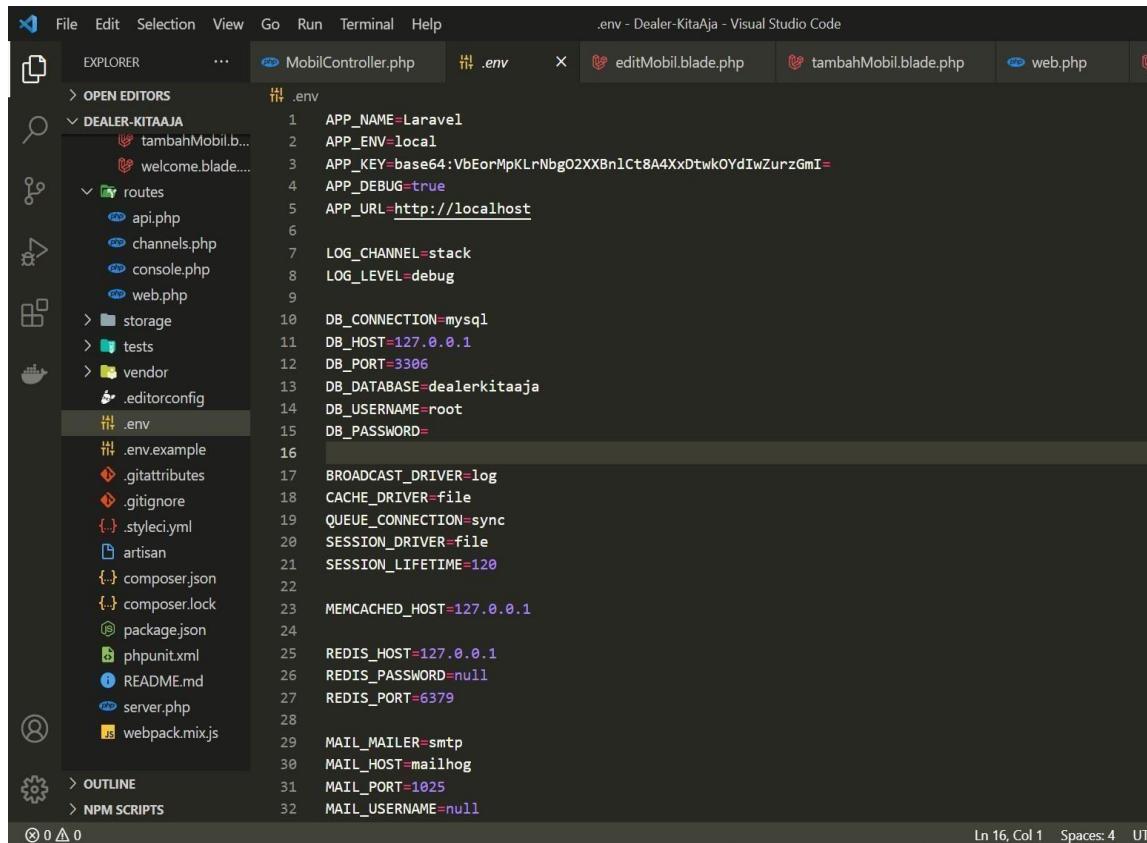
Ikuti Langkah-langkah berikut ini untuk membuat Create & Read pada Laravel.

1. Pertama-tama kita harus membuat basis data terlebih dahulu dengan menggunakan engine mysql. Nyalakan Apache dan MySQL di xampp control, setelah itu jalankan **localhost/phpMyAdmin** di web browser.
2. Di dalam localhost/phpMyAdmin buat sebuah basis data baru dengan nama **DealerKitaAja**, kemudian klik tombol **Buat**.

The screenshot shows the phpMyAdmin interface on a web browser. The URL in the address bar is `localhost/phpmyadmin/server_databases.php?server=1`. The main window title is "Basis data". On the left, there's a sidebar with a tree view of existing databases: Baru, api-II, belajarapi, failed\_jobs, feeds, migrations, password\_resets, users, chatapp, information\_schema, mysql, pantaihatulian, performance\_schema, phpmyadmin, and test. In the center, there's a form titled "Basis data" with fields "Nama basis data" containing "DealerKitaAja" and "Collation" dropdown set to "utf8mb4\_general\_ci". Below the form is a button labeled "Buat". Underneath the form, there's a table titled "Basis data" with columns "Nama basis data", "Penyortiran", and "Tindakan". The table lists all the databases from the sidebar. A warning message at the bottom says: "Perhatian: Pengaktifan statistik basis data dapat mengakibatkan lalu lintas besar antara webserver dan server MySQL." and "Aktifkan Statistik".

3. Setelah database telah dibuat, kita beralih ke project yang kita buat pada instalasi Laravel di

atas. Buka project tersebut di code editor anda. Hal yang pertama kita lakukan adalah konfigurasi project laravel dengan database yang kita buat, buka file .env pada project dan ubah konfigurasi seperti gambar berikut.



```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:VbEorMpKLrNbgO2XXBn1Ct8A4XxDtwkOYdIwZurzGmI=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dealerkitaja
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
MEMCACHED_HOST=127.0.0.1
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
```

4. Setelah itu, Pada folder public buat sebuah folder yang bernama asset, dan di dalam folder asset tersebut buat 3 folder dengan nama, css, js, dan gambar.
5. Lalu, untuk memudahkan kita mengetahui apa penyebab dari kesalahan atau error yang mungkin terjadi. Aktifkan app\_debug pada file app.php yang ada pada folder **app/config/app.php**. Pada file tersebut ubah nilai debug dari false menjadi true seperti gambar di bawah.

```
'debug' => (bool) env('APP_DEBUG', true),
```

6. Kemudian kita akan buat sebuah tabel migration. Di dalam code editor vs code, buka terminal dan ketikkan **php artisan make:migration create\_mobil\_tabel** kemudian tekan enter.

```
C:\Users\Kevin\Documents\033_kevin_yoyada_tambunan\asdos d3 ti\Dealer-KitaAja>php artisan make:migration create_mobil_tabel
Created Migration: 2021_03_22_083651_create_mobil_tabel

C:\Users\Kevin\Documents\033_kevin_yoyada_tambunan\asdos d3 ti\Dealer-KitaAja>
Ln 12, Col 20  Spaces: 4  UTF-8  LF  PHP  ⚡  ⚡
```

- Setelah itu, kita akan atur tabel yang akan kita buat di database melalui file yang sudah kita buat, buka file tersebut di

**database>migrations>2021\_03\_22\_083651\_create\_mobil\_tabel.php** dan konfigurasikan kode seperti gambar berikut.

```
File Edit Selection View Go Run Terminal Help 2021_03_22_083651_create_mobil_tabel.php - Dealer-KitaAja - Visual Studio Code
EXPLORER OPEN EDITORS ... 2021_03_22_083651_create_mobil_tabel.php X
DEALER-KITAJA database > migrations > 2021_03_22_083651_create_mobil_tabel.php > CreateMobilTabel > up
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateMobilTabel extends Migration
8 {
9
10 /**
11 * Run the migrations.
12 */
13
14 * @return void
15 */
16 public function up()
17 {
18     Schema::create('mobil_tabel', function (Blueprint $table) {
19         $table->id();
20         $table->string('name');
21         $table->longText('deskripsi');
22         $table->string("gambar");
23         $table->bigInteger('harga');
24         $table->integer('stok');
25         $table->timestamps();
26     });
27 }
28
29 /**
30 * Reverse the migrations.
31 */
32
33 * @return void
34 */
35 public function down()
36 }
```

- Setelah selesai dengan tabelnya, ketikkan **php artisan migrate** di terminal dan tekan enter.

The screenshot shows the Visual Studio Code interface. In the Explorer sidebar, there are several migration files in the 'migrations' folder, including '2014\_10\_12\_00000\_create\_users\_table.php', '2014\_10\_12\_100000\_create\_password\_resets\_table.php', '2019\_08\_19\_00000\_create\_failed\_jobs\_table.php', and '2021\_03\_22\_083651\_create\_mobil\_tabel.php'. The current file being edited is '2021\_03\_22\_083651\_create\_mobil\_tabel.php'. The code in this file defines a migration class 'CreateMobilTabel' that creates a table named 'mobil\_tabel' with columns for id, name, deskripsi, gambar, harga, stok, and timestamps.

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateMobilTabel extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('mobil_tabel', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('deskripsi');
            $table->string("gambar");
            $table->bigInteger('harga');
            $table->integer('stok');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
    }
}

```

In the terminal window, the command 'php artisan migrate' is run, and the output shows the successful migration of the 'mobil\_tabel' table.

9. Jika kita lihat di localhost/phpMyAdmin di browser, maka akan muncul 5 tabel baru di database **DealerKitaAja**. Abaikan semua tabel kecuali tabel mobil\_tabel yang akan kita gunakan. Selanjutnya kita akan buat controller dan model. Ketikkan **php artisan make:controller MobilController -r -m Mobil** dan tekan enter di terminal anda, controller dan model sudah dibuatkan dan sudah siap kita gunakan.
10. Buka file MobilController yang ada di folder **app>Http>Controllers> MobilController.php** kemudian lengkapi code seperti gambar berikut.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "DEALER-KITAJA". Key files include `Controller.php`, `MobilController.php`, `Mobil.php`, and `User.php`.
- Code Editor (Center):** Displays the `MobilController.php` file content. The code defines a `create()` method that returns a view named `tambahMobil`. It also contains a `store()` method that validates input fields (`nama`, `deskripsi`, `gambar`, `stok`, `harga`) and saves a file to the `asset/gambar` directory. Finally, it saves a new `Mobil` instance with the provided data.
- Status Bar (Bottom):** Shows "Ln 36, Col 41" and "PHP".

11. Sekarang Model Mobil yang akan kita ubah code nya, buka model Mobil di folder **app>Models>Mobil.php** dan ubah kodennya seperti gambar berikut.

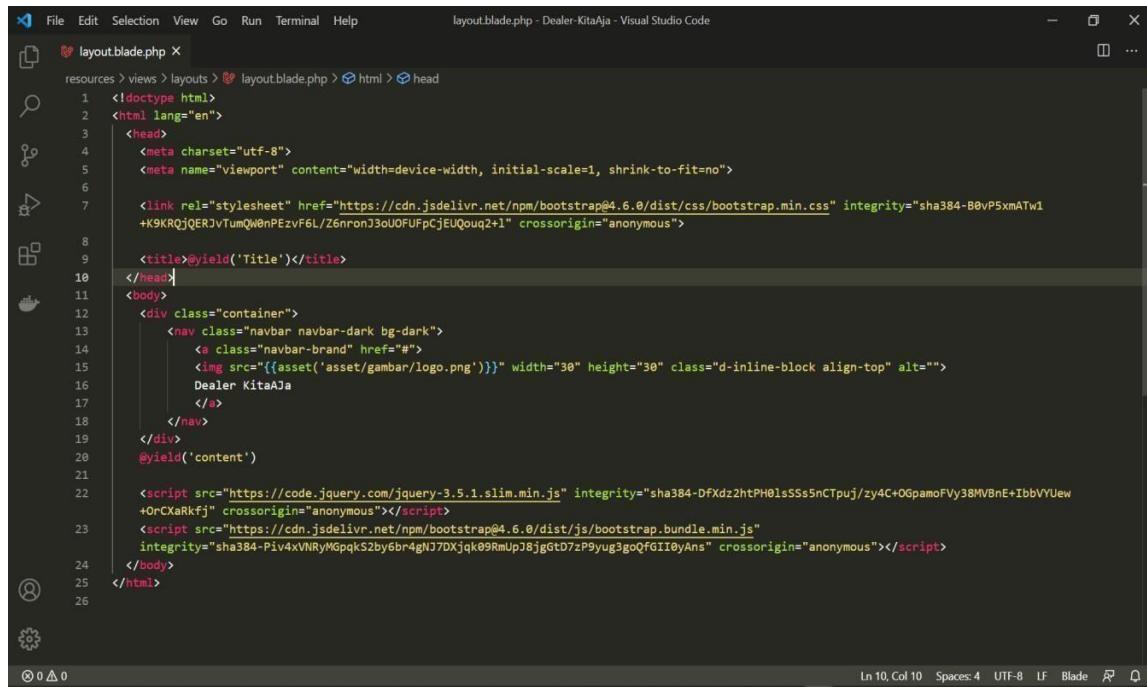
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under the "app" folder:
  - Console
  - Exceptions
  - Http
  - Models (selected)
  - User.php
  - Providers
  - bootstrap
  - config
  - database (selected)
  - factories
  - migrations
    - 2014\_10\_12\_00...
    - 2014\_10\_12\_10...
    - 2019\_08\_19\_00...
    - 2021\_03\_22\_08...
  - seeders
  - .gitignore
  - public
  - resources
    - css
    - js
    - lang
    - views (selected)
- Mobil.php File Preview (Center):** The code editor displays the following PHP code:

```
<?php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Mobil extends Model  
{  
    use HasFactory;  
    protected $table = 'mobil_tabel';  
}
```
- Bottom Status Bar:** Shows "Ln 13, Col 1" and other standard status bar information.

12. Setelah selesai dengan controller dan model, kita akan pindah ke view. Salah satu kelebihan laravel, view yang sama dapat kita gunakan berulang kali(re-usable). Untuk melakukan itu, di dalam folder **resources>views** tambahkan sebuah folder dengan nama layouts dan didalam folder layouts tambahkan sebuah file dengan nama **layout.blade.php**. Di dalam file

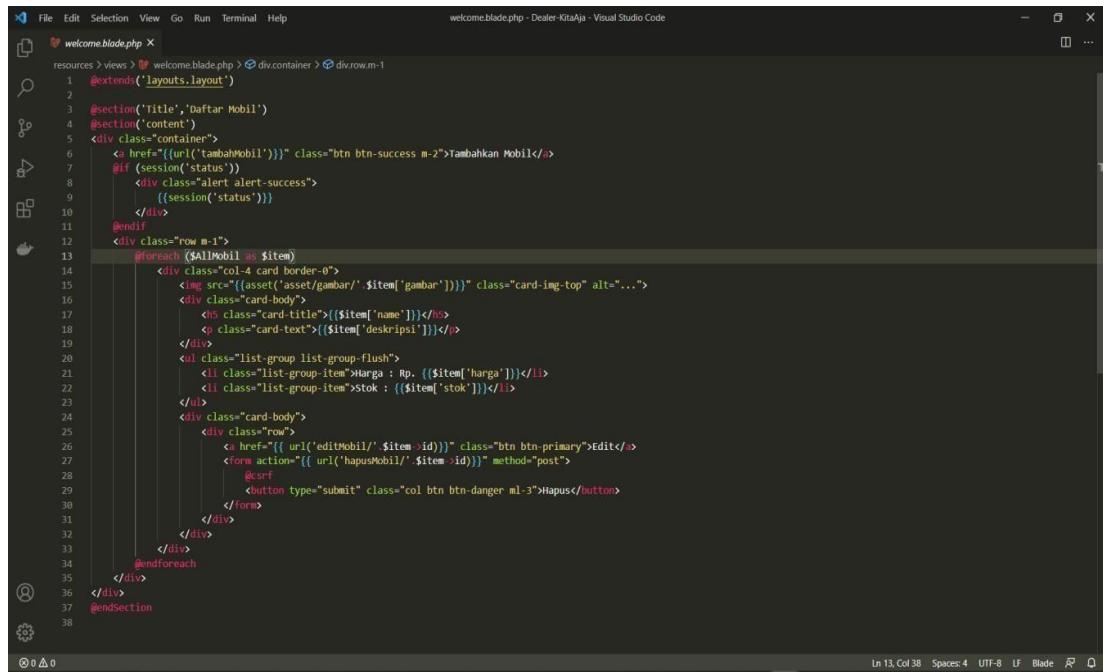
layout.blade.php lengkapi code seperti gambar berikut.



```
resources > views > layouts > layout.blade.php > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9RQjQERJvTmQWnPzF6L/Z6nronJ3oUOFpcCjEUQouq2+l" crossorigin="anonymous">
8
9
10 <title>@yield('Title')</title>
11 </head>
12 <body>
13     <div class="container">
14         <nav class="navbar navbar-dark bg-dark">
15             <a class="navbar-brand" href="#">
16                 
17                 Dealer KitaAja
18             </a>
19         </nav>
20     </div>
21     @yield('content')
22
23     <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXzd2htPH0lsSS5nCTpuj/zY4C+OGpamoFV38MVBrE+IbbVVUew+OrCaRkfj" crossorigin="anonymous"></script>
24     <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-Piv4xVRyMGpqkS2by6br4gNJDXjqk09RmuJ8jGtD7zP9yug3goQfGI0yAns" crossorigin="anonymous"></script>
25 </body>
26 </html>
```

Ln 10, Col 10 Spaces: 4 UTF-8 LF Blade ⚡

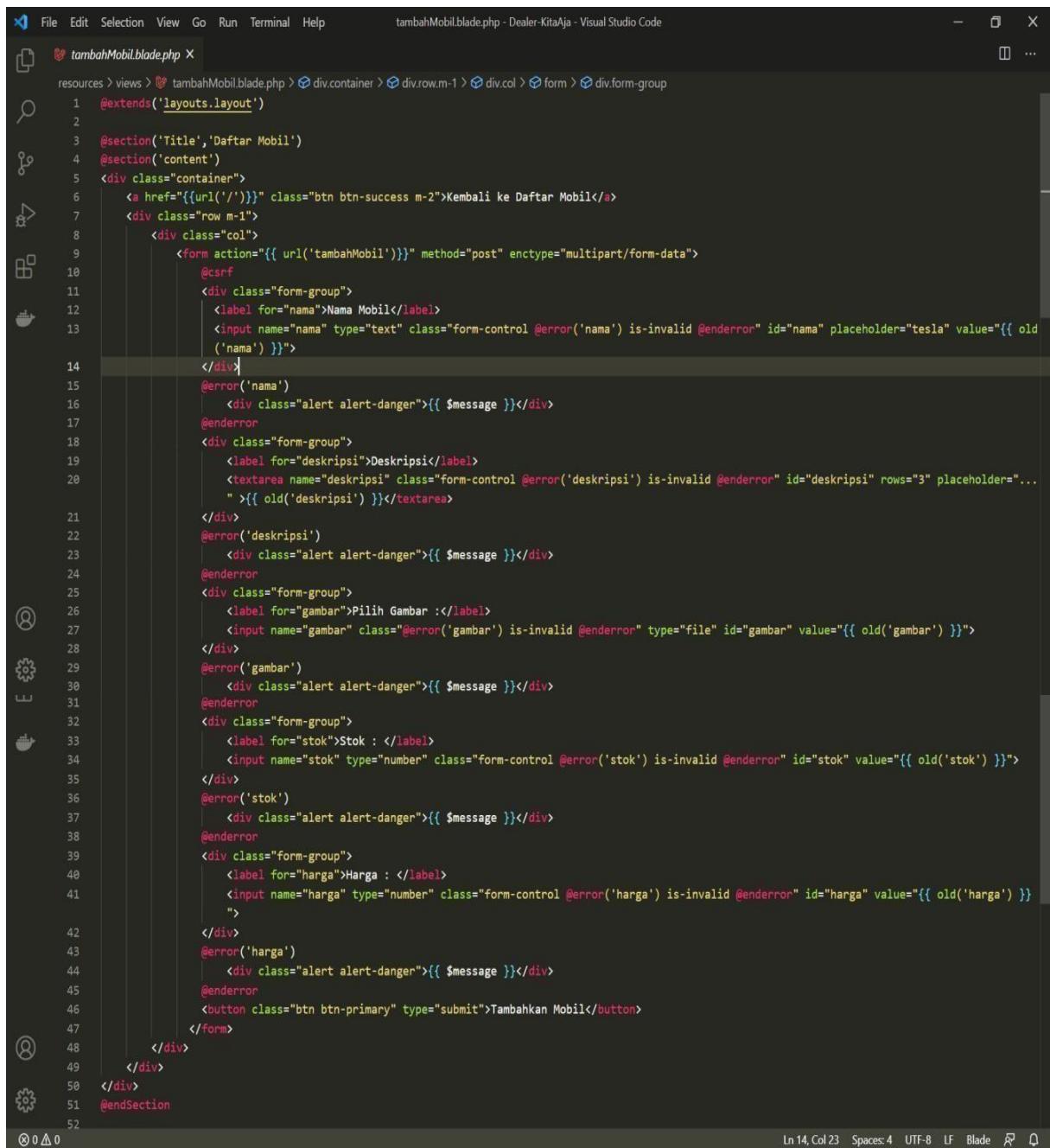
13. Untuk mengimplementasikan layout tersebut, di dalam file **welcome.blade.php** yang berada difolder **resources>views**, lengkapi code seperti gambar berikut.



```
resources > views > welcome.blade.php > div.container > div.row.m-1
1 @extends('layouts.layout')
2
3 @section('title','Daftar Mobil')
4 @section('content')
5 <div class="container">
6     <a href="{{url('tambahMobil')}}" class="btn btn-success m-2">Tambahkan Mobil</a>
7     @if (session('status'))
8         <div class="alert alert-success">
9             {{session('status')}}
10        </div>
11    @endif
12    <div class="row m-1">
13        @foreach ($allMobil as $item)
14            <div class="col-4 card border-0">
15                
16                <div class="card-body">
17                    <h5 class="card-title">{{ $item['name'] }}</h5>
18                    <p class="card-text">{{ $item['deskripsi'] }}</p>
19                </div>
20                <ul class="list-group list-group-flush">
21                    <li class="list-group-item">Harga : Rp. {{ $item['harga'] }}</li>
22                    <li class="list-group-item">stok : {{ $item['stok'] }}</li>
23                </ul>
24                <div class="card-body">
25                    <div class="row">
26                        <a href="{{ url('editmobil/' . $item->id)}}" class="btn btn-primary">Edit</a>
27                        <form action="{{ url('hapusMobil/' . $item->id)}}" method="post">
28                            @csrf
29                            <button type="submit" class="col btn btn-danger ml-3">Hapus</button>
30                        </form>
31                    </div>
32                </div>
33            @endforeach
34        </div>
35    @endSection
36 </div>
37 @endSection
38
```

Ln 13, Col 38 Spaces: 4 UTF-8 LF Blade ⚡

14. Potongan code diatas digunakan untuk menampilkan data dan belum untuk menambahkan data, untuk menambahkan data, buat file baru dengan nama **tambahMobil.blade.php**



```

1  @extends('layouts.layout')
2
3  @section('Title','Daftar Mobil')
4  @section('content')
5  <div class="container">
6      <a href="{{url('/')}}" class="btn btn-success m-2">Kembali ke Daftar Mobil</a>
7      <div class="row m-1">
8          <div class="col">
9              <form action="{{ url('tambahMobil') }}" method="post" enctype="multipart/form-data">
10                 @csrf
11                 <div class="form-group">
12                     <label for="nama">Nama Mobil</label>
13                     <input name="nama" type="text" class="form-control @error('nama') is-invalid @enderror" id="nama" placeholder="tesla" value="{{ old('nama') }}>
14                 </div>
15                 @error('nama')
16                     <div class="alert alert-danger">{{ $message }}</div>
17                 @enderror
18                 <div class="form-group">
19                     <label for="deskripsi">Deskripsi</label>
20                     <textarea name="deskripsi" class="form-control @error('deskripsi') is-invalid @enderror" id="deskripsi" rows="3" placeholder="...>
21                         >{{ old('deskripsi') }}</textarea>
22                 </div>
23                 @error('deskripsi')
24                     <div class="alert alert-danger">{{ $message }}</div>
25                 @enderror
26                 <div class="form-group">
27                     <label for="gambar">Pilih Gambar :</label>
28                     <input name="gambar" class="form-control @error('gambar') is-invalid @enderror" type="file" id="gambar" value="{{ old('gambar') }}>
29                 </div>
30                 @error('gambar')
31                     <div class="alert alert-danger">{{ $message }}</div>
32                 @enderror
33                 <div class="form-group">
34                     <label for="stok">Stok :</label>
35                     <input name="stok" type="number" class="form-control @error('stok') is-invalid @enderror" id="stok" value="{{ old('stok') }}>
36                 </div>
37                 @error('stok')
38                     <div class="alert alert-danger">{{ $message }}</div>
39                 @enderror
40                 <div class="form-group">
41                     <label for="harga">Harga :</label>
42                     <input name="harga" type="number" class="form-control @error('harga') is-invalid @enderror" id="harga" value="{{ old('harga') }}>
43                 </div>
44                 @error('harga')
45                     <div class="alert alert-danger">{{ $message }}</div>
46                 @enderror
47                 <button class="btn btn-primary" type="submit">Tambahkan Mobil</button>
48             </form>
49         </div>
50     </div>
51     @endSection
52 
```

didalam folder **resources>views** dan lengkap code seperti gambar berikut.

15. Kita sudah selesai dengan view, sekarang kita mengatur routes pada file web.php yang berada di folder routes. Lengkapi code seperti gambar berikut.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like `tambahMobil.blade.php`, `welcome.blade.php`, `api.php`, `channels.php`, `console.php`, and `web.php` (which is currently selected).
- Code Editor (Center):** Displays the `routes/web.php` file content. The code defines routes for a mobile application controller, including methods for index, create, store, destroy, edit, and update.
- Bottom Status Bar:** Shows the current file is `web.php - Dealer-KitaAja - Visual Studio Code`, and the editor is at Line 20, Column 81, with 81 spaces and LF as the line separator.

16. Kita sudah selesai dengan C (Create), sekarang kita akan buat R (read). Pada file MobilController dan lengkapi code seperti gambar berikut.



```
File Edit Selection View Go Run Terminal Help MobilController.php - Dealer-KitaAja - Visual Studio Code

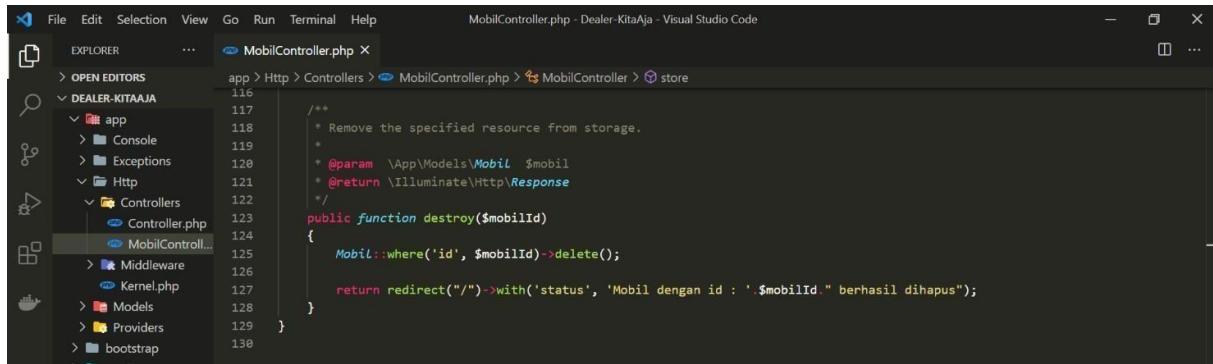
EXPLORER ... MobilController.php X
OPEN EDITORS app > Http > Controllers > MobilController.php > MobilController > store
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Mobil;
6 use Illuminate\Http\Request;
7
8 class MobilController extends Controller
9 {
10     /**
11      * Display a listing of the resource.
12      *
13      * @return \Illuminate\Http\Response
14      */
15     public function index()
16     {
17         $AllMobil = Mobil::all();
18
19         return view('welcome', ['AllMobil' => $AllMobil]);
20     }
21 }
```

17. Create dan Read sudah selesai, untuk menjalankan aplikasi ketikkan **php artisan serve** di terminal anda, setelah itu, ketikkan <http://127.0.0.1:8000>.

## B. Proses Update dan Delete

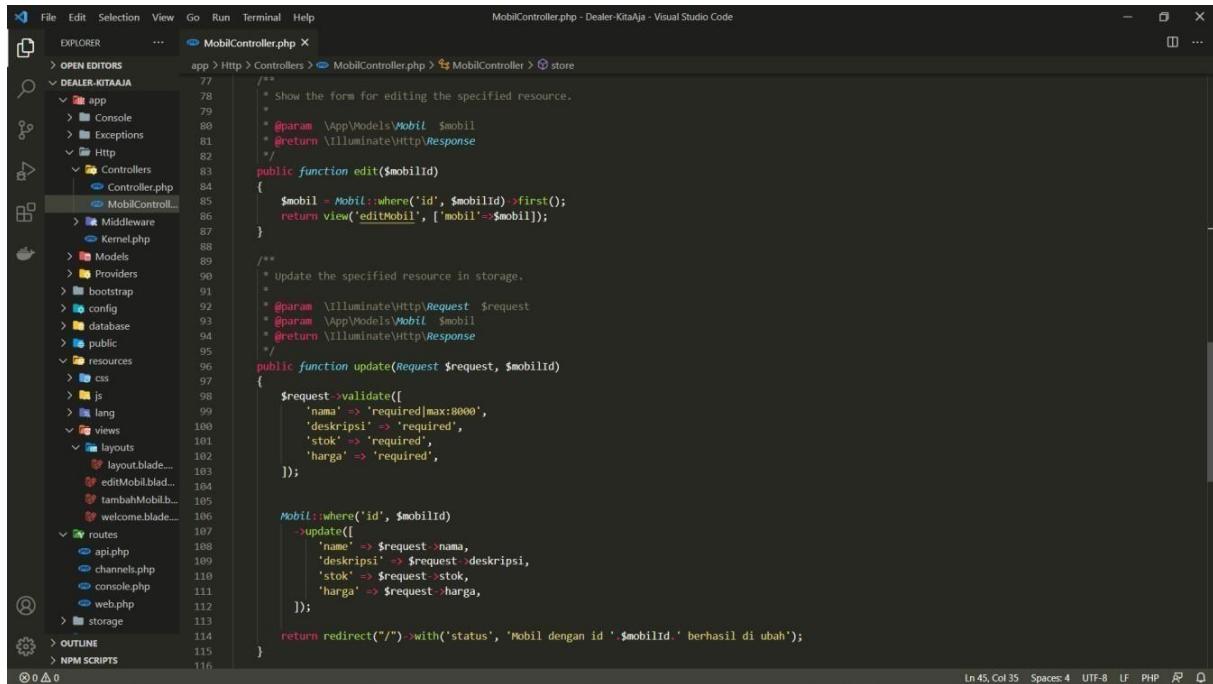
Setelah kita membuat Create dan Read, sekarang kita akan membuat Update dan Delete. Untuk itu ikuti langkah-langkah berikut ini.

1. Pada Create dan Read, sebenarnya kita sudah membuat tombol untuk delete dan edit, namun itu belum berfungsi sebagaimana mestinya. Untuk membuatnya berfungsi, buka file **MobilController** dan pada method destroy lengkapi code seperti gambar berikut.



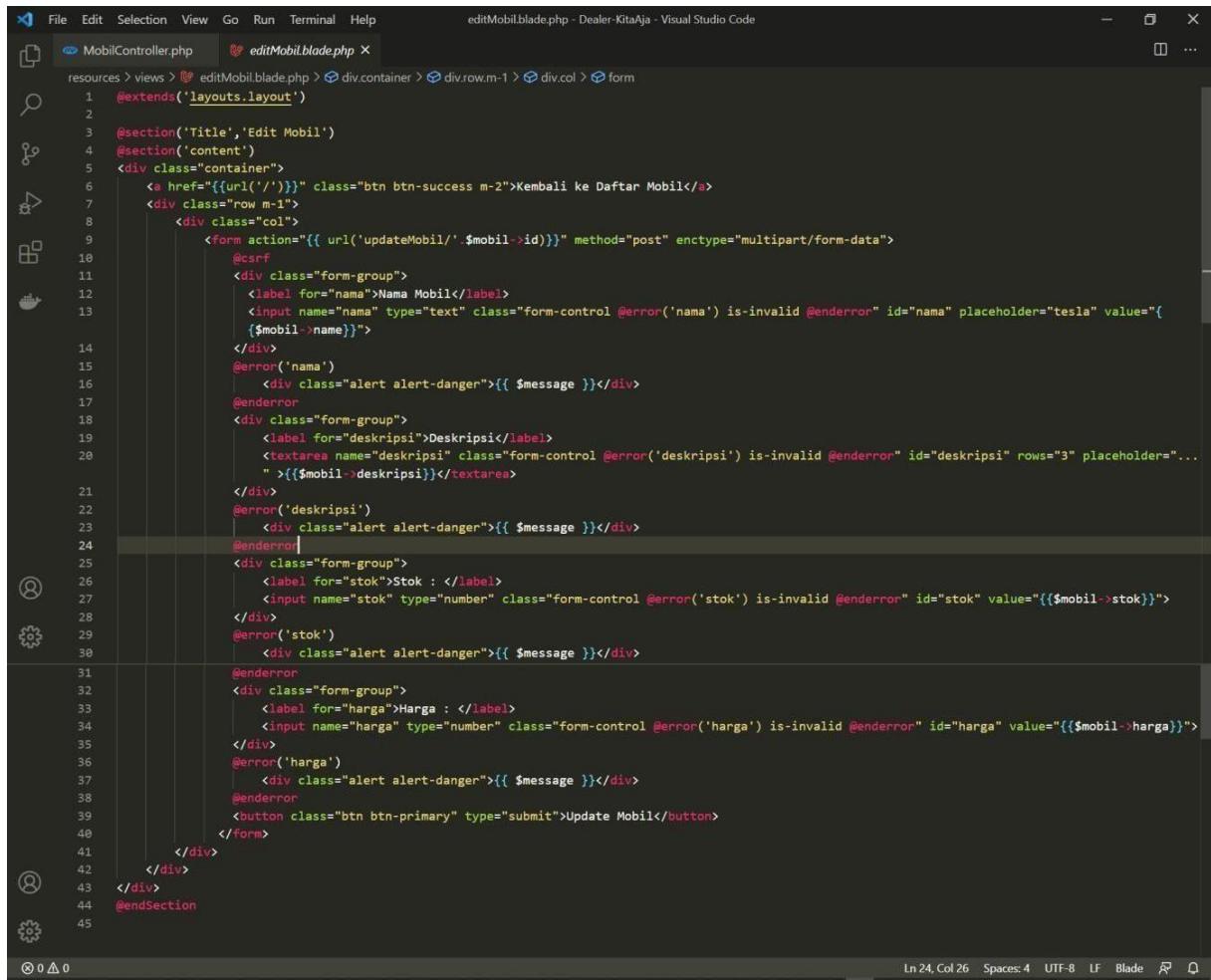
```
File Edit Selection View Go Run Terminal Help
MobilController.php - Dealer-KitaAja - Visual Studio Code
EXPLORER OPEN EDITORS
DEALER-KITAAJA app > Http > Controllers > MobilController.php > MobilController > store
116     /**
117      * Remove the specified resource from storage.
118      *
119      * @param \App\Models\Mobil $mobil
120      *
121      * @return \Illuminate\Http\Response
122      */
123     public function destroy($mobilId)
124     {
125         $mobil::where('id', $mobilId)->delete();
126
127         return redirect('/')->with('status', 'Mobil dengan id : '.$mobilId.' berhasil dihapus');
128     }
129 }
130 }
```

2. Delete telah selesai, kemudian kita akan memperbaiki Update. Pada file MobilController perbaiki method update dan edit hingga seperti gambar berikut ini.



```
File Edit Selection View Go Run Terminal Help
MobilController.php - Dealer-KitaAja - Visual Studio Code
EXPLORER OPEN EDITORS
DEALER-KITAAJA app > Http > Controllers > MobilController.php > MobilController > store
77     /**
78      * Show the form for editing the specified resource.
79      *
80      * @param \App\Models\Mobil $mobil
81      *
82      * @return \Illuminate\Http\Response
83      */
84     public function edit($mobilId)
85     {
86         $mobil = $mobil::where('id', $mobilId)->first();
87         return view('editMobil', ['mobil' => $mobil]);
88     }
89
90     /**
91      * Update the specified resource in storage.
92      *
93      * @param \Illuminate\Http\Request $request
94      * @param \App\Models\Mobil $mobil
95      *
96      * @return \Illuminate\Http\Response
97      */
98     public function update(Request $request, $mobilId)
99     {
100         $request->validate([
101             'nama' => 'required|max:8000',
102             'deskripsi' => 'required',
103             'stok' => 'required',
104             'harga' => 'required',
105         ]);
106
107         $mobil::where('id', $mobilId)
108             ->update([
109                 'name' => $request->nama,
110                 'deskripsi' => $request->deskripsi,
111                 'stok' => $request->stok,
112                 'harga' => $request->harga,
113             ]);
114
115         return redirect('/')->with('status', 'Mobil dengan id '.$mobilId.' berhasil di ubah');
116     }
117 }
```

3. Kita telah selesai di controller, sekarang kita beralih ke view. Di dalam folder **resources>view** buat sebuah file baru dengan nama **editMobil.blade.php** dan lengkapi kode berikut ini.



```
File Edit Selection View Go Run Terminal Help
MobilController.php editMobil.blade.php
resources > views > editMobil.blade.php > div.container > div.row.m-1 > div.col > form
1  @extends('layouts.layout')
2
3  @section('title','Edit Mobil')
4  @section('content')
5  <div class="container">
6      <a href="{{url('/')}}" class="btn btn-success m-2">Kembali ke Daftar Mobil</a>
7      <div class="row m-1">
8          <div class="col">
9              <form action="{{ url('updateMobil/'.$mobil->id) }}" method="post" enctype="multipart/form-data">
10                 @csrf
11                 <div class="form-group">
12                     <label for="nama">Nama Mobil</label>
13                     <input name="nama" type="text" class="form-control @error('nama') is-invalid @enderror" id="nama" placeholder="tesla" value="{{$mobil->name}}>
14                 </div>
15                 @error('nama')
16                     <div class="alert alert-danger">{{ $message }}</div>
17                 @enderror
18                 <div class="form-group">
19                     <label for="deskripsi">Deskripsi</label>
20                     <textarea name="deskripsi" class="form-control @error('deskripsi') is-invalid @enderror" id="deskripsi" rows="3" placeholder="...>{{($mobil->deskripsi)}}</textarea>
21                 </div>
22                 @error('deskripsi')
23                     <div class="alert alert-danger">{{ $message }}</div>
24                 @enderror
25                 <div class="form-group">
26                     <label for="stok">Stok : </label>
27                     <input name="stok" type="number" class="form-control @error('stok') is-invalid @enderror" id="stok" value="{{$mobil->stok}}>
28                 </div>
29                 @error('stok')
30                     <div class="alert alert-danger">{{ $message }}</div>
31                 @enderror
32                 <div class="form-group">
33                     <label for="harga">Harga : </label>
34                     <input name="harga" type="number" class="form-control @error('harga') is-invalid @enderror" id="harga" value="{{$mobil->harga}}>
35                 </div>
36                 @error('harga')
37                     <div class="alert alert-danger">{{ $message }}</div>
38                 @enderror
39                 <button class="btn btn-primary" type="submit">Update Mobil</button>
40             </form>
41         </div>
42     </div>
43 </div>
44 @endSection
45
```

Ln 24, Col 26 Spaces: 4 UTF-8 LF Blade ⚙

4. Update telah selesai, sekarang jalankan aplikasi Kembali dengan cara ketik **php artisan serve** di terminal, dan ketik <http://127.0.0.1:8000> di browser Anda.

## Tugas

Simpan hasil pekerjaan Anda dengan nama **w11s02\_NimLengkap.zip**. Pada folder archived tersebut terdapat **Observasi Praktikum sesi 02** yang Anda kerjakan.

**Selamat Belajar dan Bekerja** 😊