

**Tugas Pendahuluan Modul 13 – *IT* dan *RPL*
Perancangan dan Pemrograman Web – Ganjil 2025/2026
„Laravel Part 2 (Update dan Delete)“**

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara Individu.
2. TP ini bersifat WAJIB, tidak mengerjakan = PENGURANGAN POIN JURNAL / TES ASESMEN.
3. Hanya MENGUMPULKAN tetapi TIDAK MENGERJAKAN = PENGURANGAN POIN JURNAL / TES ASESMEN.
4. **Deadline** pengumpulan TP Modul 13 adalah **Senin, 8 Oktober 2025 pukul 06.00 WIB**.
5. TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.
6. DILARANG PLAGIAT (PLAGIAT = E).
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format PDF dengan ketentuan: TP_MOD_[XX]_NIM_NAMA.pdf

CP (WA):

- Febry (08152052546)

B. Soal Tugas Pendahuluan

Buatlah sebuah aplikasi To Do List / Task Management menggunakan Laravel 11/12 yang dapat melakukan operasi CRUD (Create, Read, Update, dan Delete) data-data task yang ada. Implementasikan dengan arsitektur MVC (Model, View, Controller), menggunakan Bootstrap 5 untuk desain tampilan yang menarik dan responsif dan menggunakan MySql sebagai databasenya.

Spesifikasi aplikasi:

- User dapat menambahkan tugas/task baru (create).
- User dapat melihat tugas/task yang pernah ia tambahkan (read).
- User dapat mengubah tugas/task (update).
- User dapat menghapus tugas/task (delete).

Persyaratan:

1. Database dan Tabel Task

- Buat sebuah database baru dengan nama `task_management` pada PhpMyAdmin atau sejenisnya.
- Buatlah tabel `tasks` yang memiliki attribute berikut:
 - `Id` (Primary Key, Auto Increment)
 - `Name` (String)
 - `Description` (text)
 - `Status` (Enum dari “pending” dan “completed”, dengan nilai defaultnya adalah “pending”).
 - `Created_At` dan `Updated_At` (timestamp)

2. Model Task

- Buat model `task` yang berfungsi untuk menghubungkan aplikasi dengan tabel `tasks` yang ada pada database.
- Pastikan untuk mendefinisikan nilai dari variabel `fillable` dengan attribute yang dapat diisi seperti `name`, `description`, dan `status`.

3. Controller Task Controller

- Buatlah sebuah controller bernama `TaskController` yang memiliki function berikut:
 - `Index`: fungsi yang mengambil semua task yang ada pada db dan mengirim data tersebut ke halaman utama.
 - `Store`: fungsi untuk menyimpan data task baru ke database.
 - `Edit`: fungsi untuk berpindah dan menampilkan form edit task berdasarkan task yang dipilih pada halaman utama.

- Update: fungsi untuk memperbaharui data task yang dipilih.

- Delete: fungsi untuk menghapus data task yang dipilih.

4. Routing (web.php)

- Gunakan route dengan method GET / untuk menampilkan daftar tugas dihalaman utama aplikasi.
- Gunakan route dengan method POST /tasks untuk menyimpan tugas baru yang dikirim dari form tambah task.
- Gunakan route GET /tasks/{id}/edit untuk menampilkan halaman yang berisi form edit.
- Gunakan route PUT /tasks/{id} untuk memperbaharui data task berdasarkan id yang ada pada url.
- Gunakan route DELTE /tasks/{id} untuk menghapus data task.

5. Tampilan (View)

- Buatlah sebuah tampilan utama (index.blade.php) yang berisi:
 - Form untuk menambahkan tugas baru yang terdapat kolom name dan description.
 - List daftar yang telah ditambahkan dengan informasi nama tugas, deskripsi tugas, status tugas, dan tombol untuk mengedit atau menghapus tugas.

Tampilan awal ketika belum ada task.

Selamat Datang, admin

[Logout](#)

To Do List Gwehj

Add New Task

Task Name

Description

[Add Task](#)

No task available

Tampilan awal ketika sudah ada task.

Selamat Datang, admin [Logout](#)

To Do List Gwehj

Add New Task

Task Name

Description

[Add Task](#)

Mencuci Baju Baju dicuci ehehe	Pending	Delete
Mengerjakan TP Mod 13 Menyelesaikan TP Mod 13 Sebelum hari senin	Pending	Delete

- Buatlah sebuah tampilan baru untuk menampilkan form edit yang berisi name, description, dan dropdown status (antara pending dan completed).

Edit Task

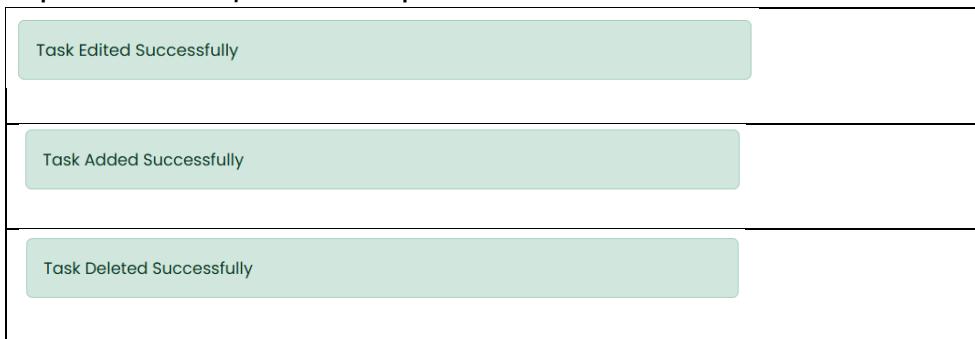
Task Name

Description

Status

[Cancel](#) [Edit Task](#)

- Tampilkan pesan sukses saat task berhasil ditambahkan, diperbaharui, dan dihapus.



- Gunakan Bootstrap 5 agar lebih mempercepat proses pengerjaan.

Langkah Pengerjaan

1. Persiapan Proyek

- Membuat proyek laravel baru:

```
composer create-project laravel/laravel  
[nama_project_kalian]
```

- Sesuaikan konfigurasi database pada file .env

2. Perisapan Database

- Buat sebuah database baru bernama task_management atau sejenisnya pada panel PhpMyAdmin (sesuaikan dengan nama yang ada pada file .env).
- Buat migration untuk tabel tasks sesuai attribute yang telah dispesifikan diatas.

```
php artisan make:migration [nama_file_migration]
```

atau

```
php artisan make:model Task -m (membuat model sekalian  
dengan migration)
```

- Jalankan migration.

```
php artisan migrate
```

3. Buat Model Task

- Buat model Task

```
php artisan make:model Task
```

- Definisikan attribute yang dapat diisi pada variabel fillable pada model Task (name, description, dan status).

4. Buat Controller

- Buat controller bernama TaskController

```
php artisan make:controller TaskController
```

- Tambahkan method berikut sesuai dengan deskripsi diatas:
 - Index()
 - Store(Request \$request)
 - Edit(\$id)
 - Update(Request \$request, \$id)
 - Delete(\$id)

5. Buat Routing

- Tambahkan route baru pada file web.php

6. Buat Tampilan

- Buat tampilan utama yang berisi form tambah task dan lisk task yang telah ditambahkan (index.blade.php).
- Buat tampilan edit yang berisi form edit task (edit.blade.php).

7. Menguji aplikasi

- Jalankan aplikasi.
- Screenshot tampilan ketika menguji fungsionalitasnya.

C. Soal Bonus [Optional] (sebaiknya dikerjakan)

- Implementasikan middleware untuk memastikan hanya pengguna yang telah terautentifikasi yang dapat melakukan fungsionalitas CRUD (cari middleware sanctum).

Hal yang harus diperhatikan

1. Struktur Aplikasi (Pengorganisiran dengan arsitektur MVC)
2. Menggunakan Eloquent untuk melakukan query ke DB.
3. Desain tampilan.
4. Validasi input (seperti tidak boleh kosong dan lain sebagainya).
5. Fungsionalitas.