

Laporan Contoh Program
CRUD LARAVEL

Dosen pengajar:
RUDHI WAHYUDI FEBRIANTO



Disusun Oleh:

Nama : Ade Hikmat Pauji Ridwan

Kelas : 222K

NPM : 22552011130

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS TEKNOLOGI BANDUNG

2024

BAB I

PENDAHULUAN

1.1. Latar Belakang

Laravel adalah salah satu framework PHP yang paling populer dan serbaguna yang digunakan dalam berbagai jenis aplikasi web. Dengan perubahan zaman yang membawa segala sesuatu menjadi global, teknologi juga mengalami perkembangan yang pesat, dan pengembangan web menjadi salah satu aspek yang sangat krusial dalam era digital ini.

Framework Laravel dikembangkan oleh Taylor Otwell pada tahun 2011 sebagai alternatif yang lebih elegan dan ekspresif untuk framework PHP yang ada pada saat itu. Laravel sangat dipengaruhi oleh sintaksis dan konsep dari berbagai framework lain seperti Ruby on Rails, ASP.NET MVC, dan Django, tetapi dengan penyederhanaan dan fleksibilitas yang lebih tinggi. Hal ini menjadikan Laravel sebagai framework yang mudah dipelajari dan dipahami. Sebagai framework yang berorientasi objek, Laravel dirancang tidak hanya untuk membantu pengembang dalam menulis kode yang efisien tetapi juga agar dapat dimengerti oleh manusia.

Dalam penggunaannya, Laravel menggunakan berbagai konsep dan fitur seperti Eloquent ORM (Object-Relational Mapping), Blade templating engine, routing, middleware, dan lain-lain. Konsep-konsep ini memungkinkan pengembang untuk membuat kode yang lebih terstruktur dan modular, sehingga memudahkan dalam pengembangan dan pemeliharaan aplikasi. Hingga saat ini, mayoritas aplikasi web modern menggunakan Laravel sebagai salah satu komponen utama dari arsitektur mereka, menunjukkan betapa pentingnya peran Laravel dalam perkembangan teknologi informasi global.

1.2. Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Mampu memahami CRUD di Laravel.
2. Mampu membuat program CRUD.

BAB II

DASAR TEORI

2.1 Pengertian CRUD di Laravel

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan operasi dasar dalam manipulasi data. Dalam konteks Laravel, CRUD digunakan untuk mengelola data dalam aplikasi web melalui model yang berinteraksi dengan database. Berikut adalah penjelasan teori dasar tentang CRUD di Laravel:

1. **Create (Membuat)**
Operasi Create digunakan untuk menambahkan data baru ke dalam database. Dalam Laravel, operasi ini biasanya dilakukan melalui form input yang dikirimkan oleh pengguna.
2. **Read (Membaca)**
Operasi Read digunakan untuk mengambil dan menampilkan data dari database. Dalam Laravel, operasi ini biasanya dilakukan dengan menggunakan Eloquent ORM untuk memudahkan pengambilan data.
3. **Update (Memperbarui)**
Operasi Update digunakan untuk memperbarui data yang sudah ada dalam database. Dalam Laravel, operasi ini biasanya melibatkan form edit yang menampilkan data saat ini dan memungkinkan pengguna untuk memperbaruinya.
4. **Delete (Menghapus)**
Operasi Delete digunakan untuk menghapus data dari database. Dalam Laravel, operasi ini biasanya dilakukan melalui permintaan yang dikirimkan dari pengguna, seperti klik tombol hapus.

BAB III METOLOGI PERCOBAAN

3.1 Alat dan Bahan

1. PC / Laptop
2. Visual Studio Code
3. Laravel Project

3.2 Preparation

1. **Install Composer** Laravel menggunakan Composer untuk manajemen dependensi. Pastikan Composer sudah terinstall di sistem Anda. Jika belum, Anda bisa mengunduhnya dari [situs resmi Composer](#).
2. **Buat Proyek Laravel Baru** Setelah Composer terinstall, jalankan perintah berikut di terminal Anda untuk membuat proyek Laravel baru:

```
composer create-project --prefer-dist laravel/laravel project-name
```

Gantilah project-name dengan nama proyek Anda.

3. **Selesai Instalasi** Setelah proses instalasi selesai, Anda dapat mulai menggunakan Laravel. Arahkan terminal ke direktori proyek baru Anda dan jalankan server pengembangan dengan perintah:

```
php artisan serve
```

Akses proyek Anda di browser melalui alamat <http://localhost:8000>.

3.3 Setting up Migration in Laravel

1. **Pastikan File .env Tersedia** Pastikan di dalam proyek Laravel Anda terdapat file .env. File ini adalah file konfigurasi proyek yang berisi berbagai variabel global statis yang nantinya dapat digunakan dalam aplikasi. Jika file .env tidak tersedia, Anda bisa membuatnya dengan menyalin file .env.example:

```
cp .env.example .env
```

2. **Konfigurasi Database** Laravel menyediakan kunci env untuk konfigurasi database. Buka file .env Anda dan cari bagian yang terkait dengan pengaturan database. Anda bisa mengubah nilai sesuai dengan konfigurasi database Anda:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=username_database
DB_PASSWORD=password_database
```

Gantilah nama_database, username_database, dan password_database dengan informasi yang sesuai dengan pengaturan database Anda.

3. **Jalankan Migrasi** Setelah konfigurasi database disesuaikan, Anda dapat menjalankan migrasi untuk membuat tabel-tabel di database. Jalankan perintah berikut di terminal Anda:

```
php artisan migrate
```

```
Macbooks-MacBook-Pro:crud-basic macbook$ php artisan migrate
INFO  Preparing database.
Creating migration table ..... 42.42ms DONE
INFO  Running migrations.
0001_01_01_000000_create_users_table ..... 40.38ms DONE
0001_01_01_000001_create_cache_table ..... 17.32ms DONE
0001_01_01_000002_create_jobs_table ..... 36.64ms DONE
```

Perintah ini akan menjalankan semua file migrasi yang terdapat di folder database/migrations dan membuat tabel-tabel yang diperlukan di database Anda.

3.3 Membuat CRUD

1. Create controller

```
Macbooks-MacBook-Pro:crud-basic macbook$ php artisan make:controller UserController --resource
INFO  Controller [app/Http/Controllers/UserController.php] created successfully.
Macbooks-MacBook-Pro:crud-basic macbook$
```

2. Register router

```
crud-basic > routes >  web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\UserController;
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 Route::resource('users', UserController::class);
11
```

3. Create View

```
make: view Create a new view
● Macbooks-MacBook-Pro:crud-basic macbook$ php artisan make:view users
  INFO View [resources/views/users.blade.php] created successfully.
● Macbooks-MacBook-Pro:crud-basic macbook$ php artisan make:view create
  INFO View [resources/views/create.blade.php] created successfully.
● Macbooks-MacBook-Pro:crud-basic macbook$ php artisan make:view edit
  INFO View [resources/views/edit.blade.php] created successfully.
● Macbooks-MacBook-Pro:crud-basic macbook$ php artisan make:view show
  INFO View [resources/views/show.blade.php] created successfully.
○ Macbooks-MacBook-Pro:crud-basic macbook$
```

4. Write controller code

BAB IV HASIL DAN ANALISA

4.1 Hasil Percobaan

Ini adalah hasil dari percobaan membuat crud <https://github.com/adehikmatfr/utb-semseter4/tree/main/Web2/Laravel/crud-basic>

BAB V KESIMPULAN

Implementasi CRUD (Create, Read, Update, Delete) di Laravel menunjukkan cara dasar untuk mengelola data dalam aplikasi web. Dengan menggunakan Laravel, pengembang dapat membuat, membaca, memperbarui, dan menghapus data pengguna dengan efisien. Proses ini memanfaatkan berbagai fitur Laravel, seperti migrasi, model, controller, dan blade templating, untuk menciptakan aplikasi yang terstruktur dan mudah dipelihara.

Poin Penting:

1. **Konfigurasi Proyek:**
 - **File .env:** File konfigurasi ini digunakan untuk menyimpan variabel global statis, seperti pengaturan database, yang diperlukan oleh aplikasi.
 - **Konfigurasi Database:** Menyesuaikan pengaturan database di file .env memastikan bahwa aplikasi Laravel dapat terhubung dengan database yang benar untuk penyimpanan data.
2. **Pembuatan Migrasi dan Model:**
 - **Migrasi:** Migrasi digunakan untuk mendefinisikan struktur tabel di database. Dengan menggunakan migrasi, pengembang dapat mengelola perubahan pada struktur database secara versioned dan terstruktur.
 - **Model:** Model Laravel mewakili tabel database dan menyediakan cara interaksi dengan data di tabel tersebut. Model User, misalnya, mengatur logika bisnis terkait pengguna.
3. **Controller dan Routing:**
 - **Controller:** Controller mengatur logika CRUD. Dengan memisahkan logika aplikasi dari tampilan, controller membuat kode lebih bersih dan mudah untuk dikelola.
 - **Routing:** Route resource digunakan untuk mendefinisikan URL endpoint untuk operasi CRUD. Ini memudahkan pengaturan navigasi dalam aplikasi.
4. **Tampilan (Views):**
 - **Blade Templating:** Blade adalah mesin templating yang disediakan oleh Laravel untuk membuat tampilan yang dinamis dan dapat digunakan kembali. Dengan blade, pengembang dapat dengan mudah membuat dan mengelola antarmuka pengguna untuk operasi CRUD.
5. **Pengelolaan Kesalahan:**
 - **Validasi:** Laravel menyediakan cara yang mudah untuk memvalidasi input pengguna, memastikan bahwa data yang diterima aplikasi valid sebelum diproses atau disimpan.

- **Pengecualian:** Penanganan pengecualian penting untuk memastikan bahwa aplikasi dapat menangani kesalahan dengan baik dan memberikan umpan balik yang berguna kepada pengguna.

6. **Pengelolaan Sumber Daya:**

- **Penggunaan Eloquent:** Eloquent ORM memudahkan pengelolaan data dengan menyediakan metode sederhana dan intuitif untuk berinteraksi dengan database. Ini memastikan bahwa pengelolaan sumber daya dilakukan dengan efisien.