

PEMROGRAMAN WEB LANJUT

JOBSHEET 3



Dosen Pengampu:

M. Hasyim Ratsanjani, S.Kom., M.Kom.

Disusun Oleh:

Nafi'ul Alam Dary Vega (2241720048)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2023

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

Link Github Project PWL_2024 : https://github.com/NafiulAlamDaryVega/PWL_2024

Link Github Project POS : <https://github.com/NafiulAlamDaryVega/POS>

SOAL PRAKTIKUM

1. Pada **Praktikum 1 - Tahap 5**, apakah fungsi dari **APP_KEY** pada *file setting .env* Laravel?

Jawab:

Pada *file setting .env* Laravel, **APP_KEY** adalah variabel yang menyimpan kunci rahasia yang digunakan untuk enkripsi data, seperti sesi pengguna dan pengiriman email, serta untuk menghasilkan tanda tangan hash yang diperlukan oleh beberapa fitur keamanan Laravel.

2. Pada **Praktikum 1**, bagaimana kita men-*generate* nilai untuk **APP_KEY**?

Jawab:

Untuk meng-*generate* nilai untuk **APP_KEY**, dapat menggunakan perintah **php artisan key:generate**. Ini akan menghasilkan nilai yang unik untuk **APP_KEY** dan secara otomatis memperbarui nilai tersebut dalam *file .env*.

3. Pada **Praktikum 2.1 - Tahap 1**, secara *default* Laravel memiliki berapa *file migrasi*? dan untuk apa saja *file migrasi* tersebut?

Jawab:

- **2014_10_12_000000_create_users_table.php**
File migrasi ini bertanggung jawab untuk membuat tabel **users** di database. Tabel **users** sering digunakan untuk menyimpan informasi pengguna, seperti nama pengguna, alamat email, dan sandi terenkripsi.
- **2014_10_12_100000_create_password_reset_tokens_table.php**
File migrasi ini digunakan untuk membuat tabel **password_reset_tokens** di database. Tabel ini digunakan untuk menyimpan token reset sandi yang diperlukan saat pengguna lupa sandi mereka. Ini membantu dalam mengelola proses reset sandi pengguna dengan aman.
- **2019_08_19_000000_create_failed_jobs_table**
File migrasi ini bertanggung jawab untuk membuat tabel **failed_jobs** di database. Tabel ini digunakan oleh sistem antrian (queue) untuk mencatat pekerjaan yang gagal dieksekusi. Ini memungkinkan Anda untuk memeriksa dan menangani pekerjaan yang gagal dengan lebih baik.
- **2019_12_14_000001_create_personal_access_tokens_table.php**
File migrasi ini digunakan untuk membuat tabel **personal_access_tokens** di database. Tabel ini diperlukan ketika kita menggunakan sistem autentikasi token untuk memberikan akses ke API Laravel. Tabel ini menyimpan informasi tentang token akses pribadi yang dikeluarkan kepada pengguna untuk mengakses API kita.

4. Secara *default*, *file migrasi* terdapat kode **\$table->timestamps();**, apa tujuan/output dari fungsi tersebut?

Jawab:

Kode **\$table->timestamps();**, itu digunakan untuk menambahkan dua kolom ke dalam tabel yang sedang dibuat, yaitu **created_at** dan **updated_at**.

- **created_at**: Kolom ini secara otomatis akan menyimpan tanggal dan waktu ketika baris data pertama kali dimasukkan ke dalam tabel.
- **updated_at**: Kolom ini akan secara otomatis diperbarui setiap kali ada perubahan pada baris data yang sudah ada dalam tabel.

5. Pada *File Migrasi*, terdapat fungsi **\$table->id();** Tipe data apa yang dihasilkan dari fungsi tersebut?

Jawab:

Fungsi **\$table->id();** dalam *file migrasi* Laravel digunakan untuk menambahkan kolom yang bertipe data **BIGINT UNSIGNED** dengan sifat **AUTO_INCREMENT PRIMARY KEY** ke dalam tabel yang sedang dibuat.

6. Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');` ?

Jawab:

Perbedaannya terletak pada cara penamaan kolom primary key yang dihasilkan:

- `$table->id();`: Dengan menggunakan ini, Laravel akan secara otomatis menambahkan kolom primary key ke tabel dengan nama `id`. Misalnya, jika menggunakan `$table->id();` dalam migrasi untuk membuat tabel `m_level`, maka Laravel akan menambahkan kolom `id` sebagai primary key.
- `$table->id('level_id');`: Dengan menggunakan ini, secara eksplisit memberikan nama kolom primary key, yaitu `level_id`. Jadi, jika menggunakan `$table->id('level_id');` dalam migrasi untuk membuat tabel `m_level`, Laravel akan menambahkan kolom `level_id` sebagai primary key.

7. Pada migration, Fungsi `->unique()` digunakan untuk apa?

Jawab:

Fungsi `->unique()` dalam sebuah migration digunakan untuk menetapkan kolom tertentu sebagai kolom unik di dalam tabel database. Ini berarti bahwa setiap nilai di kolom tersebut harus unik, tidak boleh ada duplikat di antara entri-entri dalam tabel.

8. Pada **Praktikum 2.2 - Tahap 2**, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$table->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$table->id('level_id')` ?

Jawab:

Kolom `level_id` pada tabel `m_user` menggunakan `$table->unsignedBigInteger('level_id')` karena kolom `level_id` tersebut adalah sebuah foreign key yang merujuk ke primary key dari tabel `m_level`. Oleh karena itu, tipe data yang digunakan adalah `unsignedBigInteger` untuk menyesuaikan dengan tipe data primary key yang menggunakan `BIGINT UNSIGNED`. Sementara itu, pada tabel `m_level`, kolom `level_id` diatur sebagai primary key dan akan secara otomatis menggunakan tipe data `BIGINT UNSIGNED`, sehingga tidak perlu menentukan tipe data secara eksplisit saat menggunakan `$table->id('level_id')`. Penggunaan `$table->id('level_id')` dalam hal ini memungkinkan untuk menetapkan nama kolom primary key secara eksplisit.

9. Pada **Praktikum 3 - Tahap 6**, apa tujuan dari Class `Hash`? dan apa maksud dari kode program `Hash::make('1234');`?

Jawab:

Class `Hash` dalam Laravel digunakan untuk melakukan hashing atau pengacakan terhadap data. Tujuan utamanya adalah untuk mengamankan data, khususnya data sensitif seperti password. Kode program `Hash::make('1234');` pada tahap tersebut berfungsi untuk membuat `hash` dari `string '1234'`. Proses hashing akan mengonversi string tersebut menjadi representasi yang sulit dibaca dan tidak terbalik (one-way), sehingga tidak mungkin untuk mengembalikan nilai aslinya dari hasil hashing.

10. Pada **Praktikum 4 - Tahap 3/5/7**, pada *query builder* terdapat tanda tanya (`?`), apa kegunaan dari tanda tanya (`?`) tersebut?

Jawab:

Tanda tanya (`?`) dalam query builder Laravel digunakan untuk menandai parameter dalam sebuah query. Saat Anda menggunakan tanda tanya dalam query builder dan menyediakan parameter menggunakan array, nilai dari setiap elemen array tersebut akan digunakan sebagai nilai yang akan diganti di tempat tanda tanya yang sesuai dalam query.

11. Pada **Praktikum 6 - Tahap 3**, apa tujuan penulisan kode `protected $table = 'm_user';` dan `protected $primaryKey = 'user_id';` ?

Jawab:

- **\$table**: Mendefinisikan nama tabel yang digunakan oleh model ini. Dalam hal ini, dengan mendefinisikan **\$table** menjadi **m_user**, model **UserModel** akan mengikat dirinya ke tabel dengan nama **m_user**. Ini memungkinkan Laravel untuk mengetahui secara eksplisit bahwa model ini berhubungan dengan tabel tertentu dalam database.
- **\$primaryKey**: Mendefinisikan primary key dari tabel yang digunakan. Dalam contoh tersebut, primary key dari tabel **m_user** adalah kolom **user_id**. Dengan mendefinisikan **\$primaryKey** menjadi **user_id**, Laravel akan menggunakan kolom ini sebagai primary key saat melakukan operasi CRUD pada model tersebut.

12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (*DB Façade / Query Builder / Eloquent ORM*) ? jelaskan

Jawab:

Dalam melakukan operasi CRUD ke database lebih mudah menggunakan *Eloquent ORM*.

Alasan kenapa lebih mudah menggunakan *Eloquent ORM*:

- Eloquent ORM adalah fitur Laravel yang menyediakan cara untuk berinteraksi dengan database menggunakan objek PHP, mengubah setiap baris dalam tabel database menjadi objek model.
- Lebih mudah digunakan dan lebih ekspresif daripada DB Façade dan Query Builder karena menyediakan abstraksi yang lebih tinggi.
- Memungkinkan pengguna untuk melakukan operasi CRUD tanpa menulis query SQL secara langsung, yang membuat kode lebih mudah dipahami dan dipelihara.
- Cocok untuk pengembangan aplikasi yang memanfaatkan prinsip ORM dan ingin meningkatkan produktivitas dengan menggunakan fitur-fitur seperti relasi antar objek.