

Pemrograman Web Lanjut
Jobsheet 3 - Migration, Seeder, DB Facade, Query Builder, dan
Eloquent ORM

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Nama : Athallah Ayudya Paramesti
NIM : 2341760061
Program Studi : D-IV Sistem Informasi Bisnis

Jurusan Teknologi Informasi
Politeknik Negeri Malang
2025

A. Pengaturan Database

Database atau basis data menjadi komponen penting dalam membangun sistem. Hal ini dikarenakan database menjadi tempat untuk menyimpan data-data transaksi yang ada pada sistem. Koneksi ke database perlu kita atur agar sesuai dengan database yang kita gunakan.

Praktikum 1 - pengaturan databases:

1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL_POS



2. Buka aplikasi VSCode dan buka folder project PWL_POS yang sudah kita buat
3. Copy file .env.example menjadi .env
4. Buka file .env, dan pastikan konfigurasi APP_KEY bernilai. Jika belum bernilai silahkan kalian generate menggunakan php artisan.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:LkSZloF1E0543KdQhwp8R9dah6jxQp04gx69r00ftQw=
APP_DEBUG=true
APP_URL=http://localhost
```

5. Edit file .env dan sesuaikan dengan database yang telah dibuat

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=PWL_POS
DB_USERNAME=root
DB_PASSWORD=
```

B. Migration

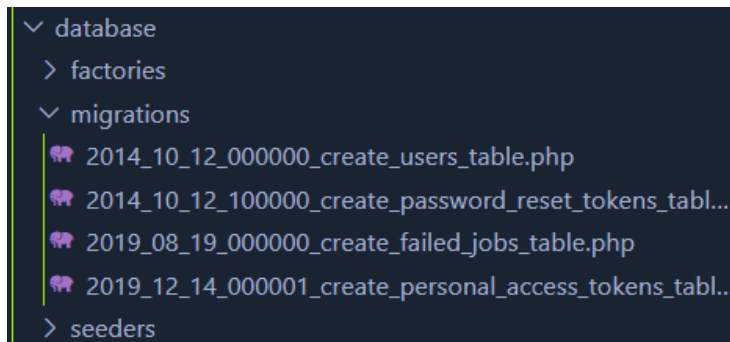
Migration pada Laravel merupakan sebuah fitur yang dapat membantu kita mengelola database secara efisien dengan menggunakan kode program. Migration membantu kita dalam membuat (create), mengubah (edit), dan menghapus (delete) struktur tabel dan kolom pada database yang sudah kita buat dengan cepat dan mudah.

Dengan Migration, kita juga dapat melakukan perubahan pada struktur database tanpa harus menghapus data yang ada.

Salah satu keunggulan menggunakan migration adalah mempermudah proses instalasi aplikasi kita, Ketika aplikasi yang kita buat akan diimplementasikan di server/komputer lain.

Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

1. Ini merupakan default dari laravel



2. Buat file migrasi untuk table m_level dengan perintah berikut:

```
λ php artisan make:migration create_m_level_table --create=m_level
```

```
Week_3 > Jobsheet3 > database > migrations > 2025_03_05_162357_create_m_level_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('m_level', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('m_level');
26     }
27 };
28
```

3. modifikasi sesuai desain database yang sudah ada

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_level', function (Blueprint $table) {
            $table->id('level_id');
            $table->string('level_kode', 10)->unique();
            $table->string('level_nama', 100);
            $table->timestamps();
        });
    }
}
```

4. jalankan perintah berikut untuk melakukan migrasi

```
λ php artisan migrate
```

```
INFO Preparing database.
Creating migration table ..... 3,175ms DONE
INFO Running migrations.
2014_10_12_000000_create_users_table ..... 1,486ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 165ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 665ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 556ms DONE
2025_03_05_162357_create_m_level_table ..... 287ms DONE
```

5. Kemudian kita cek di phpMyAdmin apakah table sudah ter-generate atau belum

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> m_level	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> password_reset_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
6 tables	Sum	5	InnoDB	utf8mb4_0900_ai_ci	96.0 KiB	0 B

6. Ok, table sudah dibuat di database
7. Buat table database dengan migration untuk table m_kategori yang sama-sama tidak memiliki foreign key

```
λ php artisan make:migration create_m_kategori_table
```

```

Week_3 > Jobsheet3 > database > migrations > 2025_03_05_204822_create_m_kategori_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('m_kategori', function (Blueprint $table) {
15             $table->id('kategori_id');
16             $table->string('kategori_kode', 10)->unique();
17             $table->string('kategori_nama', 100);
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      */
25     public function down(): void
26     {
27         Schema::dropIfExists('m_kategori');
28     }
29 };

```

```

λ php artisan migrate

```

```

INFO Running migrations.
2025_03_05_204822_create_m_kategori_table ..... 6,739ms DONE

```

Table
<input type="checkbox"/> failed_jobs
<input type="checkbox"/> migrations
<input type="checkbox"/> m_kategori
<input type="checkbox"/> m_level
<input type="checkbox"/> password_reset_tokens
<input type="checkbox"/> personal_access_tokens
<input type="checkbox"/> users

Praktikum 2.2 - Pembuatan file migrasi dengan relasi

1. Buka terminal VSCode kalian, dan buat file migrasi untuk table m_user

```
λ php artisan make:migration create_m_user_table --table=m_user
```

2. Buka file migrasi untuk table m_user, dan modifikasi seperti berikut

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_user', function (Blueprint $table) {
            $table->id('user_id');
            $table->unsignedBigInteger('level_id')->index(); //indexing untuk ForeignKey
            $table->string('username', 20)->unique(); //untuk memastikan tidak ada username yang sama
            $table->string('nama', 100);
            $table->string('password');
            $table->timestamps();

            //Mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
            $table->foreign('level_id')->references('level_id')->on('m_level');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('m_user');
    }
};
```

3. Simpan kode program Langkah 2, dan jalankan perintah php artisan migrate.
Amati apa yang terjadi pada database.

Table
<input type="checkbox"/> failed_jobs
<input type="checkbox"/> migrations
<input type="checkbox"/> m_kategori
<input type="checkbox"/> m_level
<input type="checkbox"/> m_user
<input type="checkbox"/> password_reset_tokens
<input type="checkbox"/> personal_access_tokens
<input type="checkbox"/> users

4. Buat tabel database dengan migration untuk tabel-tabel yang memiliki foreign key

m_barang
t_penjualan
t_stok
t_penjualan_detail

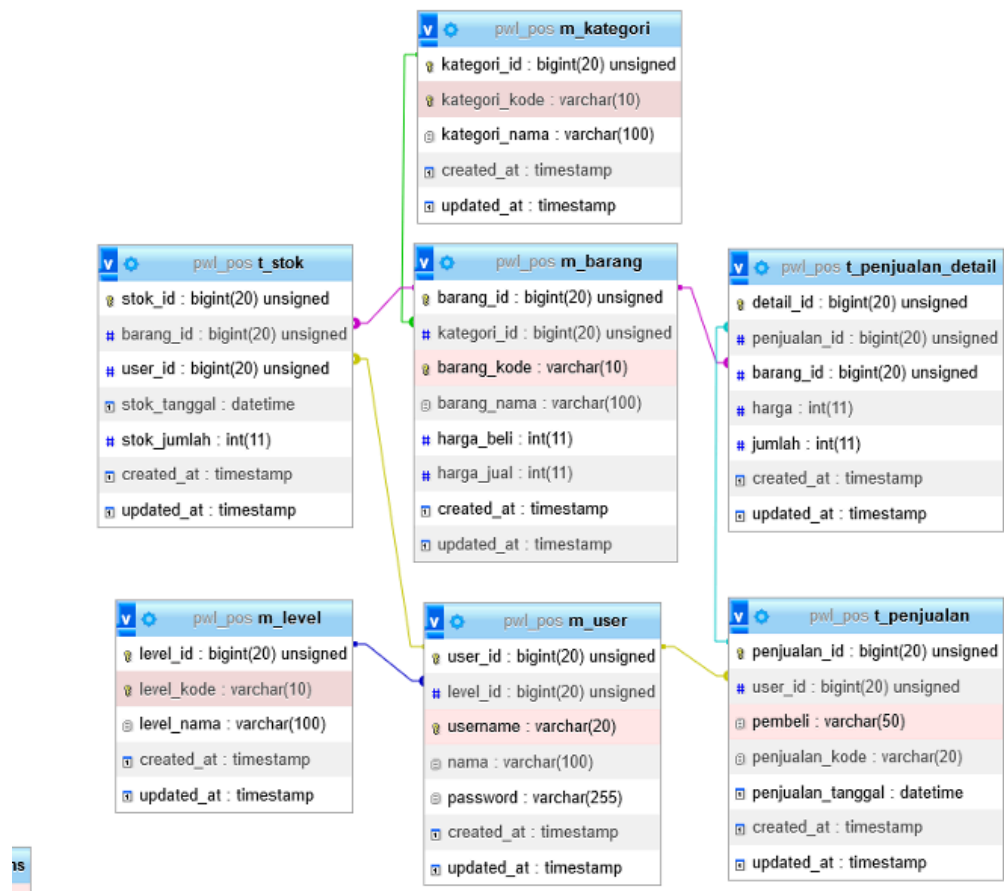
```

2025_03_05_212533_create_m_barang_table.php  U
2025_03_05_212710_create_t_stok_table.php    U
2025_03_05_220125_create_t_penjualan_table.php U
2025_03_05_224308_create_t_penjualan_detail_table.php U

```

isi file sesuai isi tabel, lalu migrasi

5. Jika semua file migrasi sudah dibuat dan dijalankan maka bisa kita lihat tampilan designer pada phpMyAdmin seperti berikut



C. Seeder

Seeder merupakan sebuah fitur yang memungkinkan kita untuk mengisi database kita dengan data awal atau data dummy yang telah ditentukan. Seeder memungkinkan kita untuk membuat data awal yang sama untuk setiap penggunaan dalam pembangunan aplikasi. Umumnya, data yang sering dibuat seeder adalah data pengguna karena data tersebut akan digunakan saat aplikasi pertama kali dijalankan dan membutuhkan aksi login.

Dalam proses pengembangan suatu aplikasi, seringkali kita membutuhkan data awal tiruan atau dummy data untuk memudahkan pengujian dan pengembangan aplikasi kita. Sehingga fitur seeder bisa kita pakai dalam membuat sebuah aplikasi web.

Praktikum 3 - Membuat file seeder

1. Buat file seeder untuk table m_level dengan mengetikkan perintah

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModel;
use Illuminate\Database\Seeder;

class LevelSeeder extends Seeder{
    public function run(): void{
        //
    }
}
```

2. Selanjutnya, untuk memasukkan data awal, kita modifikasi file tersebut di dalam function run()

```
use Illuminate\Support\Facades\DB;
```

```
class LevelSeeder extends Seeder{
    public function run(): void{
        $data = [
            ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Admi
            ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Mana
            ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staf
        ];
        DB::table('m_level')->insert($data);
    }
}
```


3. Selanjutnya, kita jalankan file seeder untuk table m_level pada terminal

```
λ php artisan db:seed --class=LevelSeeder
```

INFO Seeding database.

4. Ketika seeder berhasil dijalankan maka akan tampil data pada table m_level

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL

5. Sekarang kita buat file seeder untuk table m_user yang me-refer ke table m_level

```
λ php artisan make:seeder UserSeeder
```

INFO Seeder [C:\laragon\www\laravel\PWL_2025\Week_3\Jobsheet3\database\seeders\UserS] successfully.

6. Modifikasi file class UserSeeder seperti berikut

```

public function run(): void
{
    $data = [
        [
            'user_id' => 1,
            'level_id' => 1,
            'username' => 'admin',
            'nama' => 'Administrator',
            'password' => Hash::make('12345'), //untuk enkripsi password
        ],
        [
            'user_id' => 2,
            'level_id' => 2,
            'username' => 'manager',
            'nama' => 'Manager',
            'password' => Hash::make('12345'), //untuk enkripsi password
        ],
        [
            'user_id' => 3,
            'level_id' => 3,
            'username' => 'staff',
            'nama' => 'Staff/Kasir',
            'password' => Hash::make('12345'), //untuk enkripsi password
        ],
    ];
    DB::table('m_user')->insert($data);
}

```

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```
λ php artisan db:seed --class=UserSeeder
```

INFO Seeding database.

8. Perhatikan hasil seeder pada table m_user

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$diZXcDqSz4CluPPSTy59/nE3hsMOwxRSrPhxwmErWL...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$SH.MKJveHbZkZiYQlpeMOJQ7dVEtC7eCCJCFvkWhEE...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$84QbySK//9FZQ8h01bkhzOvrKakWv2YZO/CU.1ir0fR...

9. Ok, data seeder berhasil dimasukkan ke database.
10. Sekarang coba kalian masukkan data seeder untuk table yang lain, dengan ketentuan seperti berikut

No	Nama Tabel	Jumlah Data	Keterangan
1	m_kategori	5	5 kategori barang
2	m_barang	10	10 barang yang berbeda
3	t_stok	10	Stok untuk 10 barang
4	t_penjualan	10	10 transaksi penjualan
5	t_penjualan_detail	30	3 barang untuk setiap transaksi penjualan

```
λ php artisan make:seeder KategoriSeeder
```

```
DB::table('m_kategori')->insert([
    ['kategori_id' => 1, 'kategori_kode' => 'FOOD', 'kategori_nama' => 'Makanan'],
    ['kategori_id' => 2, 'kategori_kode' => 'DRINK', 'kategori_nama' => 'Minuman'],
    ['kategori_id' => 3, 'kategori_kode' => 'BEAUTY', 'kategori_nama' => 'Kecantikan'],
    ['kategori_id' => 4, 'kategori_kode' => 'HOME', 'kategori_nama' => 'Peralatan Rumah'],
    ['kategori_id' => 5, 'kategori_kode' => 'BABY', 'kategori_nama' => 'Perlengkapan Bayi'],
]);
```

```
λ php artisan db:seed --class=KategoriSeeder
```

<div>←T→</div>				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	DRINK	Minuman	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	BEAUTY	Kecantikan	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	HOME	Peralatan Rumah	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	BABY	Perlengkapan Bayi	NULL	NULL

```
λ php artisan make:seeder BarangSeeder
```

```
DB::table('m_barang')->insert([
    ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'BRG001', 'barang_nama' => 'Nasi Goreng', 'harga_beli' => 10000, 'harga_jual' => 15000],
    ['barang_id' => 2, 'kategori_id' => 1, 'barang_kode' => 'BRG002', 'barang_nama' => 'Mie Goreng', 'harga_beli' => 9000, 'harga_jual' => 14000],
    ['barang_id' => 3, 'kategori_id' => 2, 'barang_kode' => 'BRG003', 'barang_nama' => 'Es Teh', 'harga_beli' => 3000, 'harga_jual' => 5000],
    ['barang_id' => 4, 'kategori_id' => 2, 'barang_kode' => 'BRG004', 'barang_nama' => 'Jus Jeruk', 'harga_beli' => 5000, 'harga_jual' => 8000],
    ['barang_id' => 5, 'kategori_id' => 3, 'barang_kode' => 'BRG005', 'barang_nama' => 'Lipstik', 'harga_beli' => 25000, 'harga_jual' => 40000],
    ['barang_id' => 6, 'kategori_id' => 3, 'barang_kode' => 'BRG006', 'barang_nama' => 'Face Wash', 'harga_beli' => 15000, 'harga_jual' => 25000],
    ['barang_id' => 7, 'kategori_id' => 4, 'barang_kode' => 'BRG007', 'barang_nama' => 'Sapu Lidi', 'harga_beli' => 10000, 'harga_jual' => 20000],
    ['barang_id' => 8, 'kategori_id' => 4, 'barang_kode' => 'BRG008', 'barang_nama' => 'Pel Lantai', 'harga_beli' => 20000, 'harga_jual' => 35000],
    ['barang_id' => 9, 'kategori_id' => 5, 'barang_kode' => 'BRG009', 'barang_nama' => 'Diapers', 'harga_beli' => 50000, 'harga_jual' => 75000],
    ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'BRG010', 'barang_nama' => 'Bedak Bayi', 'harga_beli' => 10000, 'harga_jual' => 15000],
]);
```

```
λ php artisan db:seed --class=BarangSeeder
```

				barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	1	BRG001	Nasi Goreng	10000	15000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	1	BRG002	Mie Goreng	9000	14000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	2	BRG003	Es Teh	3000	5000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	2	BRG004	Jus Jeruk	5000	8000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	3	BRG005	Lipstik	25000	40000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	3	BRG006	Face Wash	15000	25000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	7	4	BRG007	Sapu Lidi	10000	20000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	8	4	BRG008	Pel Lantai	20000	35000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	9	5	BRG009	Diapers	50000	75000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	10	5	BRG010	Bedak Bayi	10000	15000	NULL	NULL

```
λ php artisan make:seeder StokSeeder
```

```
DB::table('t_stok')->insert([
    ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 50],
    ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 40],
    ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 30],
    ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 25],
    ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 20],
    ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 35],
    ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 15],
    ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 10],
    ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 20],
    ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 30],
]);
```

```
λ php artisan db:seed --class=StokSeeder
```

<div><div></div><div></div><div></div></div>				stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	1	1	1	2025-03-11 21:00:43	50	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	2	2	2	2025-03-11 21:00:43	40	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	3	3	1	2025-03-11 21:00:43	30	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	4	4	3	2025-03-11 21:00:43	25	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	5	5	2	2025-03-11 21:00:43	20	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	6	6	1	2025-03-11 21:00:43	35	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	7	7	3	2025-03-11 21:00:43	15	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	8	8	2	2025-03-11 21:00:43	10	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	9	9	3	2025-03-11 21:00:43	20	NULL	NULL
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	10	10	1	2025-03-11 21:00:43	30	NULL	NULL

```
λ php artisan db:seed --class=PenjualanSeeder
```

```
DB::table('t_penjualan')->insert([
    ['penjualan_id' => 1, 'user_id' => 1, 'pembeli' => 'Arjo', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 10],
    ['penjualan_id' => 2, 'user_id' => 2, 'pembeli' => 'Atha', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 15],
    ['penjualan_id' => 3, 'user_id' => 1, 'pembeli' => 'Andi', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 20],
    ['penjualan_id' => 4, 'user_id' => 3, 'pembeli' => 'Rina', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 25],
    ['penjualan_id' => 5, 'user_id' => 2, 'pembeli' => 'Miri', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 30],
    ['penjualan_id' => 6, 'user_id' => 1, 'pembeli' => 'Dewi', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 35],
    ['penjualan_id' => 7, 'user_id' => 3, 'pembeli' => 'Agus', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 40],
    ['penjualan_id' => 8, 'user_id' => 2, 'pembeli' => 'Lina', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 45],
    ['penjualan_id' => 9, 'user_id' => 3, 'pembeli' => 'Dayu', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 50],
    ['penjualan_id' => 10, 'user_id' => 1, 'pembeli' => 'Eka', 'penjualan_tanggal' => now(), 'penjualan_jumlah' => 55],
]);
```

```
λ php artisan db:seed --class=PenjualanSeeder
```

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Arjo	TRX001	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	Atha	TRX002	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	Andi	TRX003	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	3	Rina	TRX004	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	Miri	TRX005	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	1	Dewi	TRX006	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	3	Agus	TRX007	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	2	Lina	TRX008	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	3	Dayu	TRX009	2025-03-11 20:57:43	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	1	Eka	TRX010	2025-03-11 20:57:43	NULL	NULL

```
λ php artisan make:seeder PenjualanDetailSeeder
```

```
DB::table('t_penjualan_detail')->insert([
    ['detail_id' => 1, 'penjualan_id' => 1, 'barang_id' => 1, 'harga' =
    ['detail_id' => 2, 'penjualan_id' => 1, 'barang_id' => 2, 'harga' =
    ['detail_id' => 3, 'penjualan_id' => 1, 'barang_id' => 3, 'harga' =
    ['detail_id' => 4, 'penjualan_id' => 2, 'barang_id' => 4, 'harga' =
    ['detail_id' => 5, 'penjualan_id' => 2, 'barang_id' => 5, 'harga' =
    ['detail_id' => 6, 'penjualan_id' => 2, 'barang_id' => 6, 'harga' =
    ['detail_id' => 7, 'penjualan_id' => 3, 'barang_id' => 7, 'harga' =
    ['detail_id' => 8, 'penjualan_id' => 3, 'barang_id' => 8, 'harga' =
    ['detail_id' => 9, 'penjualan_id' => 3, 'barang_id' => 9, 'harga' =
    ['detail_id' => 10, 'penjualan_id' => 4, 'barang_id' => 10, 'harga' =
    ['detail_id' => 11, 'penjualan_id' => 4, 'barang_id' => 1, 'harga'
```

```
λ php artisan db:seed --class=PenjualanDetailSeeder
```

	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	1	15000	2	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	2	14000	1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	3	5000	3	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	2	4	8000	2	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	5	40000	1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	2	6	25000	1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	3	7	20000	1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	3	8	35000	2	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	3	9	75000	1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	4	10	15000	1	NULL	NULL

D. DB Facade

Praktikum 4

1. Kita buat controller dahulu untuk mengelola data pada table m_level

```
λ php artisan make:controller LevelController
```

2. Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php

```
Week_3 > Jobsheet3 > routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\LevelController;
4  use Illuminate\Support\Facades\Route;
5  use Monolog\Level;
6
7  Route::get('/', function () {
8      return view('welcome');
9  });
10
11 Route::get('/level', [LevelController::class, 'index']);
```

3. Selanjutnya, kita modifikasi file LevelController untuk menambahkan 1 data ke table m_level

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class LevelController extends Controller
{
    public function index(){
        DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
        return 'Insert data baru berhasil';
    }
}
```

4. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level dan amati apa yang terjadi pada table m_level di database, screenshot perubahan yang ada pada table m_level

Insert data baru berhasil

				level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	CUS	Pelanggan	2025-03-11 21:30:33	NULL

5. Selanjutnya, kita modifikasi lagi file LevelController untuk meng-update data di table m_level seperti berikut

```
public function index(){
    //DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', '']);
    //return 'Insert data baru berhasil';

    $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
    return 'Update data berhasil, jumlah data yang diupdate: ' . $row . ' baris';
}
```

6. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level lagi dan amati apa yang terjadi pada table m_level di database, screenshot perubahan yang ada pada table m_level

Update data berhasil, jumlah data yang diupdate: 1 baris

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	CUS	Customer	2025-03-11 21:30:33	NULL

7. Kita coba modifikasi lagi file LevelController untuk melakukan proses hapus data

```
public function index(){
    //DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', '']);
    //return 'Insert data baru berhasil';

    // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
    // return 'Update data berhasil, jumlah data yang diupdate: ' . $row . ' baris';

    $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
    return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
}
```

Delete data berhasil. Jumlah data yang dihapus: 1 baris

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL

8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m_level. Kita modifikasi file LevelController seperti berikut

```
$data = DB::select('select * from m_level');
return view('level', ['data' => $data]);
```

9. Coba kita perhatikan kode yang diberi tanda kotak merah, berhubung kode tersebut memanggil view('level'), maka kita buat file view pada VSCode di PWL_POS/resources/view/level.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Data Level Pengguna</title>
</head>
<body>
    <h1>Data Level Pengguna</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <th>ID</th>
            <th>Kode Level</th>
            <th>Nama Level</th>
        </tr>
        @foreach ($data as $d)
            <tr>
                <th>{{ $d->level_id }}</th>
                <th>{{ $d->level_kode }}</th>
                <th>{{ $d->level_nama }}</th>
            </tr>
        @endforeach
    </body>
</html>
```

10. Jalankan di web browser

Data Level Pengguna

ID	Kode Level	Nama Level
1	ADM	Administrator
2	MNG	Manager
3	STF	Staff/Kasir

E. Query Builder

Query builder adalah fitur yang disediakan Laravel untuk melakukan proses CRUD (create, retrieve/read, update, delete) pada database. Pada query builder perintah SQL ini diakses menggunakan method. Jadi, kita tidak menulis perintah SQL secara langsung, melainkan cukup memanggil method-method yang ada di query builder.

Perintah pertama yang dilakukan pada query builder adalah menentukan nama table yang akan dilakukan operasi CRUD. Kemudian baru disusul method yang ingin digunakan sesuai dengan peruntukannya.

Praktikum 5 - Implementasi Query Builder

1. Kita buat controller dahulu untuk mengelola data pada table m_kategori

```
λ php artisan make:controller KategoriController
```

2. Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php

```
Route::get('/kategori', [KategoriController::class, 'index']);
```

3. Selanjutnya, kita modifikasi file KategoriController untuk menambahkan 1 data ke table m_kategori

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class KategoriController extends Controller
{
    public function index(){
        $data = [
            'kategori_kode' => 'SNK',
            'kategori_nama' => 'Snack/Makanan Ringan',
            'created_at' => now()
        ];
        DB::table('m_kategori')->insert($data);
        return 'Insert data baru berhasil';
    }
}
```

4. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori dan amati apa yang terjadi pada table m_kategori di database, screenshot perubahan yang ada pada table m_kategori

Insert data baru berhasil

				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	DRINK	Minuman	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	BEAUTY	Kecantikan	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	HOME	Peralatan Rumah	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	BABY	Perlengkapan Bayi	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	SNK	Snack/Makanan Ringan	2025-03-11 23:04:13	NULL

5. Selanjutnya, kita modifikasi lagi file KategoriController untuk meng-update data di table m_kategori seperti berikut

```
$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
```

- Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori lagi dan amati apa yang terjadi pada table m_kategori di database, screenshot perubahan yang ada pada table m_kategori

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	DRINK	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	BEAUTY	Kecantikan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	HOME	Peralatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	BABY	Perlengkapan Bayi	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Snack/Makanan Ringan	2025-03-11 23:04:13	NULL

Update data berhasil. Jumlah data yang diupdate: 1 baris

- Kita coba modifikasi lagi file KategoriController untuk melakukan proses hapus data

```
$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
```

Delete data berhasil. Jumlah data yang dihapus: 1 baris

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	DRINK	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	BEAUTY	Kecantikan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	HOME	Peralatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	BABY	Perlengkapan Bayi	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Snack/Makanan Ringan	2025-03-11 23:04:13	NULL

- Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m_kategori. Kita modifikasi file KategoriController seperti berikut :

```
$data = DB::table('m_kategori')->get();
return view('kategori', ['data' => $data]);
```

- Coba kita perhatikan kode yang diberi tanda kotak merah, berhubung kode tersebut memanggil view('kategori'), maka kita buat file view pada VSCode di PWL_POS/resources/view/kategori.blade.php

```

<!DOCTYPE html>
<html>
<head>
  <title>Data Kategori Barang</title>
</head>
<body>
  <h1>Data Kategori Barang</h1>
  <table border="1" cellpadding="2" cellspacing="0">
    <tr>
      <th>ID</th>
      <th>Kode Kategori</th>
      <th>Nama Kategori</th>
    </tr>
    @foreach ($data as $d)
      <tr>
        <th>{{ $d->kategori_id }}</th>
        <th>{{ $d->kategori_kode }}</th>
        <th>{{ $d->kategori_nama }}</th>
      </tr>
    @endforeach
  </table>
</body>
</html>

```

10. Jalankan di browser

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	FOOD	Makanan
2	DRINK	Minuman
3	BEAUTY	Kecantikan
4	HOME	Peralatan Rumah
5	BABY	Perlengkapan Bayi

F. Eloquent ORM

Eloquent ORM adalah fitur bawaan dari laravel. Eloquent ORM adalah cara mengakses database dimana setiap baris tabel dianggap sebagai sebuah object. Kata ORM sendiri merupakan singkatan dari Object-relational mapping, yakni suatu teknik programming untuk mengkonversi data ke dalam bentuk object.

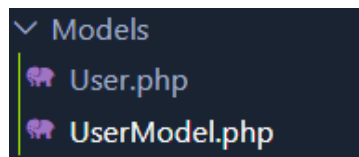
Untuk bisa melakukan operasi CRUD (create, read/retrieve, update, delete), kita harus membuat sebuah model sesuai dengan target tabel yang ingin digunakan. Jadi, dalam 1 model, merepresentasikan 1 tabel database.

Praktikum 6 - Implementasi Eloquent ORM

1. Kita buat file model untuk tabel m_user dengan mengetikkan perintah

```
λ php artisan make:model UserModel
```

- Setelah berhasil generate model, terdapat 2 file pada folder model yaitu file User.php bawaan dari laravel dan file UserModel.php yang telah kita buat. Kali ini kita akan menggunakan file UserModel.php



- Kita buka file UserModel.php dan modifikasi seperti berikut

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
}
```

- Kita modifikasi route web.php untuk mencoba routing ke controller UserController

```
Route::get('/user', [UserController::class, 'index']);
```

- Sekarang, kita buat file controller UserController dan memodifikasinya seperti berikut

```
λ php artisan make:controller UserController
```

```
<?php

namespace App\Http\Controllers;

use App\Models\UserModel;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index(){
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}
```

- Kemudian kita buat view user.blade.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Data User</title>
</head>
<body>
    <h1>Data User</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <th>ID</th>
            <th>Username</th>
            <th>Nama</th>
            <th>ID Level Pengguna</th>
        </tr>
        @foreach ($data as $d)
            <tr>
                <td>{{ $d->user_id }}</td>
                <td>{{ $d->Username }}</td>
                <td>{{ $d->nama }}</td>
                <td>{{ $d->level_id }}</td>
            </tr>
        @endforeach
    </body>
</html>

```

7. Jalankan di browser, catat dan laporkan apa yang terjadi

Data User

ID	Username	Nama	ID Level Pengguna
1		Administrator	1
2		Manager	2
3		Staff/Kasir	3

8. Setelah itu, kita modifikasi lagi file UserController

```

<?php

namespace App\Http\Controllers;

use App\Models\UserModel;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class UserController extends Controller
{
    public function index(){
        $data = [
            'username' => 'customer-1',
            'nama' => 'Pelanggan',
            'password' => Hash::make('12345'),
            'level_id' => 4
        ];
        UserModel::insert($data); //tambah data ke tabel m_user

        //coba akses model UserModel
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}

```

9. Jalankan di browser, amati dan laporkan apa yang terjadi

Data User

ID	Username	Nama	ID Level Pengguna
1		Administrator	1
2		Manager	2
3		Staff/Kasir	3
6		Pelanggan	4

10. Kita modifikasi lagi file UserController menjadi seperti berikut

```
class UserController extends Controller
{
    public function index(){
        $data = [
            'nama' => 'Pelanggan Pertama',
        ];
        // UserModel::insert($data); //tambah data ke tabel m_user
        UserModel::where('username', 'customer-1')->update($data);
        //coba akses model UserModel
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}
```

11. Jalankan di browser

Data User

ID	Username	Nama	ID Level Pengguna
1		Administrator	1
2		Manager	2
3		Staff/Kasir	3
6		Pelanggan Pertama	4

G. Penutup

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

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

Jawab:

APP_KEY digunakan untuk **enkripsi data dalam Laravel**, seperti sesi pengguna, token, dan password yang dienkripsi menggunakan hashing.

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

Jawab:

menggunakan perintah php artisan key:generate

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

Jawab:

4 file yaitu, create_users_table.php, create_password_reset_tokens_table.php, create_failed_jobs_table.php, dan create_personal_access_tokens_table.php

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

Jawab:

Untuk secara otomatis menambahkan 2 kolom yang berfungsi untuk menyimpan waktu kapan data dibuat dan menyimpan waktu kapan data diperbarui terakhir kali

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

Jawab:

menghasilkan kolom primary key dengan tipe data BIGINT (unsigned big integer) dan auto-increment.

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

Jawab:

`$table->id();` → Secara default, Laravel akan membuat primary key dengan nama id.

`$table->id('level_id');` → Primary key tetap berupa BIGINT (unsigned big integer) dan auto-increment, tetapi namanya diubah menjadi level_id.

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

Jawab:

untuk memastikan nilai dalam kolom tidak boleh duplikat

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:

Pada `m_level`, `$table->id('level_id')`; karena kolom ini adalah primary key, sehingga bersifat auto-increment.

Pada `m_user`, `$table->unsignedBigInteger('level_id')`; karena kolom ini adalah foreign key yang merujuk ke `level_id` pada tabel `m_level`.

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

Jawab:

Class `Hash` digunakan untuk melakukan hashing pada data sensitif, seperti password.

Kode `Hash::make('1234')`; berarti password "1234" akan diubah menjadi hash terenkripsi, sehingga lebih aman saat disimpan di database

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

Jawab:

digunakan untuk binding parameter dalam query SQL agar lebih aman dari serangan SQL Injection

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

Jawab:

`protected $table = 'm_user'`; → Memberitahu Eloquent bahwa model ini menggunakan tabel `m_user`.

`protected $primaryKey = 'user_id'`; → Mengatur primary key tabel ini agar menggunakan kolom `user_id`, bukan `id` (default Laravel).

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

Jawab:

Eloquent ORM lebih mudah digunakan untuk CRUD karena lebih singkat dan berbasis OOP.