



Nama : Nimas Septiandini

No : 17

NIM : 2341760087

Kelas : SIB-2B

Mata Kuliah : Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 6 (enam)

Pertemuan ke- : 1 (satu)

JOBSHEET 03

MIGRATION, SEEDER, DB FAÇADE, QUERY BUILDER, dan ELOQUENT ORM

A. PENGATURAN DATABASE

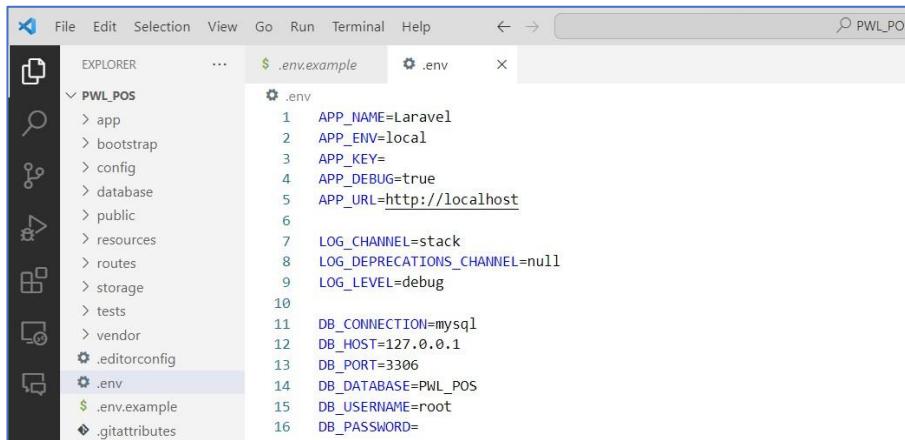
Praktikum 1 - pengaturan database:

1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama [PWL_POS](#)

The screenshot shows the phpMyAdmin interface for a MySQL server named 'localhost'. The main menu bar includes 'Databases', 'SQL', 'Status', 'User accounts', 'Export', and 'Import'. On the left, there's a sidebar with a tree view of databases. The main panel is titled 'Databases' and contains a 'Create database' button. Below it, two input fields are shown: 'PWL_POS' in the first and 'utf8mb4_unicode_ci' in the second. A 'Create' button is located to the right of the second field.



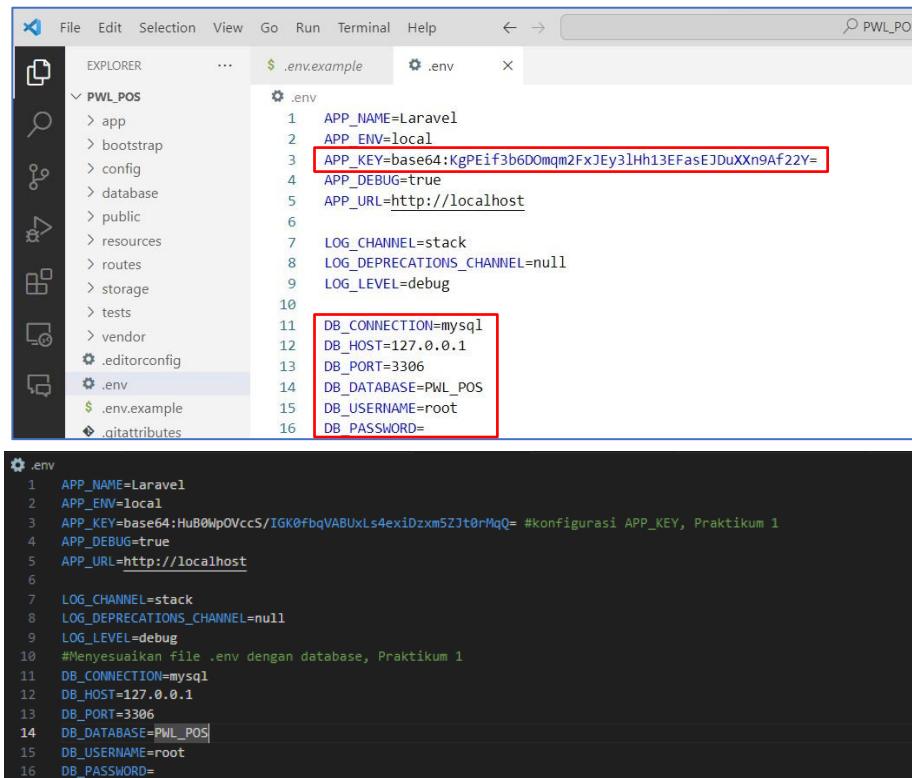
2. Buka aplikasi VSCode dan buka folder project [PWL_POS](#) yang sudah kita buat
3. Copy file [.env.example](#) menjadi [.env](#)
4. Buka file [.env](#), dan pastikan konfigurasi **APP_KEY** bernilai. Jika belum bernilai silahkan kalian generate menggunakan [php artisan](#).



```
File Edit Selection View Go Run Terminal Help ⌘ ⌘ PWL_POS .env.example .env .env

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=PWL_POS
15 DB_USERNAME=root
16 DB_PASSWORD=
```

5. Edit file [.env](#) dan sesuaikan dengan database yang telah dibuat



```
File Edit Selection View Go Run Terminal Help ⌘ ⌘ PWL_POS .env.example .env .env

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:KgPEif3b6D0mqm2FxJy3lHh13EFasEJDUXXn9Af22Y=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=PWL_POS
15 DB_USERNAME=root
16 DB_PASSWORD=
```



```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:HuB0MpOVcS/IGK0fbqVABUxLs4exiDzxm5Zjt0rMqQ= #konfigurasi APP_KEY, Praktikum 1
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10 #Menyesuaikan file .env dengan database, Praktikum 1
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=PWL_POS
15 DB_USERNAME=root
16 DB_PASSWORD=
```

6. Laporkan hasil Praktikum-1 ini dan *commit* perubahan pada *git*.
GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



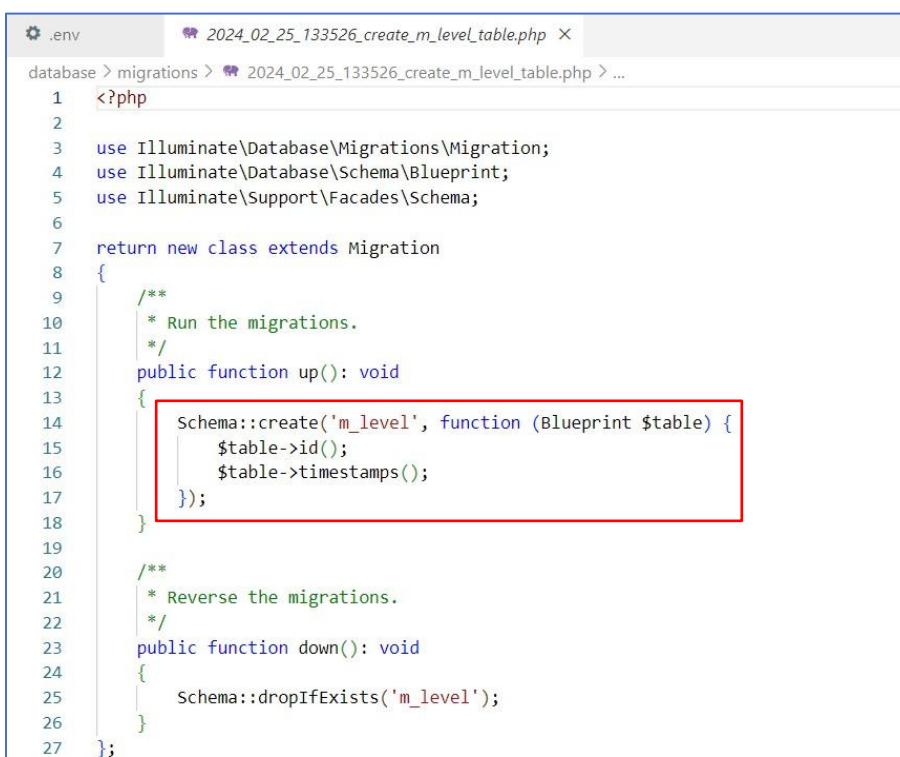
B. MIGRATION

1. Kita buat file migrasi untuk table `m_level` dengan perintah

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

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan
make:migration create_m_level_table --create=m_level

[INFO] Migration [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL
_POS\database\migrations\2025_02_142108_create_m_level_table.php] created successfully.
```



```
.env 2024_02_25_133526_create_m_level_table.php ×

database > migrations > 2024_02_25_133526_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 };
```



2. Kita perhatikan bagian yang di kotak merah, bagian tersebut yang akan kita modifikasi sesuai desain database yang sudah ada

```
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('level_id');
16            $table->string('level_kode', 10)->unique();
17            $table->string('level_nama', 100);
18            $table->timestamps();
19        });
20    }
21
22    /**
23     * Reverse the migrations.
24     */
25    public function down(): void
26    {
27        Schema::dropIfExists('m_level');
28    }
29};
```

```
public function up(): void
{
    // Modifikasi schema, praktikum 2
    Schema::create('m_level', function (Blueprint $table) {
        $table->id('level_id');
        $table->string('level_kode', 10)->unique();
        $table->string('level_nama', 100);
        $table->timestamps();
    });
}
```

3. Simpan kode pada tahapan 4 tersebut, kemudian jalankan perintah ini pada terminal VSCode untuk melakukan migrasi



Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

```
php artisan migrate
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\_laragon\www\PWL_POS> php artisan migrate

[INFO] Preparing database.

Creating migration table ..... 12ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 16ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 6ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 42ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE
2024_02_25_133526 create_m_level_table ..... 13ms DONE

PS D:\_laragon\www\PWL_POS>
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan migrate

[INFO] Preparing database.

Creating migration table ..... 62ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 102ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 21ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 66ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 85ms DONE
2025_03_02_142108_create_m_level_table ..... 66ms DONE
```



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

The screenshot shows two tables of database information:

Table	Action	Rows
failed_jobs	Browse Structure Search Insert Empty Drop	0
migrations	Browse Structure Search Insert Empty Drop	5
m_level	Browse Structure Search Insert Empty Drop	0
password_reset_tokens	Browse Structure Search Insert Empty Drop	0
personal_access_tokens	Browse Structure Search Insert Empty Drop	0
users	Browse Structure Search Insert Empty Drop	0

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
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

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

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_P
OS> php artisan make:migration create_m_kategori_table --create=m_kategori
```

```
[INFO] Migration [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2
025\Minggu3\PWL_POS\database\migrations\2025_03_02_144734_create_m_kategori_ta
ble.php] created successfully.
```

```
public function up(): void
{
    Schema::create('m_kategori', function (Blueprint $table) {
        $table->id('kategori_id');
        $table->string('kategori_kode', 10)->unique();
        $table->string('kategori_nama', 100);
        $table->timestamps();
    });
}
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\P
WL_POS> php artisan migrate
```

```
[INFO] Running migrations.
```

```
2025_03_02_144734_create_m_kategori_table ..... 83ms DONE
```



Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_kategori	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	112.0 KiB	0 B
7 tables	Sum					

7. Laporkan hasil Praktikum-2.1 ini dan *commit* perubahan pada *git*.
GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



Praktikum 2.2 - Pembuatan file migrasi dengan relasi

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

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

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

```
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('m_user', function (Blueprint $table) {
15             $table->id('user_id');
16             $table->unsignedBigInteger('level_id')->index(); // indexing untuk ForeignKey
17             $table->string('username', 20)->unique(); // unique untuk memastikan tidak ada username yang sama
18             $table->string('nama', 100);
19             $table->string('password');
20             $table->timestamps();
21
22             // Mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
23             $table->foreign('level_id')->references('level_id')->on('m_level');
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('m_user');
33     }
34 }
```

```
database > migrations > 2025_03_03_012209_create_m_user_table.php
8
9 /**
10
11     public function up(): void
12     {
13         Schema::create('m_user', function (Blueprint $table) {
14             $table->id('user_id');
15             $table->unsignedBigInteger('level_id')->index(); //indexing untuk foreignkey
16             $table->string('username', 20)->unique(); //unique memastikan tidak ada username yang sama
17             $table->string('nama', 100);
18             $table->string('password');
19             $table->timestamps();
20
21             //Mendefinisikan Foreign key pada kolom level_id mengacu pada kolom level_id di tabel m_level
22             $table->foreign('level_id')->references('level_id')->on('m_level');
23         });
24     }
25
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('m_user');
33     }
34 }
```



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

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan migrate
INFO  Running migrations.
2025_03_03_012209 create m_user table ..... 314ms DONE
```

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_kategori	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_user	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
8 tables	Sum	7	InnoDB	utf8mb4_0900_ai_ci	144.0 KiB	0 B

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

<u>m_barang</u>
<u>t_penjualan</u>
<u>t_stok</u>
<u>t_penjualan_detail</u>

- Tabel m_barang

```
database > migrations > 2025_03_03_033740_create_m_barang_table.php
 7   return new class extends Migration
 8   {
 9     /**
10      * Run the migrations.
11     */
12     public function up(): void
13     {
14       // tabel barang
15       Schema::create('m_barang', function (Blueprint $table) {
16         $table->id('barang_id');
17         $table->unsignedBigInteger('kategori_id')->index(); //indexing foreign key
18         $table->string('barang.kode', 10)->unique();
19         $table->string('barang.nama', 100);
20         $table->integer('harga_beli');
21         $table->integer('harga_jual');
22         $table->timestamps();
23
24         $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori');
25       });
26     }
27
28     /**
29      * Reverse the migrations.
30     */
31     public function down(): void
32     {
33       Schema::dropIfExists('m_barang');
34     }
35   };
```



Table	Action	Rows	Type
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB
migrations	Browse Structure Search Insert Empty Drop	9	InnoDB
m_barang	Browse Structure Search Insert Empty Drop	0	InnoDB
m_kategori	Browse Structure Search Insert Empty Drop	0	InnoDB
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB
m_user	Browse Structure Search Insert Empty Drop	0	InnoDB
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB
users	Browse Structure Search Insert Empty Drop	0	InnoDB
9 tables		Sum	9 InnoDB

- Tabel t_penjualan

```
public function up(): void
{
    Schema::create('t_penjualan', function (Blueprint $table) {
        $table->id('penjualan_id');
        $table->unsignedBigInteger('user_id')->index(); //index untuk foreign key
        $table->string('pembeli', 20);
        $table->string('penjualan_kode', 20);
        $table->string('penjualan_tanggal');
        $table->timestamps();

        //mendefinisikan foreign key pada kolom level_id mengacu pada kolom user_id di tabel m_user
        $table->foreign('user_id')->references('user_id')->on('m_user');
    });
}

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

```
INFO: Running migrations.
2025_03_03_042049_create_t_penjualan_table ..... 149ms DONE
```



Table	Action	Rows	Type
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB
migrations	Browse Structure Search Insert Empty Drop	10	InnoDB
m_barang	Browse Structure Search Insert Empty Drop	0	InnoDB
m_kategori	Browse Structure Search Insert Empty Drop	0	InnoDB
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB
m_user	Browse Structure Search Insert Empty Drop	0	InnoDB
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB
t_penjualan	Browse Structure Search Insert Empty Drop	0	InnoDB
users	Browse Structure Search Insert Empty Drop	0	InnoDB
10 tables	Sum		10 InnoDB

- Tabel t_stok

```
public function up(): void
{
    Schema::create('t_stok', function (Blueprint $table) {
        $table->id('stok_id');
        $table->unsignedBigInteger('barang_id')->index();
        $table->unsignedBigInteger('user_id')->index();
        $table->datetime('stok_tanggal');
        $table->integer('stok_jumlah');
        $table->timestamps();

        //foreign key
        $table->foreign('barang_id')->references('barang_id')->on('m_barang');
        $table->foreign('user_id')->references('user_id')->on('m_user');
    });
}

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

```
INFO Running migrations.

2025_03_03_042922_create_t_stok_table ..... 281ms DONE
```

t_stok	Browse Structure Search Insert Empty Drop	0	InnoDB
------------------------	---	---	--------

- Tabel_t_penjualan_detail



```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('t_penjualan_detail', function (Blueprint $table) {
            $table->bigIncrements('detail_id');
            $table->unsignedBigInteger('penjualan_id')->index(); //indexing foreignkey
            $table->unsignedBigInteger('barang_id')->index(); //indexing foreignkey
            $table->integer('harga');
            $table->integer('jumlah');
            $table->timestamps(); //otomatisasi kolom created_at dan updated_at

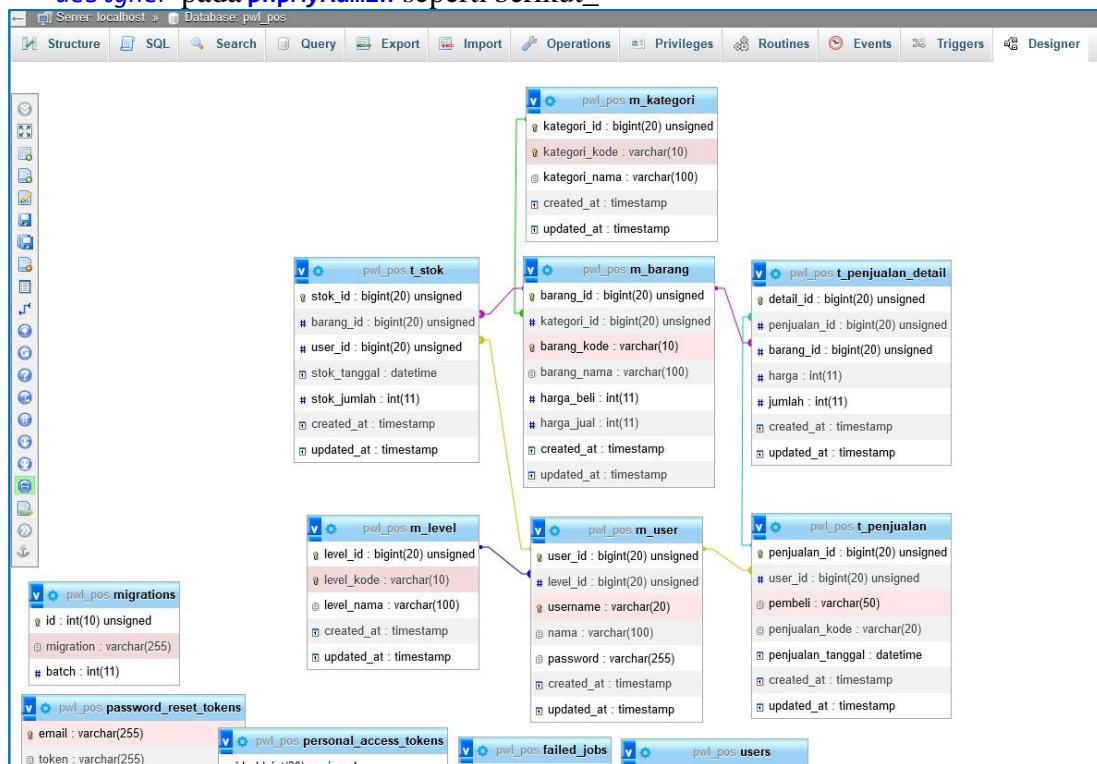
            //foreign key
            $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan');
            $table->foreign('barang_id')->references('barang_id')->on('m_barang');
        });
    }

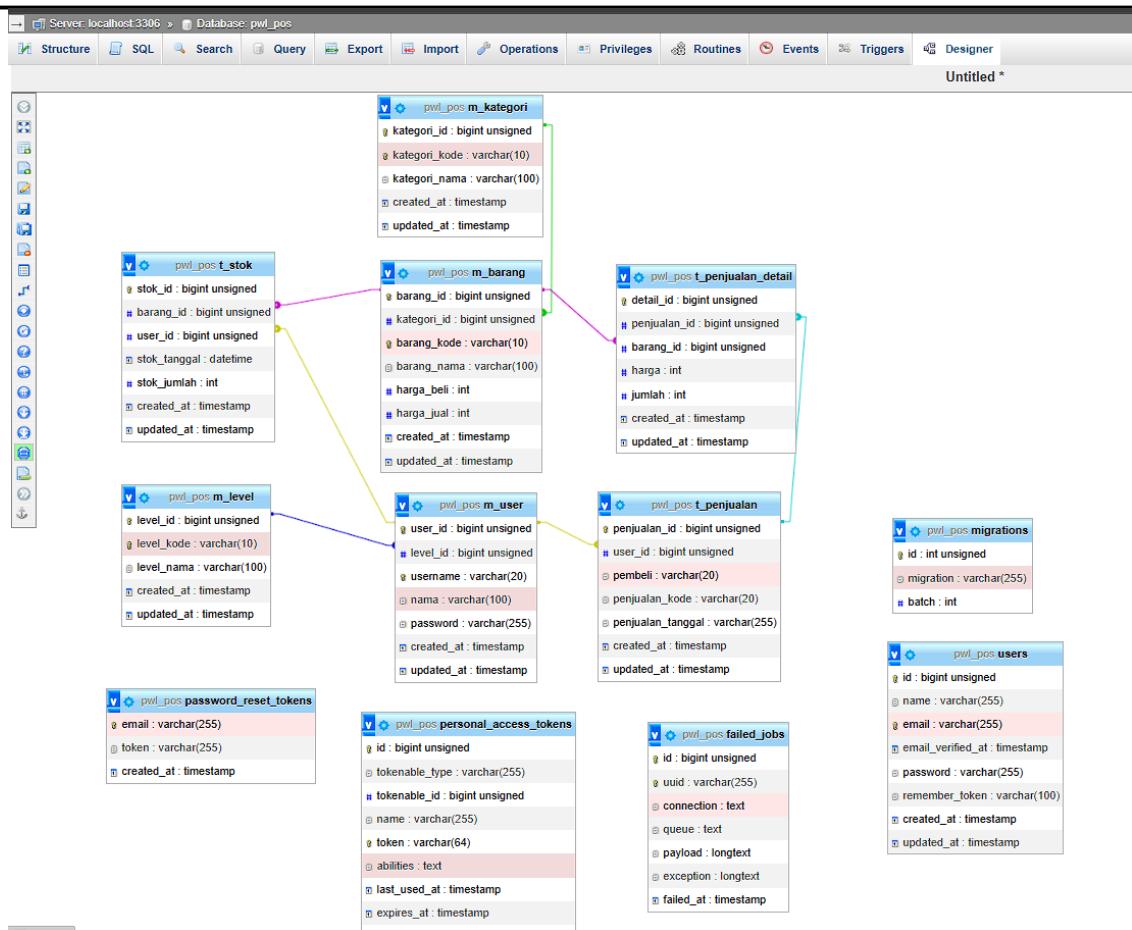
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('t_penjualan_detail');
    }
};
```

```
INFO Running migrations.
2025_03_03_043850_create_t_penjualan_detail_table ..... 262ms DONE
```

| [t_penjualan_detail](#) [Browse](#) [Structure](#) [Search](#) [Insert](#) [Empty](#) [Drop](#) [InnoDB](#)

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





= Ya, relasi sudah sesuai

6. Laporkan hasil Praktikum-2.2 ini dan *commit* perubahan pada *git*.

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



C. SEEDER

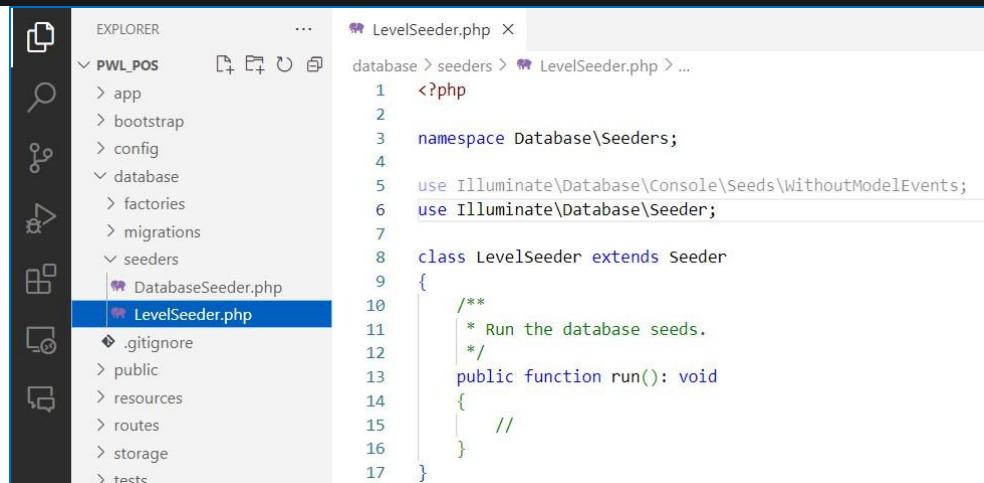
Praktikum 3 – Membuat file seeder

1. Kita akan membuat file seeder untuk table `m_level` dengan mengetikkan perintah

```
php artisan make:seeder LevelSeeder
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan make:seeder LevelSeeder
```

```
INFO Seeder [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS\database\seeders\LevelSeeder.php] created successfully.
```



```
<?php
namespace Database\Seeders;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
class LevelSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
    }
}
```



```
<?php
namespace Database\Seeders;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
class LevelSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
            ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
            ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
        ];
        DB::table('m_level')->insert($data);
    }
}
```



```
public function run(): void
{
    $data = [
        ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
        ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
        ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
    ];
    DB::table('m_level')->insert($data);
}
```

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

2. Selanjutnya, kita jalankan file *seeder* untuk table `m_level` pada terminal

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:_laragon\www\PWL_POS> php artisan db:seed --class=LevelSeeder

INFO Seeding database.

PS D:_laragon\www\PWL_POS> █

PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> p
hp artisan db:seed --class=LevelSeeder

INFO Seeding database.

3. Ketika *seeder* berhasil dijalankan maka akan tampil data pada table `m_level`

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

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

4. Sekarang kita buat file *seeder* untuk table `m_user` yang me-refer ke table `m_level`

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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan make:seeder UserSeeder
```

```
INFO Seeder [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS\database\seeders\UserSeeder.php] created successfully.
```



5. Modifikasi file **class UserSeeder** seperti berikut

```
9 class UserSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             [
15                 'user_id' => 1,
16                 'level_id' => 1,
17                 'username' => 'admin',
18                 'nama' => 'Administrator',
19                 'password' => Hash::make('12345'), // class untuk mengenkripsi/hash password
20             ],
21             [
22                 'user_id' => 2,
23                 'level_id' => 2,
24                 'username' => 'manager',
25                 'nama' => 'Manager',
26                 'password' => Hash::make('12345'),
27             ],
28             [
29                 'user_id' => 3,
30                 'level_id' => 3,
31                 'username' => 'staff',
32                 'nama' => 'Staff/Kasir',
33                 'password' => Hash::make('12345'),
34             ],
35         ];
36         DB::table('m_user')->insert($data);
37     }
38 }
```



```
$data = [
    ['user_id' => 1, 'level_id' => 1, 'username' => 'admin', 'nama' => 'Administrator', 'password' => Hash::make('12345'),], // Hash::make' class untuk mengenkripsi/hash password
    ['user_id' => 2, 'level_id' => 2, 'username' => 'manager', 'nama' => 'Manager', 'password' => Hash::make('12345'),],
    ['user_id' => 3, 'level_id' => 3, 'username' => 'staff', 'nama' => 'Staff/Kasir', 'password' => Hash::make('12345'),],
];
DB::table('m_user')->insert($data);
```

6. Jalankan perintah untuk mengeksekusi class **UserSeeder**

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

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS>
php artisan db:seed --class=UserSeeder

INFO Seeding database.
```

7. Perhatikan hasil seeder pada table **m_user**

	user_id	level_id	username	nama	password	
<input type="checkbox"/>	1	1	admin	Administrator	\$2y\$12\$Tevu4dDO1CUAQpeM6H.Vp.LySwhY.4oAKU7FzwS6fXV...	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>
<input type="checkbox"/>	2	2	manager	Manager	\$2y\$12\$AjfnS20/FdPTeUgghz31muEhlFaruLxkh5wvZ9NGRpu...	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>
<input type="checkbox"/>	3	3	staff	Staff/Kasir	\$2y\$12\$G123TqGclW5pYeR0VL4o5OxPwb3Osk99VMy/BHnbJ9W...	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>

	user_id	level_id	username	nama	password	creat
<input type="checkbox"/>	1	1	admin	Administrator	\$2y\$12\$hO/Q/F.imB.mnqqX52p3Y.rJFnRjYEUHJOnBcKtz5v...	NULL
<input type="checkbox"/>	2	2	manager	Manager	\$2y\$12\$V8Q5JA8DylakVET7/yZde6O4.OO45cLXdYE38uTnw...	NULL
<input type="checkbox"/>	3	3	staff	Staff/Kasir	\$2y\$12\$3XHgqduCoDhcOZ.gHhjOtes5NkFL2X/1q3H35c79NjH...	NULL

8. Ok, data seeder berhasil di masukkan ke database.

9. Sekarang coba kalian masukkan data *seeder* untuk table ya ng 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

- Seeder kategori

```
$data = [
    ['kategori_id' => 1, 'kategori_kode' => 'KAT001', 'kategori_nama' => 'Pakaian'],
    ['kategori_id' => 2, 'kategori_kode' => 'KAT002', 'kategori_nama' => 'Elektronik'],
    ['kategori_id' => 3, 'kategori_kode' => 'KAT003', 'kategori_nama' => 'Makanan'],
    ['kategori_id' => 4, 'kategori_kode' => 'KAT004', 'kategori_nama' => 'Minuman'],
    ['kategori_id' => 5, 'kategori_kode' => 'KAT005', 'kategori_nama' => 'Perabotan'],
];

// Insert ke tabel `m_kategori`
DB :: table('m_kategori')->insert($data);
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan db:seed --class=KategoriSeeder
INFO Seeding database.
```

	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	1	KAT001	Pakaian	NULL	NULL
<input type="checkbox"/>	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/>	3	KAT003	Makanan	NULL	NULL
<input type="checkbox"/>	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/>	5	KAT005	Perabotan	NULL	NULL

- Seeder barang

```
1 $data = [
2     //1 pakaian, 2 elektronik, makanan, minuman, perabotan
3     ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'BRG001', 'barang_nama' => 'Kemeja Katun', 'harga_beli' => 70000, 'harga_jual' => 75000],
4     ['barang_id' => 2, 'kategori_id' => 1, 'barang_kode' => 'BRG002', 'barang_nama' => 'Celana Jeans', 'harga_beli' => 80000, 'harga_jual' => 85000],
5     ['barang_id' => 3, 'kategori_id' => 2, 'barang_kode' => 'BRG003', 'barang_nama' => 'Blender', 'harga_beli' => 200000, 'harga_jual' => 205000],
6     ['barang_id' => 4, 'kategori_id' => 2, 'barang_kode' => 'BRG004', 'barang_nama' => 'Televisi', 'harga_beli' => 700000, 'harga_jual' => 705000],
7     ['barang_id' => 5, 'kategori_id' => 3, 'barang_kode' => 'BRG005', 'barang_nama' => 'Mi Instan', 'harga_beli' => 5000, 'harga_jual' => 7000],
8     ['barang_id' => 6, 'kategori_id' => 3, 'barang_kode' => 'BRG006', 'barang_nama' => 'Roti Tawar', 'harga_beli' => 10000, 'harga_jual' => 15000],
9     ['barang_id' => 7, 'kategori_id' => 4, 'barang_kode' => 'BRG007', 'barang_nama' => 'Air Mineral', 'harga_beli' => 5000, 'harga_jual' => 6000],
10    ['barang_id' => 8, 'kategori_id' => 4, 'barang_kode' => 'BRG008', 'barang_nama' => 'Ten Manis', 'harga_beli' => 5000, 'harga_jual' => 6000],
11    ['barang_id' => 9, 'kategori_id' => 5, 'barang_kode' => 'BRG009', 'barang_nama' => 'Gayung', 'harga_beli' => 5000, 'harga_jual' => 6000],
12    ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'BRG010', 'barang_nama' => 'Kemoceng', 'harga_beli' => 7000, 'harga_jual' => 9000],
13];
14];
15
16 DB::table('m_barang')->insert($data);
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan db:seed --class=BarangSeeder
INFO Seeding database.
```



		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	Kemeja Katun	70000	75000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	1	BRG002	Celana Jeans	80000	85000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	2	BRG003	Blender	200000	205000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	2	BRG004	Televisi	700000	705000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	3	BRG005	Mi Instan	5000	7000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	3	BRG006	Roti Tawar	10000	15000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	4	BRG007	Air Mineral	5000	6000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	4	BRG008	Teh Manis	5000	6000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	9	5	BRG009	Gayung	5000	6000	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	10	5	BRG010	Kemoceng	7000	9000	NULL	NULL

- Seeder stok

```
1 $data = [
2     ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 100],
3     ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 50],
4     ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 30],
5     ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 75],
6     ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 200],
7     ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 120],
8     ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 150],
9     ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 90],
10    ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 60],
11    ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 40],
12 ];
13
14
15 // Insert data ke tabel 't_stok'
16 DB::table('t_stok')->insert($data);
17 }
```

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan db:seed --class=StokSeeder
INFO Seeding database.
```

		stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	1	1	2025-03-03 09:03:24	100	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	2	1	2025-03-03 09:03:24	50	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	3	1	2025-03-03 09:03:24	30	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	4	1	2025-03-03 09:03:24	75	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	5	1	2025-03-03 09:03:24	200	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	6	1	2025-03-03 09:03:24	120	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	7	1	2025-03-03 09:03:24	150	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	8	1	2025-03-03 09:03:24	90	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	9	9	1	2025-03-03 09:03:24	60	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	10	10	1	2025-03-03 09:03:24	40	NULL	NULL

- Seeder penjualan



```
1 public function run(): void
2 {
3     $data = [
4         ['penjualan_id' => 1, 'user_id' => 1, 'pembeli' => 'Nimas', 'penjualan_kode' => 'TRX001', 'penjualan_tanggal' => '2024-01-02 10:15:00'],
5         ['penjualan_id' => 2, 'user_id' => 1, 'pembeli' => 'Septian', 'penjualan_kode' => 'TRX002', 'penjualan_tanggal' => '2024-01-05 14:30:00'],
6         ['penjualan_id' => 3, 'user_id' => 1, 'pembeli' => 'Dini', 'penjualan_kode' => 'TRX003', 'penjualan_tanggal' => '2024-01-07 09:45:00'],
7         ['penjualan_id' => 4, 'user_id' => 1, 'pembeli' => 'Desi', 'penjualan_kode' => 'TRX004', 'penjualan_tanggal' => '2024-01-10 16:00:00'],
8         ['penjualan_id' => 5, 'user_id' => 1, 'pembeli' => 'Eko', 'penjualan_kode' => 'TRX005', 'penjualan_tanggal' => '2024-01-12 11:20:00'],
9         ['penjualan_id' => 6, 'user_id' => 1, 'pembeli' => 'Fajar', 'penjualan_kode' => 'TRX006', 'penjualan_tanggal' => '2024-01-15 17:10:00'],
10        ['penjualan_id' => 7, 'user_id' => 1, 'pembeli' => 'Galuh', 'penjualan_kode' => 'TRX007', 'penjualan_tanggal' => '2024-01-18 13:40:00'],
11        ['penjualan_id' => 8, 'user_id' => 1, 'pembeli' => 'Hadi', 'penjualan_kode' => 'TRX008', 'penjualan_tanggal' => '2024-01-20 08:50:00'],
12        ['penjualan_id' => 9, 'user_id' => 1, 'pembeli' => 'Cika', 'penjualan_kode' => 'TRX009', 'penjualan_tanggal' => '2024-01-25 15:05:00'],
13        ['penjualan_id' => 10, 'user_id' => 1, 'pembeli' => 'Joko', 'penjualan_kode' => 'TRX010', 'penjualan_tanggal' => '2024-01-28 12:25:00'],
14    ];
15
16    // Insert data ke tabel 't_penjualan'
17    DB::table('t_penjualan')->insert($data);
18 }
19
20
21
22
23
24
25
26
27
28
29 }
```

	← T →	▼	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	1	Nimas	TRX001	2024-01-02 10:15:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	1	Septian	TRX002	2024-01-05 14:30:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	1	Dini	TRX003	2024-01-07 09:45:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	1	Desi	TRX004	2024-01-10 16:00:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	1	Eko	TRX005	2024-01-12 11:20:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	1	Fajar	TRX006	2024-01-15 17:10:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	7	1	Galuh	TRX007	2024-01-18 13:40:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	8	1	Hadi	TRX008	2024-01-20 08:50:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	9	1	Cika	TRX009	2024-01-25 15:05:00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	10	1	Joko	TRX010	2024-01-28 12:25:00	NULL

- Seeder penjualan_detail

```
1 public function run(): void
2 {
3     $penjualan_ids = DB::table('t_penjualan')->pluck('penjualan_id'); //Mengambil semua id penjualan
4     $barang_ids = DB::table('m_barang')->pluck('barang_id'); //Mengambil semua id barang
5
6     $data = [];
7     $detail_id = 1; //memulai detail_id dari urutan 1
8
9     foreach ($penjualan_ids as $penjualan_id) {
10        // Ambil acak 3 barang untuk setiap transaksi
11        $barang_acak = $barang_ids->shuffle()->take(3);
12
13        foreach ($barang_acak as $barang_id) {
14            $harga = rand(10000, 50000); // Harga acak antara 10.000 - 50.000
15            $jumlah = rand(1, 5); // Jumlah barang antara 1 - 5
16
17            $data[] = [
18                'detail_id' => $detail_id++, // Auto-increment manual
19                'penjualan_id' => $penjualan_id,
20                'barang_id' => $barang_id,
21                'harga' => $harga,
22                'jumlah' => $jumlah,
23            ];
24        }
25    }
26
27    // Insert data ke tabel `t_penjualan_detail`
28    DB::table('t_penjualan_detail')->insert($data);
29 }
```



	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input checked="" type="checkbox"/> Edit Copy Delete	9	3	5	29191	4	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	10	4	6	10786	1	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	11	4	9	49235	2	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	12	4	7	31662	5	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	13	5	10	19143	3	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	14	5	3	34524	2	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	15	5	5	39865	2	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	16	6	9	34349	3	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	17	6	5	11406	1	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	18	6	8	35359	1	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	19	7	5	34580	5	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	20	7	2	24476	5	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	21	7	10	15337	2	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	22	8	6	45102	1	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	23	8	2	22821	5	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	24	8	4	28301	5	NULL	NULL
<input checked="" type="checkbox"/> Edit Copy Delete	25	9	9	46718	5	NULL	NULL

10. Jika sudah, laporkan hasil Praktikum-3 ini dan *commit* perubahan pada *git*

Github: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



D. DB FACADE

Praktikum 4 – Implementasi DB Facade

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

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

```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan make:controller LevelController
```

```
[INFO] Controller [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS\app\Http\Controllers\LevelController.php] created successfully.
```

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

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

```
use App\Http\Controllers\LevelController;
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/level', [LevelController::class, 'index']);
```

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

```
LevelController.php > web.php
app > Http > Controllers > LevelController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class LevelController extends Controller
9  {
10     public function index()
11     {
12         DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['cus', 'Pelanggan', now()]);
13
14         return 'Insert data baru berhasil';
15     }
16 }
```



	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4	CUS	Pelanggan	2024-02-26 08:20:00	NULL

	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4	CUS	Pelanggan	2025-03-03 14:35:00	NULL

← → ⌛ ⓘ 127.0.0.1:8000/level

Insert data baru berhasil

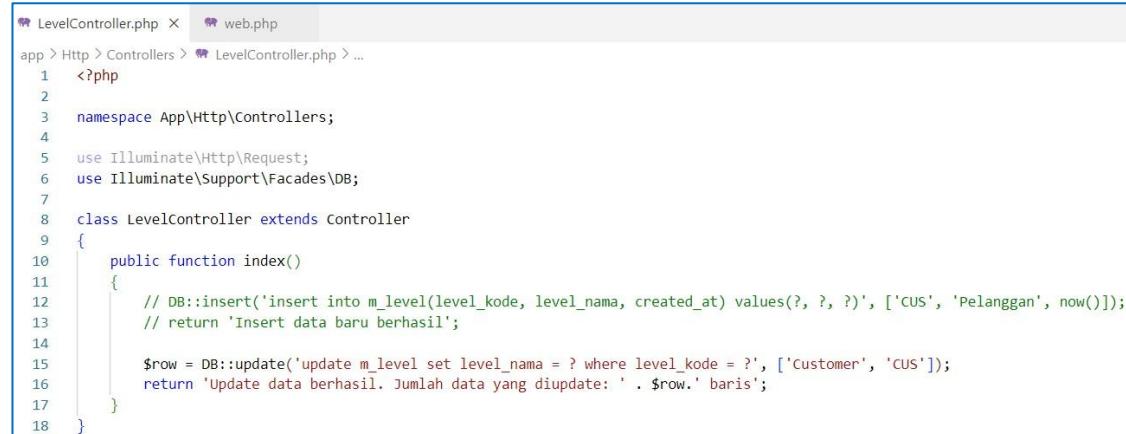
```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13
14         return 'Insert data baru berhasil';
15     }
16 }
17
```

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`

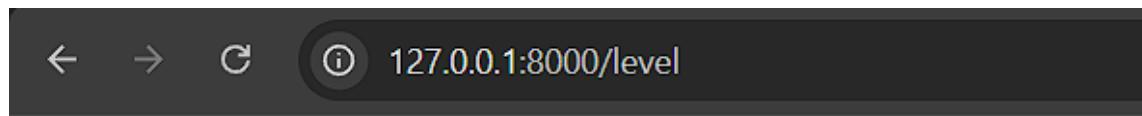
4. Selanjutnya, kita modifikasi lagi file `LevelController` untuk meng-*update* data di table



`m_level` seperti berikut



```
LevelController.php x web.php
app > Http > Controllers > LevelController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
16         return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
17     }
18 }
```



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	2025-03-03 14:35:00	NULL

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

5. 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`
6. Kita coba modifikasi lagi file `LevelController` untuk melakukan proses hapus data



```
LevelController.php X web.php
app > Http > Controllers > LevelController.php > LevelController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['cus', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
16         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
17
18         $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
19         return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
20     }
21 }
```

		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

← → ⌂ ⓘ 127.0.0.1:8000/level

Delete data berhasil. Jumlah data yang dihapus: 1 baris



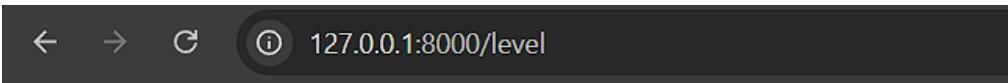
7. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table `m_level`. Kita modifikasi file `LevelController` seperti berikut

```
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
16         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
17
18         // $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
19         // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
20
21         $data = DB::select('select * from m_level');
22         return view('level') ['data' => $data];
23     }
24 }
```

8. 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`

```
LevelController.php      level.blade.php X      web.php
resources > views > level.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Data Level Pengguna</title>
5      </head>
6      <body>
7          <h1>Data Level Pengguna</h1>
8          <table border="1" cellpadding="2" cellspacing="0">
9              <tr>
10                 <th>ID</th>
11                 <th>Kode Level</th>
12                 <th>Nama Level</th>
13             </tr>
14             @foreach ($data as $d)
15             <tr>
16                 <td>{{ $d->level_id }}</td>
17                 <td>{{ $d->level_kode }}</td>
18                 <td>{{ $d->level_nama }}</td>
19             </tr>
20             @endforeach
21         </table>
22     </body>
23 </html>
```

9. Silahkan dicoba pada browser dan amati apa yang terjadi



Data Level Pengguna

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

10. Laporkan hasil Praktikum-4 ini dan *commit* perubahan pada *git*.

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



E. QUERY BUILDER

Praktikum 5 – Implementasi *Query Builder*

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

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

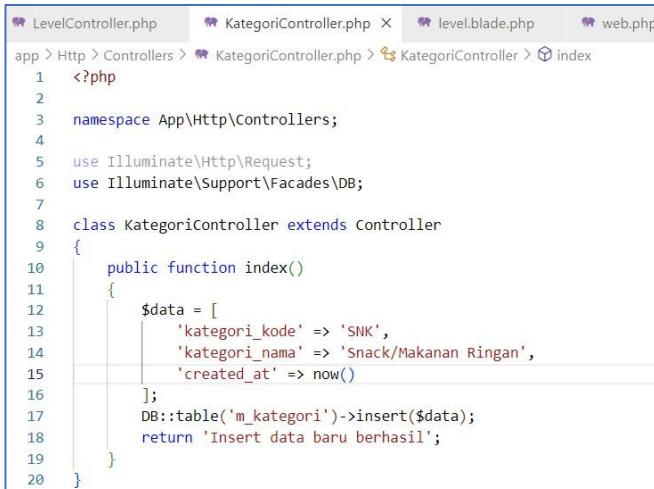
```
INFO Controller [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.
```

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

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

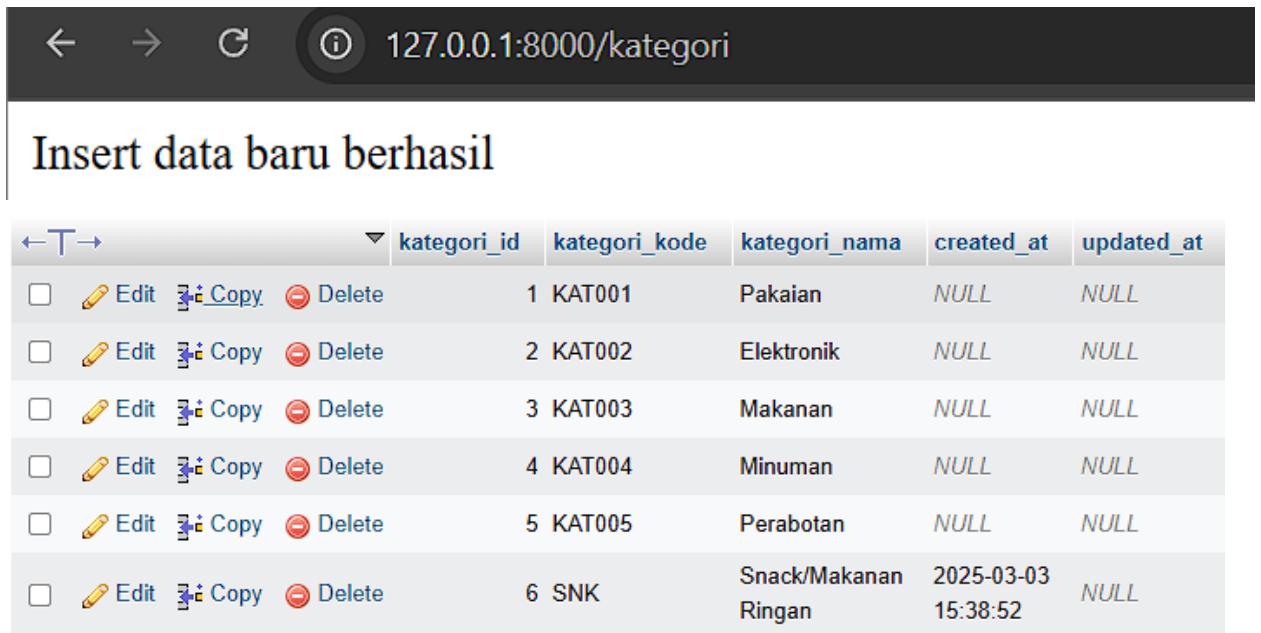
```
Route::get('/level', [LevelController::class, 'index']);
Route::get('/kategori', [KategoriController::class, 'index']);
```

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



```
LevelController.php KategoriController.php X level.blade.php web.php
app > Http > Controllers > KategoriController.php > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class KategoriController extends Controller
9 {
10     public function index()
11     {
12         $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17         DB::table('m_kategori')->insert($data);
18         return 'Insert data baru berhasil';
19     }
20 }
```

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`



← → ⌂ 127.0.0.1:8000/kategori

Insert data baru berhasil

kategori_id	kategori.kode	kategori.nama	created_at	updated_at
1	KAT001	Pakaian	NULL	NULL
2	KAT002	Elektronik	NULL	NULL
3	KAT003	Makanan	NULL	NULL
4	KAT004	Minuman	NULL	NULL
5	KAT005	Perabotan	NULL	NULL
6	SNK	Snack/Makanan Ringan	2025-03-03 15:38:52	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';
```

6. 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`



← → ⌂ 127.0.0.1:8000/kategori

Update data berhasil. Jumlah data yang diupdate: 1 baris

	← ↑ →	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	KAT001	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KAT003	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KAT005	Perabotan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Camilan	2025-03-03 15:38:52	NULL

7. 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';
```

← → ⌂ 127.0.0.1:8000/kategori

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	KAT001	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KAT003	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KAT005	Perabotan	NULL	NULL

8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table [m_kategori](#). Kita modifikasi file [KategoriController](#) seperti berikut



```
10 public function index()
11 {
12     /* $data = [
13         'kategori_kode' => 'SNK',
14         'kategori_nama' => 'Snack/Makanan Ringan',
15         'created_at' => now()
16     ];
17     DB::table('m_kategori')->insert($data);
18     return 'Insert data baru berhasil'; */
19
20     // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
21     // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
22
23     // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
24     // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
25
26     $data = DB::table('m_kategori')->get();
27     return view('kategori', ['data' => $data]);
28 }
```



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

9. 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`

```
resources > views > kategori.blade.php > html > body > table > tr > td
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Data Kategori Barang</title>
5      </head>
6      <body>
7          <h1>Data Kategori Barang</h1>
8          <table border="1" cellpadding="2" cellspacing="0">
9              <tr>
10                 <th>ID</th>
11                 <th>Kode Kategori</th>
12                 <th>Nama Kategori</th>
13             </tr>
14             @foreach ($data as $d)
15             <tr>
16                 <td>{{ $d->kategori_id }}</td>
17                 <td>{{ $d->kategori_kode }}</td>
18                 <td>{{ $d->kategori_nama }}</td>
19             </tr>
20             @endforeach
21         </table>
22     </body>
23 </html>
```



```
resources > views > kategori.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Data Kategori Barang</title>
5  </head>
6  <body>
7  |   <h1>Data Kategori Barang</h1>
8  |   <table border="1" cellpadding="2" cellspacing="0">
9  |       <tr>
10 |           <th>ID</th>
11 |           <th>Kode Kategori</th>
12 |           <th>Nama Kategori</th>
13 |       </tr>
14 |       @foreach ($data as $d)
15 |           <tr>
16 |               <td>{{ $d->kategori_id }}</td>
17 |               <td>{{ $d->kategori_kode }}</td>
18 |               <td>{{ $d->kategori_nama }}</td>
19 |           </tr>
20 |       @endforeach
21 |   </table>
22 | </body>
23 | </html>
24 |
```

10. Silahkan dicoba pada browser dan amati apa yang terjadi.

← → ⌂ 127.0.0.1:8000/kategori

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	KAT001	Pakaian
2	KAT002	Elektronik
3	KAT003	Makanan
4	KAT004	Minuman
5	KAT005	Perabotan

11. Laporkan hasil Praktikum-5 ini dan *commit* perubahan pada *git*

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git

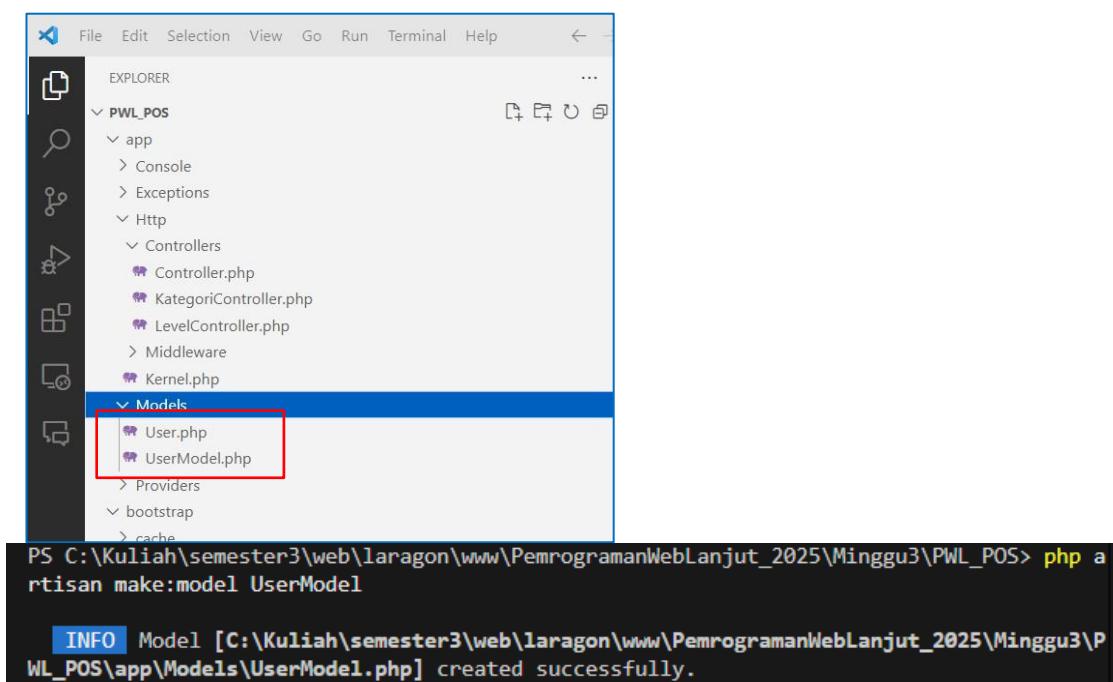


F. ELOQUENT ORM

Praktikum 6 – Implementasi Eloquent ORM

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

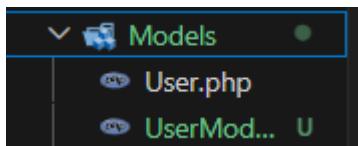
```
php artisan make:model UserModel
```



```
PS C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS> php artisan make:model UserModel
INFO Model [C:\Kuliah\semester3\web\laragon\www\PemrogramanWebLanjut_2025\Minggu3\PWL_POS\app\Models\UserModel.php] created successfully.
```



2. 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`



3. Kita buka file `UserModel.php` dan modifikasi seperti berikut

```
app > Models > UserModel.php > UserModel
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user';           // Mendefinisikan nama tabel yang digunakan oleh model ini
13     protected $primaryKey = 'user_id';    // Mendefinisikan primary key dari tabel yang digunakan
14 }
15
```

```
app > Models > UserModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user'; //mendefinisikan nama tabel yang digunakan oleh model
13     protected $primaryKey = 'user_id'; //Mendefinisikan primary key dari tabel yang digunakan
14 }
15
```

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

```
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\KategoriController;
4  use App\Http\Controllers\LevelController;
5  use App\Http\Controllers\UserController;
6  use Illuminate\Support\Facades\Route;
7
8
9  Route::get('/', function () {
10     return view('welcome');
11 });
12
13 Route::get('/level', [LevelController::class, 'index']);
14 Route::get('/kategori', [KategoriController::class, 'index']);
15 Route::get('/user', [UserController::class, 'index']);
```



```
routes > web.php
1  <?php
2
3  use App\Http\Controllers\LevelController;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\KategoriController;
6  use App\Http\Controllers\UserController;
7
8  /*
9  |-----|
10 | Web Routes
11 |-----|
12 |
13 | Here is where you can register web routes for your application. These
14 | routes are loaded by the RouteServiceProvider and all of them will
15 | be assigned to the "web" middleware group. Make something great!
16 |
17 */
18
19 <?php
20 Route::get('/', function () {
21     return view('welcome');
22 });
23
24 Route::get('/level', [LevelController::class, 'index']);
25 Route::get('/kategori', [KategoriController::class, 'index']);
26 Route::get('/user', [UserController::class, 'index']);
```

5. Sekarang, kita buat file controller **UserController** dan memodifikasinya seperti berikut

```
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function index()
11     {
12         // coba akses model UserModel
13         $user = UserModel::all(); // ambil semua data dari tabel m_user
14         return view('user', ['data' => $user]);
15     }
16 }
```

```
app > Http > Controllers > UserController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function index()
11     {
12         //mencoba mengakses UserModel
13         $user = UserModel::all(); //ambil semua data dari tabel m_user
14         return view('user', ['data' => $user]);
15     }
16 }
```



6. Kemudian kita buat view [user.blade.php](#)

```
resources > views > user.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |       <title>Data User</title>
5  |   </head>
6  |   <body>
7  |   |       <h1>Data User</h1>
8  |   |       <table border="1" cellpadding="2" cellspacing="0">
9  |   |           <tr>
10 |   |               <th>ID</th>
11 |   |               <th>Username</th>
12 |   |               <th>Nama</th>
13 |   |               <th>ID Level Pengguna</th>
14 |   |           </tr>
15 |   |           @foreach ($data as $d)
16 |   |           <tr>
17 |   |               <td>{{ $d->user_id }}</td>
18 |   |               <td>{{ $d->username }}</td>
19 |   |               <td>{{ $d->nama }}</td>
20 |   |               <td>{{ $d->level_id }}</td>
21 |   |           </tr>
22 |   |           @endforeach
23 |   |       </table>
24 |   </body>
25 </html>
```

```
resources > views > user.blade.php > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |     <title>Data User</title>
5  </head>
6  <body>
7  |     <h1>Data User</h1>
8  |     <table border="1" cellpadding="2" cellspacing="0">
9  |         <tr>
10 |         |             <th>ID</th>
11 |         |             <th>Username</th>
12 |         |             <th>Nama</th>
13 |         |             <th>ID Level Pengguna</th>
14 |         |         </tr>
15 |         |         @foreach ($data as $d)
16 |         |         <tr>
17 |         |             <td>{{ $d->user_id }}</td>
18 |         |             <td>{{ $d->username }}</td>
19 |         |             <td>{{ $d->nama }}</td>
20 |         |             <td>{{ $d->level_id }}</td>
21 |         |         </tr>
22 |         |         @endforeach
23 |     </table>
24 </body>
25 </html>
```

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

← → ⌛ 127.0.0.1:8000/user ⏷ ⏸ ⏴

Data User

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

= Muncul tabel berisi data m_user, dimana setiap baris tabel menampilkan user_id, username, nama, dan level_id.

8. Setelah itu, kita modifikasi lagi file [UserController](#)



```
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data user dengan Eloquent Model
14         $data = [
15             'username' => 'customer-1',
16             'nama' => 'Pelanggan',
17             'password' => Hash::make('12345'),
18             'level_id' => 4
19         ];
20         userModel::insert($data); // tambahkan data ke tabel m_user
21
22         // coba akses model UserModel
23         $user = UserModel::all(); // ambil semua data dari tabel m_user
24         return view('user', ['data' => $user]);
25     }
26 }
```

```
app > Http > Controllers > UserController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index(){
12         //tambah data dengan Eloquent Model
13         $data = [
14             'username' => 'customer-1',
15             'nama' => 'Pelanggan',
16             'password' => Hash::make('12345'),
17             'level_id' => 4
18         ];
19         UserModel::insert($data); //tambah data ke tabel m_user
20
21         //mencoba mengakses UserModel
22         $user = UserModel::all(); //ambil semua data dari tabel m_user
23         return view('user', ['data' => $user]);
24     }
25 }
```

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

Illuminate\Database\QueryException PHP 8.1.10 10.48.28

SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails (`pwl_pos`.`m_use...

```
INSERT INTO `m_user`(`username`, `nama`, `password`, `level_id`) VALUES ('cu...
```

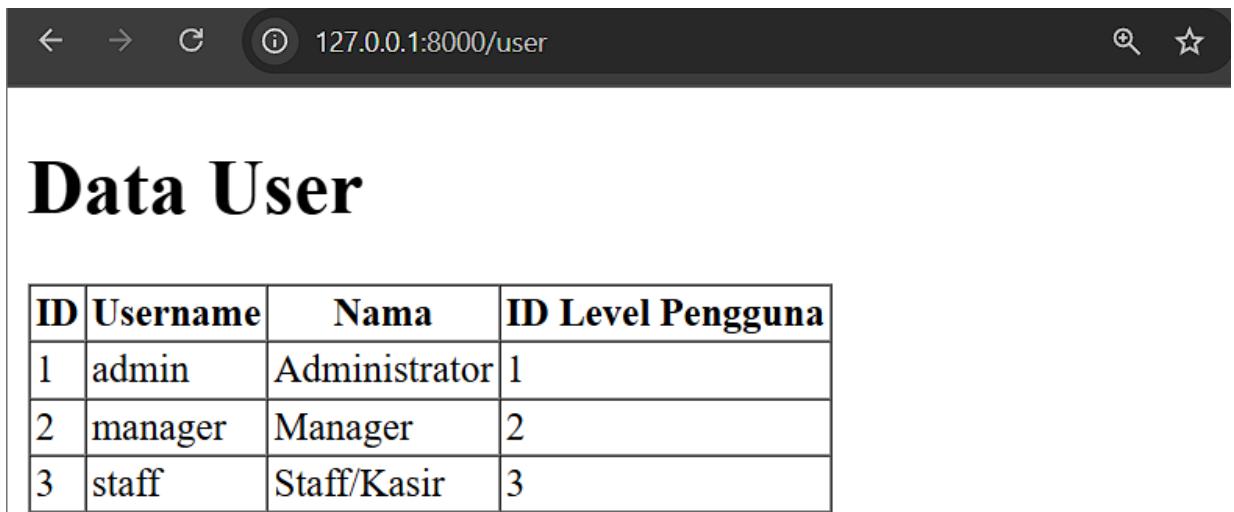
= Terjadi error, karena level_id = 4 tidak ada dalam tabel m_level. Seharusnya, data level_id = 4 di tabel m_level harus ada dulu sebelum memasukkan data ke tabel m_user.

10. Kita modifikasi lagi file **UserController** menjadi seperti berikut



```
app > Http > Controllers > UserController.php
 1  <?php
 2
 3  namespace App\Http\Controllers;
 4
 5  use App\Models\UserModel;
 6  use Illuminate\Http\Request;
 7  use Illuminate\Support\Facades\Hash;
 8
 9  class UserController extends Controller
10 {
11     public function index(){
12         //tambah data dengan Eloquent Model
13         $data = [
14             'nama' => 'Pelanggan Pertama',
15         ];
16         UserModel::where('username', 'customer-1')->update($data); //update data user
17
18         //mencoba mengakses UserModel
19         $user = UserModel::all(); //ambil semua data dari tabel m_user
20         return view('user', ['data' => $user]);
21     }
22 }
23
```

11. Jalankan di browser, amati dan laporan apa yang terjadi



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

= Menampilkan sama seperti sebelumnya, hal ini karena di tabel User username customer-1 tidak ada jadi tidak ada yang diupdate atau dirubah.

12. Jika sudah, laporan hasil Praktikum-6 ini dan *commit* perubahan pada *git*
GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



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?
= Digunakan untuk keamanan enkripsi data sensitif seperti session, password, dan token dalam Laravel.
2. Pada **Praktikum 1**, bagaimana kita men-*generate* nilai untuk `APP_KEY`?
= Untuk menggenerate APP_KEY bisa 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?
= Ada tiga file migrasi. `create_users_table.php` untuk membuat tabel users. `create_password_reset_tokens_table.php` untuk menyimpan token reset password. `create_failed_jobs_table.php` untuk menyimpan informasi job yang gagal diproses.
4. Secara *default*, file migrasi terdapat kode `$table->timestamps();`, apa tujuan/*output* dari fungsi tersebut?
= kode `$table->timestamps();`; nantinya akan menambahkan kolom `created_at` dan `updated_at` pada tabel.
5. Pada File Migrasi, terdapat fungsi `$table->id();` Tipe data apa yang dihasilkan dari fungsi tersebut?
= Fungsi ``$table->id();`` digunakan untuk membuat kolom primary key yang secara otomatis berfungsi sebagai auto-increment dengan tipe data *BIGINT UNSIGNED*. Secara default, tipe data yang dihasilkan adalah *BIGINT* tanpa tanda (*unsigned*), sehingga hanya dapat menyimpan nilai positif dalam jumlah besar. Hal ini menjadikannya ideal sebagai primary key dalam sebuah tabel.
6. Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');` ?
= `$table->id();` → Secara default, perintah ini akan membuat kolom id sebagai primary key dengan tipe data BIGINT dan memiliki atribut AUTO_INCREMENT. `$table->id('level_id');` → Perintah ini akan menghasilkan kolom dengan nama level_id (bukan id), namun tetap bertipe BIGINT, memiliki atribut AUTO_INCREMENT, dan berperan sebagai primary key.
7. Pada migration, Fungsi `->unique()` digunakan untuk apa?
= Fungsi `->unique()` pada *migration* digunakan untuk menetapkan bahwa nilai dalam kolom tersebut harus unik, sehingga tidak boleh ada duplikasi data.
8. Pada **Praktikum 2.2 - Tahap 2**, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$tabel->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$tabel->id('level_id')` ?
- ``$table->unsignedBigInteger('level_id');`` Digunakan dalam tabel `m_user` karena `level_id` berfungsi sebagai *Foreign Key* yang mengacu pada tabel `m_level`.



Penggunaan `unsignedBigInteger` memastikan tipe data sesuai dengan *Primary Key* yang dirujuk.

- `'\$table->id('level_id');` Digunakan dalam tabel `m_level` sebagai *Primary Key* dengan tipe data `BIGINT AUTO_INCREMENT`.
9. Pada **Praktikum 3 - Tahap 6**, apa tujuan dari Class `Hash`? dan apa maksud dari kode program `Hash::make('1234');`?
- Tujuan Class Hash: Digunakan untuk mengenkripsi password sebelum disimpan ke dalam database guna meningkatkan keamanan.
 - Kode `Hash::make('1234');`: Kode ini menghasilkan hash dari string "1234" menggunakan algoritma bcrypt. Hasilnya berupa string terenkripsi seperti `$2y$10$abcdefg....`. Password yang telah di-hash tidak dapat dikembalikan ke bentuk aslinya.
10. Pada **Praktikum 4 - Tahap 3/5/7**, pada *query builder* terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
- = Berfungsi sebagai placeholder untuk parameter binding dalam query SQL. Fitur ini merupakan bagian dari prepared statements, yang membantu melindungi query dari serangan SQL injection serta meningkatkan efisiensi eksekusi query.
11. Pada **Praktikum 6 - Tahap 3**, apa tujuan penulisan kode `protected $table = 'm_user';` dan `protected $primaryKey = 'user_id';`?
- `protected $table = 'm_user';`: Menetapkan nama tabel yang digunakan oleh model Eloquent sebagai m_user. Ini berguna jika nama tabel tidak sesuai dengan konvensi penamaan bawaan Laravel.
 - `protected $primaryKey = 'user_id';`: Menentukan bahwa kolom user_id digunakan sebagai primary key pada tabel, bukan id seperti standar Laravel. Ini berguna jika nama kolom primary key berbeda dari konvensi default Laravel.
12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (*DB Façade / Query Builder / Eloquent ORM*) ? jelaskan
- = Eloquent ORM lebih mudah digunakan karena sintaksnya lebih sederhana, intuitif, dan berbasis objek, sehingga memudahkan manipulasi data tanpa menulis query SQL secara manual. Namun, Query Builder lebih fleksibel untuk query kompleks, sedangkan DB Façade cocok untuk operasi langsung dengan SQL mentah.

*** Sekian, dan selamat belajar ***