

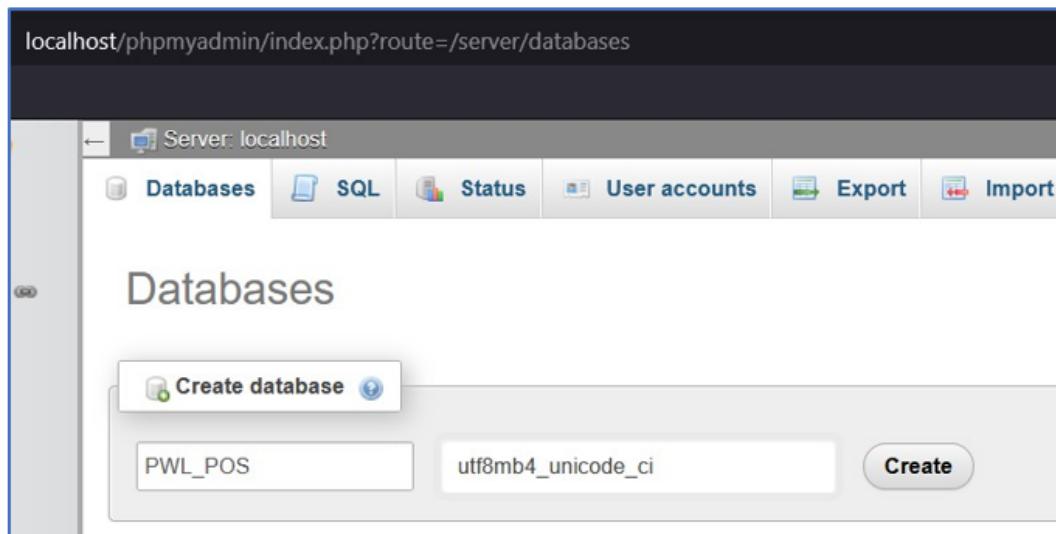
## PEMROGRAMAN WEB LANJUT

### JOBSHEET 03

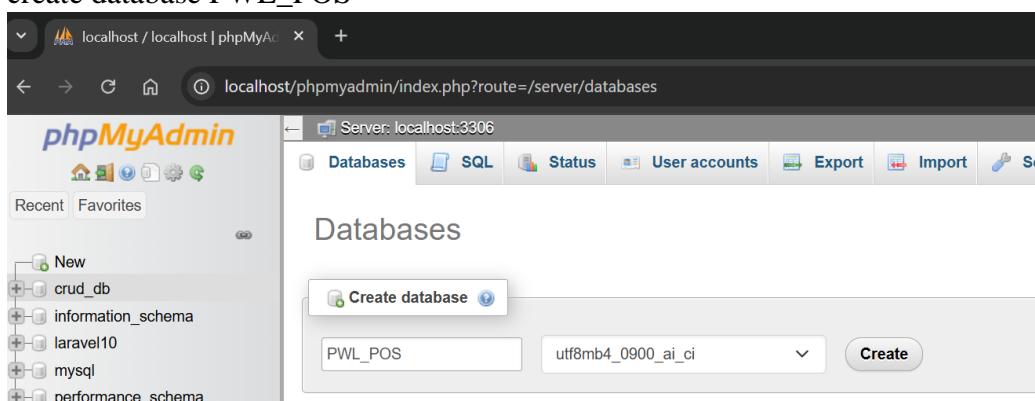
#### MIGRATION, SEEDER, DB FAÇADE, QUERY BUILDER, dan ELOQUENT FORM

#### A. Pengaturan Database

1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama **PWL\_POS**

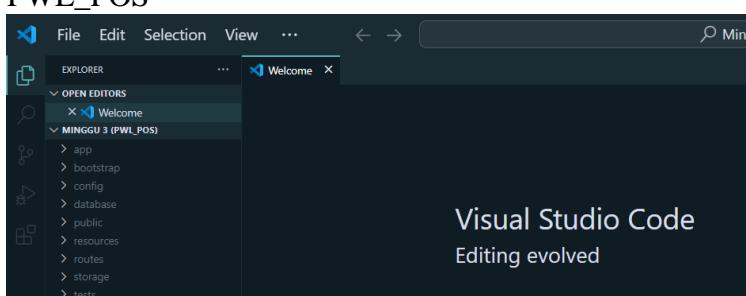


- ⇒ create database PWL\_POS

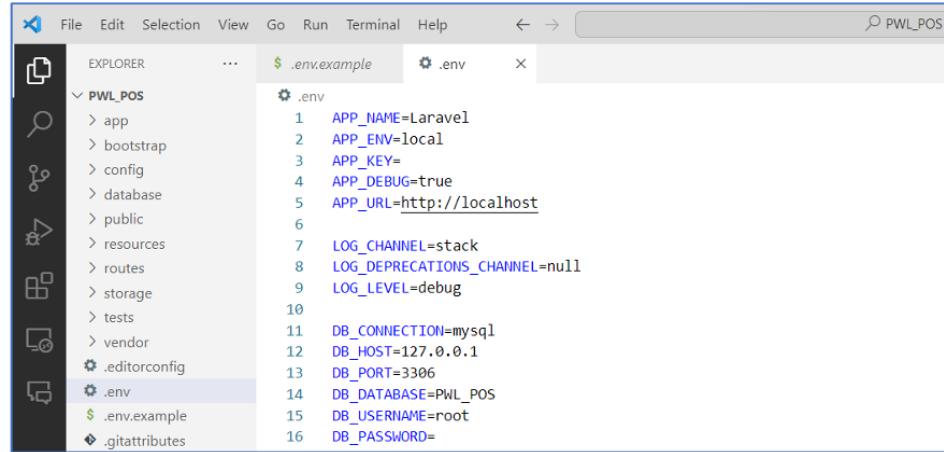


2. Buka aplikasi VSCode dan buka folder project **PWL\_POS** yang sudah kita buat

- ⇒ PWL\_POS



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
EXPLORER ... $ .env.example .env x
PWL_POS
> app
> bootstrap
> config
> database
> public
> resources
> routes
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
$ .gitattributes

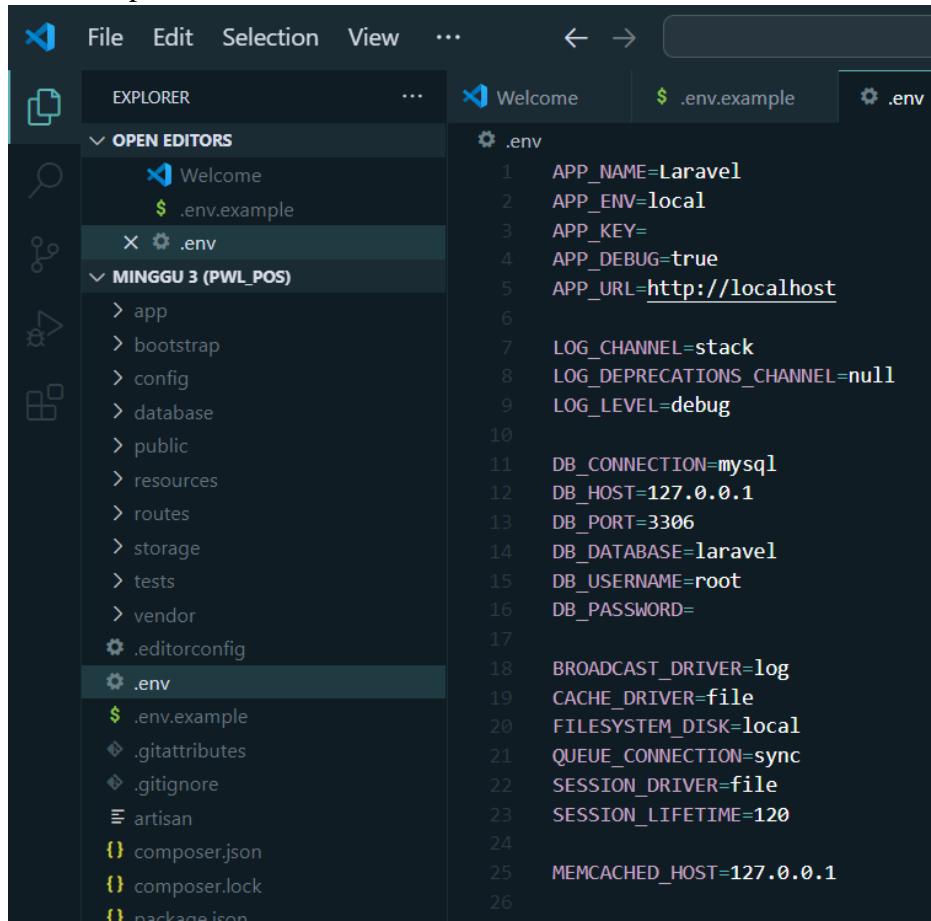
```

```

.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=

```

⇒ env.example => env.



```

File Edit Selection View ... ← →
EXPLORER ... Welcome $ .env.example .env
OPEN EDITORS
> Welcome
$ .env.example
X .env
MINGGU 3 (PWL_POS)
> app
> bootstrap
> config
> database
> public
> resources
> routes
> storage
> tests
> vendor
> .editorconfig
> .env
$ .env.example
$ .gitattributes
$ .gitignore
artisan
composer.json
composer.lock
package.json

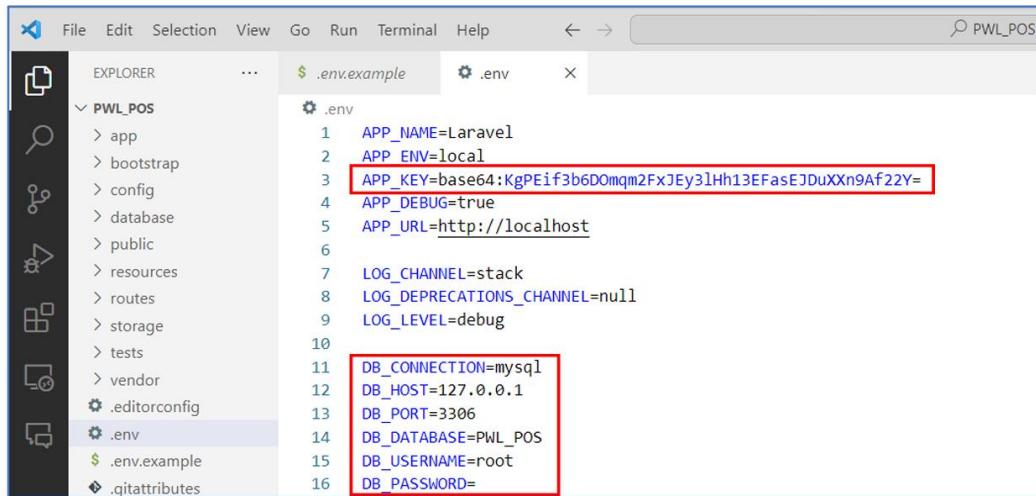
```

```

.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=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26

```

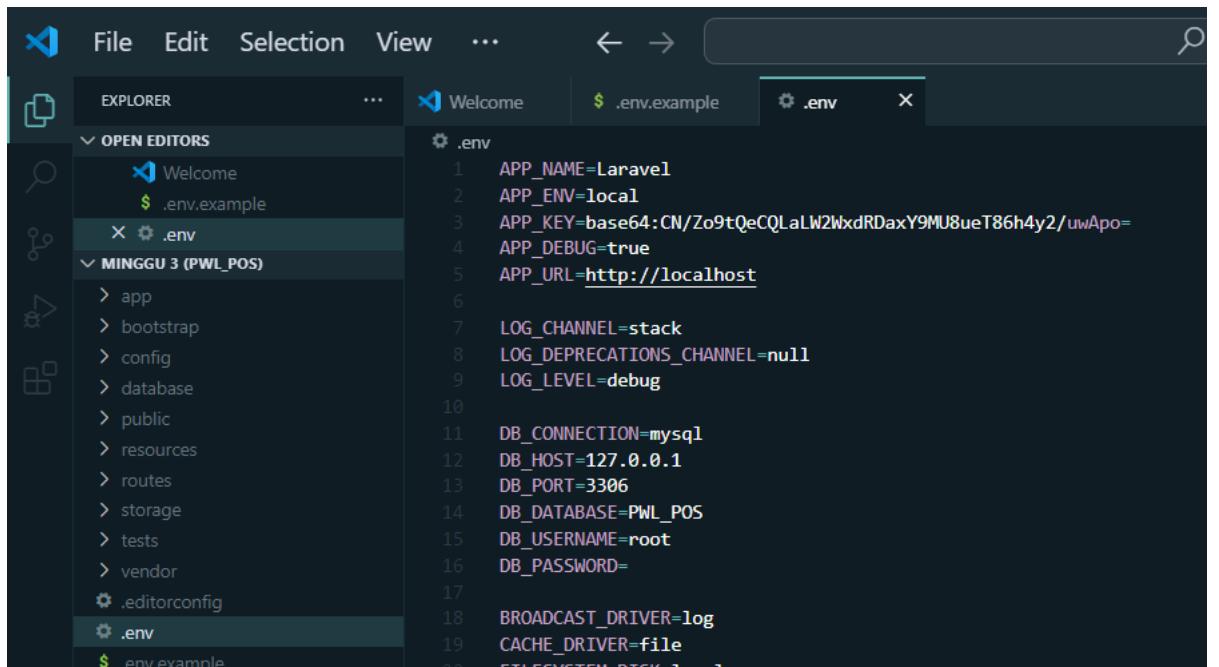
5. Edit file `.env` dan sesuaikan dengan database yang telah dibuat



A screenshot of the Visual Studio Code interface. The left sidebar shows a project structure for 'PWL\_POS' with files like app, bootstrap, config, database, public, resources, routes, storage, tests, vendor, .editorconfig, .env, .env.example, and .gitattributes. The right panel displays the contents of the '.env' file. A red box highlights the database configuration section:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:KgPEif3b6DOmqm2FxJEy3lHh13EFasEJDuXXn9Af22Y=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=PWL_POS
DB_USERNAME=root
DB_PASSWORD=
```

⇒ env.



A screenshot of the Visual Studio Code interface. The left sidebar shows a project structure for 'MINGGU 3 (PWL\_POS)' with files like app, bootstrap, config, database, public, resources, routes, storage, tests, vendor, .editorconfig, .env, and .env.example. The right panel displays the contents of the '.env' file. Compared to the previous screenshot, it includes additional variables at the bottom:

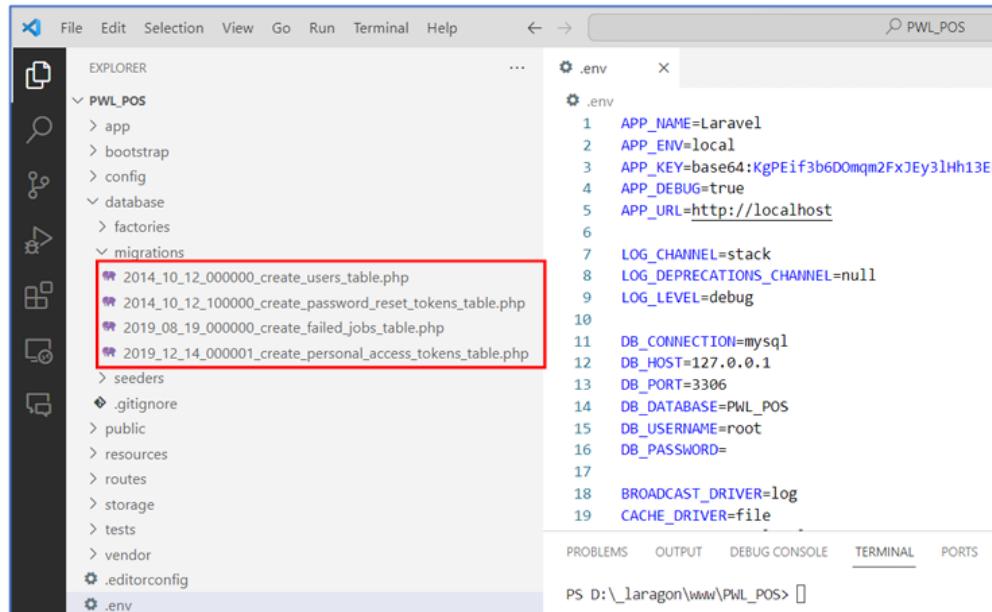
```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:CN/Zo9tQeCQlaLW2WxdRDaxY9MU8ueT86h4y2/uwApo=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=PWL_POS
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
```

6. Laporkan hasil Praktikum-1 ini dan *commit* perubahan pada *git*.

## B. Migration

### Praktikum 2.1 Pembuatan file migrasi tanpa relasi

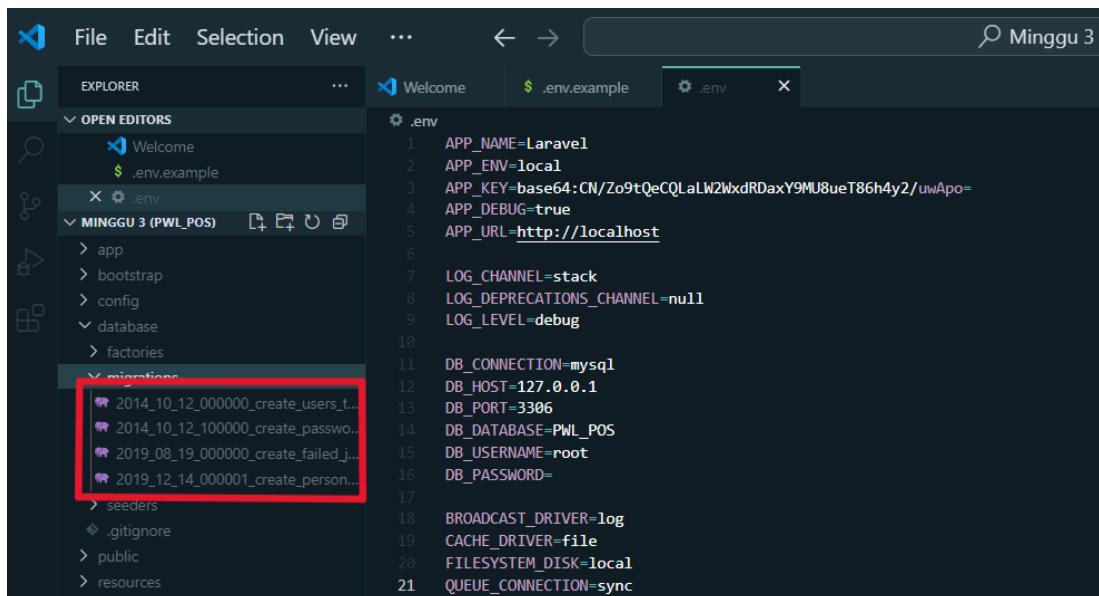
1. Buka terminal VSCode kalian, untuk yang di kotak merah adalah default dari laravel



The screenshot shows the VSCode interface with the Explorer and Editor tabs open. In the Explorer sidebar, under the 'PWL\_POS' project, the 'migrations' folder is expanded, showing several migration files: '2014\_10\_12\_000000\_create\_users\_table.php', '2014\_10\_12\_100000\_create\_password\_reset\_tokens\_table.php', '2019\_08\_19\_000000\_create\_failed\_jobs\_table.php', and '2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table.php'. These files are highlighted with a red box. The main editor window displays the '.env' configuration file with various environment variables defined.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:KgPEif3b6D0mqm2FxJEy3lHh13EF
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=PWL_POS
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
```

⇒ Laravel default migrations

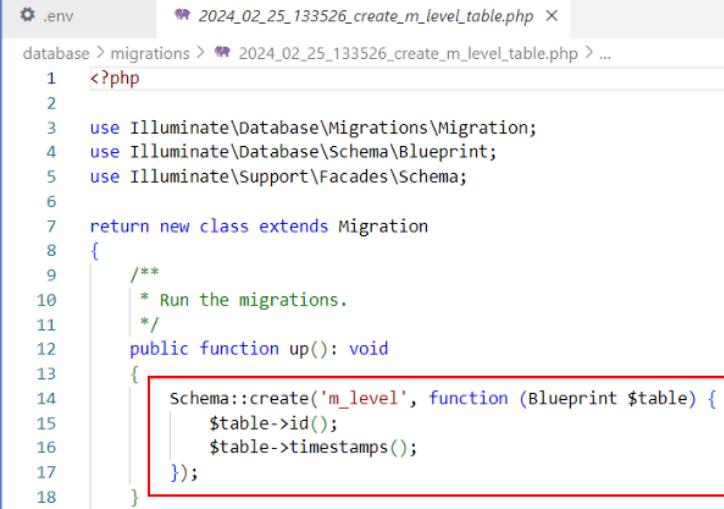


The screenshot shows the VSCode interface with the Explorer and Editor tabs open. In the Explorer sidebar, under the 'MINGGU 3 (PWL\_POS)' project, the 'migrations' folder is expanded, showing the same four migration files as the previous screenshot. These files are highlighted with a red box. The main editor window displays the '.env' configuration file with various environment variables defined.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:CN/Zo9tQeCQLaLw2WxdRDaxY9MU8ueT86h4y2/uwApo=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=PWL_POS
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
```

2. Kita abaikan dulu yang di kotak merah (jangan di hapus)
3. Kita buat file migrasi untuk table `m_level` dengan perintah

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



```

.env  2024_02_25_133526_create_m_level_table.php X
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 };

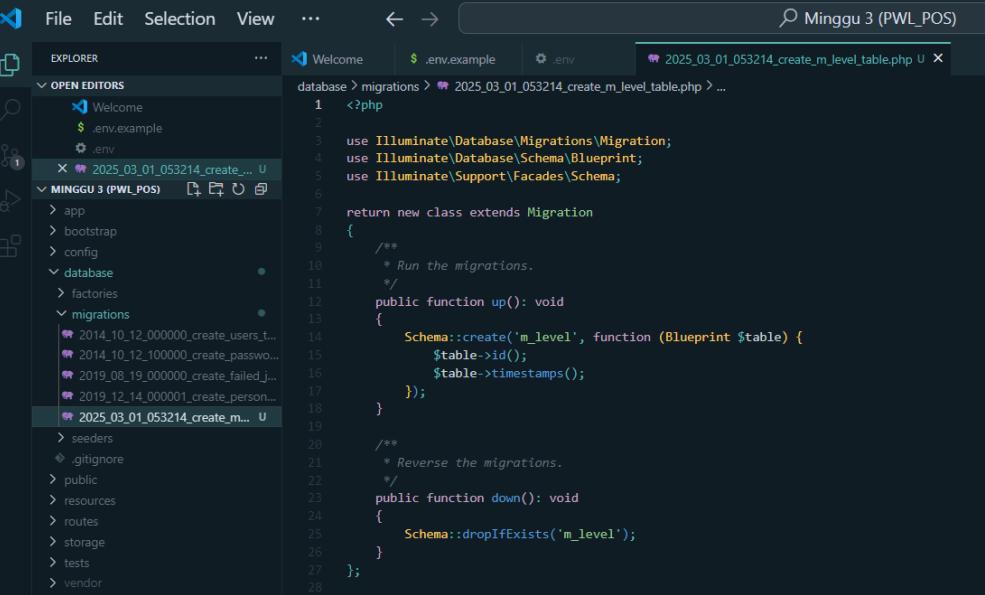
```

⇒ Make migration

```

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_m_level_table --create=m_level
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_053214_create_m_level_table.php] created successfully.

```



```

File Edit Selection View ... ← → Minggu 3 (PWL_POS)
EXPLORER .env.example .env 2025_03_01_053214_create_m_level_table.php U
OPEN EDITORS Welcome
MINGGU 3 (PWL_POS) 2025_03_01_053214_create_m_level_table.php U
database > migrations > 2025_03_01_053214_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 };

```

4. 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  };

```

⇒ Modified

The screenshot shows the VSCode interface with the following details:

- File Bar:** File, Edit, Selection, View, ...
- Editor:** The main editor window displays the PHP migration code. The file path is "database > migrations > 2025\_03\_01\_053214\_create\_m\_level\_table.php".
- Side Panels:**
  - Left Panel:** Shows icons for Explorer, Search, and others.
  - Right Panel:** Shows icons for Welcome, .env.example, .env, and a preview of the migration code.
- Code Content:**

```

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('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  };

```

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

```
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>
```

## ⇒ Migrate

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu_3 (PWL_POS)> php artisan migrate  
WARN The database 'PWL_POS' does not exist on the 'mysql' connection.  
Would you like to create it? (yes/no) [no]  
yes  
INFO Preparing database.  
Creating migration table ..... 55ms DONE  
INFO Running migrations.  
2014_10_12_000000_create_users_table ..... 87ms DONE  
2014_10_12_100000_create_password_reset_tokens_table ..... 21ms DONE  
2019_08_19_000000_create_failed_jobs_table ..... 72ms DONE  
2019_12_14_000001_create_personal_access_tokens_table ..... 79ms DONE  
2025_03_01_053214_create_m_level_table ..... 59ms DONE
```

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

Table	Action	Rows
failed_jobs		0
migrations		5
<b>m_level</b>		0
password_reset_tokens		0
personal_access_tokens		0
users		0

## ⇒ PhpMyAdmin

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs		0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
migrations		5	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
<b>m_level</b>		0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
password_reset_tokens		0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
personal_access_tokens		0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
users		0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	
6 tables	Sum	5	InnoDB	utf8mb4_0900_ai_ci	96.0 Kib	0.1

7. Ok, table sudah dibuat di database
  8. Buat table *database* dengan *migration* untuk table **m\_kategori** yang sama-sama tidak memiliki *foreign key*
- ⇒ Create m\_kategori

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_m_kategori_table --create=m_kategori
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_063056_create_m_kategori_table.php] created successfully.

VS Code Editor
  Welcome  ×  $ .env.example  ⚙ .env  2025_03_01_063056_create_m_kategori_table.php U ×

database > migrations > 2025_03_01_063056_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 };
30
31 |
```

9. Laporkan hasil Praktikum-2.1 ini dan *commit* perubahan pada *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
```

- ⇒ Create file migration m\_user

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_m_user_table --table=m_user
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_063606_create_m_user_table.php] created successfully.
```

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 };

```

## ⇒ Modifikasi

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use Illuminate\Validation\Rules\Unique;

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

            //men definisikan foreign key pada kolom Level_id mengacu pada kolom Level_id di table m_Level
            $table->foreign('level_id')->references('level_id')->on('m_level');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('m_user', function (Blueprint $table) {
            //
        });
    }
};

```

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

⇒ `php artisan migrate`

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan migrate
[INFO] Running migrations.

2025_03_01_063056_create_m_kategori_table ..... 84ms DONE
2025_03_01_063066_create_m_user_table ..... 201ms DONE

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)>
```

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

<code>m_barang</code>
<code>t_penjualan</code>
<code>t_stok</code>
<code>t_penjualan_detail</code>

⇒ buat table database

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_m_barang_table --table=m_barang
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_065748_create_m_barang_table.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_t_penjualan_table --table=t_penjualan
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_065821_create_t_penjualan_table.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_t_stok_table --table=t_stok
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_065859_create_t_stok_table.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:migration create_t_penjualan_detail_table --table=t_penjualan_detail
[INFO] Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\migrations\2025_03_01_065935_create_t_penjualan_detail_table.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan migrate
[INFO] Running migrations.

2025_03_01_065748_create_m_barang_table ..... 202ms DONE
2025_03_01_065821_create_t_penjualan_table ..... 152ms DONE
2025_03_01_065859_create_t_stok_table ..... 278ms DONE
2025_03_01_065935_create_t_penjualan_detail_table ..... 233ms DONE

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)>
```

Table m\_barang

```
database > migrations > 2025_03_01_065748_create_m_barang_table.php > class
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_barang', function (Blueprint $table) {
15              $table->id('barang_id');
16              $table->string('barang_kode', 10)->unique();
17              $table->string('barang_nama', 100);
18              $table->integer('harga_beli');
19              $table->integer('harga_jual');
20              $table->unsignedBigInteger('kategori_id')->index(); // indexing untuk foreign key
21              $table->timestamps();
22
23              $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori');
24          });
25      }
26
27      /**
28      * Reverse the migrations.
29      */
30      public function down(): void
31  {
32          Schema::table('m_barang', function (Blueprint $table) {
33              //
34          });
35      }
36  };
37 };
```

### Table t\_penjualan

```
database > migrations > 2025_03_01_065821_create_t_penjualan_table.php > class > up
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('t_penjualan', function (Blueprint $table) {
15             $table->id('penjualan_id');
16             $table->string('pembeli', 50);
17
18             $table->string('penjualan_kode', 20)->unique();
19             $table->dateTime('penjualan_tanggal');
20             $table->timestamps();
21             $table->unsignedBigInteger('user_id')->index();
22             $table->foreign('user_id')->references('user_id')->on('m_user');
23         });
24     }
25
26     /**
27     * Reverse the migrations.
28     */
29     public function down(): void
30     {
31         Schema::table('t_penjualan', function (Blueprint $table) {
32             //
33         });
34     }
35 };
36
```

### Table t\_stok

```
database > migrations > 2025_03_01_065859_create_t_stok_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      public function up(): void
10     {
11         Schema::create('t_stok', function (Blueprint $table) {
12             $table->id('stok_id'); // primary key auto increment
13             $table->integer('stok_jumlah');
14             $table->dateTime('stok_tanggal');
15             $table->timestamps();
16             $table->unsignedBigInteger('barang_id')->index();
17             $table->unsignedBigInteger('user_id')->index();
18
19             $table->foreign('barang_id')->references('barang_id')->on('m_barang');
20             $table->foreign('user_id')->references('user_id')->on('m_user');
21         });
22     }
23
24     public function down(): void
25     {
26         Schema::dropIfExists('t_stok');
27     }
28 };
29
```

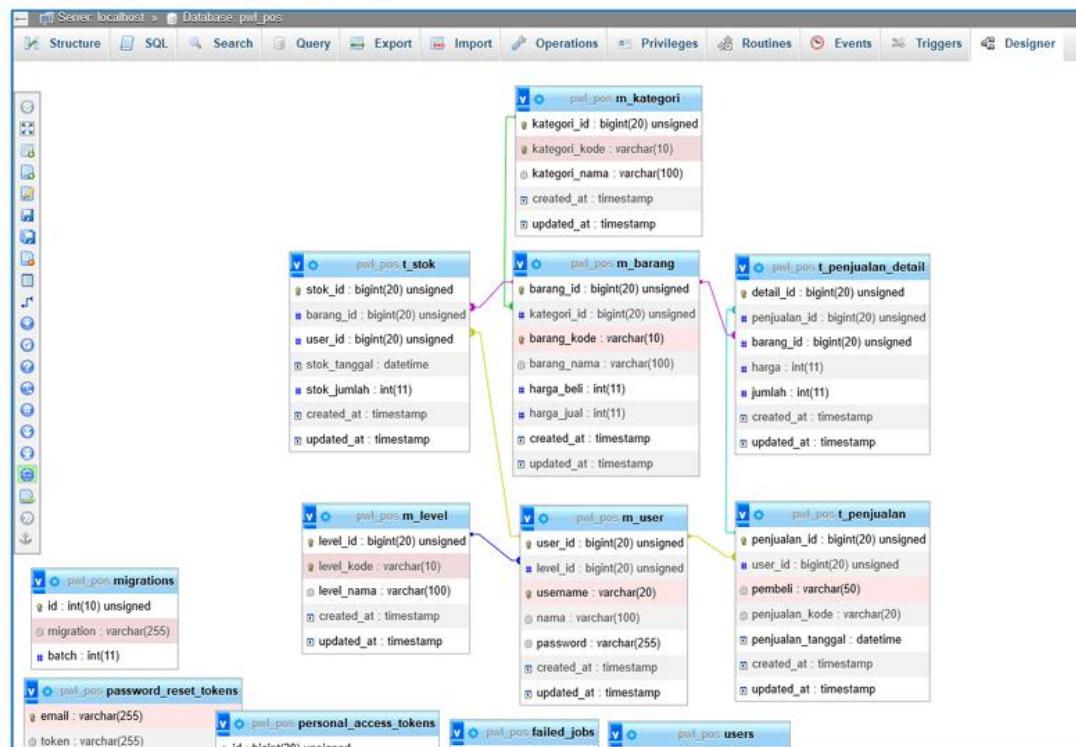
Table t\_penjualan\_detail

```

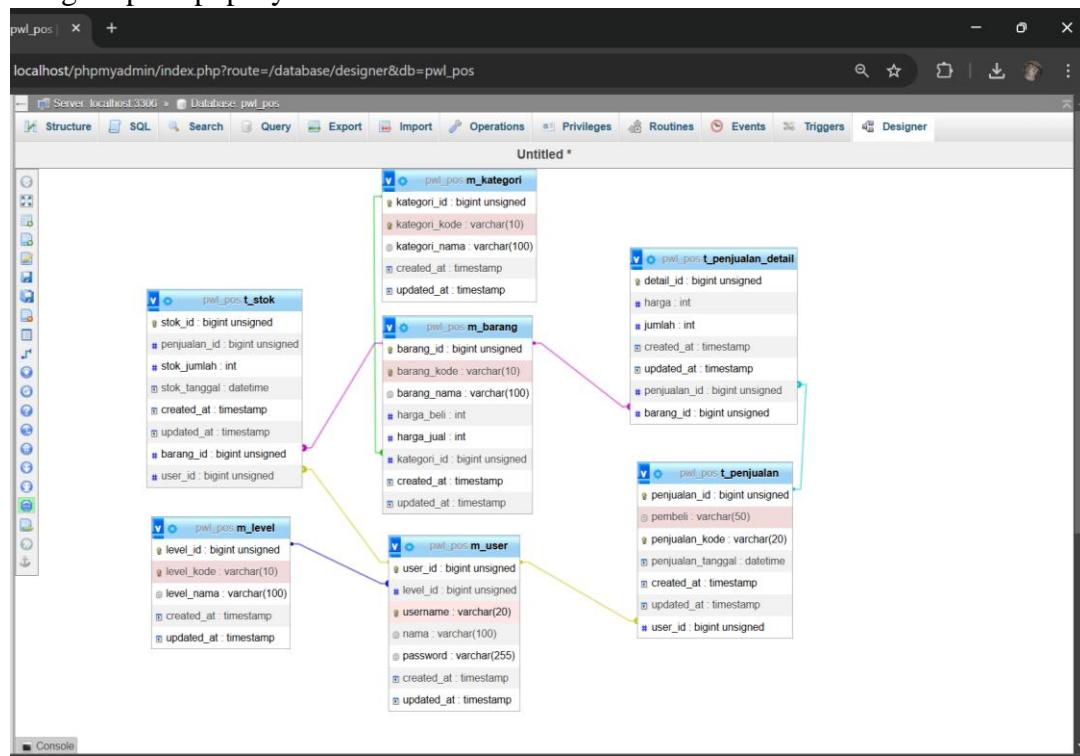
database > migrations > 2025_03_01_065935_create_t_penjualan_detail_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      public function up(): void
10     {
11         Schema::create('t_penjualan_detail', function (Blueprint $table) {
12             $table->id('detail_id');
13             $table->integer('harga');
14             $table->integer('jumlah');
15             $table->timestamps();
16             $table->unsignedBigInteger('penjualan_id')->index();
17             $table->unsignedBigInteger('barang_id')->index();
18             $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan');
19             $table->foreign('barang_id')->references('barang_id')->on('m_barang');
20         });
21     }
22
23     public function down(): void
24     {
25         Schema::dropIfExists('t_penjualan_detail');
26     }
27 };
28

```

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



⇒ designer pada phpMyAdmin

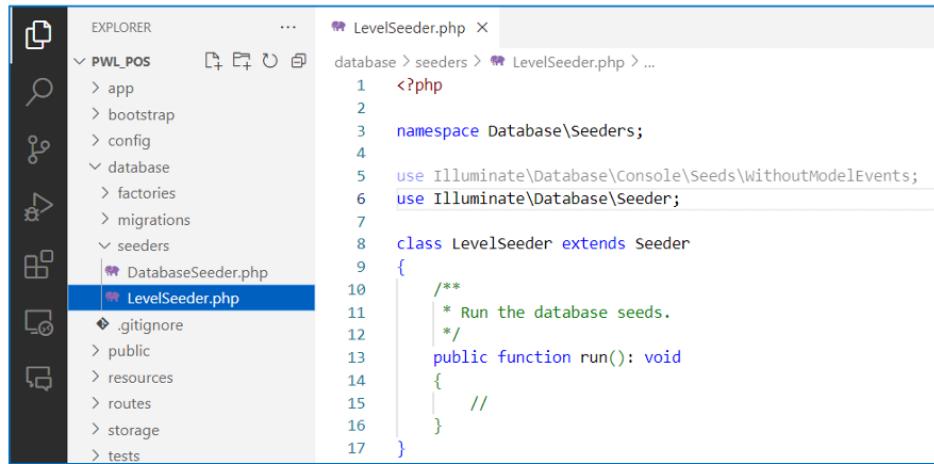


6. Laporkan hasil Praktikum-2.2 ini dan *commit* perubahan pada git.

## C. Seeder

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

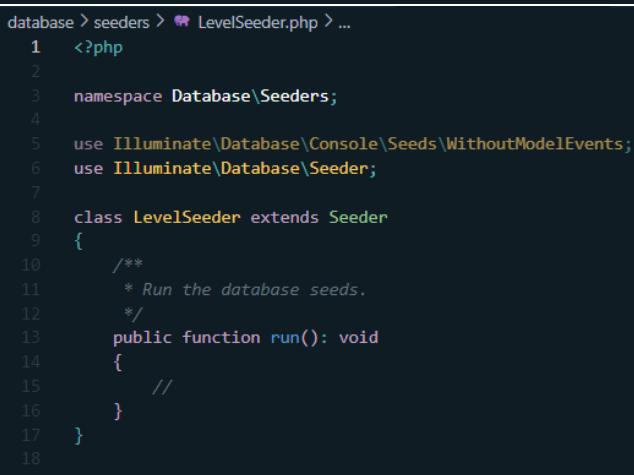
```
php artisan make:seeder LevelSeeder
```



```
database > seeders > LevelSeeder.php < ...  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  
8  class LevelSeeder extends Seeder  
9  {  
10     /**  
11      * Run the database seeds.  
12      */  
13     public function run(): void  
14     {  
15         //  
16     }  
17 }  
18 }
```

⇒ buat file seeder

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:seeder LevelSeeder  
[INFO] Seeder [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\seeders\LevelSeeder.php] created successfully.
```



```
database > seeders > LevelSeeder.php < ...  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  
8  class LevelSeeder extends Seeder  
9  {  
10     /**  
11      * Run the database seeds.  
12      */  
13     public function run(): void  
14     {  
15         //  
16     }  
17 }  
18 }
```

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

```

EXPLORER          ...  LevelSeeder.php X
PWL_POS
  > app
  > bootstrap
  > config
  > database
    > factories
    > migrations
    > seeders
      DatabaseSeeder.php
      LevelSeeder.php
        .gitignore
        public
        resources
        routes
        storage
        tests
        vendor
        .editorconfig
        .env
        .env.example
        .gitattributes
database > seeders > LevelSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class LevelSeeder extends Seeder
{
10
11  /**
12   * Run the database seeds.
13   */
14  public function run(): void
15  {
16      $data = [
17          ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
18          ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
19          ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
20      ];
21      DB::table('m_level')->insert($data);
22  }
23 }

```

⇒ modifikasi `run`

```

database > seeders > LevelSeeder.php > LevelSeeder > run
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class LevelSeeder extends Seeder
{
10
11  /**
12   * Run the database seeds.
13   */
14  public function run(): void
15  {
16      $data = [
17          ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
18          ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
19          ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
20      ];
21      DB::table('m_level')->insert($data);
22  }
23 }

```

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

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

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

```
INFO Seeding database.
```

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

⇒ run file seeder

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan db:seed --class=LevelSeeder
```

```
INFO Seeding database.
```

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

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

Check all    With selected:  Edit  Copy  Delete  Export

⇒ table `m_level`

Server: localhost:3306 » Database: pwl\_pos » Table: m\_level

Browse Structure SQL Search Insert Export Import Privileges Operation

Showing rows 0 - 2 (3 total, Query took 0.0012 seconds.)

SELECT \* FROM `m\_level`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

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

Check all    With selected:  Edit  Copy  Delete  Export

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

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

⇒ buat seeder untuk table `m_user`

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:seeder UserSeeder
INFO Seeder [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\seeders\UserSeeder.php] created successfully.
```

## 6. 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 }
```

⇒ Modifikasi UserSeeder

```
database > seeders > UserSeeder.php > UserSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8  use Illuminate\Support\Facades\Hash;
9
10 class UserSeeder extends Seeder
11 {
12     public function run(): void
13     {
14         $data = [
15             [
16                 'user_id' => 1,
17                 'level_id' => 1,
18                 'username' => 'admin',
19                 'nama' => 'Administrator',
20                 'password' => Hash::make('12345'), // class untuk mengenkripsi/hash password
21             ],
22             [
23                 'user_id' => 2,
24                 'level_id' => 3,
25                 'username' => 'manager',
26                 'nama' => 'Manager',
27                 'password' => Hash::make('12345'),
28             ],
29             [
30                 'user_id' => 3,
31                 'level_id' => 3,
32                 'username' => 'staff',
33                 'nama' => 'Staff/Kasir',
34                 'password' => Hash::make('12345'),
35             ],
36         ];
37         DB::table('m_user')->insert($data);
38     }
39 }
```

7. Jalankan perintah untuk mengeksekusi class `UserSeeder`

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

- ⇒ Execute UserSeeder

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan db:seed --class=UserSeeder
[INFO] Seeding database.
```

8. Perhatikan hasil seeder pada table `m_user`

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	user_id	level_id	username	nama	password
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$Tevu4dDO1CUAQpeM6H.Vp.LySwhY4oAKU7FzwS6XV...
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$Ajfnz20/FdTTeUgghz31muEhIFaruLxkh5wvZ9NGRp...
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$Gi23TqGclW5pYeR0VL4o5OxPwb3Osk99VMy/BHnbJ9W...

- ⇒ Hasil seeder

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** localhost:3306
  - Database:** pwl\_pos
  - Table:** m\_user
- Toolbar:** Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers.
- Message Bar:** Showing rows 0 - 2 (3 total, Query took 0.0012 seconds.)
- SQL Editor:** SELECT \* FROM `m\_user`
- Table Data:**

	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/>	1	1	admin	Administrator	\$2y\$12\$RtgYVo0HMUmmMpNVkrFhO2lsrFUJtO9wVjFXrp2lg...	NULL	NULL
<input type="checkbox"/>	2	3	manager	Manager	\$2y\$12\$nj/6WZEYMWQqs4uWa4wOD EvlyvnFNbYb/eGRyy5Gwz...	NULL	NULL
<input type="checkbox"/>	3	3	staff	Staff/Kasir	\$2y\$12\$lvvZ7HwotroD5VFU3K3yeVKsaWx8wNa10qZ4jRp1qV...	NULL	NULL

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

10. Sekarang coba kalian masukkan data *seeder* untuk table yang lain, dengan ketentuan seperti berikut

No	Nama Tabel	Jumlah Data	Keterangan
1	<a href="#">m_kategori</a>	5	5 kategori barang
2	<a href="#">m_barang</a>	10	10 barang yang berbeda
3	<a href="#">t_stok</a>	10	Stok untuk 10 barang
4	<a href="#">t_penjualan</a>	10	10 transaksi penjualan
5	<a href="#">t_penjualan_detail</a>	30	3 barang untuk setiap transaksi penjualan

- ⇒ Data seeder untuk 5 table lainnya

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:seeder StokSeeder
[INFO] Seeder [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\seeders\StokSeeder.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:seeder PenjualanSeeder
[INFO] Seeder [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\seeders\PenjualanSeeder.php] created successfully.

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:seeder PenjualanDetailSeeder
[INFO] Seeder [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\database\seeders\PenjualanDetailSeeder.php] created successfully.
```

## KategoriSeeder

```
database > seeders > KategoriSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class KategoriSeeder extends Seeder
10 {
11
12     public function run(): void
13     {
14         $data = [
15             [
16                 'kategori_id' => 1,
17                 'kategori_kode' => 'KAT001',
18                 'kategori_nama' => 'Elektronik',
19             ],
20             [
21                 'kategori_id' => 2,
22                 'kategori_kode' => 'KAT002',
23                 'kategori_nama' => 'Pakaian',
24             ],
25             [
26                 'kategori_id' => 3,
27                 'kategori_kode' => 'KAT003',
28                 'kategori_nama' => 'Alat Tulis',
29             ],
30             [
31                 'kategori_id' => 4,
32                 'kategori_kode' => 'KAT004',
33                 'kategori_nama' => 'Makanan',
34             ],
35             [
36                 'kategori_id' => 5,
37                 'kategori_kode' => 'KAT005',
38                 'kategori_nama' => 'Minuman',
39             ],
40         ];
41     }
42 }
```

The screenshot shows the phpMyAdmin interface for the 'm\_kategori' table. The table has columns: kategori\_id, kategori\_kode, kategori\_nama, created\_at, and updated\_at. The data is as follows:

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

## BarangSeeder

```
database > seeders > BarangSeeder.php > BarangSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class BarangSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             [
15                 'barang_id' => 1,
16                 'barang_kode' => 'BRG001',
17                 'barang_nama' => 'Laptop',
18                 'harga_beli' => 5000000,
19                 'harga_jual' => 7000000,
20                 'kategori_id' => 1,
21             ],
22             [
23                 'barang_id' => 2,
24                 'barang_kode' => 'BRG002',
25                 'barang_nama' => 'Smartphone',
26                 'harga_beli' => 3000000,
27                 'harga_jual' => 4000000,
28                 'kategori_id' => 1,
29             ],
30             [
31                 'barang_id' => 3,
32                 'barang_kode' => 'BRG003',
33                 'barang_nama' => 'T-Shirt',
34                 'harga_beli' => 50000,
35                 'harga_jual' => 100000,
36                 'kategori_id' => 2,
37             ],
38         ];
39     }
40 }
```

Server: localhost:3306 » Database: pwl\_pos » Table: m\_barang

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 9 (10 total, Query took 0.0009 seconds.)

SELECT \* FROM `m\_barang`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	barang_id	barang_kode	barang_nama	harga_beli	harga_jual	kategori_id	created_at	updated_at
<input type="checkbox"/>	1	BRG001	Laptop	5000000	7000000	1	NULL	NULL
<input type="checkbox"/>	2	BRG002	Smartphone	3000000	4000000	1	NULL	NULL
<input type="checkbox"/>	3	BRG003	T-Shirt	50000	100000	2	NULL	NULL
<input type="checkbox"/>	4	BRG004	Celana Jeans	150000	250000	2	NULL	NULL
<input type="checkbox"/>	5	BRG005	Pulpen	5000	7500	3	NULL	NULL
<input type="checkbox"/>	6	BRG006	Buku Tulis	3000	5000	3	NULL	NULL
<input type="checkbox"/>	7	BRG007	Mie Instan	2500	3500	4	NULL	NULL
<input type="checkbox"/>	8	BRG008	Air Mineral	1500	3000	5	NULL	NULL
<input type="checkbox"/>	9	BRG009	Keyboard	150000	250000	1	NULL	NULL
<input type="checkbox"/>	10	BRG010	Mouse	100000	150000	1	NULL	NULL

## StokSeeder

```
database > seeders > StokSeeder.php > StokSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class StokSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             [
15                 'stok_id' => 1,
16                 'stok_jumlah' => 100,
17                 'stok_tanggal' => '2024-02-27 08:00:00',
18                 'barang_id' => 1,
19                 'user_id' => 1,
20             ],
21             [
22                 'stok_id' => 2,
23                 'stok_jumlah' => 120,
24                 'stok_tanggal' => '2024-02-27 08:00:00',
25                 'barang_id' => 2,
26                 'user_id' => 1,
27             ],
28             [
29                 'stok_id' => 3,
30                 'stok_jumlah' => 10,
31                 'stok_tanggal' => '2024-02-27 08:00:00',
32                 'barang_id' => 3,
33                 'user_id' => 1,
34             ],
35             [
36                 'stok_id' => 4,
37                 'stok_jumlah' => 30,
38                 'stok_tanggal' => '2024-02-27 08:00:00',
39                 'barang_id' => 4,
40                 'user_id' => 1,
41             ],
42             [
43                 'stok_id' => 5,
44                 'stok_jumlah' => 50,
45                 'stok_tanggal' => '2024-02-27 08:00:00',
46                 'barang_id' => 5,
47                 'user_id' => 1,
48             ],
49             [
50                 'stok_id' => 6,
51                 'stok_jumlah' => 75,
52                 'stok_tanggal' => '2024-02-27 08:00:00',
53                 'barang_id' => 6,
54                 'user_id' => 1,
55             ],
56             [
57                 'stok_id' => 7,
58                 'stok_jumlah' => 200,
59                 'stok_tanggal' => '2024-02-27 08:00:00',
60                 'barang_id' => 7,
61                 'user_id' => 1,
62             ],
63             [
64                 'stok_id' => 8,
65                 'stok_jumlah' => 300,
66                 'stok_tanggal' => '2024-02-27 08:00:00',
67                 'barang_id' => 8,
68                 'user_id' => 1,
69             ],
70             [
71                 'stok_id' => 9,
72                 'stok_jumlah' => 45,
73                 'stok_tanggal' => '2024-02-27 08:00:00',
74                 'barang_id' => 9,
75                 'user_id' => 1,
76             ],
77             [
78                 'stok_id' => 10,
79                 'stok_jumlah' => 60,
80                 'stok_tanggal' => '2024-02-27 08:00:00',
81                 'barang_id' => 10,
82                 'user_id' => 1,
83             ],
84         ];
85         DB::table('t_stok')->insert($data);
86     }
87 }
```

Server: localhost:3306 » Database: pwl\_pos » Table: t\_stok

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 9 (10 total, Query took 0.0008 seconds.)

SELECT \* FROM `t\_stok`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	stok_id	stok_jumlah	stok_tanggal	created_at	updated_at	barang_id	user_id
<input type="checkbox"/>	Edit Copy Delete	1	100	2024-02-27 08:00:00	NULL	NULL	1
<input type="checkbox"/>	Edit Copy Delete	2	120	2024-02-27 08:00:00	NULL	NULL	2
<input type="checkbox"/>	Edit Copy Delete	3	10	2024-02-27 08:00:00	NULL	NULL	3
<input type="checkbox"/>	Edit Copy Delete	4	30	2024-02-27 08:00:00	NULL	NULL	4
<input type="checkbox"/>	Edit Copy Delete	5	50	2024-02-27 08:00:00	NULL	NULL	5
<input type="checkbox"/>	Edit Copy Delete	6	75	2024-02-27 08:00:00	NULL	NULL	6
<input type="checkbox"/>	Edit Copy Delete	7	200	2024-02-27 08:00:00	NULL	NULL	7
<input type="checkbox"/>	Edit Copy Delete	8	300	2024-02-27 08:00:00	NULL	NULL	8
<input type="checkbox"/>	Edit Copy Delete	9	45	2024-02-27 08:00:00	NULL	NULL	9
<input type="checkbox"/>	Edit Copy Delete	10	60	2024-02-27 08:00:00	NULL	NULL	10

## PenjualanSeeder

```
database > seeders > PenjualanSeeder.php > PenjualanSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class PenjualanSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             [
15                 'penjualan_id' => 1,
16                 'pembeli' => 'John Doe',
17                 'penjualan_kode' => 'PJLAN001',
18                 'penjualan_tanggal' => '2024-02-27 08:00:00',
19                 'user_id' => 1,
20             ],
21             [
22                 'penjualan_id' => 2,
23                 'pembeli' => 'Jane Smith',
24                 'penjualan_kode' => 'PJLAN002',
25                 'penjualan_tanggal' => '2024-02-27 08:30:00',
26                 'user_id' => 1,
27             ],
28             [
29                 'penjualan_id' => 3,
30                 'pembeli' => 'Michael Johnson',
31                 'penjualan_kode' => 'PJLAN003',
32                 'penjualan_tanggal' => '2024-02-27 09:00:00',
33                 'user_id' => 1,
34             ],
35             [
36                 'penjualan_id' => 4,
37                 'pembeli' => 'Sarah Williams',
38                 'penjualan_kode' => 'PJLAN004',
39                 'penjualan_tanggal' => '2024-02-27 09:15:00'
40             ]
41         ];
42     }
43 }
```

Server: localhost:3306 » Database: pw1\_pos » Table: t\_penjualan

Browse Structure SQL Search Insert Export Import Privileges Operations Trigger

Showing rows 0 - 9 (10 total, Query took 0.0007 seconds.)

SELECT \* FROM `t\_penjualan`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	penjualan_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at	user_id
<input type="checkbox"/>	1	John Doe	PJLAN001	2024-02-27 08:00:00	NULL	NULL	1
<input type="checkbox"/>	2	Jane Smith	PJLAN002	2024-02-27 08:30:00	NULL	NULL	1
<input type="checkbox"/>	3	Michael Johnson	PJLAN003	2024-02-27 09:00:00	NULL	NULL	1
<input type="checkbox"/>	4	Sarah Williams	PJLAN004	2024-02-27 09:15:00	NULL	NULL	1
<input type="checkbox"/>	5	Robert Brown	PJLAN005	2024-02-27 10:00:00	NULL	NULL	1
<input type="checkbox"/>	6	Emily Davis	PJLAN006	2024-02-27 10:30:00	NULL	NULL	1
<input type="checkbox"/>	7	David Wilson	PJLAN007	2024-02-27 11:00:00	NULL	NULL	1
<input type="checkbox"/>	8	Jessica Miller	PJLAN008	2024-02-27 11:30:00	NULL	NULL	1
<input type="checkbox"/>	9	Thomas Taylor	PJLAN009	2024-02-27 12:00:00	NULL	NULL	1
<input type="checkbox"/>	10	Amanda Anderson	PJLAN010	2024-02-27 12:30:00	NULL	NULL	1

## PenjualanDetailSeeder

```
database > seeders > PenjualanDetailSeeder.php > PenjualanDetailSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Seeder;
6  use Illuminate\Support\Facades\DB;
7
8  class PenjualanDetailSeeder extends Seeder
9  {
10     public function run(): void
11     {
12         $data = [];
13
14         $jumlah_transaksi = 10; // Total transaksi penjualan
15
16         for ($i = 1; $i <= $jumlah_transaksi; $i++) {
17             for ($barang_id = 1; $barang_id <= 3; $barang_id++) {
18                 $data[] = [
19                     'harga' => rand(100000, 2000000), // Harga acak
20                     'jumlah' => rand(1, 5),           // Jumlah acak
21                     'penjualan_id' => $i,
22                     'barang_id' => $barang_id,
23                 ];
24             }
25         }
26
27         DB::table('t_penjualan_detail')->insert($data);
28     }
29 }
30 }
```

Server: localhost:3306 > Database: pwl\_pos > Table: t\_penjualan\_detail

Browse Structure SQL Search Insert Export Import Privileges C

Showing rows 0 - 24 (30 total, Query took 0.0005 seconds.)

SELECT \* FROM `t\_penjualan\_detail`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

1 > >> Show all Number of rows: 25 Filter rows: Search this table Sort by

Extra options

	detail_id	harga	jumlah	created_at	updated_at	penjualan_id	barang_id
<input type="checkbox"/>	271	1555361	4	NULL	NULL	1	1
<input type="checkbox"/>	272	1413125	3	NULL	NULL	1	2
<input type="checkbox"/>	273	396404	3	NULL	NULL	1	3
<input type="checkbox"/>	274	197920	2	NULL	NULL	2	1
<input type="checkbox"/>	275	192665	1	NULL	NULL	2	2
<input type="checkbox"/>	276	953260	4	NULL	NULL	2	3
<input type="checkbox"/>	277	965380	2	NULL	NULL	3	1
<input type="checkbox"/>	278	267107	5	NULL	NULL	3	2
<input type="checkbox"/>	279	1662947	2	NULL	NULL	3	3
<input type="checkbox"/>	280	1733206	1	NULL	NULL	4	1
<input type="checkbox"/>	281	882299	5	NULL	NULL	4	2
<input type="checkbox"/>	282	1778815	3	NULL	NULL	4	3
<input type="checkbox"/>	283	1167776	2	NULL	NULL	5	1
<input type="checkbox"/>	284	1330788	3	NULL	NULL	5	2
<input type="checkbox"/>	285	973853	2	NULL	NULL	5	3
<input type="checkbox"/>	286	1960555	4	NULL	NULL	6	1
<input type="checkbox"/>	287	720436	4	NULL	NULL	6	2
<input type="checkbox"/>	288	621977	1	NULL	NULL	6	3
<input type="checkbox"/>	289	296381	4	NULL	NULL	7	1
<input type="checkbox"/>	290	720288	1	NULL	NULL	7	2

## D. DB Façade

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

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

- ⇒ create controller

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:controller LevelController
[INFO] Controller [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (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']);
```

- ⇒ routing



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

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

```

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 }

```

⇒ modifikasi LevelController

```

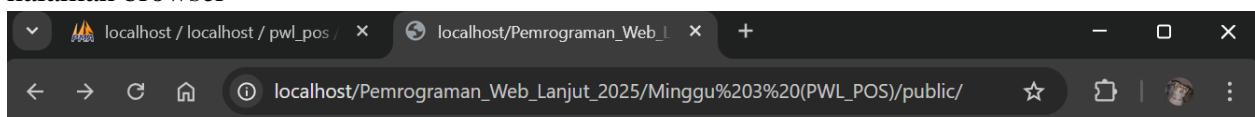
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         DB::insert('insert into m_level (level_kode, level_nama, created_at) values(?)', ['CUS',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

```

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`

		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	2024-02-26 08:20:00	NULL

⇒ halaman browser



insert data berhasil

Table `m_level` di database

Server: localhost:3306 » Database: pwl_pos » Table: m_level																																									
Browse		Structure	SQL	Search	Insert	Export																																			
<span style="color: green;">✓ Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)</span>																																									
<code>SELECT * FROM `m_level`</code>																																									
<input type="checkbox"/> Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]																																									
<input type="checkbox"/> Show all   Number of rows: 25		Filter rows:	Search this table	Sort by key:	None																																				
<a href="#">Extra options</a>																																									
<table border="1"> <thead> <tr> <th>← T →</th> <th>▼</th> <th>level_id</th> <th>level_kode</th> <th>level_nama</th> <th>created_at</th> <th>updated_at</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td><a href="#">Edit</a></td> <td><a href="#">Copy</a></td> <td><a href="#">Delete</a></td> <td>1 ADM</td> <td>Administrator</td> <td>NULL</td> </tr> <tr> <td><input type="checkbox"/></td> <td><a href="#">Edit</a></td> <td><a href="#">Copy</a></td> <td><a href="#">Delete</a></td> <td>2 MNG</td> <td>Manager</td> <td>NULL</td> </tr> <tr> <td><input type="checkbox"/></td> <td><a href="#">Edit</a></td> <td><a href="#">Copy</a></td> <td><a href="#">Delete</a></td> <td>3 STF</td> <td>Staff/Kasir</td> <td>NULL</td> </tr> <tr> <td><input type="checkbox"/></td> <td><a href="#">Edit</a></td> <td><a href="#">Copy</a></td> <td><a href="#">Delete</a></td> <td>4 CUS</td> <td>Pelanggan</td> <td>2025-03-01 14:08:26 NULL</td> </tr> </tbody> </table>							← T →	▼	level_id	level_kode	level_nama	created_at	updated_at	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1 ADM	Administrator	NULL	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2 MNG	Manager	NULL	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3 STF	Staff/Kasir	NULL	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4 CUS	Pelanggan	2025-03-01 14:08:26 NULL
← T →	▼	level_id	level_kode	level_nama	created_at	updated_at																																			
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1 ADM	Administrator	NULL																																			
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2 MNG	Manager	NULL																																			
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3 STF	Staff/Kasir	NULL																																			
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4 CUS	Pelanggan	2025-03-01 14:08:26 NULL																																			
<a href="#">Up</a> <input type="checkbox"/> <a href="#">Check all</a> With selected: <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> <a href="#">Export</a>																																									

5. Selanjutnya, kita modifikasi lagi file [LevelController](#) untuk meng-update data di table `m_level` seperti berikut

```
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 }
```

#### ⇒ Modifikasi LevelController

```
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
14         // return
15         // 'insert data berhasil';
16
17         $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
18         return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
19     }
20 }
```

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`

⇒ Halaman browser

Update data berhasil. Jumlah data yang diupdate: 1 baris

### Database

	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
<input type="checkbox"/>	4	CUS	Customer	2025-03-01 14:08:26	NULL

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

```

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 }

```

⇒ Modifikasi LevelController

```

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
14         // return
15         // 'insert data berhasil';
16
17         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
18         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
19
20         $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
21         return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
22     }
23 }

```

Delete data berhasil. Jumlah data yang dihapus: 1 baris

	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

  
 With selected:    
    
   

8. 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 }

```

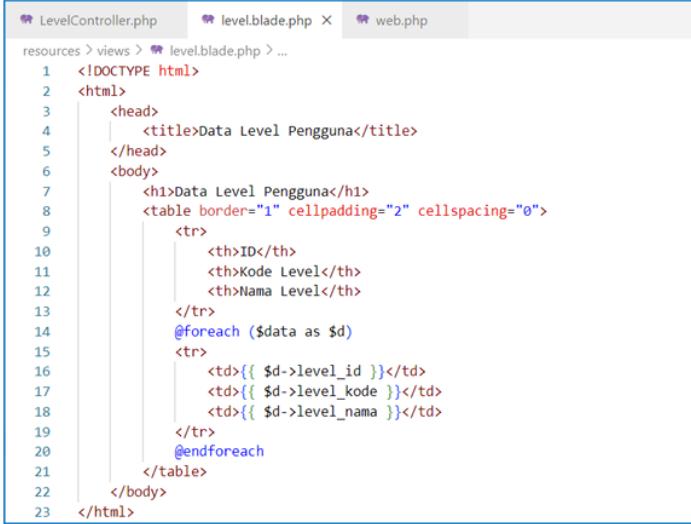
⇒ Menampilkan data pada m\_level

```

app > Http > Controllers > LevelController.php > LevelController > index
1
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         // DB::insert('insert into m_level (Level_kode, Level_nama, created_at) values(?, ?, ?)', ['CUS', 'PeLanggan', now()]);
12
13         // return
14         // 'insert data berhasil';
15
16         // $row = DB::update('update m_Level set Level_nama = ? where Level_kode = ?', ['Customer', 'CUS']);
17         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row. ' baris';
18
19         // $row = DB::delete('delete from m_Level where Level_kode = ?', ['CUS']);
20         // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row. ' baris';
21
22         $data = DB::select('select * from m_level');
23         return view('level', ['data' => $data]);
24
25     }
26 }

```

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`



```

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>

```

⇒ Create view

```

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:view level
[INFO] View [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\resources\views\level.blade.php] created successfully.

```

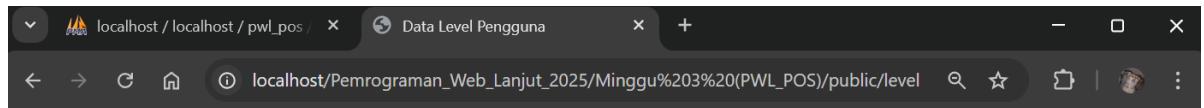
```

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>
24

```

10. Silahkan dicoba pada browser dan amati apa yang terjadi

⇒ Halaman browser



### Data Level Pengguna

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

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

## E. Query Builder

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

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

- ⇒ Buat controller `m_kategori`

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:controller KategoriController
[INFO] Controller [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (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']);
```

- ⇒ Modifikasi route

```
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  |--------------------------------------------------------------------------
9  | Web Routes
10 |--------------------------------------------------------------------------
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider and all of them will
14 | be assigned to the "web" middleware group. Make something great!
15 |
16 */
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
22
23 Route::get('/level', [LevelController::class, 'index']);
24 Route::get('/kategori', [KategoriController::class, 'index']);
```

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

```

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 }

```

⇒ KategoriController

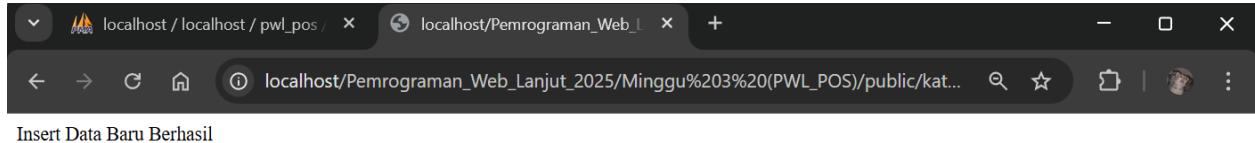
```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8
9 class KategoriController extends Controller
10 {
11     public function index(){
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`

⇒ Halaman browser



Insert Data Baru Berhasil

Database

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	KAT001	Elektronik	NULL	NULL
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	KAT002	Pakalan	NULL	NULL
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3	KAT003	Alat Tulis	NULL	NULL
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4	KAT004	Makanan	NULL	NULL
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	5	KAT005	Minuman	NULL	NULL
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	6	SNK	Snack/Makanan Ringan	2025-03-01 14:48:45	NULL

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

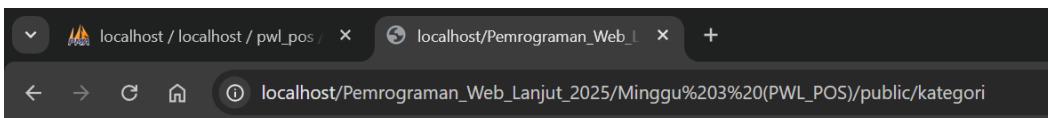
```
app > Http > Controllers > KategoriController.php > KategoriController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class KategoriController extends Controller
9 {
10     public function index()
11     {
12         /*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

⇒ Modifikasi

```
app > Http > Controllers > KategoriController.php > KategoriController > 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         // $data = [
12         //     'kategori.kode' => 'SNK',
13         //     'kategori.nama' => 'Snack/Makanan Ringan',
14         //     'created_at' => now()
15         // ];
16         // DB::table('m_kategori')->insert($data);
17         // return 'Insert Data Baru Berhasil';
18
19         $row = DB::table('m_kategori')->where('kategori.kode', 'SNK')->update(['kategori.nama' => 'Camilan']);
20         return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
21
22     }
23 }
```

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`

⇒ Halaman browser



Update data berhasil. Jumlah data yang diupdate: 1 baris

## Database

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	KAT001	Elektronik	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2	KAT002	Pakaian	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3	KAT003	Alat Tulis	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4	KAT004	Makanan	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5	KAT005	Minuman	NULL	NULL
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	6	SNK	Camilan	2025-03-01 14:48:45	NULL

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

```

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     }

```

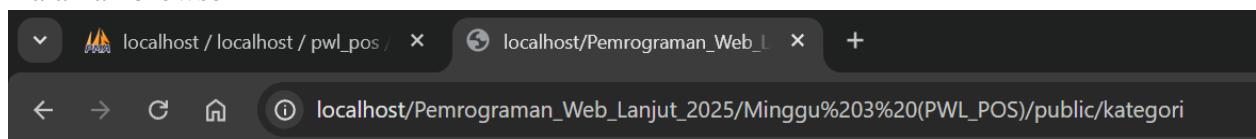
⇒ Modified

```

8
9 class KategoriController extends Controller
10 {
11     public function index(){
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')
24             ->where('kategori.kode', 'SNK')
25             ->delete();
26
27         return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
28
29     }
30 }

```

Halaman browser



Delete data berhasil. Jumlah data yang dihapus: 1 baris

## Database

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

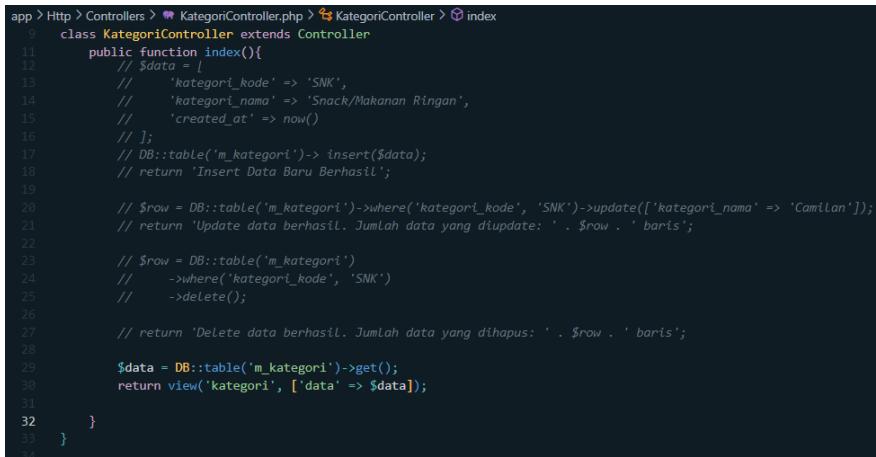
8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table `m_kategori`. Kita modifikasi file `KategoriController` seperti berikut

```

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    }

```

⇒ Modifikasi



```

app > Http > Controllers > KategoriController.php > KategoriController > index
9 class KategoriController extends Controller
10 public function index(){
11     /* $data = [
12         'kategori_kode' => 'SNK',
13         'kategori_nama' => 'Snack/Makanan Ringan',
14         'created_at' => now()
15     ];
16     DB::table('m_kategori')->insert($data);
17     return 'Insert Data Baru Berhasil';
18
19     // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
20     // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
21
22     // $row = DB::table('m_kategori')
23     //     ->where('kategori_kode', 'SNK')
24     //     ->delete();
25
26     // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
27
28     $data = DB::table('m_kategori')->get();
29     return view('kategori', ['data' => $data]);
30
31 }
32
33 }

```

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`

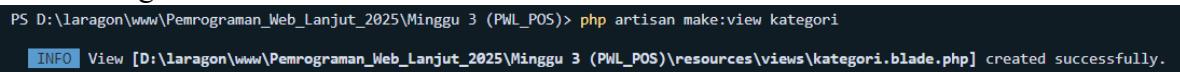


```

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>

```

⇒ View 'kategori'



```

PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:view kategori
INFO  View [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)\resources\views\kategori.blade.php] created successfully.

```

```

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.

⇒ Halaman browser



## Data Kategori Barang

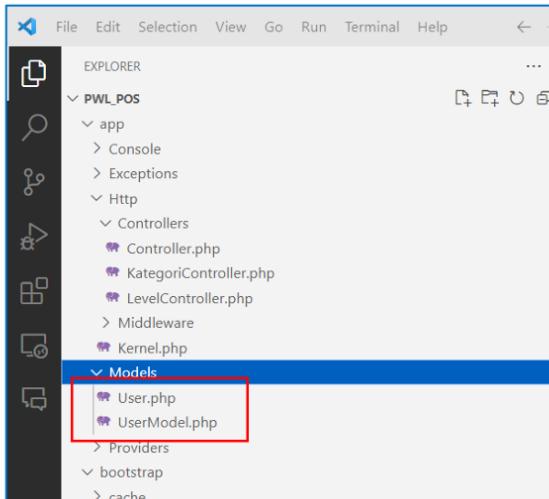
ID	Kode Kategori	Nama Kategori
1	KAT001	Elektronik
2	KAT002	Pakaian
3	KAT003	Alat Tulis
4	KAT004	Makanan
5	KAT005	Minuman

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

## F. Eloquent Form

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

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



⇒ Make model

```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (PWL_POS)> php artisan make:model UserModel
[INFO] Model [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 3 (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
```

⇒ Modifikasi

```
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 ini
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']);
16
17 */
```

⇒ Modifikasi route

```
routes > 📄 web.php > ...
4  use App\Http\Controllers\LevelController;
5  use App\Http\Controllers\UserController;
6  use Illuminate\Support\Facades\Route;
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 Route::get('/', function () {
20     return view('welcome');
21 });
22
23
24 Route::get('/level', [LevelController::class, 'index']);
25 Route::get('/kategori', [KategoriController::class, 'index']);
26 Route::get('/user', [UserController::class, 'index']);
27 */
```

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 }
```

⇒ Create 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
8  class UserController extends Controller
9  {
10     public function show($id, $name) {
11         //coba akses model UserModel
12         $user = UserModel::all(); //ambil semua data dari m_user
13         return view('user', ['data' => $user]);
14     }
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>

```

⇒ User view

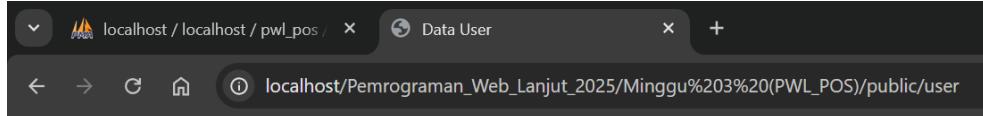
```

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>

```

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

⇒ Halaman browser



## Data User

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

Dari yang dapat dilihat halaman browser berhasil menampilkan data user dari tabel m\_user dalam bentuk tabel. Proses ini menunjukkan bahwa koneksi ke database, pengambilan data melalui model, serta pengiriman data ke view sudah berjalan dengan baik.

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 }
```

⇒ Modifikasi

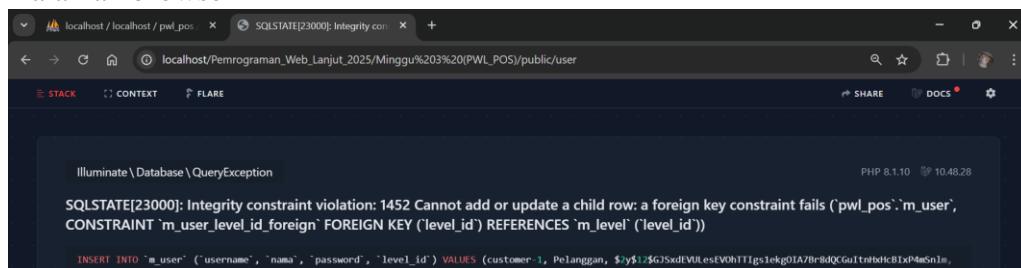
```

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 }
27

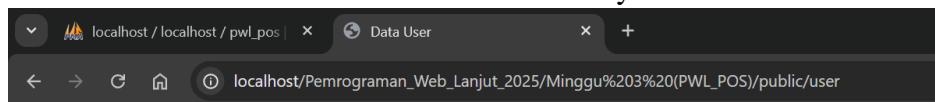
```

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

⇒ Halaman browser



Terjadi error, dikarenakan level\_id 4 belum ada di database, sehingga perlu menambahkan dulu secara manual di databasenya sendiri



## Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	3
3	staff	Staff/Kasir	3
10	customer-1	Pelanggan	4

Ketika kode dijalankan, sistem berhasil menambahkan data user baru bernama *Pelanggan* dengan username *customer-1* ke dalam tabel m\_user, lalu ditampilkan di halaman *Data User*, setelah level\_id yang digunakan ditambahkan terlebih dahulu.

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

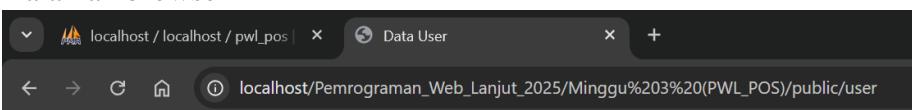
```
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data user dengan Eloquent Model
14         $data = [
15             'nama' => 'Pelanggan Pertama',
16         ];
17         UserModel::where('username', 'customer-1')->update($data); // update data user
18
19         // coba akses model UserModel
20         $user = UserModel::all(); // ambil semua data dari tabel m_user
21         return view('user', ['data' => $user]);
22     }
23 }
```

⇒ Modifikasi

```
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
13         // $data = [
14             // 'username' => 'customer-1',
15             // 'nama' => 'Pelanggan',
16             // 'password' => Hash::make('12345'),
17             // 'level_id' => 4
18             // ];
19
20         //tambah data dengan Eloquent Model
21         $data = [
22             'nama' => 'Pelanggan Pertama',
23         ];
24
25         // UserModel::insert($data); //tambahkan data ke table m_user
26         UserModel::where('username', 'customer-1')->update($data); //update data user
27
28
29         // // Ambil semua data dari tabel m_user (sesuai model UserModel)
30         $user = UserModel::all();
31
32         // Kirim data ke view user.blade.php
33         return view('user', ['data' => $user]);
34     }
35 }
```

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

⇒ Halaman browser



## Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	3
3	staff	Staff/Kasir	3
10	customer-1	Pelanggan Pertama	4

12. Jika sudah, lapor hasil Praktikum-6 ini dan *commit* perubahan pada *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?
2. Pada **Praktikum 1**, bagaimana kita men-generate nilai untuk `APP_KEY`?
3. Pada **Praktikum 2.1 - Tahap 1**, secara *default* Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?
4. Secara *default*, file migrasi terdapat kode `$table->timestaps();`, apa tujuan/*output* dari fungsi tersebut?
5. Pada File Migrasi, terdapat fungsi `$table->id();` Tipe data apa yang dihasilkan dari fungsi tersebut?
6. Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');` ?
7. Pada migration, Fungsi `->unique()` digunakan untuk apa?
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')` ?
9. Pada **Praktikum 3 - Tahap 6**, apa tujuan dari Class `Hash`? dan apa maksud dari kode program `Hash::make('1234');`?
10. Pada **Praktikum 4 - Tahap 3/5/7**, pada *query builder* terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
11. Pada **Praktikum 6 - Tahap 3**, apa tujuan penulisan kode `protected $table = 'm_user';` dan `protected $primaryKey = 'user_id';` ?
12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (*DB Façade / Query Builder / Eloquent ORM*) ? jelaskan

Jawaban ;

1. APP\_KEY adalah sebuah konfigurasi pada file ‘.env’ di laravel yang digunakan untuk menyimpan kunci enkripsi aplikasi. Kunci ini digunakan untuk menjaga keamanan data yang dienkripsi, seperti session cookies dan data lainnya.
2. Untuk generate nilai ‘APP\_KEY’, kita dapat menggunakan perintah artisan didalam terminal dengan menggunakan ‘php artisan key:generate’.
3. Secara default, laravel memiliki tiga file migrasi: ‘create\_users\_table.php’ ‘create\_password\_resets\_table.php’, dan ‘cretae\_failed\_jobs\_table’. File ini digunakan untuk membuat table pengguna (users), reset password, dan table log untuk failed jobs.
4. Kode ‘\$table->timestaps();’ secara otomatis menambahkan dua kolom, yaitu ‘created\_at’ dan ‘updated\_at’ yang digunakan untuk melacak waktu pembuatan dan pembaruan suatu record dalam tabel.

5. Fungsi ‘\$table->ide();’ menghasilkan tipe data ‘unsigned big integer’ yang secara otomatis bertambah nilainya (auto-incrementing). Digunakan untuk menentukan kolom primary key dalam sebuah tabel.
6. Perbedaan antara menggunakan ‘\$table->id();’ dan ‘\$table->id(‘level\_id’);’ pada hasil migrasi pada tabel ‘m\_level’ menggunakan ‘->id(‘level\_id’);’ memberi nama kolom primary key secara eksplisit, yaitu ‘level\_id’. Sedangkan ‘->id();’ secara implisit.
7. Fungsi ‘->unique()’ pada migration digunakan untuk menentukan nilai pada kolom tersebut harus unik, tidak boleh ada duplikat nilai antara baris-baris dalam tabel.
8. Kolom ‘level\_id’ pada tabel ‘m\_user’ menggunakan ‘\$table->unsignedBigInteger(‘level\_id’);’ diasumsikan sebagai foreign key yang merujuk ke primary key di tabel lain. Sedangkan pada tabel ‘m\_level’, ‘\$table->id(‘level\_id’);’ digunakan untuk mendefinisikan primary key dengan nama kolom ‘level\_id’.
9. Class ‘Hash’ digunakan untuk melakukan enkripsi data, seperti enkripsi password pengguna. Kode program ‘Hash::make(‘12345’);’ menghasilkan hasil enkripsi dari string ‘12345’ yang biasanya digunakan saat membuat dan menyimpan password baru pengguna ke dalam database.
10. Tanda tanya (?) pada query builder digunakan untuk menandai parameter yang akan di-bind secara dinamis ke dalam query, memungkinkan penggunaan parameter query dan melindungi terhadap serangan SQL Injection.
11. Penulisan kode berikut adalah bagian dari model Eloquent dalam framework laravel. Tujuan dari kode ini adalah untuk menghubungkan model tersebut dengan tabel tertentu dalam basis data dan menetapkan primary key dari tabel tersebut.

```
$table->id();


|  |  |
|--|--|
|  |  |
|--|--|


```
12. Menurut saya pribadi lebih mudah jika menggunakan Query Builder dikarenakan memiliki sintaks yang digunakan untuk mempermudah penggunaan query dan juga sintaks tersebut mudah dipahami dibanding dengan Eloquent Orm yang sintaks nya sulit untuk dipahami dan juga pada DB Facade yang penulisan sintaksnya masih manual.