



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 6 (enam)
Pertemuan ke- : 1 (satu)

Nama : Aqila Nur Azza

JOBSHEET 03

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

Sebelumnya kita sudah membahas mengenai *Routing*, *Controller*, dan *View* yang ada di Laravel. Sebelum kita masuk pada pembuatan aplikasi berbasis website, alangkah baiknya kita perlu menyiapkan Basis data sebagai tempat menyimpan data-data pada aplikasi kita nanti. Selain itu, umumnya kita perlu menyiapkan juga data awal yang kita gunakan sebelum membuat aplikasi, seperti data user administrator, data pengaturan sistem, dll.

Untuk itu, kita memerlukan teknik untuk merancang/membuat table basis data sebelum membuat aplikasi. Laravel memiliki fitur dalam pengelolaan basis data seperti, migration, seeder, model, dll.

Sebelum kita masuk materi, kita buat dulu project baru yang akan kita gunakan untuk membangun aplikasi sederhana dengan topik *Point of Sales (PoS)*, sesuai dengan **Studi Kasus PWL.pdf**.

Jadi kita bikin project Laravel 10 dengan nama **PWL_POS**.

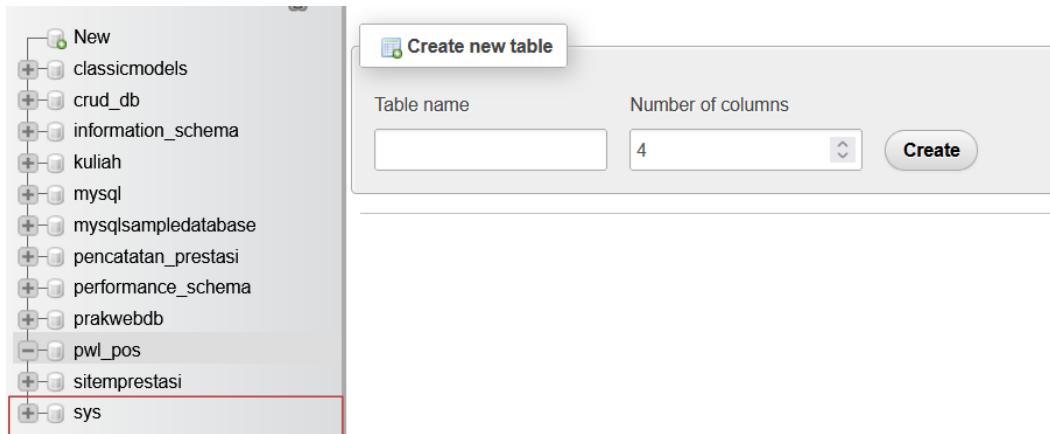
Project PWL_POS akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajarai

A. PENGATURAN DATABASE

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

Praktikum 1 - pengaturan database:

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



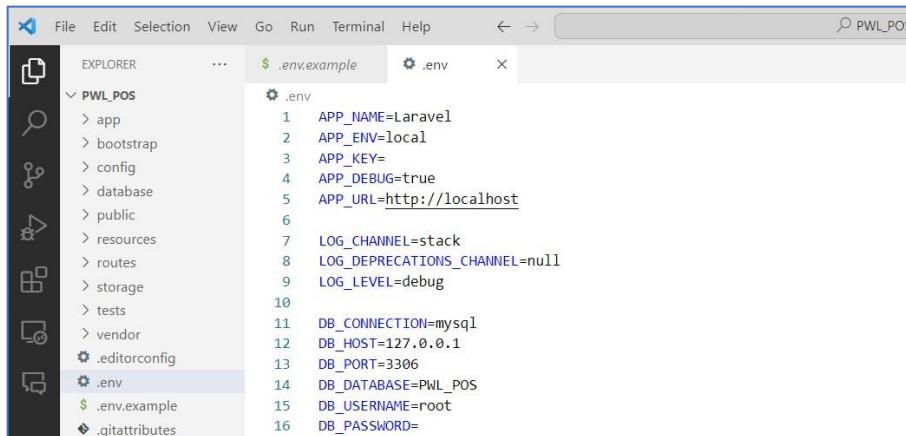
The screenshot shows the MySQL Workbench interface. On the left, a tree view displays various databases: New, classicmodels, crud_db, information_schema, kuliah, mysql, mysqlsampledatabse, pencatatan_prestasi, performance_schema, prakwebdb, pwli_pos, sitemprestasi, and sys. The 'sys' database is highlighted with a red box. To the right, a modal window titled 'Create new table' is open, prompting for a 'Table name' (which is empty) and a 'Number of columns' (set to 4), with a 'Create' button.

2. Buka aplikasi VSCode dan buka folder project **PWL_POS** yang sudah kita buat

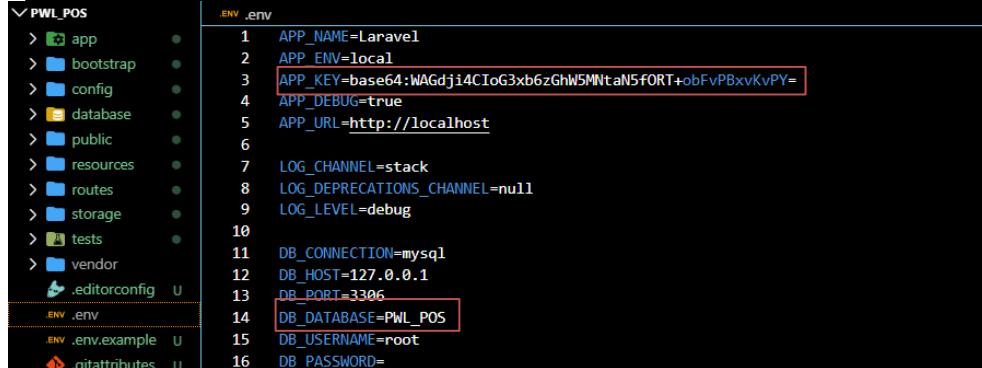
```
Bl packages you are using are looking for funding.  
Use the 'composer fund' command to find out more!  
> @php artisan vendor:publish --tag=laravel-assets --ansi --force  
  
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
INFO No publishable resources for tag [laravel-assets].  
  
No security vulnerability advisories found.  
> @php artisan key:generate --ansi  
  
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
INFO Application key set successfully.
```



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**.



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



The image shows a file explorer window with a sidebar and a main content area. The sidebar contains a tree view of a directory structure:

- PWL_POS
 - app
 - bootstrap
 - config
 - database
 - public
 - resources
 - routes
 - storage
 - tests
 - vendor
 - .editorconfig
 - .env
 - .env.example
 - .gitattributes

The main content area displays the contents of the '.env' file:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:WAGdji4CIoG3xb6zGhW5MNtaN5f0RT+obFvPBxvKvPY=
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=
```

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



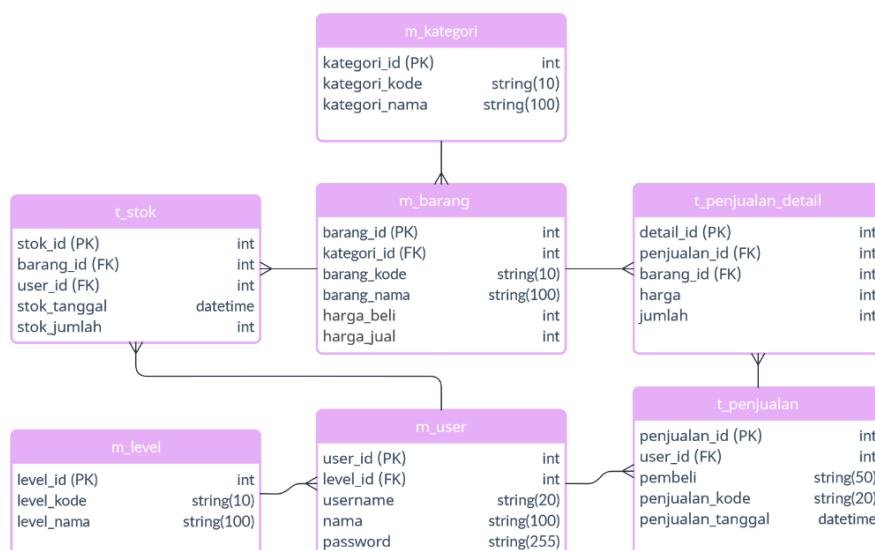
B. MIGRATION

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

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

Sesuai dengan topik pembelajaran kita untuk membangun sistem *Point of Sales (PoS)* sederhana, maka kita perlu membuat migration sesuai desain database yang sudah didefinisikan pada file

Studi Kasus PWL.pdf



Dalam membuat file migration di Laravel, yang perlu kita perhatikan adalah struktur table yang ingin kita buat.

TIPS MIGRATION

Buatlah file migration untuk table yang tidak memiliki relasi (table yang tidak ada *foreign key*) dulu, dan dilanjutkan dengan membuat file migrasi yang memiliki relasi yang sedikit, dan dilanjut ke file migrasi dengan table yang memiliki relasi yang banyak.

Dari tips di atas, kita dapat melakukan cek untuk desain database yang sudah ada dengan mengetahui jumlah *foreign key* yang ada. Dan kita bisa menentukan table mana yang akan kita buat migrasinya terlebih dahulu.



No Urut	Nama Tabel	Jumlah FK
1	m_level	0
2	m_kategori	0
3	m_user	1
4	m_barang	1
5	t_penjualan	1
6	t_stok	2
7	t_penjualan_detail	2

INFO

Secara default Laravel sudah ada table [users](#) untuk menyimpan data pengguna, tapi pada praktikum ini, kita gunakan table sesuai dari file [Studi Kasus PWL.pdf](#) yaitu [m_user](#).

Pembuatan file migrasi bisa menggunakan 2 cara, yaitu

- Menggunakan [artisan](#) untuk membuat *file migration*

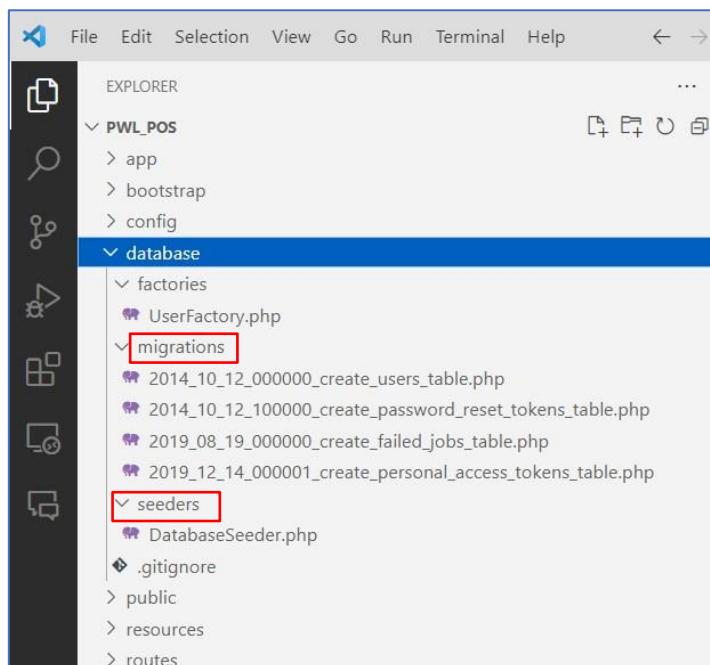
```
php artisan make:migration <nama-file-tabel> --create=<nama-tabel>
```

- Menggunakan [artisan](#) untuk membuat *file model + file migration*

```
php artisan make:model <nama-model> -m
```

Perintah [-m](#) di atas adalah *shorthand* untuk opsi membuat file migrasi berdasarkan model yang dibuat.

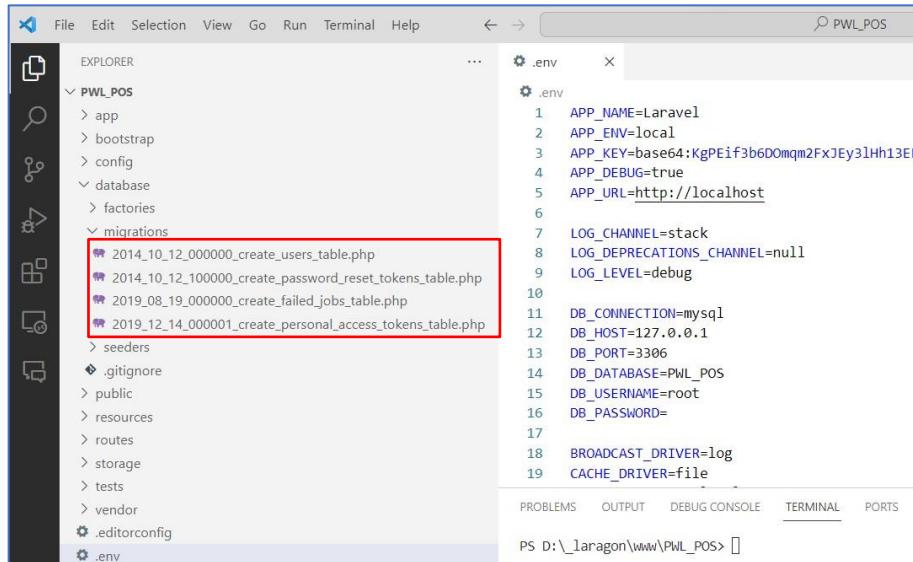
Pada Laravel, file-file *migration* ataupun *seeder* berada pada folder [PWL_POS/database](#)





Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

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



```
File Edit Selection View Go Run Terminal Help
EXPLORER .env
PWL_POS
  app
  bootstrap
  config
  database
    factories
    migrations
      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
      2019_12_14_000001_create_personal_access_tokens_table.php
    seeders
    .gitignore
    public
    resources
    routes
    storage
    tests
    vendor
  .editorconfig
  .env
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\_laragon\www\PWL_POS>
```

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

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration create_m_level_table --create=m_level
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_150322_create_m_level_table.php] created successfully.
```



```
database > migrations > 2025_03_150322_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     /**
11      * Run the migrations.
12      */
13
14     public function up(): void
15     {
16         Schema::create('m_level', function (Blueprint $table) {
17             $table->id();
18             $table->timestamps();
19         });
20
21         /**
22          * Reverse the migrations.
23          */
24         public function down(): void
25         {
26             Schema::dropIfExists('m_level');
27         }
28     };

```



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

```
database > migrations > 2025_03_03_162617_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('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  };
```

INFO

Dalam fitur migration Laravel, terdapat berbagai macam function untuk membuat kolom di table database. Silahkan cek disini

<https://laravel.com/docs/10.x/migrations#available-column-types>

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

```
php artisan migrate
```

```
INFO Preparing database.
Creating migration table ..... 445ms DONE
INFO Running migrations.

2014_10_12_000000_create_users_table ..... 166ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 28ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 105ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 68ms DONE
2025_03_03_150406_create_m_level_table ..... 62ms DONE
```



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

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	-
m_level		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 B

7. Ok, table sudah dibuat di database

8. Buat table *database* dengan *migration* untuk table **m_kategori** yang sama-sama tidak memiliki *foreign key*

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration create_m_kategori_table --create=m_kategori
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Migration [C:/laragon/www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_03_160328_create_m_kategori_table.php] created successfully.
```

```
database > migrations > 2025_03_03_160328_create_m_kategori_table.php > class > up > Closure
 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      /**
11      * Run the migrations.
12      */
13      public function up(): void
14      {
15          Schema::create('m_kategori', function (Blueprint $table) {
16              $table->id('kategori_id');
17              $table->string('kategori_kode', 10)->unique();
18              $table->string('kategori_nama', 100);
19              $table->timestamps();
20          });
21      }
22
23      /**
24      * Reverse the migrations.
25      */
26      public function down(): void
27      {
28          Schema::dropIfExists('m_kategori');
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**

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration create_m_user_table --table=m_user
```

```
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
```

```
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
```

```
INFO Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_03_161010_create_m_user_table.php] created successfully.
```

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

```
database > migrations > 2025_03_03_161010_create_m_user_table.php > class > up > Closure
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::table('m_user', function (Blueprint $table) {
15            $table->id('user_id');
16            $table->unsignedBigInteger('level_id')->index(); //index untuk foreignkey
17            $table->string('username', 20)->unique(); //unique untuk memastikan tidak ada username yang sama
18            $table->string('name', 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    /**
29     * Reverse the migrations.
30     */
31    public function down(): void
32    {
33        Schema::table('m_user', function (Blueprint $table) {
34            // ...
35        });
36    }
37 };
38
```



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

```
2025_03_03_164030_tion_create_m_user_table--create=m_user 120
5 DONE
```

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

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

m_barang
t_penjualan
t_stok
t_penjualan_detail

➤ **m_barang**

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration create_m_barang_table --table=m_barang
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations/2025_03_03_164925_create_m_barang_table.php] created successfully.
```



```
database > migrations > 2025_03_03_164925_create_m_barang_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_barang', function (Blueprint $table) {
15            $table->id('barang_id');
16            $table->unsignedBigInteger('kategori_id');
17            $table->string('barang_kode', 10)->unique();
18            $table->string('barang_nama', 100);
19            $table->integer('harga_beli');
20            $table->integer('harga_jual');
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::dropIfExists('m_barang');
33    }
34};|
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan migrate
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Running migrations.
2025_03_03_164925_create_m_barang_table ..... 81ms DONE
```

➤ t_penjualan

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration
  create_t_penjualan_table --table=t_penjualan
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_03_165300_create_t_penjualan_table.php] created successfully.
```



```
database > migrations > 2025_03_03_165300_craete_t_penjualan_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('t_penjualan', function (Blueprint $table) {
15            $table->id('penjualan_id');
16            $table->unsignedBigInteger('user_id');
17            $table->string('pembeli', 50);
18            $table->string('penjualan_kode', 20)->unique();
19            $table->dateTime('penjualan_tanggal');
20            $table->timestamps();
21
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::dropIfExists('t_penjualan');
32    }
33};
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan migrate
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO | Running migrations.
2025_03_03_165300_craete_t_penjualan_table ..... 80ms DONE
```

➤ t_stok

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration
create_t_stok_table --table=t_stok
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO | Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_03_165440_create_t_stok_table.php] created successfully.
```



```
database > migrations > 2025_03_03_165440_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      /**
10      * Run the migrations.
11      */
12      public function up(): void
13      {
14          Schema::create('t_stok', function (Blueprint $table) {
15              $table->id('stok_id');
16              $table->unsignedBigInteger('barang_id');
17              $table->unsignedBigInteger('user_id');
18              $table->dateTime('stok_tanggal');
19              $table->integer('stok_jumlah');
20              $table->timestamps();
21
22              $table->foreign('barang_id')->references('barang_id')->on('m_barang');
23              $table->foreign('user_id')->references('user_id')->on('m_user');
24          });
25      }
26
27      /**
28      * Reverse the migrations.
29      */
30      public function down(): void
31      {
32          Schema::dropIfExists('t_stok');
33      }
34  };
```

```
C:\Laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan migrate
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Running migrations.
2025_03_03_165440_create_t_stok_table ..... 103ms DONE
```

➤ **t_penjualan_detail**

```
C:\Laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:migration create_t_penjualan_detail_table --table=t_penjualan_detail
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Migration [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\migrations\2025_03_03_165935_create_t_penjualan_detail_table.php] created successfully.
```

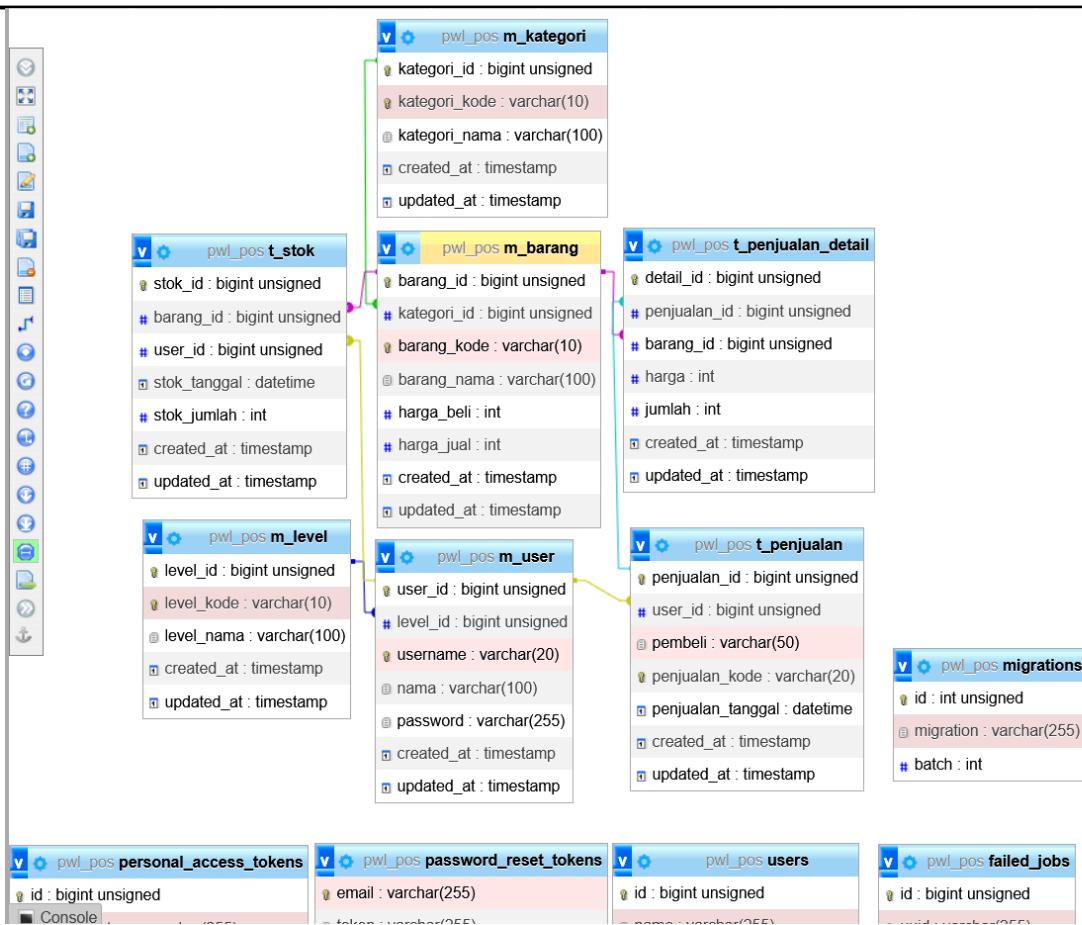


```
database > migrations > 2025_03_03_165935_create_t_penjualan_detail_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9
10    /**
11     * Run the migrations.
12     */
13    public function up(): void
14    {
15        Schema::create('t_penjualan_detail', function (Blueprint $table) {
16            $table->id('detail_id');
17            $table->unsignedBigInteger('penjualan_id');
18            $table->unsignedBigInteger('barang_id');
19            $table->integer('harga');
20            $table->integer('jumlah');
21            $table->timestamps();
22
23            $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan');
24            $table->foreign('barang_id')->references('barang_id')->on('m_barang');
25        });
26
27        /**
28         * Reverse the migrations.
29         */
30        public function down(): void
31        {
32            Schema::dropIfExists('t_penjualan_detail');
33        }
34    };
}
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan migrate
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO: Running migrations.
2025_03_03_165935_create_t_penjualan_detail_table 106ms DONE
```

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

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations		11	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_barang		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
m_kategori		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_user		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
password_reset_tokens		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
t_penjualan		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
t_penjualan_detail		0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
t_stok		0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
users		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
12 tables	Sum	11	InnoDB	utf8mb4_0900_ai_ci	304.0 KiB	0 B



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



C. SEEDER

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

1. Perintah umum dalam membuat **file seeder** adalah seperti berikut

```
php artisan make:seeder <nama-class-seeder>
```

Perintah tersebut akan men-generate file seeder pada folder **PWL_POS/database/seeders**

2. Dan perintah untuk **menjalankan file seeder** seperti berikut

```
php artisan db:seed --class=<nama-class-seeder>
```

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

Praktikum 3 – Membuat file seeder

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

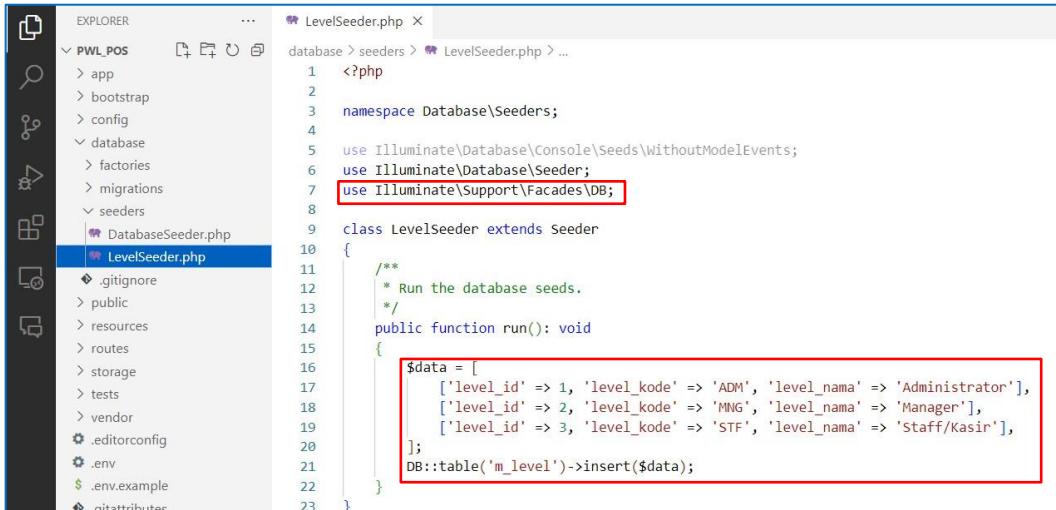
```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder LevelSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeder [C:\Laragon\www\PWL_2025\PWL_WEEK3\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.



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



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

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

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=LevelSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeding database.
```

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

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

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

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder UserSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeders\UserSeeder.php] created successfully.
```



7. Modifikasi file `class UserSeeder` seperti berikut

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

8. Jalankan perintah untuk mengeksekusi class `UserSeeder`

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=UserSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
[INFO] Seeding database.
```

9. Perhatikan hasil seeder pada table `m_user`

	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$uxsRMuP/zbBC3QuuduxLleyKYK2DP1WyYqVv7w5kNS7...	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$yaEH3ZHzPrGD95Vui06leEtVsWIVBuByAjpy17eOym...	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$6kkJK0Hc0GxuZL2je4HesyhP/p6eutx/NOAR6UgJ2...	NULL	NULL

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

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

No	Nama Tabel	Jumlah Data	Keterangan
----	------------	-------------	------------



1	m_kategori	5	5 kategori barang
2	m_barang	10	10 barang yang berbeda
3	t_stok	10	Stok untuk 10 barang
4	t_penjualan	10	10 transaksi penjualan
5	t_penjualan_detail	30	3 barang untuk setiap transaksi penjualan

➤ [m_kategori](#)

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seede
r KategoriSeeder

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeds\KategoriSeeder.php] created successfully.
```

```
database > seeders >  KategoriSeeder.php >  KategoriSeeder
 1  <?php
 2
 3  namespace Database\Seeders;
 4
 5  use Illuminate\Database\Seeder;
 6  use Illuminate\Support\Facades\DB;
 7
 8  class KategoriSeeder extends Seeder
 9  {
10      public function run(): void
11      {
12          $data = [
13              ['kategori_id' => 1, 'kategori_kode' => 'FB', 'kategori_nama' => 'Food & Beverage'],
14              ['kategori_id' => 2, 'kategori_kode' => 'BH', 'kategori_nama' => 'Beauty & Health'],
15              ['kategori_id' => 3, 'kategori_kode' => 'HC', 'kategori_nama' => 'Home Care'],
16              ['kategori_id' => 4, 'kategori_kode' => 'BK', 'kategori_nama' => 'Baby & Kids'],
17              ['kategori_id' => 5, 'kategori_kode' => 'EL', 'kategori_nama' => 'Electronics'],
18          ];
19
20          DB::table('m_kategori')->insert($data);
21      }
22  }
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed
--class=KategoriSeeder

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Seeding database.
```

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
	Edit Copy Delete	1	FB	Food & Beverage	NULL	NULL
	Edit Copy Delete	2	BH	Beauty & Health	NULL	NULL
	Edit Copy Delete	3	HC	Home Care	NULL	NULL
	Edit Copy Delete	4	BK	Baby & Kids	NULL	NULL
	Edit Copy Delete	5	EL	Electronics	NULL	NULL

➤ [m_barang](#)



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
http://www.polinema.ac.id

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder BarangSeeder

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeders\BarangSeeder.php] created successfully.
```

```
database > seeders > BarangSeeder.php > BarangSeeder

1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use Illuminate\Support\Facades\DB;
7
8 class BarangSeeder extends Seeder
9 {
10     public function run(): void
11     {
12         $data = [
13             ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'B001', 'barang_nama' => 'Air Mineral', 'harga_beli' => 2000, 'harga_jual' => 3000],
14             ['barang_id' => 2, 'kategori_id' => 1, 'barang_kode' => 'B002', 'barang_nama' => 'Roti Tawar', 'harga_beli' => 10000, 'harga_jual' => 12000],
15             ['barang_id' => 3, 'kategori_id' => 2, 'barang_kode' => 'B003', 'barang_nama' => 'Sabun Mandi', 'harga_beli' => 5000, 'harga_jual' => 7000],
16             ['barang_id' => 4, 'kategori_id' => 2, 'barang_kode' => 'B004', 'barang_nama' => 'Shampoo', 'harga_beli' => 15000, 'harga_jual' => 18000],
17             ['barang_id' => 5, 'kategori_id' => 3, 'barang_kode' => 'B005', 'barang_nama' => 'Pembersih Lantai', 'harga_beli' => 10000, 'harga_jual' => 13000],
18             ['barang_id' => 6, 'kategori_id' => 3, 'barang_kode' => 'B006', 'barang_nama' => 'Tisu', 'harga_beli' => 8000, 'harga_jual' => 10000],
19             ['barang_id' => 7, 'kategori_id' => 4, 'barang_kode' => 'B007', 'barang_nama' => 'Popok Bayi', 'harga_beli' => 50000, 'harga_jual' => 55000],
20             ['barang_id' => 8, 'kategori_id' => 4, 'barang_kode' => 'B008', 'barang_nama' => 'Susu Formula', 'harga_beli' => 60000, 'harga_jual' => 65000],
21             ['barang_id' => 9, 'kategori_id' => 5, 'barang_kode' => 'B009', 'barang_nama' => 'Lampu LED', 'harga_beli' => 25000, 'harga_jual' => 30000],
22             ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'B010', 'barang_nama' => 'Charger HP', 'harga_beli' => 35000, 'harga_jual' => 40000],
23         ];
24
25         DB::table('m_barang')->insert($data);
26     }
27 }
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=BarangSeeder

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Seeding database.
```

	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
Edit Copy Delete	1	1	B001	Air Mineral	2000	3000	NULL	NULL
Edit Copy Delete	2	1	B002	Roti Tawar	10000	12000	NULL	NULL
Edit Copy Delete	3	2	B003	Sabun Mandi	5000	7000	NULL	NULL
Edit Copy Delete	4	2	B004	Shampoo	15000	18000	NULL	NULL
Edit Copy Delete	5	3	B005	Pembersih Lantai	10000	13000	NULL	NULL
Edit Copy Delete	6	3	B006	Tisu	8000	10000	NULL	NULL
Edit Copy Delete	7	4	B007	Popok Bayi	50000	55000	NULL	NULL
Edit Copy Delete	8	4	B008	Susu Formula	60000	65000	NULL	NULL
Edit Copy Delete	9	5	B009	Lampu LED	25000	30000	NULL	NULL
Edit Copy Delete	10	5	B010	Charger HP	35000	40000	NULL	NULL

➤ t_stok

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder StokSeeder

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeders\StokSeeder.php] created successfully.
```



```
database > seeders > 🐾 StokSeeder.php > 📁 StokSeeder
  1  <?php
  2
  3  namespace Database\Seeders;
  4
  5  use Illuminate\Database\Seeder;
  6  use Illuminate\Support\Facades\DB;
  7  use Carbon\Carbon;
  8
  9  class StokSeeder extends Seeder
 10 [
 11     public function run(): void
 12     {
 13         $data = [];
 14         for ($i = 1; $i <= 10; $i++) {
 15             $data[] = [
 16                 'barang_id' => $i,
 17                 'user_id' => 1, // Admin yang melakukan input stok
 18                 'stok_tanggal' => Carbon::now(),
 19                 'stok_jumlah' => rand(10, 100),
 20             ];
 21         }
 22         DB::table('t_stok')->insert($data);
 23     }
 24 }
 25 ]
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=StokSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO  Seeding database.
```

← →	▼	stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	Edit			1	1	2025-03-03 17:53:00	35	NULL
<input type="checkbox"/>	Edit			2	2	1 2025-03-03 17:53:00	17	NULL
<input type="checkbox"/>	Edit			3	3	1 2025-03-03 17:53:00	90	NULL
<input type="checkbox"/>	Edit			4	4	1 2025-03-03 17:53:00	86	NULL
<input type="checkbox"/>	Edit			5	5	1 2025-03-03 17:53:00	72	NULL
<input type="checkbox"/>	Edit			6	6	1 2025-03-03 17:53:00	92	NULL
<input type="checkbox"/>	Edit			7	7	1 2025-03-03 17:53:00	83	NULL
<input type="checkbox"/>	Edit			8	8	1 2025-03-03 17:53:00	32	NULL
<input type="checkbox"/>	Edit			9	9	1 2025-03-03 17:53:00	51	NULL
<input type="checkbox"/>	Edit			10	10	1 2025-03-03 17:53:00	56	NULL

➤ t_penjualan

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder PenjualanSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO  Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeders\PenjualanSeeder.php] created successfully.
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
http://www.polinema.ac.id

```
database > seeders > PenjualanSeeder.php > PenjualanSeeder
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use Illuminate\Support\Facades\DB;
7 use Carbon\Carbon;
8
9 class PenjualanSeeder extends Seeder
10
11     public function run(): void
12     {
13         $data = [];
14         for ($i = 1; $i <= 10; $i++) {
15             $data[] = [
16                 'user_id' => rand(1, 3), // Kasir yang berbeda
17                 'pembeli' => 'Customer ' . $i,
18                 'penjualan_kode' => 'TRX00' . $i,
19                 'penjualan_tanggal' => Carbon::now()->subDays(rand(1, 30)),
20             ];
21         }
22
23         DB::table('t_penjualan')->insert($data);
24     }
25 }
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=PenjualanSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeding database.
```

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
 	1	1	Customer 1	TRX001	2025-02-17 17:56:15	NULL	NULL
 	2	3	Customer 2	TRX002	2025-02-27 17:56:15	NULL	NULL
 	3	1	Customer 3	TRX003	2025-02-20 17:56:15	NULL	NULL
 	4	1	Customer 4	TRX004	2025-02-23 17:56:15	NULL	NULL
 	5	3	Customer 5	TRX005	2025-02-21 17:56:15	NULL	NULL
 	6	1	Customer 6	TRX006	2025-02-23 17:56:15	NULL	NULL
 	7	1	Customer 7	TRX007	2025-02-22 17:56:15	NULL	NULL
 	8	2	Customer 8	TRX008	2025-02-20 17:56:15	NULL	NULL
 	9	1	Customer 9	TRX009	2025-02-04 17:56:15	NULL	NULL
 	10	1	Customer 10	TRX0010	2025-02-11 17:56:15	NULL	NULL

➤ t_penjualan_detail

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:seeder PenjualanDetailSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeder [C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

```
database > seeders > PenjualanDetailSeeder.php > PenjualanDetailSeeder
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         $data = [];
13         for ($i = 1; $i <= 10; $i++) { // 10 transaksi
14             for ($j = 1; $j <= 3; $j++) { // 3 barang per transaksi
15                 $barang_id = rand(1, 10);
16                 $barang = DB::table('m_barang')->where('barang_id', $barang_id)->first();
17
18                 $data[] = [
19                     'penjualan_id' => $i,
20                     'barang_id' => $barang_id,
21                     'harga' => $barang->harga_jual,
22                     'jumlah' => rand(1, 5),
23                 ];
24             }
25         }
26
27         DB::table('t_penjualan_detail')->insert($data);
28     }
29 }
```

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan db:seed --class=PenjualanDetailSeeder
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO Seeding database.
```

Table: t_penjualan_detail						
	detail_id	penjualan_id	barang_id	harga	jumlah	created_at
	1	1	3	12000	5	NULL
	2	1	3	7000	1	NULL
	3	1	4	18000	4	NULL
	4	2	4	18000	2	NULL
	5	2	2	12000	4	NULL
	6	2	9	30000	3	NULL
	7	3	2	12000	4	NULL
	8	3	10	40000	1	NULL
	9	3	9	30000	4	NULL
	10	4	4	18000	4	NULL
	11	4	10	40000	1	NULL
	12	4	1	3000	1	NULL
	13	5	9	30000	4	NULL
	14	5	1	3000	4	NULL
	15	5	8	65000	1	NULL
	16	6	10	40000	2	NULL
	17	6	1	3000	4	NULL
	18	6	3	7000	2	NULL
	19	7	5	13000	5	NULL
	20	7	6	10000	3	NULL
	21	7	9	30000	5	NULL
	22	8	2	12000	2	NULL

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



D. DB FACADE

DB Façade merupakan fitur dari Laravel yang digunakan untuk melakukan *query* secara langsung dengan mengetikkan perintah SQL secara utuh (*raw query*). Disebut *raw query* (query mentah) karena penulisan query pada DB Façade langsung ditulis sebagaimana yang biasa dituliskan pada database, seperti “`select * from m_user`” atau “`insert into m_user...`” atau “`update m_user set ... Where ...`”

Raw query adalah cara paling dasar dan tradisional yang ada di Laravel. Raw query terasa familiar karena biasa kita pakai ketika melakukan query langsung ke database.

INFO

Dokumentasi penggunaan DB Façade bisa dicek di laman ini

<https://laravel.com/docs/10.x/database#running-queries>

Terdapat banyak method yang bisa digunakan pada DB Façade ini. Akan tetapi yang kita pelajari cukup 4 (empat) method yang umum dipakai, yaitu

a. `DB::select()`

Method ini digunakan untuk mengambil data dari database. Method ini **mengembalikan (return)** data hasil *query*. Contoh

```
DB::select('select * from m_user'); //Query semua data pada tabel m_user
```

```
DB::select('select * from m_user where level_id = ?', [1]); //Query tabel m_user dengan level_id = 1
```

```
DB::select('select * from m_user where level_id = ? and username = ?', [1, 'admin']);
```

b. `DB::insert()`

Method ini digunakan untuk memasukkan data pada table database. Method ini **tidak memiliki nilai pengembalian (no return)**. Contoh

```
DB::insert('insert into m_level(level_kode, level_nama) values(?,?)', ['cus', 'Pelanggan']);
```

c. `DB::update()`

Method ini digunakan saat menjalankan *raw query* untuk meng-update data pada database. Method ini **memiliki nilai pengembalian (return)** berupa jumlah baris data yang ter-update. Contoh

```
DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'cus']);
```



d. **DB::delete()**

Method ini digunakan saat menjalankan *raw query* untuk menghapus data dari table.

Method ini **memiliki nilai pengembalian (return)** berupa jumlah baris data yang telah dihapus. Contoh

```
DB::delete('delete from m_level where level_kode = ?', ['CUS']);
```

Ok, sekarang mari kita coba praktikkan menggunakan DB Façade pada project kita

Praktikum 4 – Implementasi DB Facade

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

```
Microsoft Windows [Version 10.0.22631.4979]
(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:controller LevelController

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO: Controller [C:\laragon\www\PWL_2025\PWL_WEEK3\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  | -----
8  | Web Routes
9  | -----
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "web" middleware group. Make something great!
14 |
15 */
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
21 Route::get('/level', [LevelController::class, 'index']);
22
```

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

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



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



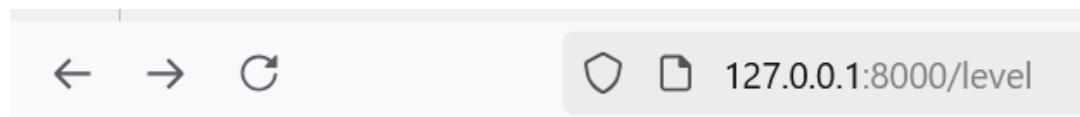
Insert data baru berhasil!

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	CUS	Pelanggan	2025-03-04 02:32:03	NULL

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

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

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



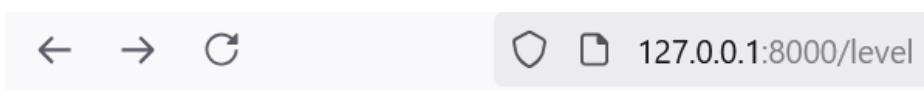
Update data berhasil. Jumlah data yang diupdate: 1 baris



	<input type="button" value="←"/> <input type="button" value="→"/>		level_id	level_kode	level_nama	created_at	updated_at	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	CUS	Customer	2025-03-04 02:32:03	NULL

7. Kita coba modifikasi lagi file [LevelController](#) untuk melakukan proses hapus data

```
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',
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 }
22 }
```



Delete data berhasil. Jumlah data yang dihapus: 1 baris

Tampilan pada database

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



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

```
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         $data = DB::select('select * from m_level');
22         return view('level', ['data' => $data]);
23     }
24 }
```

```
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>
```

10. Silahkan dicoba pada browser dan amati apa yang terjadi



← → ⌂

🛡️ 📁 127.0.0.1:8000/level

Data Level Pengguna

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

Penjelasan :

Pada laman 27.0.0.1:8000/level berhasil menampilkan data dari tabel m_level dengan format tabel HTML dengan :

- Routing dan controller sudah berjalan dengan benar
- View (level.blade.php) berhasil dirender dengan data dari database
- Database sudah memiliki data dan berhasil ditampilkan

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



E. QUERY BUILDER

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

Query builder membuat kode kita menjadi rapi dan lebih mudah dibaca. Selain itu *query builder* tidak terikat ke satu jenis database, jadi query builder bisa digunakan untuk mengakses berbagai jenis database seperti MySQL, MariaDB, PostgreSQL, SQL Server, dll. Jika suatu saat ingin beralih dari database MySQL ke PostgreSQL, tidak akan banyak kendala. Namun kelemahan dari *query builder* adalah kita harus mengetahui method-method apa saja yang ada di *query builder*.

INFO

Dokumentasi penggunaan Query Builder pada Laravel bisa dicek di laman ini

<https://laravel.com/docs/10.x/queries>

Ciri khas *query builder* Laravel adalah kita tentukan dahulu target table yang akan kita akses untuk operasi CRUD.

```
DB::table('<nama-tabel>'); // query builder untuk melakukan operasi CRUD pada tabel yang dituju
```

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

- Perintah untuk *insert* data dengan method `insert()`

```
DB::table('m_kategori')->insert(['kategori_kode' => 'SMP', 'kategori_nama' => 'Smartphone']);
```

Query yang dihasilkan dari kode di atas adalah

```
insert into m_kategori(kategori_kode, kategori_nama) values('SMP', 'Smartphone');
```

- Perintah untuk *update* data dengan method `where()` dan `update()`

```
DB::table('m_kategori')->where('kategori_id', 1)->update(['kategori_nama' => 'Makanan Ringan']);
```

Query yang dihasilkan dari kode di atas adalah

```
update m_kategori set kategori_nama = 'Makanan Ringan' where kategori_id = 1;
```



- c. Perintah untuk *delete* data dengan method `where()` dan `delete()`

```
DB::table('m_kategori')->where('kategori_id', 9) ->delete();
```

Query yang dihasilkan dari kode di atas adalah

```
delete from m_kategori where kategori_id = 9;
```

- d. Perintah untuk ambil data

Method Query Builder	Query yang dihasilkan
<code>DB::table('m_kategori')->get();</code>	<code>select * from m_kategori</code>
<code>DB::table('m_kategori')</code> <code>->where('kategori_id', 1)->get();</code>	<code>select * from m_kategori where kategori_id = 1;</code>
<code>DB::table('m_kategori')</code> <code>->select('kategori_kode')</code> <code>->where('kategori_id', 1)->get();</code>	<code>select kategori_kode from m_kategori where kategori_id = 1;</code>

Praktikum 5 – Implementasi *Query Builder*

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

```
C:\laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:controller KategoriController
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0
INFO | Controller [C:\laragon\www\PWL_2025\PWL_WEEK3\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 |-----
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 Route::get('/level', [LevelController::class, 'index']);
23 Route::get('/kategori', [KategoriController::class, 'index']);
```

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



table [m_kategori](#)

```
app > Http > Controllers >  KategoriController.php >  KategoriController > f 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         $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17
18         DB::table('m_kategori')->insert($data);
19
20         return 'Insert data baru berhasil!';
21     }
22 }
```



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



Insert data baru berhasil!

	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	kategori_id	kategori.kode	kategori.nama	created_at	updated_at
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	FB	Food & Beverage	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	BH	Beauty & Health	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3	HC	Home Care	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4	BK	Baby & Kids	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	5	EL	Electronics	NULL	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	6	SNK	Snack/Makanan Ringan	2025-03-04 03:44:11	NULL

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

```
app > Http > Controllers > KategoriController.php > index
  1  <?php
  2
  3  namespace App\Http\Controllers;
  4
  5  use Illuminate\Http\Request;
  6  use Illuminate\Support\Facades\DB;
  7
  8  class KategoriController extends Controller
  9  {
10      public function index()
11      {
12          /* $data = [
13          'kategori.kode' => 'SNK',
14          'kategori.nama' => 'Snack/Makanan Ringan',
15          'created_at' => now()
16          ];
17
18          DB::table('m_kategori')->insert($data);
19          return 'Insert data baru berhasil!';*/
20
21          $row = DB::table('m_kategori')->where('kategori.kode', 'SNK')->update(['kategori.nama' => 'Camilan']);
22          return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
23
24      }
25  }
```

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`



Update data berhasil. Jumlah data yang diupdate: 1 baris



	Edit	Copy	Delete	kategori_id	kategori.kode	kategori.nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	FB	Food & Beverage	<i>NULL</i>	<i>NULL</i>
<input type="checkbox"/>	Edit	Copy	Delete	2	BH	Beauty & Health	<i>NULL</i>	<i>NULL</i>
<input type="checkbox"/>	Edit	Copy	Delete	3	HC	Home Care	<i>NULL</i>	<i>NULL</i>
<input type="checkbox"/>	Edit	Copy	Delete	4	BK	Baby & Kids	<i>NULL</i>	<i>NULL</i>
<input type="checkbox"/>	Edit	Copy	Delete	5	EL	Electronics	<i>NULL</i>	<i>NULL</i>
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Camilan	2025-03-04 03:44:11	<i>NULL</i>

7. Kita coba modifikasi lagi file [KategoriController](#) untuk melakukan proses hapus data



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

8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table `m_kategori`. Kita modifikasi file `KategoriController` 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         /* $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17
18         DB::table('m_kategori')->insert($data);
19         return 'Insert data baru berhasil!';
20
21         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'camilan']);
22         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
23
24         // $row = DB::table("m_kategori")->where('kategori_kode', 'SNK')->delete();
25         // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
26
27         $data = DB::table("m_kategori")->get();
28         return view('kategori', ['data' => $data]);
29     }
30 }
```

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`

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



```
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 |
```

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	FB	Food & Beverage
2	BH	Beauty & Health
3	HC	Home Care
4	BK	Baby & Kids
5	EL	Electronics
6	SNK	Camilan

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



F. ELOQUENT ORM

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

INFO

Eloquent ORM memerlukan Model untuk proses konversi data pada tabel menjadi object. Object inilah yang nantinya akan kita akses dari dalam controller. Oleh karena itu **membuat Model pada Laravel berarti menggunakan Eloquent ORM**. Silahkan cek disini

<https://laravel.com/docs/10.x/eloquent>

Perintah untuk membuat model adalah sebagai berikut

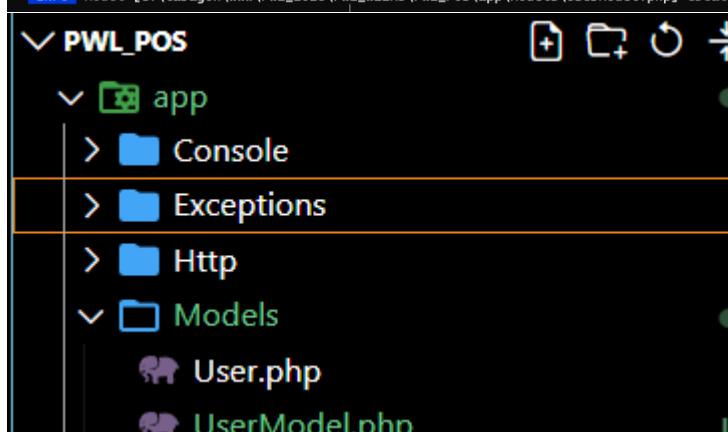
```
php artisan make:model <nama-model-CamelCase>
```

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

Praktikum 6 – Implementasi Eloquent ORM

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

```
C:\Laragon\www\PWL_2025\PWL_WEEK3\PWL_POS>php artisan make:model  
UserModel  
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
INFO Model [C:\laragon\www\PWL_2025\PWL_WEEK3\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 > ...
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 }
```

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

```
routes > web.php > fClosure
1  <?php
2
3  use App\Http\Controllers\UserController;
4  use App\Http\Controllers\KategoriController;
5  use App\Http\Controllers\LevelController;
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 Route::get('/level', [LevelController::class, 'index']);
24 Route::get('/kategori', [KategoriController::class, 'index']);
25 Route::get('/user', [UserController::class, 'index']);
26
```

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
14        return view('user', ['data' => $user]);
15    }
16 }
```

6. Kemudian kita buat view `user.blade.php`



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

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

Data User

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

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

```
app > Http > Controllers > UserController.php > UserController > index
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         $data = [
15             'username' => 'customer-1',
16             'name' => 'Pelanggan',
17             'password' => Hash::make('12345'),
18             'level_id' => 4
19         ];
20
21         UserModel::insert($data); // tambahkan data ke tabel m_user
22
23         // coba akses model UserModel
24         $user = UserModel::all(); // ambil semua data dari tabel m_user
25         return view('user', ['data' => $user]);
26     }
27 }
```

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



```
SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails ('pwl_pos`.`m_user', CONSTRAINT 'm_user_lev...
16     'nama' => 'Pelanggan',
17     'password' => Hash::make('12345'),
18     'level_id' => 4
19 ];
20
21 UserModel::insert($data); // tambahkan data ke tabel m_user
22
23 // coba akses model UserModel
24 $user = UserModel::all(); // ambil semua data dari tabel m_user
25 return view('user', ['data' => $user]);
26 }
27 }
28 }
```

Masalah terjadi karena adanya foreign key constraint saat melakukan insert atau update data. Hal ini disebabkan oleh nilai level_id = 4 yang akan dimasukkan tidak ditemukan dalam database, sehingga terjadi pelanggaran aturan referensial pada tabel.

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

```
app > Http > Controllers > UserController.php > UserController > index
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         $data = [
15             'nama' => 'Pelanggan Pertama',
16         ];
17
18         UserModel::where('username', 'customer-1')->update($data); // update data user
19
20         // coba akses model UserModel
21         $user = UserModel::all(); // ambil semua data dari tabel m_user
22         return view('user', ['data' => $user]);
23     }
24 }
```



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

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

- Penjelasan :
Kode ini mencari data di tabel m_user yang memiliki username = 'customer-1'. Jika ditemukan, kolom nama akan diperbarui menjadi "Pelanggan Pertama". Jika tidak ada customer-1 dalam database, tidak ada data yang diubah, tetapi kode tetap berjalan tanpa error.

12. Jika sudah, laporan 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?

Jawaban :

APP_KEY pada Laravel digunakan sebagai kunci untuk mengenkripsi informasi penting dan sensitif seperti session, token dan password. Dengan APP_KEY, laravel bisa memastikan bahwa data yang disimpan atau dikirim tetap aman dan tidak bisa dibaca oleh orang yang tidak mendapatkan akses.

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

Jawaban :

Untuk menhasilkan nilai APP_KEY secara otomatis dan menyimpan di file .env dapat menggunakan sintaks :

```
php artisan key:generate
```

```
C:\laragon\www\PWL_2625\PWL_WEEK3\PWL_POS>php artisan key:generate
te

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

INFO Application key set successfully.
```

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

Jawaban :



Secara Default, Laravel memiliki tiga file migrasi yaitu :

`create_users_table.php` : digunakan untuk membuat tabel pengguna

`create_tokens_table.php` : digunakan untuk menyimpan token reset password

`create_failed_jobs_table.php` : digunakan untuk menyimpan log pekerja (jobs) yang gagal

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

Jawaban :

Fungsi `$table->timestamps();`, dalam migration Laravel digunakan untuk menambahkan dua kolom otomatis di dalam database, yaitu :

1. `created_at` → akan menyimpan waktu saat data pertama kali dibuat (ditambahkan ke database).
2. `updated_at` → akan menyimpan waktu saat data terakhir diperbarui (diubah).

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

Jawaban :

Tipe data yang dihasilkan adalah big integer (BIGINT) yang bersifat auto-increment. Laravel mendesain `$table->id();` sebagai shortcut dari `$table->bigIncrements('id');`, yang secara default menghasilkan kolom id bertipe BIGINT (unsigned) dan auto-increment.

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

Jawaban :

Perbedannya yaitu terletak pada kolom Primary Key yang dihasilkan :

- Jika menggunakan `$table->id();` secara default akan membuat kolom primary key bernama id dengan tipe BIGINT AUTO_INCREMENT
- Jika menggunakan `$table->id('level_id');` Laravel tetap akan membuat kolom primary key dengan tipe BIGINT (unsigned) AUTO_INCREMENT, tetapi dengan nama level_id, bukan id.

Dapat disimpulkan bawahwa fungsinya akan tetap sama yaitu sebagai Primary Key namun dengan nama yang berbeda.

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

Jawaban :

Fungsi `-> unique()` dalam migration Laravel digunakan untuk menjadikan kolom memiliki nilai unik, artinya tidak boleh ada duplikat data dalam kolom tersebut.

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')` ?

Jawaban :

- `m_level.level_id` menggunakan `$table->id('level_id')`; karena itu primary key dengan tipe BIGINT unsigned AUTO_INCREMENT.
- `m_user.level_id` menggunakan `$table->unsignedBigInteger('level_id')`; agar kompatibel dengan primary key `m_level.level_id` dan dapat dijadikan foreign key.

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

Jawaban :

Class Hash dalam Laravel digunakan untuk mengubah (enkripsi) password atau data sensitif. Tujuan utamanya adalah untuk melindungi password pengguna dengan cara menyimpannya dalam bentuk terenkripsi, sehingga meskipun database diretas, password asli tetap tidak bisa diketahui.

Maksud dari Kode `Hash::make('1234');` digunakan untuk mengubah teks password (1234) menjadi hash yang aman sebelum disimpan ke database.

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

Jawaban :

Tanda tanya (?) digunakan sebagai tempat sementara untuk menggantikan nilai yang akan dimasukkan ke dalam query. Laravel akan menggantinya dengan nilai yang sesuai saat query dijalankan.

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

Jawaban :

- `$table = 'm_user';` → agar Laravel tahu bahwa model ini menggunakan tabel `m_user` bukan tabel default users.
- `$primaryKey = 'user_id';` → Supaya Laravel tahu bahwa primary key di tabel ini adalah `user_id`, bukan `id`.
- `protected` digunakan supaya properti ini hanya bisa dipakai di dalam model ini dan model turunannya, tidak bisa diakses sembarangan.

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

Jawaban :

Menurut saya lebih mudah menggunakan *Eloquent ORM* karena kodennya lebih singkat,



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

rapi dan jelas serta dapat dengan mudah dibaca dan dipahami .

*** *Sekian, dan selamat belajar ****