



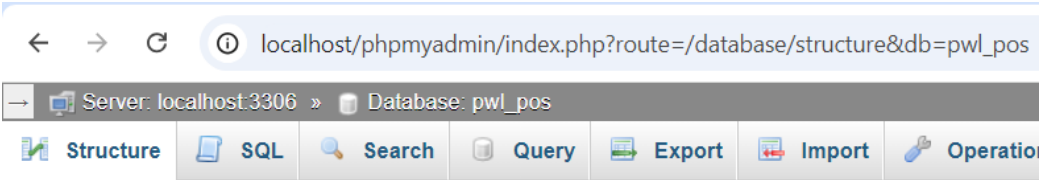
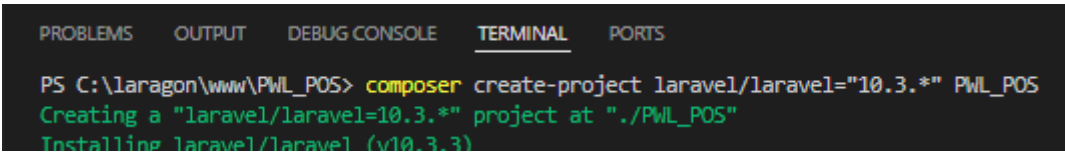
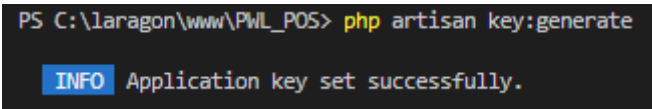
Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Sistem Informasi Bisnis  
Semester : 5

Kelas : SIB  
NIM : 2241760026  
Nama : Siti Faiqoh  
Jobsheet Ke- : 3

Link Github : [https://github.com/faiqoh04/PWL\\_POS](https://github.com/faiqoh04/PWL_POS)

## Laporan Jobsheet

### Praktikum Ke-1 pengaturan database

Langkah	Jawaban/Deskripsi
1	<p>Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL_POS</p>  <p><a href="http://localhost/phpmyadmin/index.php?route=/database/structure&amp;db=pwl_pos">http://localhost/phpmyadmin/index.php?route=/database/structure&amp;db=pwl_pos</a></p>
2	<p>Buka aplikasi VSCode dan buka folder project PWL_POS yang sudah kita buat</p>  <pre>PS C:\laragon\www\PWL_POS&gt; composer create-project laravel/laravel="10.3.*" PWL_POS Creating a "laravel/laravel=10.3.*" project at "./PWL_POS" Installing laravel/laravel (v10.3.3)</pre>
3	<p>Copy file .env.example menjadi .env</p>
4	<p>Buka file .env, dan pastikan konfigurasi APP_KEY bernilai. Jika belum bernilai silahkan kalian <i>generate</i> menggunakan php artisan.</p> <p>Edit file .env dan sesuaikan dengan database yang telah dibuat</p>  <pre>PS C:\laragon\www\PWL_POS&gt; php artisan key:generate INFO Application key set successfully.</pre>

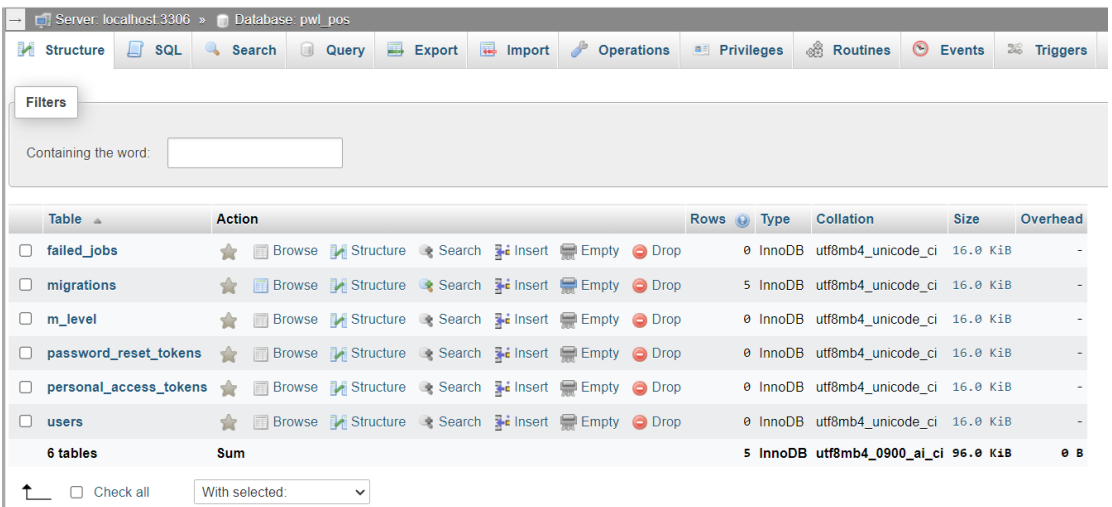


	<pre>.env 1 APP_NAME=Laravel 2 APP_ENV=local 3 APP_KEY=base64:RPMYXQWQ5n96petLg3q19u1SbAoc19S8wZWj1CQpUv8= 4 APP_DEBUG=true 5 APP_URL=http://localhost 6 7 LOG_CHANNEL=stack 8 LOG_DEPRECATIONS_CHANNEL=null 9 LOG_LEVEL=debug 10 11 DB_CONNECTION=mysql 12 DB_HOST=127.0.0.1 13 DB_PORT=3306 14 DB_DATABASE=laravel 15 DB_USERNAME=root 16 DB_PASSWORD= 17 18 BROADCAST_DRIVER=log</pre>
5	Laporkan hasil Praktikum-1 ini dan <i>commit</i> perubahan pada <i>git</i> .

### Praktikum Ke-2.1 - Pembuatan file migrasi tanpa relasi

Langkah	Jawaban/Deskripsi
1	Buka <i>terminal</i> VSCode kalian, untuk yang di kotak merah adalah default dari laravel
2	Kita abaikan dulu yang di kotak merah (jangan di hapus)
3	Kita buat file migrasi untuk table <i>m_level</i> dengan perintah <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:migration create_m_level_table --create=m_level</pre> 
4	Kita perhatikan bagian yang di kotak merah, bagian tersebut yang akan kita modifikasi sesuai desain database yang sudah ada <pre>database &gt; migrations &gt; 2024_09_14_013343_create_m_level_table.php &gt; ... 1 &lt;?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-&gt;id('level_id'); 16             \$table-&gt;string('level_kode',10)-&gt;unique(); 17             \$table-&gt;string('level_nama',100); 18             \$table-&gt;timestamps(); 19         }); 20     } 21 22     /** 23      * Reverse the migrations. 24      */ 25     public function down(): void 26     { 27         Schema::dropIfExists('m_level'); 28     } 29 }</pre>
5	Simpan kode pada tahapan 4 tersebut, kemudian jalankan perintah ini pada terminal VSCode untuk melakukan migrasi



	<pre>PS C:\laragon\www\PWL_POS&gt; php artisan migrate  [WARN] The database 'laravel' does not exist on the 'mysql' connection.  Would you like to create it? (yes/no) [no] &gt; yes  [INFO] Preparing database.  Creating migration table ..... 158ms DONE  [INFO] Running migrations.  2014_10_12_000000_create_users_table ..... 121ms DONE 2014_10_12_100000_create_password_reset_tokens_table ..... 34ms DONE 2019_08_19_000000_create_failed_jobs_table ..... 74ms DONE 2019_12_14_000001_create_personal_access_tokens_table ..... 120ms DONE 2024_09_14_013343_create_m_level_table ..... 112ms DONE  PS C:\laragon\www\PWL_POS&gt;</pre>
6	<p>Kemudian kita cek di phpMyAdmin apakah table sudah ter-generate atau belum</p> <pre>.env 11 DB_CONNECTION=mysql 12 DB_HOST=127.0.0.1 13 DB_PORT=3306 14 DB_DATABASE=pwl_pos 15 DB_USERNAME=root 16 DB_PASSWORD= 17 18 BROADCAST_DRIVER=log 19 CACHE_DRIVER=file 20 FILESYSTEM_DISK=local</pre> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan migrate  [INFO] Preparing database.  Creating migration table ..... 37ms DONE  [INFO] Running migrations.  2014_10_12_000000_create_users_table ..... 77ms DONE 2014_10_12_100000_create_password_reset_tokens_table ..... 44ms DONE 2019_08_19_000000_create_failed_jobs_table ..... 70ms DONE 2019_12_14_000001_create_personal_access_tokens_table ..... 130ms DONE 2024_09_14_013343_create_m_level_table ..... 165ms DONE</pre> 
7	Ok, table sudah dibuat di database
8	Buat table <i>database</i> dengan <i>migration</i> untuk table m_kategori dan m_supplier yang sama-sama tidak memiliki <i>foreign key</i>



```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_kategori_table

[INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_14_015218_create_m_kategori_table.php] created successfully.

PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_supplier_table

[INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_14_015236_create_m_supplier_table.php] created successfully.

PS C:\laragon\www\PWL_POS> php artisan migrate

[INFO] Running migrations.

2024_09_14_015218_create_m_kategori_table ..... 50ms DONE
2024_09_14_015236_create_m_supplier_table ..... 32ms DONE
```

```
database > migrations > 2024_09_14_015218_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       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('m_kategori', function (Blueprint $table) {
15             $table->id('kategori_id'); // Menggunakan id() untuk primary key
16             $table->string('kategori_kode', 10)->unique(); // Menggunakan string() untuk kode kategori
17             $table->string('kategori_nama', 100); // Menggunakan string() untuk nama kategori
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      */
25     public function down(): void
26     {
27         Schema::dropIfExists('m_kategori');
28     }
29 };
```

```
database > migrations > 2024_09_14_015236_create_m_supplier_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_supplier', function (Blueprint $table) {
15             $table->id('supplier_id');
16             $table->string('supplier_kode',10)->unique();
17             $table->string('supplier_nama',100);
18             $table->string('supplier_alamat',255);
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists('m_supplier');
29     }
30 };
```



9	Laporkan hasil Praktikum-2.1 ini dan <i>commit</i> perubahan pada <i>git</i>

## Praktikum Ke-2.2 – Pembuatan dile migrasi degan relasi

Langkah	Jawaban/Deskripsi
1	Buka <i>terminal</i> VSCode kalian, dan buat file migrasi untuk table <i>m_user</i> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:migration create_m_user_table --table=m-user</pre> <p>INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_09_14_020459_create_m_user_table.php] created successfully.</p>
2	Buka file migrasi untuk table <i>m_user</i> , dan modifikasi seperti berikut <pre>database &gt; migrations &gt; 2024_09_14_020459_create_m_user_table.php &gt; ... 1  &lt;?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_user', function (Blueprint \$table) { 15               \$table-&gt;id('user_id'); // Primary key 16               \$table-&gt;unsignedBigInteger('level_id'); // Foreign key ke level_id 17               \$table-&gt;string('username', 20)-&gt;unique(); // Unique username 18               \$table-&gt;string('nama', 100); // Nama pengguna 19               \$table-&gt;string('password', 255); // Password pengguna 20               \$table-&gt;timestamps(); // Created_at dan Updated_at 21 22               // Menambahkan foreign key constraint 23               \$table-&gt;foreign('level_id')-&gt;references('level_id')-&gt;on('m_level')-&gt;onDelete('cascade'); 24           }); 25       } 26 27       /** 28        * Reverse the migrations. 29        */ 30       public function down(): void 31       { 32           Schema::table('m_user', function (Blueprint \$table) { 33               // Menghapus foreign key constraint sebelum menghapus tabel 34               \$table-&gt;dropForeign(['level_id']); 35           }); 36 37           Schema::dropIfExists('m_user'); 38       } 39  };</pre>
3	Simpan kode program Langkah 2, dan jalankan perintah <i>php artisan migrate</i> . Amati apa yang terjadi pada database.



	<div><pre>PS C:\laragon\www\pwl_pos&gt; php artisan migrate</pre><div><div>INFO</div>Running migrations. 2024_09_14_020459_create_m_user_table ..... 266ms <b>DONE</b></div></div> <div><div>Server: localhost:3306 » Database: pwl_pos » Table: m_user</div><div><div>Browse</div><div>Structure</div><div>SQL</div><div>Search</div><div>Insert</div><div>Export</div><div>Import</div></div><div>✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)</div><div>SELECT * FROM `m_user`</div><div><input type="checkbox"/> Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]</div><div><div>user_id</div><div>level_id</div><div>username</div><div>nama</div><div>password</div><div>created_at</div><div>updated_at</div></div><div><div>Query results operations</div><div><div>Create view</div></div></div></div> <p>Pada table user terdapat kolom baru dengan nama level_id yang menyambungkan table user dengan table level</p>
4	<p>Buat table <i>database</i> dengan <i>migration</i> untuk table-tabel yang memiliki <i>foreign key</i></p> <div><pre>PS C:\laragon\www\pwl_pos&gt; php artisan make:migration create_m_barang_table --table=m-barang</pre><div><div>INFO</div>Migration [C:\laragon\www\pwl_pos\database\migrations\2024_09_14_030723_create_m_barang_table.php] created successfully.</div></div> <div><pre>PS C:\laragon\www\pwl_pos&gt; php artisan make:migration create_t_penjualan_table --table=t-penjualan</pre><div><div>INFO</div>Migration [C:\laragon\www\pwl_pos\database\migrations\2024_09_14_030857_create_t_penjualan_table.php] created successfully.</div></div> <div><pre>PS C:\laragon\www\pwl_pos&gt; php artisan make:migration create_t_stok_table --table=t-stok</pre><div><div>INFO</div>Migration [C:\laragon\www\pwl_pos\database\migrations\2024_09_14_030918_create_t_stok_table.php] created successfully.</div></div> <div><pre>PS C:\laragon\www\pwl_pos&gt; php artisan make:migration create_t_penjualan_detail_table --table=t-penjualan-detail</pre><div><div>INFO</div>Migration [C:\laragon\www\pwl_pos\database\migrations\2024_09_14_030956_create_t_penjualan_detail_table.php] created successfully.</div></div> <p>- m_barang</p>



```
database > migrations > 2024_09_14_030723_create_m_barang_table.php > class > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('m_barang', function (Blueprint $table) {
15             $table->id('barang_id');
16             $table->unsignedBigInteger('kategori_id'); // Foreign key ke 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             // Menambahkan foreign key constraint
24             $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori')->onDelete('cascade');
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      */
31     public function down(): void
32     {
33         Schema::table('m_barang', function (Blueprint $table) {
34             // Menghapus foreign key constraint sebelum menghapus tabel
35             $table->dropForeign(['kategori_id']);
36         });
37
38         Schema::dropIfExists('m_barang');
39     }
40 };
```

- t\_penjualan

```
database > migrations > 2024_09_14_030857_create_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'); // Foreign key ke m_user
17             $table->string('pembeli', 50);
18             $table->string('penjualan_kode', 20);
19             $table->dateTime('penjualan_tanggal'); // Tanggal penjualan
20             $table->timestamps();
21
22             // Menambahkan foreign key constraint
23             $table->foreign('user_id')->references('user_id')->on('m_user')->onDelete('cascade');
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::table('t_penjualan', function (Blueprint $table) {
33             // Menghapus foreign key constraint sebelum menghapus tabel
34             $table->dropForeign(['user_id']);
35         });
36
37         Schema::dropIfExists('t_penjualan');
38     }
39 };
```

- t\_stok



```
database > migrations > 2024_09_14_030918_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'); // Primary key
16              $table->unsignedBigInteger('supplier_id')->index(); // Foreign key ke tabel supplier
17              $table->unsignedBigInteger('barang_id')->index(); // Foreign key ke tabel barang
18              $table->unsignedBigInteger('user_id')->index(); // Foreign key ke tabel user
19              $table->dateTime('stok_tanggal'); // Tanggal stok dalam format datetime
20              $table->integer('stok_jumlah'); // Jumlah stok dengan tipe integer
21              $table->timestamps(); // Created_at dan Updated_at
22
23              // Menambahkan foreign key constraints
24              $table->foreign('supplier_id')->references('supplier_id')->on('m_supplier')->onDelete('cascade');
25              $table->foreign('barang_id')->references('barang_id')->on('m_barang')->onDelete('cascade');
26              $table->foreign('user_id')->references('user_id')->on('m_user')->onDelete('cascade');
27          });
28      }
29
30      /**
31       * Reverse the migrations.
32       */
33      public function down(): void
34      {
35          Schema::dropIfExists('t_stok'); // Menghapus tabel jika rollback
36      }
37  };
```

- **t\_penjualan\_detail**

```
database > migrations > 2024_09_14_030956_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       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create('t_penjualan_detail', function (Blueprint $table) {
15              $table->id('detail_id'); // Primary key
16              $table->unsignedBigInteger('penjualan_id')->index(); // Foreign key ke tabel penjualan
17              $table->unsignedBigInteger('barang_id')->index(); // Foreign key ke tabel barang
18              $table->integer('harga'); // Harga dengan tipe integer
19              $table->integer('jumlah'); // Jumlah barang yang dibeli
20              $table->timestamps(); // Created_at dan Updated_at
21          });
22
23          // Menambahkan foreign key constraints (Optional)
24          Schema::table('t_penjualan_detail', function (Blueprint $table) {
25              $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan')->onDelete('cascade');
26              $table->foreign('barang_id')->references('barang_id')->on('m_barang')->onDelete('cascade');
27          });
28      }
29
30      /**
31       * Reverse the migrations.
32       */
33      public function down(): void
34      {
35          Schema::dropIfExists('t_penjualan_detail');
36      }
37  };
```

- **php artisan migrate setelah error**

```
PS C:\laragon\www\PWL_POS> php artisan migrate

[INFO] Running migrations.

2024_09_14_015236_create_m_supplier_table ..... 64ms DONE
2024_09_14_020459_create_m_user_table ..... 175ms DONE
2024_09_14_030723_create_m_barang_table ..... 165ms DONE
2024_09_14_030857_create_t_penjualan_table ..... 122ms DONE
2024_09_14_030918_create_t_stok_table ..... 423ms DONE
2024_09_14_030956_create_t_penjualan_detail_table ..... 345ms DONE
```

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





6	Laporkan hasil Praktikum-2.2 ini dan <i>commit</i> perubahan pada <i>git</i>

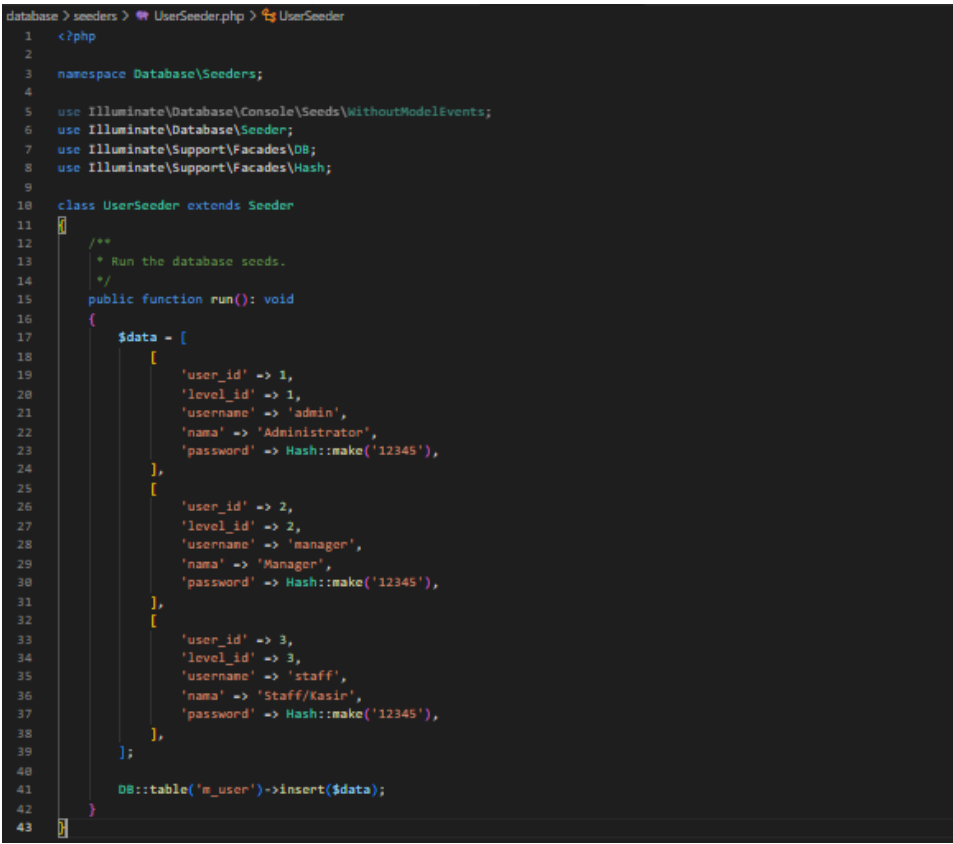
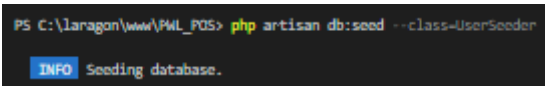
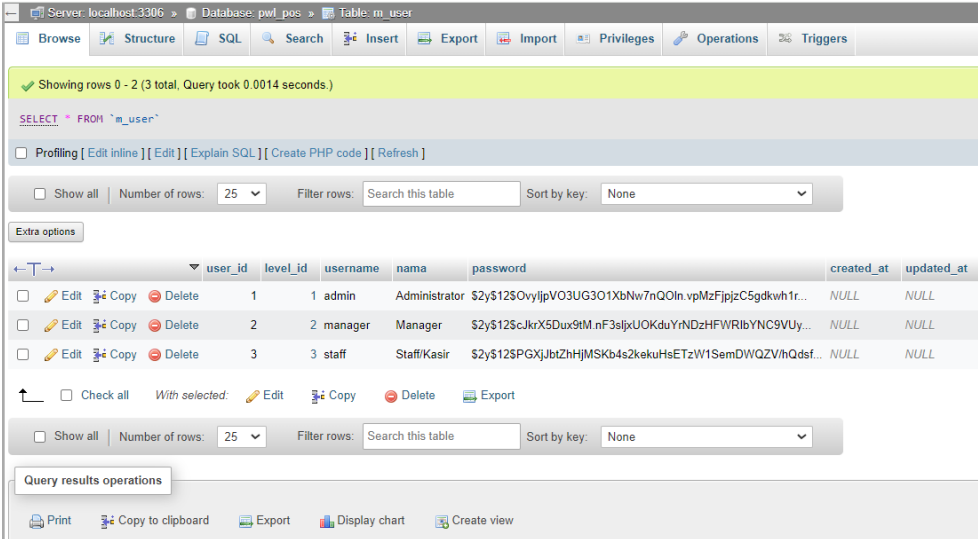
### Praktikum Ke-3 – Membuat file seeder

Langkah	Jawaban/Deskripsi
1	Kita akan membuat file seeder untuk table m_level dengan mengetikkan perintah <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:seeder LevelSeeder</pre> <p>INFO: Seeder [C:\laragon\www\PWL_POS\database\seeders\LevelSeeder.php] created successfully.</p>
2	Selanjutnya, untuk memasukkan data awal, kita modifikasi file tersebut di dalam function run() <pre>database &gt; seeders &gt; LevelSeeder.php &gt; LevelSeeder 1  &lt;?php 2 3  namespace Database\Seeders; 4 5  use Illuminate\Database\Console\Seeds\WithoutModelEvents; 6  use Illuminate\Database\Seeder; 7  use Illuminate\Support\Facades\DB; 8 9  class LevelSeeder extends Seeder 10 { 11     /** 12      * Run the database seeds. 13      */ 14     public function run(): void 15     { 16         \$data = [ 17             ['level_id' =&gt; 1, 'level_kode' =&gt; 'ADM', 'level_nama' =&gt; 'Administrator'], 18             ['level_id' =&gt; 2, 'level_kode' =&gt; 'MNG', 'level_nama' =&gt; 'Manager'], 19             ['level_id' =&gt; 3, 'level_kode' =&gt; 'STF', 'level_nama' =&gt; 'Staff/Kasir'], 20         ]; 21 22         DB::table('m_level')-&gt;insert(\$data); 23     } 24 }</pre>



3	<p>Selanjutnya, kita jalankan file <i>seeder</i> untuk table <i>m_level</i> pada terminal</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan db:seed --class=levelSeeder</pre> <p><b>INFO</b> Seeding database.</p>
4	<p>Ketika <i>seeder</i> berhasil dijalankan maka akan tampil data pada table <i>m_level</i></p>  <p>The screenshot shows the phpMyAdmin interface for the 'pwl_pos' database. The 'Table: m_level' is selected. The SQL tab is active, showing a 'SELECT * FROM `m_level`' query. Below the query, there are options for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (Search this table), and 'Sort by key' (None). The table data is displayed with columns: level_id, level_kode, level_nama, created_at, and updated_at. The data rows are: 1 ADM Administrator, 2 MNG Manager, and 3 STF Staff/Kasir. All created_at and updated_at fields are NULL. At the bottom, there are options for 'Query results operations' including Print, Copy to clipboard, Export, Display chart, and Create view.</p>
5	<p>Sekarang kita buat file <i>seeder</i> untuk table <i>m_user</i> yang me-<i>refer</i> ke table <i>m_level</i></p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:seeder UserSeeder</pre> <p><b>INFO</b> Seeder [C:\laragon\www\PWL_POS\database\seeders\UserSeeder.php] created successfully.</p>
6	<p>Modifikasi file class UserSeeder seperti berikut</p>



	
7	Jalankan perintah untuk mengeksekusi class UserSeeder 
8	Perhatikan hasil seeder pada table m_user 
9	Ok, data seeder berhasil di masukkan ke database.



10

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

- m\_kategori

```
PS C:\laragon\www\PWL_POS> php artisan make:seeder KategoriSeeder
INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\KategoriSeeder.php] created successfully.

database > seeders > KategoriSeeder.php > KategoriSeeder > run
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     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         $data = [
16             ['kategori_id' => 1, 'kategori_kode' => 'ELEC', 'kategori_nama' => 'Elektronik'],
17             ['kategori_id' => 2, 'kategori_kode' => 'FURN', 'kategori_nama' => 'Furniture'],
18             ['kategori_id' => 3, 'kategori_kode' => 'FOOD', 'kategori_nama' => 'Makanan'],
19             ['kategori_id' => 4, 'kategori_kode' => 'CLOT', 'kategori_nama' => 'Pakaian'],
20             ['kategori_id' => 5, 'kategori_kode' => 'BOOK', 'kategori_nama' => 'Buku'],
21         ];
22
23         DB::table('m_kategori')->insert($data);
24     }
25 }
```

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

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	ELEC	Elektronik	NULL	NULL
2	FURN	Furniture	NULL	NULL
3	FOOD	Makanan	NULL	NULL
4	CLOT	Pakaian	NULL	NULL
5	BOOK	Buku	NULL	NULL

- m\_supplier

```
PS C:\laragon\www\PWL_POS> php artisan make:seeder SupplierSeeder
INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\SupplierSeeder.php] created successfully.
```



database > seeders > SupplierSeeder.php > SupplierSeeder > run

```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use Illuminate\Support\Facades\DB;
7
8 class SupplierSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         $data = [
16             [
17                 'supplier_id' => 1, // Menentukan supplier_id secara manual
18                 'supplier_kode' => 'SUP001',
19                 'supplier_nama' => 'Supplier A',
20                 'supplier_alamat' => 'Jl. Meangga No. 1',
21             ],
22             [
23                 'supplier_id' => 2, // Menentukan supplier_id secara manual
24                 'supplier_kode' => 'SUP002',
25                 'supplier_nama' => 'Supplier B',
26                 'supplier_alamat' => 'Jl. Pepaya No. 2',
27             ],
28             [
29                 'supplier_id' => 3, // Menentukan supplier_id secara manual
30                 'supplier_kode' => 'SUP003',
31                 'supplier_nama' => 'Supplier C',
32                 'supplier_alamat' => 'Jl. Timun No. 3',
33             ]
34         ];
35
36         DB::table('m_supplier')->insert($data);
37     }
38 }
```

PS C:\laragon\www\PWL\_POS> php artisan db:seed --class=SupplierSeeder

INFO Seeding database.

Server: localhost:3306 > Database: pwl\_pos > Table: m\_supplier

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

SELECT \* FROM `m\_supplier`

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

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

Extra options

	supplier_id	supplier_kode	supplier_nama	supplier_alamat	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	SUP001	Supplier A	Jl. Meangga No. 1	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	SUP002	Supplier B	Jl. Pepaya No. 2	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	SUP003	Supplier C	Jl. Timun No. 3	NULL	NULL

Check all | With selected: Edit Copy Delete Export

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

Query results operations

Print Copy to clipboard Export Display chart Create view

- m\_barang

PS C:\laragon\www\PWL\_POS> php artisan make:seeder BarangSeeder

INFO Seeder [C:\laragon\www\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     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         $data = [];
16         $kategori = [1, 2, 3, 4, 5]; // Kategori IDs
17
18         for ($i = 1; $i <= 15; $i++) {
19             $data[] = [
20                 'barang_id' => $i,
21                 'kategori_id' => $kategori[array_rand($kategori)],
22                 'barang_kode' => 'BRG' . $i,
23                 'barang_nama' => 'Barang ' . $i,
24                 'harga_beli' => rand(10000, 50000),
25                 'harga_jual' => rand(60000, 100000),
26             ];
27         }
28
29         DB::table('m_barang')->insert($data);
30     }
31 }
```

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

INFO Seeding database.

Server: localhost:3306 Database: pwil\_pos Table: m\_barang

Showing rows 0 - 14 (15 total, Query took 0.0005 seconds.)

SELECT \* FROM 'm\_barang'

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

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

Extra options

	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	2	BRG1	Barang 1	40951	90560	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	BRG2	Barang 2	16217	68037	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	2	BRG3	Barang 3	27940	93297	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	BRG4	Barang 4	13874	93160	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	BRG5	Barang 5	47666	98674	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	5	BRG6	Barang 6	15320	60427	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	2	BRG7	Barang 7	29443	96774	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	5	BRG8	Barang 8	12567	81765	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	4	BRG9	Barang 9	24793	86328	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	4	BRG10	Barang 10	36806	76463	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	3	BRG11	Barang 11	15701	68971	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	2	BRG12	Barang 12	15676	78478	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	5	BRG13	Barang 13	13007	70478	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	1	BRG14	Barang 14	26410	76723	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	1	BRG15	Barang 15	27080	82758	NULL	NULL

Check all With selected: Edit Copy Delete Export

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

- t\_stok

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

INFO Seeder [C:\laragon\www\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     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [];
17         $now = Carbon::now();
18
19         for ($i = 1; $i <= 15; $i++) {
20             $data[] = [
21                 'stok_id' => $i,
22                 'supplier_id' => ($i % 3) + 1,
23                 'barang_id' => $i,
24                 'user_id' => 1,
25                 'stok_tanggal' => $now->subDays(rand(1, 30))->toDateTimeString(),
26                 'stok_jumlah' => rand(10, 100),
27             ];
28         }
29
30         DB::table('t_stok')->insert($data);
31     }
32 }
```

```
PS C:\laragon\www\PM_L_POS> php artisan db:seed --class=StokSeeder
```

INFO Seeding database.

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

Showing rows 0 - 14 (15 total. Query took 0.0006 seconds.)

SELECT \* FROM `t\_stok`

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

	stok_id	supplier_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	1	2	1	1	2024-09-12 05:57:24	90	NULL	NULL
<input type="checkbox"/>	2	3	2	1	2024-08-13 05:57:24	80	NULL	NULL
<input type="checkbox"/>	3	1	3	1	2024-07-15 05:57:24	89	NULL	NULL
<input type="checkbox"/>	4	2	4	1	2024-06-15 05:57:24	91	NULL	NULL
<input type="checkbox"/>	5	3	5	1	2024-05-19 05:57:24	49	NULL	NULL
<input type="checkbox"/>	6	1	6	1	2024-05-08 05:57:24	95	NULL	NULL
<input type="checkbox"/>	7	2	7	1	2024-04-24 05:57:24	34	NULL	NULL
<input type="checkbox"/>	8	3	8	1	2024-04-04 05:57:24	50	NULL	NULL
<input type="checkbox"/>	9	1	9	1	2024-03-10 05:57:24	77	NULL	NULL
<input type="checkbox"/>	10	2	10	1	2024-02-21 05:57:24	91	NULL	NULL
<input type="checkbox"/>	11	3	11	1	2024-01-23 05:57:24	18	NULL	NULL
<input type="checkbox"/>	12	1	12	1	2024-01-16 05:57:24	99	NULL	NULL
<input type="checkbox"/>	13	2	13	1	2023-12-21 05:57:24	15	NULL	NULL
<input type="checkbox"/>	14	3	14	1	2023-11-24 05:57:24	45	NULL	NULL
<input type="checkbox"/>	15	1	15	1	2023-11-19 05:57:24	76	NULL	NULL

Check all With selected: Edit Copy Delete Export

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

- t\_penjualan

```
PS C:\laragon\www\PM_L_POS> php artisan make:seeder PenjualanSeeder
```

INFO Seeder [C:\laragon\www\PM\_L\_POS\database\seeders\PenjualanSeeder.php] created successfully.



```
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     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [];
17         $now = Carbon::now();
18
19         for ($i = 1; $i <= 10; $i++) {
20             $data[] = [
21                 'penjualan_id' => $i,
22                 'user_id' => 1,
23                 'pembeli' => 'Pembeli ' . $i,
24                 'penjualan_kode' => 'PNJ' . $i,
25                 'penjualan_tanggal' => $now->subDays(rand(1, 30))>toDateTimeString(),
26             ];
27         }
28
29         DB::table('t_penjualan')->insert($data);
30     }
31 }
```

```
PS C:\laragon\www\PWL_POS> php artisan db:seed --class=PenjualanSeeder
```

INFO Seeding database.

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

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

SELECT \* FROM 't\_penjualan'

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

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

Extra options

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Pembeli 1	PNJ1	2024-09-09 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	Pembeli 2	PNJ2	2024-09-06 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	Pembeli 3	PNJ3	2024-08-21 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	1	Pembeli 4	PNJ4	2024-07-25 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	1	Pembeli 5	PNJ5	2024-07-01 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	1	Pembeli 6	PNJ6	2024-06-26 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	1	Pembeli 7	PNJ7	2024-06-09 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	1	Pembeli 8	PNJ8	2024-05-23 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	1	Pembeli 9	PNJ9	2024-05-10 06:03:49	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	1	Pembeli 10	PNJ10	2024-04-25 06:03:49	NULL	NULL

Check all | With selected: Edit Copy Delete Export

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

- t\_penjualan\_detail

```
PS C:\laragon\www\PWL_POS> php artisan make:seeder PenjualanDetailSeeder
```

INFO Seeder [C:\laragon\www\PWL\_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.





```
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     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         $data = [];
16
17         for ($i = 1; $i <= 10; $i++) {
18             for ($j = 1; $j <= 3; $j++) {
19                 $data[] = [
20                     'detail_id' => ($i - 1) * 3 + $j,
21                     'penjualan_id' => $i,
22                     'barang_id' => rand(1, 15),
23                     'harga' => rand(60000, 100000),
24                     'jumlah' => rand(1, 10),
25                 ];
26             }
27         }
28
29         DB::table('t_penjualan_detail')->insert($data);
30     }
31 }
```

```
PS C:\laragon\www\PWL_POS> php artisan db:seed --class=PenjualanDetailSeeder
```

INFO Seeding database.

	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/>	1	1	11	60342	3	NULL	NULL
<input type="checkbox"/>	2	1	6	86860	7	NULL	NULL
<input type="checkbox"/>	3	1	13	93165	4	NULL	NULL
<input type="checkbox"/>	4	2	9	90996	6	NULL	NULL
<input type="checkbox"/>	5	2	12	82428	8	NULL	NULL
<input type="checkbox"/>	6	2	10	90872	4	NULL	NULL
<input type="checkbox"/>	7	3	6	66065	8	NULL	NULL
<input type="checkbox"/>	8	3	5	70862	6	NULL	NULL
<input type="checkbox"/>	9	3	8	60456	4	NULL	NULL
<input type="checkbox"/>	10	4	12	94209	1	NULL	NULL
<input type="checkbox"/>	11	4	8	72625	7	NULL	NULL
<input type="checkbox"/>	12	4	13	82066	7	NULL	NULL
<input type="checkbox"/>	13	5	6	68275	3	NULL	NULL
<input type="checkbox"/>	14	5	2	64768	10	NULL	NULL
<input type="checkbox"/>	15	5	2	99720	6	NULL	NULL
<input type="checkbox"/>	16	6	10	73720	1	NULL	NULL
<input type="checkbox"/>	17	6	11	72935	7	NULL	NULL
<input type="checkbox"/>	18	6	2	90244	4	NULL	NULL
<input type="checkbox"/>	19	7	3	68656	10	NULL	NULL
<input type="checkbox"/>	20	7	15	66788	10	NULL	NULL
<input type="checkbox"/>	21	7	1	92415	7	NULL	NULL
<input type="checkbox"/>	22	8	1	78374	5	NULL	NULL
<input type="checkbox"/>	23	8	15	68943	8	NULL	NULL
<input type="checkbox"/>	24	8	8	60140	2	NULL	NULL
<input type="checkbox"/>	25	9	3	79336	8	NULL	NULL

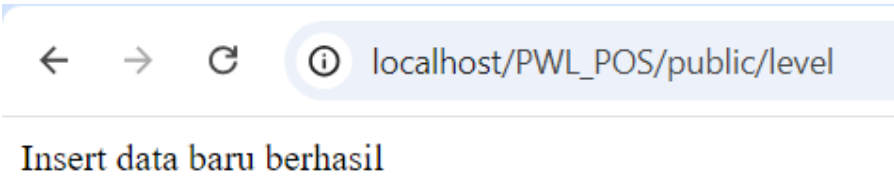
11

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

## Praktikum Ke-4 – Implementasi DB Facade

Langkah	Jawaban/Deskripsi
---------	-------------------



1	<p>Kita buat controller dahulu untuk mengelola data pada table m_level</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:controller LevelController</pre> <p>INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\LevelController.php] created successfully.</p>
2	<p>Kita modifikasi dulu untuk <i>routing</i>-nya, ada di PWL_POS/routes/web.php</p> <pre>routes &gt; web.php &gt; ... 1  &lt;?php 2  use App\Http\Controllers\LevelController; 3  use App\Http\Controllers\KategoriController; 4  use App\Http\Controllers\UserController; 5  use Illuminate\Support\Facades\Route; 6 7  Route::get('/', function() { 8      return view('welcome'); 9  }); 10 11 Route::get('/level', [LevelController::class, 'index']);</pre>
3	<p>Selanjutnya, kita modifikasi file LevelController untuk menambahkan 1 data ke table m_level</p> <pre>app &gt; Http &gt; Controllers &gt; LevelController.php &gt; LevelController 1  &lt;?php 2 3  namespace App\Http\Controllers; 4 5  use Illuminate\Http\Request; 6  use Illuminate\Support\Facades\DB; 7  use App\Http\Controllers\Controller; // Pastikan import Controller seperti ini 8 9  class LevelController extends Controller 10 { 11     public function index(){ 12         //INSERT DATA 13         DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?,?,?)', ['CUS', 'Pelanggan', now()]); 14         return 'Insert data baru berhasil'; 15     } 16     //UPDATE DATA 17     // \$row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']); 18     //return 'Update data berhasil. Jumlah data yang diupdate: '.\$row.' baris'; 19 20     //DELETE DATA 21     // \$row = DB::delete('delete from m_level where level_kode = ?', ['CUS']); 22     //return 'Delete data berhasil. Jumlah data yang dihapus: '.\$row.' baris'; 23 24     //VIEW DATA 25     // \$data = DB::select('select * from m_level'); 26     // return view('level', ['data'=&gt;\$data]); 27 } 28</pre>
4	<p>Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level dan amati apa yang terjadi pada table m_level di database, <i>screenshot</i> perubahan yang ada pada table m_level</p> 



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

Showing rows 0 - 3 (4 total, Query took 0.0014 seconds.)

SELECT \* FROM `m\_level`

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

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

Extra options

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	CUS	Pelanggan	2024-09-14 06:38:19	NULL

Check all | With selected: Edit Copy Delete Export

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

akan menambah data baru dengan level\_kode 'CUS' dan level\_nama 'Pelanggan' juga akan mengisi kolom created\_at yang diisi waktu saat kita membuat data tersebut

- 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 use App\Http\Controllers\Controller; // Pastikan import Controller seperti ini
8
9 class LevelController extends Controller
10 {
11     public function index(){
12         //INSERT DATA
13         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?,?,?)', ['CUS','Pelanggan', now()]);
14         // return 'Insert data baru berhasil';
15
16         //UPDATE DATA
17         $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
18         return 'Update data berhasil. Jumlah data yang diupdate: '.$row.' baris';
19
20         //DELETE DATA
21         // $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
22         //return 'Delete data berhasil. Jumlah data yang dihapus: '.$row.' baris';
23
24         //VIEW DATA
25         // $data = DB::select('select * from m_level');
26         // return view('level', ['data'=>$data]);
27     }
28 }
```

- 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

localhost/PWL\_POS/public/level

Update data berhasil. Jumlah data yang diupdate: 1 baris



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

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

SELECT \* FROM `m\_level`

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

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

Extra options

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	CUS	Customer	2024-09-14 06:38:19	NULL

Check all | With selected: Edit Copy Delete Export

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

level\_nama yang sebelumnya pelanggan berubah menjadi customer

7

Kita coba modifikasi lagi file LevelController untuk melakukan proses hapus data

```
app > Http > Controllers > LevelController.php > LevelController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7 use App\Http\Controllers\Controller; // Pastikan import Controller seperti ini
8
9 class LevelController extends Controller
10 {
11     public function index(){
12         //INSERT DATA
13         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?,?,?)', ['CUS','Pelanggan', now()]);
14         // return 'Insert data baru berhasil';
15
16         //UPDATE DATA
17         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
18         // return 'Update data berhasil. Jumlah data yang diupdate: '.$row.' baris';
19
20         //DELETE DATA
21         $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
22         return 'Delete data berhasil. Jumlah data yang dihapus : '.$row.' baris';
23
24         //VIEW DATA
25         // $data = DB::select('select * from m_level');
26         // return view('level', ['data'=>$data]);
27     }
28 }
```

localhost/PWL\_POS/public/level

Delete data berhasil. Jumlah data yang dihapus : 1 baris



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



```
resources > views > level.blade.php > html
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

← → ↻ ⓘ localhost/PWL\_POS/public/level

## Data Level Pengguna

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

akan menampilkan table yang memiliki 3 kolom yang berisi id, kode level dan nama level.

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

### Praktikum 5 – Implementasi *Query Builder*

Langkah	Jawaban/Deskripsi
1	<p>Kita buat controller dahulu untuk mengelola data pada table m_kategori</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:controller KategoriController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.</pre>
2	Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php



	<pre>routes &gt; web.php &gt; ... 1  &lt;?php 2  use App\Http\Controllers\LevelController; 3  use App\Http\Controllers\KategoriController; 4  use App\Http\Controllers\UserController; 5  use Illuminate\Support\Facades\Route; 6 7  Route::get('/', function() { 8      return view('welcome'); 9  }); 10 11 Route::get('/level', [LevelController::class, 'index']); 12 Route::get('/kategori', [KategoriController::class, 'index']); 13 // Route::get('/user', [UserController::class, 'index']);</pre>
3	<p>Selanjutnya, kita modifikasi file KategoriController untuk menambahkan 1 data ke table m_kategori</p> <pre>app &gt; Http &gt; Controllers &gt; KategoriController.php &gt; KategoriController &gt; index 1  &lt;?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' =&gt; 'SNK', 14             'kategori_nama' =&gt; 'Snack/Makanan Ringan', 15             'created_at' =&gt; now() 16         ]; 17 18         // INSERT 19         DB::table('m_kategori')-&gt;insert(\$data); 20         return 'insert data baru berhasil'; 21 22         //UPDATE 23         // \$row = DB::table('m_kategori')-&gt;where('kategori_kode', 'SNK')-&gt;update(['kategori_nama' =&gt; 'Camilan']); 24         //return 'update data berhasil. Jumlah data yg diupdate: '.\$row.' baris'; 25 26         //DELETE 27         // \$row = DB::table('m_kategori')-&gt;where('kategori_kode', 'SNK')-&gt;delete(); 28         //return 'delete data berhasil. jumlah data yg dihapus: '.\$row.' baris'; 29 30         //VIEW 31         // \$data = DB::table('m_kategori')-&gt;get(); 32         // return view('kategori', ['data' =&gt; \$data]); 33     } 34 }</pre>
4	<p>Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori dan amati apa yang terjadi pada table m_kategori di database, <i>screenshot</i> perubahan yang ada pada table m_kategori</p> <div></div>



Server: localhost:3306 > Database: pwl\_pos > Table: m\_kategori

Showing rows 0 - 5 (6 total, Query took 0.0017 seconds)

SELECT \* FROM `m\_kategori`

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

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

Extra options

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ELEC	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	FURN	Furniture	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	CLOT	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	BOOK	Buku	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Snack/Makanan Ringan	2024-09-14 07:52:36	NULL

Check all | With selected: Edit Copy Delete Export

Pada url localhost/PWL\_POS/public/kategori akan menampilkan output 'Insert data baru berhasil'

- 5 Selanjutnya, kita modifikasi lagi file KategoriController untuk meng-update data di table m\_kategori seperti berikut

```
app > Http > Controllers > KategoriController.php > KategoriController
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 {
11     public function index()
12     {
13         $data = [
14             'kategori_kode' => 'SNK',
15             'kategori_nama' => 'Snack/Makanan Ringan',
16             'created_at' => now()
17         ];
18
19         // INSERT
20         // DB::table('m_kategori')->insert($data);
21         // return 'insert data baru berhasil';
22
23         //UPDATE
24         $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
25         return 'update data berhasil. Jumlah data yg diupdate: '.$row.' baris';
26
27         //DELETE
28         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
29         //return 'delete data berhasil. jumlah data yg dihapus: '.$row.' baris';
30
31         //VIEW
32         // $data = DB::table('m_kategori')->get();
33         // return view('kategori', ['data' => $data]);
34     }
35 }
```

- 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





←

→

↺

localhost/PWL\_POS/public/kategori

update data berhasil. Jumlah data yg diupdate: 1 baris

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

BrowserStructureSQLSearchInsertExportImportPrivilegesTools

Showing rows 0 - 5 (6 total, Query took 0.0018 seconds.)

SELECT \* FROM `m\_kategori`

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

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

Extra options

				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	ELEC	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	FURN	Furniture	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	FOOD	Makanan	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	CLOT	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	BOOK	Buku	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Camilan	2024-09-14 07:52:36	NULL

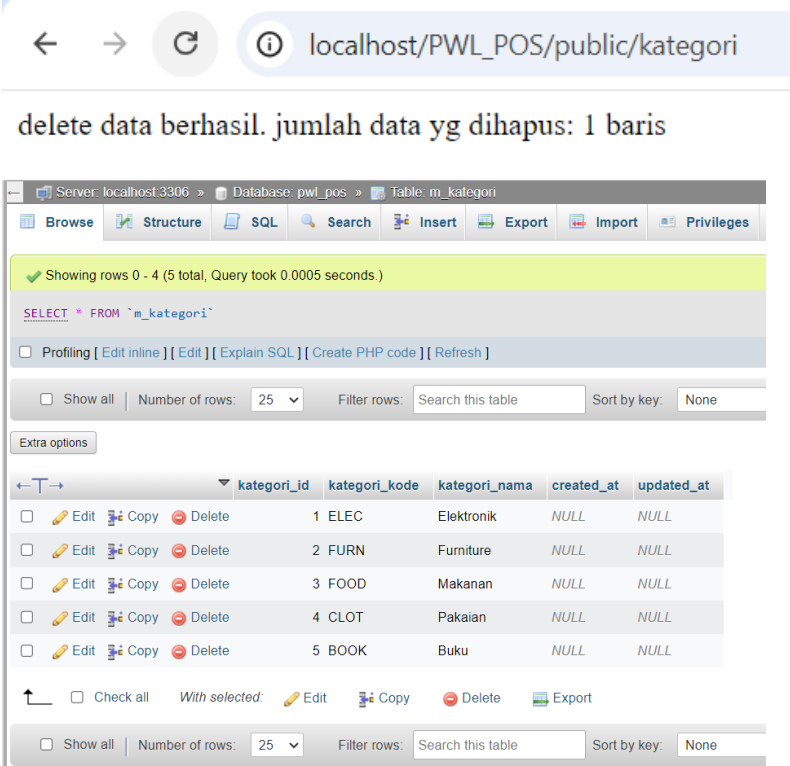
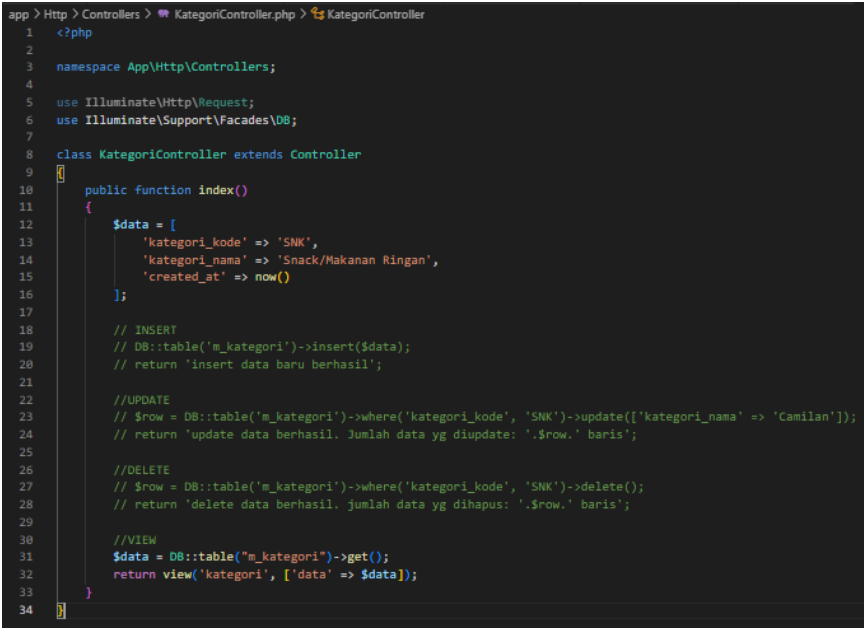
☐ Check all | With selected: Edit Copy Delete Export

Pada url localhost/PWL\_POS/public/kategori akan muncul tulisan ‘Update data berhasil. Jumlah data yang diupdate: 1 baris’. Pada database, kategori barang dengan kode SNK yang tadinya Bernama Snack/Makanan Ringan maka akan berubah menjadi Camilan.

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         $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17
18         // INSERT
19         // DB::table('m_kategori')->insert($data);
20         // return 'insert data baru berhasil';
21
22         //UPDATE
23         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
24         // return 'update data berhasil. Jumlah data yg diupdate: '.$row.' baris';
25
26         //DELETE
27         $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
28         return 'delete data berhasil. jumlah data yg dihapus: '.$row.' baris';
29
30         //VIEW
31         // $data = DB::table('m_kategori')->get();
32         // return view('kategori', ['data' => $data]);
33     }
34 }
```



	
8	<p>Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m_kategori. Kita modifikasi file KategoriController seperti berikut</p> 
9	<p>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</p>



```
resources > views > kategori.blade.php > html
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>
```

10 Silahkan dicoba pada browser dan amati apa yang terjadi.

## Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	ELEC	Elektronik
2	FURN	Furniture
3	FOOD	Makanan
4	CLOT	Pakaian
5	BOOK	Buku

akan menampilkan table yang memiliki 3 kolom yang berisi id, kode kategori dan nama kategori

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

### Praktikum 6 – Implementasi Eloquent ORM

Langkah	Jawaban/Deskripsi
1	<p>Kita buat file model untuk tabel m_user dengan mengetikkan perintah</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:model UserModel</pre> <p><b>INFO</b> Model [C:\laragon\www\PWL_POS\app\Models\UserModel.php] created successfully.</p>



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	<p>Kita buka file UserModel.php dan modifikasi seperti berikut</p> <pre>app &gt; Models &gt; UserModel.php &gt; UserModel 1  &lt;?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'; 13     protected \$primaryKey = 'user_id'; 14 }</pre>
4	<p>Kita modifikasi route web.php untuk mencoba routing ke controller UserController</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:controller UserController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\UserController.php] created successfully.</pre> <pre>routes &gt; web.php &gt; ... 1  &lt;?php 2  use App\Http\Controllers\LevelController; 3  use App\Http\Controllers\KategoriController; 4  use App\Http\Controllers\UserController; 5  use Illuminate\Support\Facades\Route; 6 7  Route::get('/', function() { 8      return view('welcome'); 9  }); 10 11  Route::get('/level', [LevelController::class, 'index']); 12  Route::get('/kategori', [KategoriController::class, 'index']); 13  Route::get('/user', [UserController::class, 'index']); 14   </pre>
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 use Illuminate\Support\Facades\Hash;
8
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data user dengan eloquent model
14         // $data = [
15             // 'username' => 'customer-1',
16             // 'nama' => 'Pelanggan',
17             // 'password' => hash::make('12345'),
18             // 'level_id' => 4
19         // ];
20
21         // UserModel::insert($data);
22
23         // 'nama' => 'Pelanggan Pertama',
24         // ];
25         // UserModel::where('username', 'customer-1')->update($data); // update data user
26
27         // Ambil semua data pengguna
28         $user = UserModel::all();
29
30         // Tampilkan view 'user' dengan data pengguna
31         return view('user', ['data' => $user]);
32     }
33 }
```

6 Kemudian kita buat view user.blade.php

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

7 Jalankan di browser, catat dan laporkan apa yang terjadi

localhost/PWL\_POS/public/user

### Data User

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

Menampilkan table dengan judul data user. Tabel memiliki 4 kolom yang berisi id, username, nama dan id level pengguna.

8 Setelah itu, kita modifikasi lagi file UserController



```
database > seeders > LevelSeeder.php > LevelSeeder > run
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8
9 class LevelSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [
17             ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
18             ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
19             ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
20             ['level_id' => 4, 'level_kode' => 'CST', 'level_nama' => 'Customer']
21         ];
22
23         DB::table('m_level')->insert($data);
24     }
25 }
26
```

```
app > Http > Controllers > UserController.php > _
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data use dengan eloquent model
14         $data = [
15             'username' => 'customer-1',
16             'nama' => 'Pelanggan',
17             'password' => hash::make('12345'),
18             'level_id' => 4
19         ];
20
21         // UserModel::insert($data);
22
23         // 'nama' => 'Pelanggan Pertama',
24         // ];
25         // UserModel::where('username', 'customer-1')->update($data); // update data user
26
27         // Ambil semua data pengguna
28         $user = UserModel::all();
29
30         // Tampilkan view 'user' dengan data pengguna
31         return view('user', ['data' => $user]);
32     }
33 }
```

9

Jalankan di browser, amati dan laporkan apa yang terjadi

← → ↺ ⓘ localhost/PWL\_POS/public/user

## Data User

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

10

Kita modifikasi lagi file UserController menjadi seperti berikut



```

app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data use dengan eloquent model
14         $data = [
15             // 'username' => 'customer-1',
16             // 'nama' => 'Pelanggan',
17             // 'password' => hash::make('12345'),
18             // 'level_id' => 4
19         ];
20
21         // UserModel::insert($data);
22         // 'nama' => 'Pelanggan Pertama',
23         ];
24         UserModel::where('username', 'customer-1')->update($data); // update data user
25
26         // Ambil semua data pengguna
27         $user = UserModel::all();
28
29         // Tampilkan view 'user' dengan data pengguna
30         return view('user', ['data' => $user]);
31     }
32 }
33
34

```

11 Jalankan di browser, amati dan laporkan apa yang terjadi

← → ↺ localhost/PWL\_POS/public/user

## Data User

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

Menampilkan table dengan judul data user. Tabel memiliki 4 kolom yang berisi id, username, nama dan id level pengguna.

12 Jika sudah, laporkan hasil Praktikum-6 ini dan *commit* perubahan pada *git*

## Penutup

Langkah	Jawaban/Deskripsi
1	<p>Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada <i>file setting</i> .env Laravel?</p> <p>APP_KEY dalam file .env Laravel berfungsi sebagai kunci enkripsi untuk melindungi data aplikasi. Jika kunci ini berubah, session dan data yang sudah terenkripsi tidak bisa lagi dibaca.</p>
2	<p>Pada Praktikum 1, bagaimana kita men-<i>generate</i> nilai untuk APP_KEY?</p> <p>Nilai untuk APP_KEY bisa dihasilkan dengan menjalankan perintah <code>php artisan key:generate</code>, yang secara otomatis mengisi kunci baru.</p>
3	<p>Pada Praktikum 2.1 - Tahap 1, secara <i>default</i> Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?</p>



	Secara default, Laravel menyediakan empat file migrasi. File-file migrasi default tersebut mencakup pembuatan tabel untuk pengguna, token reset password, pencatatan pekerjaan yang gagal dalam antrian, dan token akses pribadi untuk otentikasi API.
4	Secara <i>default</i> , file migrasi terdapat kode <code>\$table-&gt;timestamps();</code> , apa tujuan/ <i>output</i> dari fungsi tersebut? Fungsi <code>\$table-&gt;timestamps();</code> di dalam migrasi secara otomatis menambahkan dua kolom, yaitu <code>created_at</code> untuk menyimpan waktu pembuatan data dan <code>updated_at</code> untuk menyimpan waktu data terakhir diperbarui.
5	Pada File Migrasi, terdapat fungsi <code>\$table-&gt;id();</code> Tipe data apa yang dihasilkan dari fungsi tersebut? Fungsi <code>\$table-&gt;id();</code> menghasilkan kolom dengan tipe data BIGINT yang berfungsi sebagai primary key.
6	Apa bedanya hasil migrasi pada table <code>m_level</code> , antara menggunakan <code>\$table-&gt;id();</code> dengan menggunakan <code>\$table-&gt;id('level_id');</code> ? Jika menggunakan <code>\$table-&gt;id();</code> , nama kolomnya akan menjadi id secara default, sedangkan jika menggunakan <code>\$table-&gt;id('level_id');</code> , nama kolomnya akan menjadi <code>level_id</code> .
7	Pada migration, Fungsi <code>-&gt;unique()</code> digunakan untuk apa? Fungsi <code>-&gt;unique()</code> digunakan untuk memastikan nilai pada kolom tersebut tidak boleh sama, menjaga integritas data dengan mencegah duplikasi.
8	Pada Praktikum 2.2 - Tahap 2, kenapa kolom <code>level_id</code> pada tabel <code>m_user</code> menggunakan <code>\$table-&gt;unsignedBigInteger('level_id')</code> , sedangkan kolom <code>level_id</code> pada tabel <code>m_level</code> menggunakan <code>\$table-&gt;id('level_id')</code> ? Pada tabel <code>m_level</code> , <code>\$table-&gt;id('level_id')</code> digunakan untuk membuat primary key dengan nama <code>level_id</code> , sedangkan pada tabel <code>m_user</code> , <code>\$table-&gt;unsignedBigInteger('level_id')</code> digunakan untuk mendefinisikan foreign key yang menghubungkan ke kolom <code>level_id</code> di tabel <code>m_level</code> .
9	Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program <code>Hash::make('1234');</code> ? Class Hash digunakan untuk mengamankan data, terutama password, dengan mengubahnya menjadi hash yang tidak mudah dibalikkan ke bentuk aslinya. Fungsi <code>Hash::make('1234');</code> mengubah password '1234' menjadi bentuk hash.
10	Pada Praktikum 4 - Tahap 3/5/7, pada <i>query builder</i> terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut? Dalam query builder, tanda tanya (?) digunakan sebagai placeholder untuk parameter yang akan diisi ke query secara aman.





11	<p>Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode <code>protected \$table = 'm_user';</code> dan <code>protected \$primaryKey = 'user_id';</code> ?</p> <p>Kode <code>protected \$table</code> digunakan untuk memberi tahu Laravel nama tabel yang dihubungkan oleh model ini, memastikan model bekerja dengan tabel yang sesuai di database.</p>
12	<p>Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (<i>DB Façade / Query Builder / Eloquent ORM</i>) ? jelaskan</p> <p>Menggunakan DB façade lebih praktis karena kita bisa langsung bekerja dengan database. Cukup buat controller, atur routing, lalu modifikasi controller yang sudah dibuat.</p>