

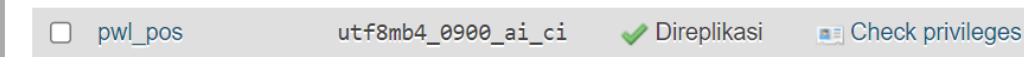
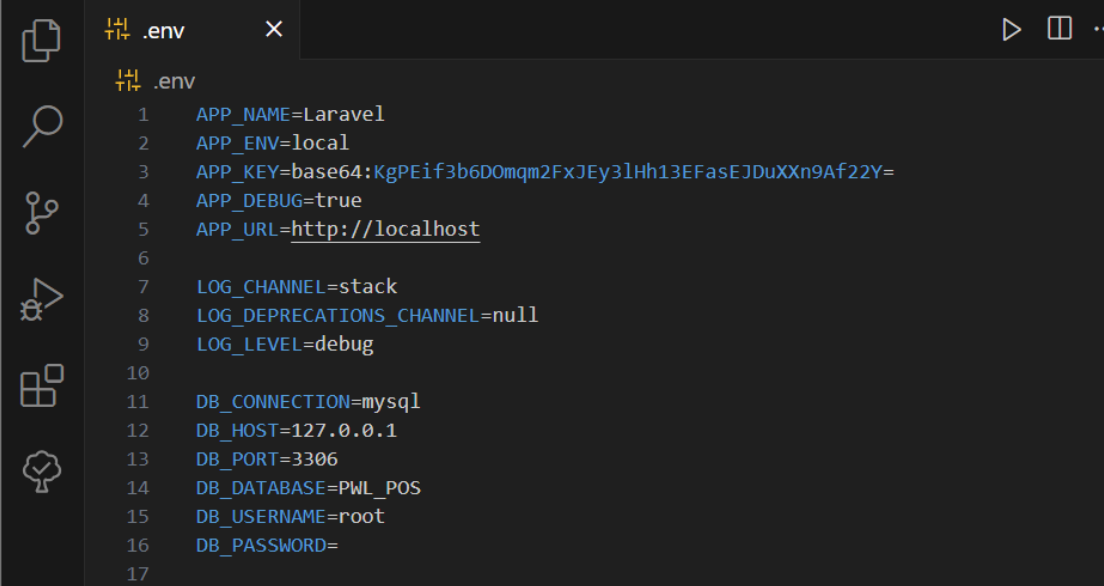


Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB
NIM : 2241760035
Nama : Fifi Novitasari
Jobsheet Ke- : 3

Laporan Jobsheet

Praktikum Ke-1: Pengaturan database


Langkah	Jawaban/Deskripsi
1	Membuat database baru “PWL_POS” 
2	Membuka folder project PWL_POS pada VsCode
3	Membuka file .env. Kemudian edit file .env dan sesuaikan dengan database yang telah dibuat. 



Praktikum Ke-2.1: Pembuatan file migrasi tanpa relasi

Langkah	Jawaban/Deskripsi
4	<p>Membuat file migrasi untuk tabel m_level dengan perintah berikut</p> <pre>C:\laragon\www\PWL_POS>php artisan make:migration create_m_level_table --create=m_level</pre> <p>INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_09_12_043600_create_m_level_table.php] created successfully.</p> <pre>C:\laragon\www\PWL_POS></pre>
5	<p>Modifikasi file dari migrasi yang telah dibuat seperti berikut</p> <pre>2024_09_12_043600_create_m_level_table.php X</pre> <pre>base > migrations > 2024_09_12_043600_create_m_level_table.php > class > up > Closure</pre> <pre>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 };</pre>
6	<p>Menjalankan perintah pada terminal untuk melakukan migrasi</p>



	<pre>Active code page: 65001 C:\laragon\www\PWL_POS>php artisan migrate INFO Preparing database. Creating migration table 69ms DONE INFO Running migrations. 2014_10_12_000000_create_users_table 87ms DONE 2014_10_12_100000_create_password_reset_tokens_table 32ms DONE 2019_08_19_000000_create_failed_jobs_table 63ms DONE 2019_12_14_000001_create_personal_access_tokens_table 95ms DONE 2024_09_12_043600_create_m_level_table 57ms DONE C:\laragon\www\PWL_POS></pre>
7	<p>Kemudian cek phpMyAdmin jika table sudah ter-generate</p> 
8	<p>Membuat table database dengan migration untuk table m_kategori dan m_supplier yang sama-sama tidak memiliki foreign key</p> <pre>C:\laragon\www\PWL_POS>php artisan make:migration m_kategori --create=m_kategori make:migration m_supplier --create=m_supplier INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_09_13_114904_m_kategori.php] created successfully. C:\laragon\www\PWL_POS></pre> <pre>C:\laragon\www\PWL_POS>php artisan make:migration m_supplier --create=m_supplier INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_09_13_115211_m_supplier.php] created successfully.</pre>



```
2024_09_13_114904_m_kategori.php X 2024_09_12_044211_create_m_user > ...

database > migrations > 2024_09_13_114904_m_kategori.php > ...

4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9
10     public function up(): void
11     {
12         Schema::create('m_kategori', function (Blueprint $table) {
13             $table->id('kategori_id');
14             $table->string('kategori_kode',100) -> unique();
15             $table->string('kategori_nama',100);
16             $table->timestamps();
17         });
18     }
19
20     public function down(): void
21     {
22         Schema::dropIfExists('m_kategori');
23     }
24 };
```

```
database > migrations > 2024_09_13_115211_m_supplier.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     public function up(): void
11     {
12         Schema::create('m_supplier', function (Blueprint $table) {
13             $table->id('supplier_id');
14             $table->string('supplier_kode',10) -> unique();
15             $table->string('supplier_nama',100);
16             $table->string('supplier_alamat',255);
17             $table->timestamps();
18         });
19     }
20
21     public function down(): void
22     {
23         Schema::dropIfExists('m_supplier');
24     }
25 };
```



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

INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_09_12_044211_create_m_user_t
able.php] created successfully.

C:\laragon\www\PWL_POS>
```

9

Kemudian migrate ke database dengan php artisan migrate

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 kategori_id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 kategori_kode	varchar(100)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	3 kategori_nama	varchar(100)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	4 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	5 updated_at	timestamp			Yes	NULL			

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 supplier_id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 supplier_kode	varchar(10)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	3 supplier_nama	varchar(100)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	4 supplier_alamat	varchar(255)	utf8mb4_unicode_ci		No	None			
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL			

Praktikum 2.2: Pembuatan file migrasi dengan relasi

10

Membuat dan memodifikasi file migrasi table 'm_user' seperti berikut

```
database > migrations > 2024_09_12_044211_create_m_user_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::table('m_user', function (Blueprint $table) {
15              $table->id('user_id');
16              $table->unsignedBigInteger('level_id')->index(); //indexing untuk ForeignKey
17              $table->string('username', 20)->unique(); // unique untuk memastikan tidak ada username yang sama
18              $table->string('nama', 100);
19              $table->string('password');
20              $table->timestamps();
21          });
22      }
23
24      /**
25       * Reverse the migrations.
26       */
27      public function down(): void
28      {
29          Schema::table('m_user', function (Blueprint $table) {
30              //
31          });
32      }
33  };
34
```

Migrate ke database



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	user_id	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	level_id	bigint(20)	UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	3	username	varchar(20)	utf8mb4_unicode_ci	No	None			Change Drop More
<input type="checkbox"/>	4	nama	varchar(100)	utf8mb4_unicode_ci	No	None			Change Drop More
<input type="checkbox"/>	5	password	varchar(255)	utf8mb4_unicode_ci	No	None			Change Drop More
<input type="checkbox"/>	6	created_at	timestamp		Yes	NULL			Change Drop More
<input type="checkbox"/>	7	updated_at	timestamp		Yes	NULL			Change Drop More

```
SQLSTATE[42S02]: Base table or view not found: 1146 Table 'pwl_pos_m_user' doesn't exist (Connection: mysql, SQL: alter table 'm_user' add 'user_id' bigint unsigned not null auto_increment primary key, add 'level_id' bigint unsigned not null, add 'username' varchar(20) not null, add 'nama' varchar(100) not null, add 'password' varchar(255) not null, add 'created_at' timestamp null, add 'updated_at' timestamp null )
at vendor\laravel\framework\src\Illuminate\Database\Connection.php:829
825         $this->getName(), $query, $this->prepareBindings($bindings), $e
826     );
827
828     throw new QueryException(
829         $this->getName(), $query, $this->prepareBindings($bindings), $e
830     );
831 }
832 }
833 }
1 vendor\laravel\framework\src\Illuminate\Database\Connection.php:587
PDOException: (SQLSTATE[42S02]: Base table or view not found: 1146 Table 'pwl_pos_m_user' doesn't exist")
2 vendor\laravel\framework\src\Illuminate\Database\Connection.php:587
PDOStatement->execute()
```

11

Melakukan setting migration untuk table lainnya yang memiliki foreign key

Active code page: 65001

C:\laragon\www\PWL_POS>php artisan make:migration m_barang --create=m_barang

INFO

 Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_13_133318_m_barang.php] created successfully.

C:\laragon\www\PWL_POS>php artisan make:migration t_penjualan --create=t_penjualan && php artisan make:migration t_penjualan_detail --create=t_penjualan_detail && php artisan make:migration t_stok --create=t_stok

INFO

 Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_13_133713_t_penjualan.php] created successfully.

INFO

 Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_13_133713_t_penjualan_detail.php] created successfully.

INFO

 Migration [C:\laragon\www\PWL_POS\database\migrations\2024_09_13_133713_t_stok.php] created successfully.



```
database > migrations > 2024_09_13_133318_m_barang.php > ...  
7   return new class extends Migration  
8   {  
9       public function up(): void  
10      {  
11          Schema::create('m_barang', function (Blueprint $table) {  
12              $table->id('barang_id');  
13              $table->unsignedBigInteger('kategori_id')->index();  
14              $table->string('barang_kode',100);  
15              $table->string('barang_nama',100);  
16              $table->integer('harga_heli');  
17              $table->integer('harga_iual');  
18              $table->timestamps();  
19  
20              $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori');  
21          });  
22      }  
23  
24      public function down(): void  
25      {  
26          Schema::dropIfExists('m_barang');  
27      }  
28  };
```

```
database > migrations > 2024_09_13_133713_t_penjualan.php > class > up > Closure  
7   return new class extends Migration  
10  {  
11      public function up(): void  
12      {  
13          Schema::create('t_penjualan', function (Blueprint $table) {  
14              $table->id('penjualan_id');  
15              $table->unsignedBigInteger('user_id')->index();  
16              $table->string('pembeli',50);  
17              $table->string('penjualan_kode',20);  
18              $table->dateTime('penjualan_tanggal');  
19              $table->timestamps();  
20  
21              $table->foreign('user_id')->references('user_id')->on('m_user');  
22          });  
23      }  
24  
25      public function down(): void  
26      {  
27          Schema::dropIfExists('t_penjualan');  
28      }  
29  };
```




	<pre>database > migrations > 2024_09_13_133713_t_penjualan_detail.php > class > up > Closure 4 use Illuminate\Database\Schema\Blueprint; 5 use Illuminate\Support\Facades\Schema; 6 7 return new class extends Migration 8 { 9 10 public function up(): void 11 { 12 Schema::create('t_penjualan_detail', function (Blueprint \$table) { 13 \$table->id('detail_id'); 14 \$table->unsignedBigInteger('penjualan_id')->index(); 15 \$table->unsignedBigInteger('barang_id')->index(); 16 \$table->integer('harga'); 17 \$table->integer('jumlah'); 18 \$table->timestamps(); 19 20 \$table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan'); 21 \$table->foreign('barang_id')->references('barang_id')->on('m_barang'); 22 }); 23 } 24 25 public function down(): void 26 { 27 Schema::dropIfExists('t_penjualan_detail'); 28 } 29 }; database > migrations > 2024_09_13_133713_t_stok.php > class > up > Closure 7 return new class extends Migration 8 { 9 10 public function up(): void 11 { 12 Schema::create('t_stok', function (Blueprint \$table) { 13 \$table->id('stok_id'); 14 \$table->unsignedBigInteger('supplier_id')->index(); 15 \$table->unsignedBigInteger('barang_id')->index(); 16 \$table->unsignedBigInteger('user_id')->index(); 17 \$table->dateTime('stok_tanggal'); 18 \$table->integer('stok_jumlah'); 19 \$table->timestamps(); 20 21 \$table->foreign('supplier_id')->references('supplier_id')->on('m_supplier'); 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 public function down(): void 28 { 29 Schema::dropIfExists('t_stok'); 30 } 31 };</pre>
12	Melakukan migrate dan cek struktur database pada designer phpMyAdmin



	<pre>graph TD m_supplier --> t_stok m_kategori --> m_barang m_barang --> t_penjualan_detail m_barang --> t_penjualan m_level --> m_user m_user --> t_penjualan</pre>
13	<pre>C:\laragon\www\PWL_POS>php artisan make:seeder <nama-class-seeder> The syntax of the command is incorrect. C:\laragon\www\PWL_POS> C:\laragon\www\PWL_POS>php artisan db:seed --class=<nama-class-seeder> The syntax of the command is incorrect. C:\laragon\www\PWL_POS></pre>

Praktikum Ke-3: Membuat file *seeder*

Langkah	Jawaban/Deskripsi
14	<p>Membuat seeder untuk table m_level</p> <pre>C:\laragon\www\PWL_POS>php artisan make:seeder LevelSeeder INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\LevelSeeder.php] created successfully. C:\laragon\www\PWL_POS></pre>



```
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
```

```
C:\laragon\www\PWL_POS>php artisan make:seeder --class=LevelSeeder
The syntax of the command is incorrect.
C:\laragon\www\PWL_POS>php artisan db:seed --class=nama-class-seeder
The syntax of the command is incorrect.
C:\laragon\www\PWL_POS>php artisan make:seeder LevelSeeder
INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\LevelSeeder.php] created successfully.
```

15

Modifikasi LevelSeeder.php

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

Menjalankan file seeder ke terminal

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

Cek data pada table m_level



	<div><div><div>← T →</div><div>level_idlevel_kodelevel_namacreated_atupdated_at</div><div><div><input type="checkbox"/></div><div></div>Ubah<div></div>Salin<div></div>Hapus</div><div>1ADMAdministratorNULLNULL</div><div><input type="checkbox"/></div><div></div>Ubah<div></div>Salin<div></div>Hapus</div><div>2MNGManagerNULLNULL</div><div><input type="checkbox"/></div><div></div>Ubah<div></div>Salin<div></div>Hapus</div> <div>3STFStaff/KasirNULLNULL</div> <div><div></div>Pilih Semua<div>Dengan pilihan:</div><div></div>Ubah<div></div>Salin<div></div>Hapus<div></div>Ekspor</div>
16	<div>Membuat file Userseeder</div> <div><pre>C:\laragon\www\PWL_POS>php artisan make:seeder UserSeeder</pre><div>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\UserSeeder.php] created successfully</div><pre>C:\laragon\www\PWL_POS></pre></div>
17	<div>Modifikasi file UserSeeder</div> <div><div>UserSeeder.php X</div><div>database > seeders > UserSeeder.php > UserSeeder > run</div><pre>1 <?php 2 namespace Database\Seeders; 3 4 use Illuminate\Database\Console\Seeds\WithoutModelEvents; 5 use Illuminate\Database\Seeder; 6 use Illuminate\Support\Facades\DB; // Import the DB facade 7 use Illuminate\Support\Facades\Hash; // Import the Hash facade 8 9 class UserSeeder extends Seeder 10 { 11 12 public function run(): void 13 { 14 \$data = [15 [16 'user_id' => 1, 17 'level_id' => 1, 18 'username' => 'admin', 19 'nama' => 'Administrator', 20 'password' => Hash::make('12345'), //class untuk mengenkripsi/hash password 21], 22 [23 'user_id' => 2, 24 'level_id' => 2, 25 'username' => 'manager', 26 'nama' => 'Manager', 27 'password' => Hash::make('12345'), 28], 29 [30 'user_id' => 3, 31 'level_id' => 3, 32 'username' => 'staff', 33 'nama' => 'Staff/Kasir', 34 'password' => Hash::make('12345'), 35], 36]; 37 DB::table('m_user')->insert(\$data); 38 } 39 }</pre></div>
18	<div>Menjalankan seeder untuk tabel m_user</div>



	<pre>C:\laragon\www\PWL_POS>php artisan db:seed --class=UserSeeder</pre> <p>INFO Seeding database.</p>																																
19	<p>Seeder berhasil masuk ke database</p> <div><div>Extra options</div><table><thead><tr><th></th><th>user_id</th><th>level_id</th><th>username</th><th>nama</th><th>password</th><th>created_at</th><th>updated_at</th></tr></thead><tbody><tr><td><input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete</td><td>1</td><td>1</td><td>admin</td><td>Administrator</td><td>\$2y\$12\$C.TibvkdIU/g53XFbXLAO.gjHfq0zQi1wLbAEHLzoQU...</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete</td><td>2</td><td>2</td><td>manager</td><td>Manager</td><td>\$2y\$12\$bUV698nf8sXy64gZwQT6KujE1thegmjQOCvyjQPRQMx...</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete</td><td>3</td><td>3</td><td>staff</td><td>Staff/Kasir</td><td>\$2y\$12\$.MQGjOczZTxfCAKiR09vqOpncUC7nh3ZGSYXgnMaxH5...</td><td>NULL</td><td>NULL</td></tr></tbody></table></div>		user_id	level_id	username	nama	password	created_at	updated_at	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$C.TibvkdIU/g53XFbXLAO.gjHfq0zQi1wLbAEHLzoQU...	NULL	NULL	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$bUV698nf8sXy64gZwQT6KujE1thegmjQOCvyjQPRQMx...	NULL	NULL	<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$.MQGjOczZTxfCAKiR09vqOpncUC7nh3ZGSYXgnMaxH5...	NULL	NULL
	user_id	level_id	username	nama	password	created_at	updated_at																										
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$C.TibvkdIU/g53XFbXLAO.gjHfq0zQi1wLbAEHLzoQU...	NULL	NULL																										
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$bUV698nf8sXy64gZwQT6KujE1thegmjQOCvyjQPRQMx...	NULL	NULL																										
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$.MQGjOczZTxfCAKiR09vqOpncUC7nh3ZGSYXgnMaxH5...	NULL	NULL																										
20	<p>Membuat data seeder untuk tabel lainnya</p> <pre>C:\laragon\www\PWL_POS>php artisan make:seeder KategoriSeeder && php artisan make:seeder SupplierSeeder && php artisan make:seeder BarangSeeder && php artisan make:seeder StokSeeder && php artisan make:seeder PenjualanSeeder && php artisan make:seeder PenjualanDetailSeeder</pre> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\KategoriSeeder.php] created successfully.</p> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\SupplierSeeder.php] created successfully.</p> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\BarangSeeder.php] created successfully.</p> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\StokSeeder.php] created successfully.</p> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\PenjualanSeeder.php] created successfully.</p> <p>INFO Seeder [C:\laragon\www\PWL_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.</p>																																
21	<p>Memodifikasi tiap file seeder</p>																																



```
database > seeders > PenjualanSeeder.php > PenjualanSeeder > run
```

```
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8  use Carbon\Carbon;
9
10 class PenjualanSeeder extends Seeder
11 {
12
13     public function run(): void
14     {
15         $data = [
16             [
17                 'penjualan_id' => 1,
18                 'user_id' => 3,
19                 'pembeli' => 'Adam',
20                 'penjualan_kode' => 'TRS001',
21                 'tanggal' => Carbon::create(2024, 9, 12)
22             ],
23             [
24                 'penjualan_id' => 2,
25                 'user_id' => 3,
26                 'pembeli' => 'Eve',
27                 'penjualan_kode' => 'TRS002',
28                 'tanggal' => Carbon::create(2024, 9, 13)
29             ]
30         ];
31     }
32 }
```

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

```
INFO Seeding database.
```

```
database > seeders > KategoriSeeder.php > ...
```

```
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class KategoriSeeder extends Seeder
10 {
11
12     public function run(): void
13     {
14         $data = [
15             [
16                 'kategori_id' => 1,
17                 'kategori_kode' => 'MKN',
18                 'kategori_nama' => 'Makanan',
19             ],
20             [
21                 'kategori_id' => 2,
22                 'kategori_kode' => 'MNM',
23                 'kategori_nama' => 'Minuman',
24             ]
25         ];
26         DB::table('kategori')->insert($data);
27     }
28 }
29 }
```



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

```
INFO Seeding database.
```

```
database > seeders > BarangSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class BarangSeeder extends Seeder
10 {
11
12     public function run(): void
13     {
14         $data = [
15             [
16                 'barang_id' => 1,
17                 'kategori_id' => 1,
18                 'barang_kode' => 'INDOF00001',
19                 'barang_nama' => 'Indomie Goreng',
20                 'harga_beli' => 2200,
21                 'harga_jual' => 2500,
22             ]
23         ];
24         DB::table('barang')->insert($data);
25     }
26 }
```

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

```
INFO Seeding database.
```



```
database > seeders > PenjualanDetailSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class PenjualanDetailSeeder extends Seeder
10 {
11
12     public function run(): void
13     {
14         $data = [
15             [
16                 'detail_id' => 1,
17                 'penjualan_id' => 1,
18                 'barang_id' => 1,
19                 'harga' => 2500,
20                 'jumlah' => 3,
21             ],
22             [
23                 'detail_id' => 2,
24                 'penjualan_id' => 1,
25                 'barang_id' => 2,
26                 'harga' => 5000,
27                 'jumlah' => 1,
28             ],
29         ];
30         DB::table('penjualan_detail')->insert($data);
31     }
32 }
33 }
```

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

```
INFO Seeding database.
```




```
database > seeders > SupplierSeeder.php > SupplierSeeder
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 SupplierSeeder extends Seeder
10 {
11
12     public function run(): void
13     {
14         $data = [
15             [
16                 'supplier_id' => 1,
17                 'supplier_kode' => 'MLG001',
18                 'supplier_nama' => 'Serba Guna Malang',
19                 'supplierAlamat' => 'Jalan Dr Cipto, Malang',
20             ],
21             [
22                 'supplier_id' => 2,
23                 'supplier_kode' => 'MLG002',
24                 'supplier_nama' => 'Serba Malang',
25                 'supplierAlamat' => 'Jalan Dr Cipto, Malang',
26             ]
27         ];
28
29         DB::table('supplier')->insert($data);
30     }
31 }
```

```
C:\laragon\www\PMI_POS>php artisan db:seed --class=SupplierSeeder
```

```
INFO Seeding database.
```



```
database > seeders > StokSeeder.php > StokSeeder
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8  use Carbon\Carbon;
9
10 class StokSeeder extends Seeder
11 {
12
13     public function run(): void
14     {
15         $data = [
16             [
17                 'stok_id' => 1,
18                 'supplier_id' => 1,
19                 'barang_id' => 1,
20                 'user_id' => 2,
21                 'stok_tanggal' => Carbon::create('2024', '9', '12'),
22                 'stok_jumlah' => 1000
23             ],
24             [
25                 'stok_id' => 2,
26                 'supplier_id' => 2,
27                 'barang_id' => 1,
28                 'user_id' => 2,
29                 'stok_tanggal' => Carbon::create('2024', '9', '12'),
30                 'stok_jumlah' => 2000
31             ]
32         ]; // Close the $data array with a semicolon
33
34         DB::table('stok')->insert($data); // Insert data into the stok table
35     }
36 }
```

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

INFO Seeding database.

Hasil di PhpMyAdmin

		▼	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at	
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	1	INDOFOOD001	Indomie Mie Goreng	2200	2500	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	2	CIMORY001	Cimory Yogurt Plain	4500	5000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	1	INDOFOOD002	Indomie Mie Kari	2200	2500	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	3	FC001	FaberCastell Pensil 2Bx10	4500	5000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	1	SASA001	Sasa GulaPasir 1kg	12000	13500	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	1	WINGS001	Mie Sedaap Goreng	2300	2600	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	2	ULTRA001	UltraMilk FullCream 1L	15000	16000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	8	4	FALCON001	Falcon Pulpen 0.5mm	2000	2500	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	5	XIAOMI001	Xiaomi MiPowerbank 10000mAh	150000	165000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	5	SAMSUNG001	Samsung USBCharger 15W	75000	80000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	1	GARUDA001	Garuda KacangGoreng 250g	15000	17500	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	12	2	AQUA001	Aqua Galon 19L	18000	20000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	13	3	COMBIPHAR001	OBH Combi 200ml	8000	10000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	14	5	SHARP001	Sharp LEDTV 32Inch	2000000	2150000	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	15	3	SANMOL001	Sanmol Paracetamol Tablet	2000	2500	NULL	NULL



<



Showing rows 0 - 14 (15 total, Query took 0.0008 seconds.)

SELECT * FROM `t_stok`

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

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

			stok_id	supplier_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>				1	1	1	2 2024-09-11 00:00:00	1000	NULL	NULL
<input type="checkbox"/>				2	2	5	2 2024-09-12 00:00:00	500	NULL	NULL
<input type="checkbox"/>				3	1	3	2 2024-09-10 00:00:00	1200	NULL	NULL
<input type="checkbox"/>				4	3	9	2 2024-09-09 00:00:00	300	NULL	NULL
<input type="checkbox"/>				5	2	14	2 2024-09-13 00:00:00	50	NULL	NULL
<input type="checkbox"/>				6	1	8	2 2024-09-07 00:00:00	2000	NULL	NULL
<input type="checkbox"/>				7	3	7	2 2024-09-06 00:00:00	1500	NULL	NULL
<input type="checkbox"/>				8	2	11	2 2024-09-14 00:00:00	3000	NULL	NULL
<input type="checkbox"/>				9	1	4	2 2024-09-08 00:00:00	500	NULL	NULL
<input type="checkbox"/>				10	3	2	2 2024-09-11 00:00:00	1200	NULL	NULL
<input type="checkbox"/>				11	2	12	2 2024-09-05 00:00:00	700	NULL	NULL
<input type="checkbox"/>				12	1	10	2 2024-09-10 00:00:00	400	NULL	NULL
<input type="checkbox"/>				13	3	13	2 2024-09-08 00:00:00	800	NULL	NULL
<input type="checkbox"/>				14	2	6	2 2024-09-12 00:00:00	2500	NULL	NULL
<input type="checkbox"/>				15	1	1	2 2024-09-13 00:00:00	3500	NULL	NULL



```
web.php X
routes > web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\LevelController;
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']);
```

24 Memodifikasi LevelController

```
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
8  class LevelController extends Controller
9  {
10     public function index() {
11         $kode = 'CUS';
12         $nama = 'Pelanggan';
13         $now = now();
14
15         DB::insert('insert into m_level(level_kode, level_nama, created_at) values (?, ?, ?)', [$kode, $nama, $now]);
16
17         return 'Insert data baru berhasil';
18     }
19 }
```

25 Mengakses URL localhost/pwl/PWL_POS/public/level dan cek data pada database

localhost/pwl/PWL_POS/public/level

Insert data baru berhasil

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Ubah <input type="button" value="Salin"/> <input type="button" value="Hapus"/>	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Ubah <input type="button" value="Salin"/> <input type="button" value="Hapus"/>	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Ubah <input type="button" value="Salin"/> <input type="button" value="Hapus"/>	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/> Ubah <input type="button" value="Salin"/> <input type="button" value="Hapus"/>	4	CUS	Pelanggan	2024-09-14 05:13:51	NULL

Data telah ditambahkan ke tabel m_level.



Modifikasi kembali file LevelController untuk update data pada tabel m_level.

```
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
8  class LevelController extends Controller
9  {
10     public function index() {
11         // $now = now();
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         $affectedRows = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
16         return 'Update data berhasil. Jumlah data yang diupdate: ' . $affectedRows . ' baris';
17     }
18
19 }
```

- 26 Jalankan URL localhost/pwl/PWL_POS/public/level dan cek database melalui PHPMYAdmin.

← ↻ ⓘ localhost/PWL_POS/public/level

Update data berhasil. Jumlah data yang diupdate: 0 baris

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	STF	Kasir	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	CUS	Customer	2024-09-10 14:00:02	NULL

Data level_nama pada tabel telah berubah.

- 27 Modifikasi LevelController untuk menghapus data pada tabel m_level.

```
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
8  class LevelController extends Controller
9  {
10     public function index() {
11         // $now = now();
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         // $affectedRows = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
16         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $affectedRows . ' baris';
17
18         $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
19         return 'hapus data berhasil. Jumlah data yang dihapus: ' . $row . ' baris.';
20     }
21
22 }
```

- 28 Memanggil URL localhost/PWL_POS/public/level



	<div><div>← ↻ ⓘ localhost/PWL_POS/public/level</div><div>Hapus data berhasil. Jumlah data yang dihapus: 1baris.</div><div>Cek database</div><div><table><tr><th></th><th></th><th></th><th></th><th>level_id</th><th>level_kode</th><th>level_nama</th><th>created_at</th><th>updated_at</th></tr><tr><td><input type="checkbox"/></td><td>Edit</td><td>Copy</td><td>Delete</td><td>1</td><td>ADM</td><td>Administrator</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/></td><td>Edit</td><td>Copy</td><td>Delete</td><td>2</td><td>MNG</td><td>Manager</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/></td><td>Edit</td><td>Copy</td><td>Delete</td><td>3</td><td>STF</td><td>Kasir</td><td>NULL</td><td>NULL</td></tr></table></div><div>Data customer telah dihapus</div></div>					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	Kasir	NULL	NULL
				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	Kasir	NULL	NULL																													
29	<div>Modifikasi LevelController untuk menampilkan data</div> <div><pre>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 // \$now = now(); 12 // DB::insert('insert into m_level(level_kode, level_nama, create 13 // return 'Insert data baru berhasil'; 14 15 // \$affectedRows = DB::update('update m_level set level_nama = ? 16 // return 'Update data berhasil. Jumlah data yang diupdate: ' . \$ 17 18 // \$row = DB::delete('delete from m_level where level_kode = ?', 19 // return 'Hapus data berhasil. Jumlah data yang dihapus: ' . \$row 20 21 \$data = DB::select('select * from m_level'); 22 return view('level', ['data'=> \$data]); 23 } 24 25 }</pre></div>																																				
30	Membuat view baru dengan nama level.blade.php																																				



```
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>
```

31

Jalankan URL pada browser:


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

LevelController mengambil data dari tabel m_level, kemudian memasukkan data tersebut kedalam variable \$data. Setelah itu, LevelController menjalankan return view “level.blade.php” dengan mengirimkan nilai \$data. Array dalam \$data ditampilkan dalam bentuk tabel.

Praktikum Ke-5: Implementasi *Query Builder*

Langkah	Jawaban/Deskripsi
1	Membuat KategoriController



	<pre>C:\laragon\www\PWL_POS>php artisan make:controller KategoriController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully. C:\laragon\www\PWL_POS></pre>
2	<p>Modifikasi route untuk menambahkan link /kategori</p> <pre>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 Route::get('/', function () { 8 return view('welcome'); 9 }); 10 11 Route::get('/level', [LevelController::class, 'index']); 12 Route::get('/kategori', [KategoriController::class, 'index']);</pre>
	<p>Selanjutnya, modifikasi file KategoriController untuk menambahkan 1 data ke table m_kategori</p> <pre>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 Route::get('/', function () { 8 return view('welcome'); 9 }); 10 11 Route::get('/level', [LevelController::class, 'index']); 12 Route::get('/kategori', [KategoriController::class, 'index']);</pre>
	<p>Akses URL localhost/PWL_POS/public/kategori dan cek database padaPHPMYAdmin.</p> <div></div>



Menjalankan URL dan Screenshot hasil perubahan pada database.

← → ↻ localhost/pwl/PWL_POS/public/kategori

Update data baru berhasil. Jumlah data yang diupdate: 1 baris.

			kategori_id	kategori_kode	kategori_nama	created_at	updated_at	
<input type="checkbox"/>	Edit	Copy	Delete	1	MKN	Makanan	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNM	Minuman	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	OBT	Obat - obatan	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	ATK	Alat tulis kantor	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	ELT	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Camilan	2024-09-10 15:45:32	NULL

Modifikasi KategoriController untuk menambahkan fungsi delete.

```
app > Http > Controllers > KategoriController.php > KategoriController > index
7
8 class KategoriController extends Controller
9 {
10     public function index()
11     {
12         // $data = [
13         //     'kategori_kode' => 'SNK',
14         //     'kategori_nama' => 'Snack/Makanan Ringan',
15         //     'created_at' => now()
16         // ];
17         // DB::table('m_kategori')->insert($data);
18         // return 'Insert data baru berhasil';
19
20         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama'
21         // return 'Update data baru berhasil. Jumlah data yang diupdate: ' . $row . ' baris.';
22
23         $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
24         return 'Delete data baru berhasil. Jumlah data yang diupdate: ' . $row . ' baris.';
25     }
26 }
```

Menjalankan URL dan screenshot tabel pada database

← → ↻ localhost/pwl/PWL_POS/public/kategori

Delete data baru berhasil. Jumlah data yang diupdate: 1 baris.

			kategori_id	kategori_kode	kategori_nama	created_at	updated_at	
<input type="checkbox"/>	Edit	Copy	Delete	1	MKN	Makanan	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNM	Minuman	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	OBT	Obat - obatan	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	ATK	Alat tulis kantor	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	ELT	Elektronik	NULL	NULL

Modifikasi KategoriController untuk menambahkan fungsi select.



	<pre>app > Http > Controllers > KategoriController.php > KategoriController 8 class KategoriController extends Controller 12 public function index() 15 16 // \$data = [17 // 'kategori_kode' => 'SNK', 18 // 'kategori_nama' => 'Snack/Makanan Ringan', 19 // 'created_at' => now() 20 //]; 21 // DB::table('m_kategori')->insert(\$data); 22 // return 'Insert data baru berhasil'; 23 24 // \$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_na 25 // return 'Update data baru berhasil. Jumlah data yang diupdate: ' . \$row . ' baris.'; 26 27 // \$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete(); 28 // return 'Delete data baru berhasil. Jumlah data yang diupdate: ' . \$row . ' baris.'; 29 30 \$data = DB::table('m_kategori') -> get(); 31 return view('kategori', ['data'=> \$data]); 32 33 } 34 35 }</pre>
	<p>Membuat view kategori.blade.php:</p> <pre>resources > views > kategori.blade.php > html > body > table > tr > td 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>Data Kategori Barang</title> 5 </head> 6 <body> 7 <h1>Data Kategori Barang</h1> 8 <table border="1" cellpadding="2" cellspacing="0"> 9 <tr> 10 <th>ID</th> 11 <th>Kode Kategori</th> 12 <th>Nama Kategori</th> 13 </tr> 14 @foreach (\$data as \$d) 15 <tr> 16 <td>{{ \$d->kategori_id }}</td> 17 <td>{{ \$d->kategori_kode }}</td> 18 <td>{{ \$d->kategori_nama }}</td> 19 </tr> 20 @endforeach 21 </table> 22 </body> 23 </html></pre>
	<p>Jalankan URL dan amati apa yang terjadi:</p>



← → ↻

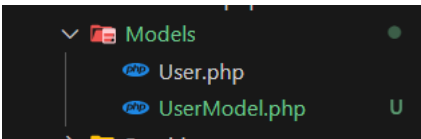
localhost/pwl/PWL_POS/public/kategori

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	MKN	Makanan
2	MNM	Minuman
3	OBT	Obat - obatan
4	ATK	Alat tulis kantor
5	ELT	Elektronik

KategoriController mengambil data dari tabel m_kategori, kemudian memasukkan data tersebut kedalam variable \$data. Setelah itu, KategoriController menjalankan return view kategori dengan mengirimkan nilai \$data. Array dalam \$data ditampilkan dalam bentuk tabel pada file kategori.blade.php.

Praktikum Ke-6: Implementasi Eloquent ORM

Langkah	Jawaban/Deskripsi
1	<p>Membuat file model untuk tabel m_user</p> <pre>C:\laragon\www\PWL_POS>php artisan make:model UserModel</pre> <p>INFO Model [C:\laragon\www\PWL_POS\app\Models\UserModel.php] created successfully.</p> <pre>C:\laragon\www\PWL_POS></pre> 
2	<p>Modifikasi file UserModel.php</p>



	<pre>app > Models > UserModel.php > UserModel 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 8 class UserModel extends Model 9 { 10 use HasFactory; 11 12 protected \$table = 'm_user'; 13 protected \$primaryKey = 'user_id'; 14 }</pre>
	<p>Menambahkan Route /user pada web.php</p> <pre>routes > web.php > ... 1 <?php 2 3 use App\Http\Controllers\KategoriController; 4 use App\Http\Controllers\LevelController; 5 use App\Http\Controllers\UserController; 6 use Illuminate\Support\Facades\Route; 7 8 Route::get('/', function () { 9 return view('welcome'); 10 }); 11 12 Route::get('/level', [LevelController::class, 'index']); 13 Route::get('/kategori', [KategoriController::class, 'index']); 14 Route::get('/user', [UserController::class, 'index']);</pre>
	<p>Membuat controller UserController dan modifikasi isinya.</p> <pre>C:\laragon\www\PWL_POS>php artisan make:controller UserController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\UserController.php] created successfully.</pre> <pre>app > Http > Controllers > UserController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use App\Models\UserModel; 7 8 class UserController extends Controller 9 { 10 public function index() { 11 \$user=UserModel::all(); 12 return view('user', ['data' => \$user]); 13 } 14 }</pre>
	<p>Membuat view baru bernama user.blade.php:</p>



```
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>
```

Jalankan URL 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

Modifikasi file UserController lagi:



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

Jalankan URL pada browser.

← → ↻ localhost/pwl/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
5	customer-1	Pelanggan	4

Modifikasi UserController lagi



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

Buka URL 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
5	customer-1	Pelanggan Pertama	4

Nama customer-1 telah diubah

Soal Praktikum: Penutup

Langkah	Jawaban/Deskripsi
1	Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada file setting .env Laravel? Jawab: APP_KEY digunakan untuk mengenkripsi data sensitif seperti password sebelum disimpan di database atau dikirim melewati internet.
2	Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP_KEY? Jawab: Dengan command “php artisan key:generate”



3	<p>Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?</p> <p>Jawab:</p> <ul style="list-style-type: none">• create_users_table : Untuk membuat tabel users pada database yang menyimpan data user Laravel.• create_passwords_reset_tokens_table: Untuk membuat tabel password_reset_token yang akan digunakan ketika user ingin mengganti passwordnya.• create_failed_jobs_table: Untuk membuat tabel yang akan menyimpan data tentang jobs yang gagal dijalankan.• create_personal_access_tokens_table: untuk membuat tabel yang akan menyimpan data token akses pribadi milik tiap user.
4	<p>Secara default, file migrasi terdapat kode <code>\$table->timestamps();</code>, apa tujuan/output dari fungsi tersebut?</p> <p>Jawab: Untuk menyimpan data waktu kapan file migration tersebut dijalankan.</p>
5	<p>Pada File Migrasi, terdapat fungsi <code>\$table->id();</code> Tipe data apa yang dihasilkan dari fungsi tersebut?</p> <p>Jawab: Tipe data <code>bigInteger</code>.</p>
6	<p>Apa bedanya hasil migrasi pada table <code>m_level</code>, antara menggunakan <code>\$table->id();</code> dengan menggunakan <code>\$table->id('level_id');</code> ?</p> <p>Jawab: Dengan memberikan parameter (<code>'level_id'</code>), fungsi akan membuat kolom untuk menyimpan nilai id dengan nama <code>"level_id"</code>. Apabila tidak diberi parameter, maka kolom akan bernama <code>"id"</code>.</p>
7	<p>Pada migration, Fungsi <code>->unique()</code> digunakan untuk apa?</p> <p>Jawab: <code>Unique()</code> digunakan agar kolom tersebut tidak memiliki nilai yang sama antara 2 data yang berbeda.</p>
8	<p>Pada Praktikum 2.2 - Tahap 2, kenapa kolom <code>level_id</code> pada tabel <code>m_user</code> menggunakan <code>\$table->unsignedBigInteger('level_id')</code> , sedangkan kolom <code>level_id</code> pada tabel <code>m_level</code> menggunakan <code>\$table->id('level_id')</code> ?</p> <p>Jawab: Karena <code>level_id</code> pada tabel <code>m_user</code> merupakan foreign key yang mereferensikan kolom <code>level_id</code> pada tabel <code>m_level</code>, sedangkan kolom <code>level_id</code> pada tabel <code>m_level</code> merupakan primary key.</p>
9	<p>Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program <code>Hash::make('1234');</code>?</p> <p>Jawab: Hash berguna untuk mengenkripsi data menggunakan algoritma Bcrypt sebelum dikirim ke database. Kode program <code>"Hash::make('1234')"</code> berarti program akan mengirim hasil enkripsi hash dari string <code>"1234"</code>.</p>
10	<p>Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?</p>



	Jawab: Tanda “?” berlaku sebagai placeholder untuk digantikan dengan nilai yang berada pada array yang diletakkan setelah query.
11	Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode <code>protected \$table = 'm_user';</code> dan <code>protected \$primaryKey = 'user_id';</code> ? Jawab: Protected berguna agar variable <code>\$table</code> dan <code>\$primaryKey</code> tidak dapat diubah oleh fungsi lain, karena UserModel hanya menggunakan tabel tersebut.
12	Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? Jelaskan! Jawab: Saya merasa lebih mudah menggunakan Query Builder, karena penggunaan fungsi lebih singkat daripada membuat query secara manual. Query builder juga memiliki struktur yang bentuknya mirip dengan query manual, dibandingkan dengan Eloquent ROM.

Link GitHub :