

Jobsheet 03: Migration, Seeder, Db Façade, Query Builder, dan Eloquent Orm

Dosen Pengampu Mata Kuliah :

M. Hasyim Ratsanjani, S.Kom., M.Kom.



Dibuat Oleh :

Daffa Yudisa Akbar : TI-2H / 10 : 2241720008

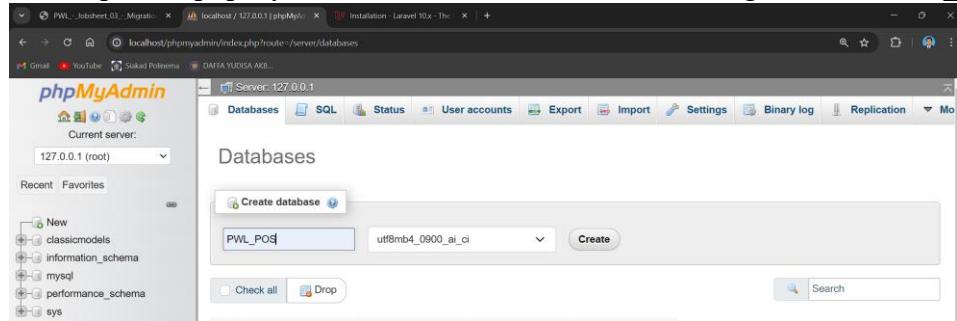
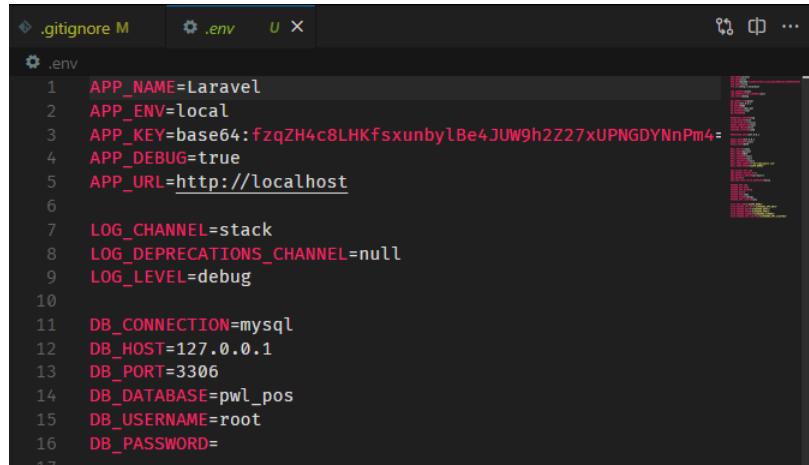
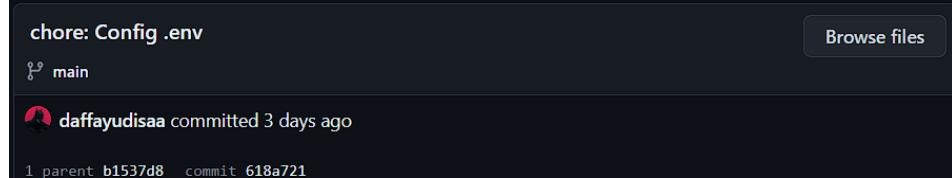
Politeknik Negeri Malang

Jurusan Teknologi Informasi

D-IV Teknik Informatika

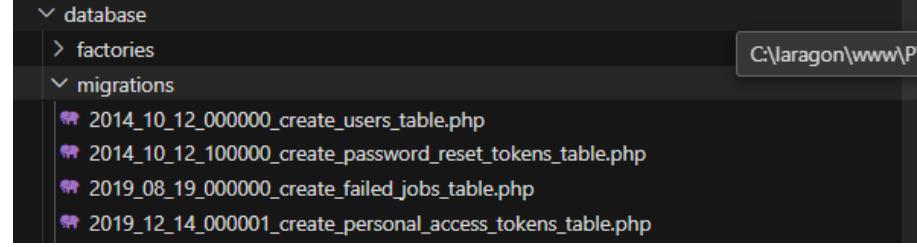
2024

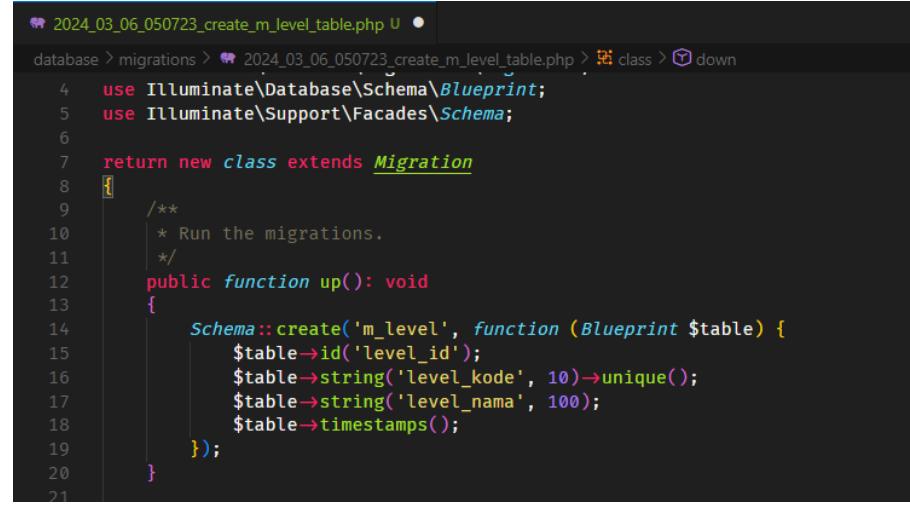
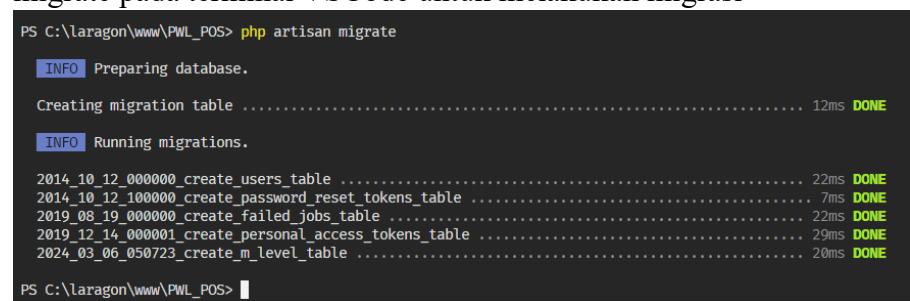
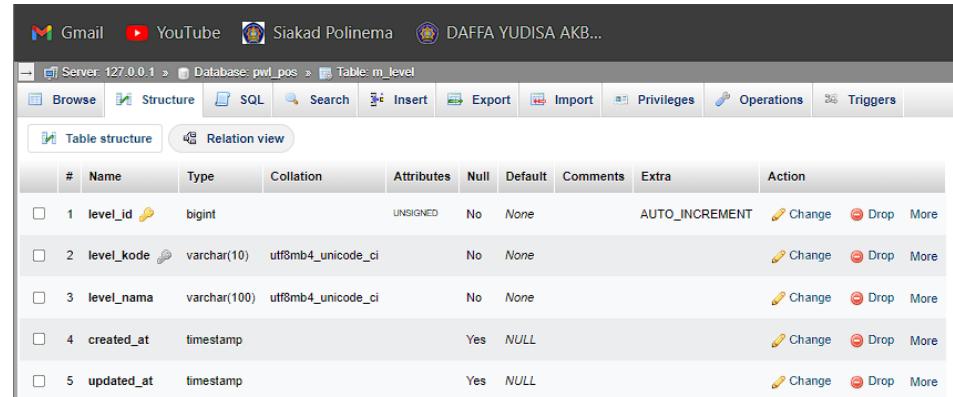
A. Praktikum 1 – Pengaturan Database

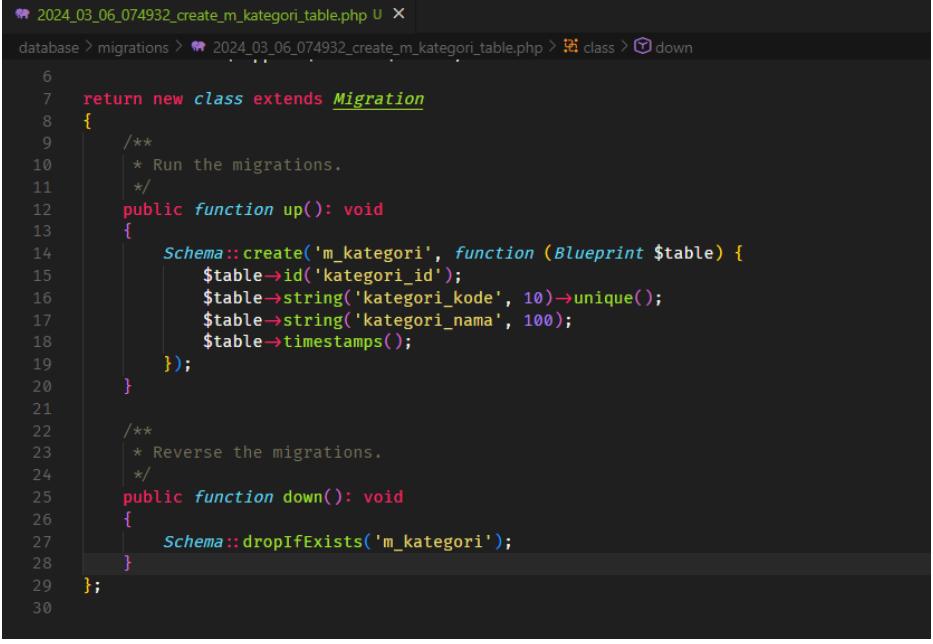
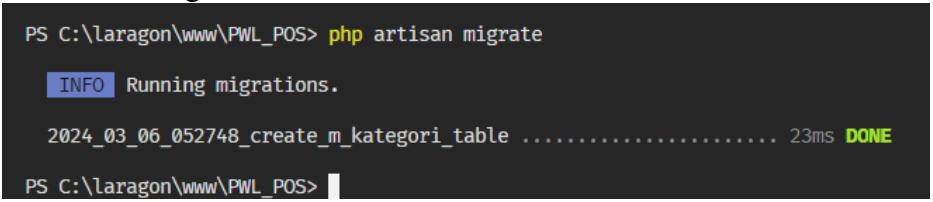
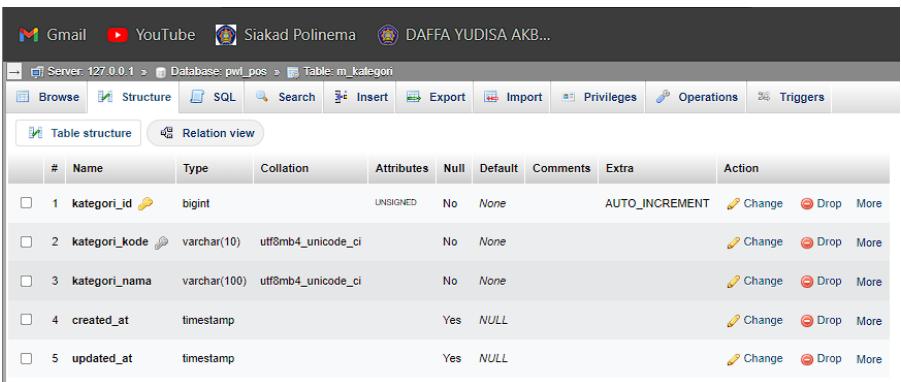
No	Keterangan dan Gambar
1.	Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL_POS 
2.	Buka file .env, dan pastikan konfigurasi APP_KEY bernilai. Lanjutkan dengan edit file .env dan sesuaikan dengan database yang telah dibuat 
3.	Laporkan hasil Praktikum-1 ini dan commit perubahan pada git. 

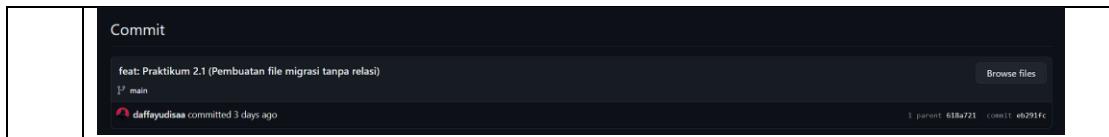
B. Praktikum 2 – Migration

- Praktikum 2.1 – Pembuatan File Migrasi Tanpa Relasi

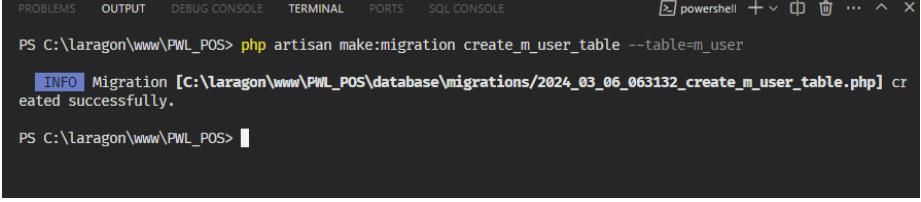
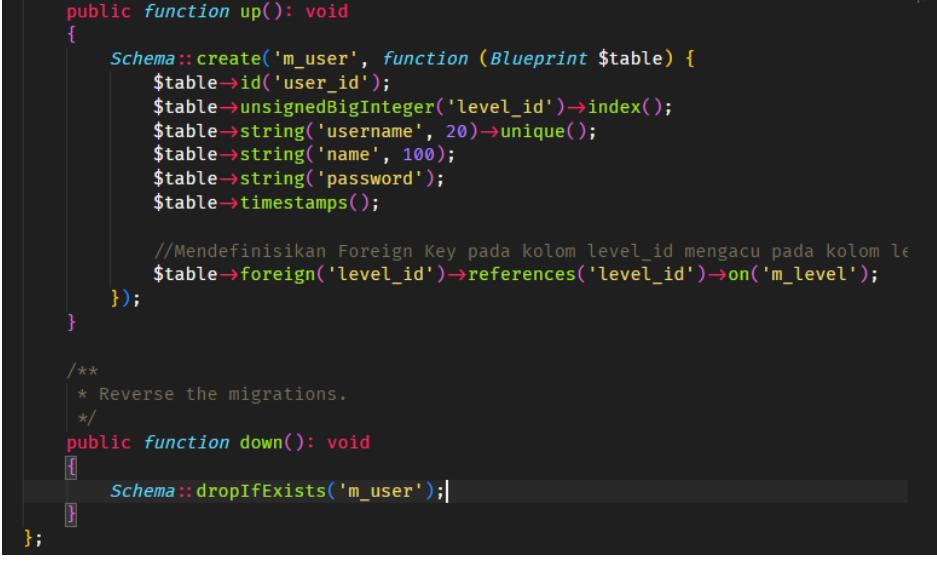
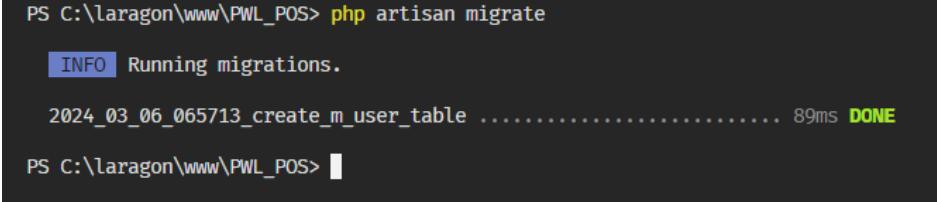
No.	Keterangan dan Gambar
1.	Buka terminal VSCode kalian, untuk yang di bawah ini adalah default dari laravel dan tidak perlu dihapus 

2.	<p>Kita buat file migrasi untuk table m_level dengan perintah</p>  <pre>PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_level_table --create=m_level INFO Migration [C:\laragon\www\PWL_POS\database\migrations\2024_03_06_050723_create_m_level_table.php] created successfully. PS C:\laragon\www\PWL_POS></pre>																																																												
3.	<p>Pada file migrasi yang kita buat, kita akan modifikasi sesuai desain database yang sudah ada</p>  <pre>2024_03_06_050723_create_m_level_table.php U • database > migrations > 2024_03_06_050723_create_m_level_table.php > class > down 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}</pre>																																																												
4.	<p>Simpan kode pada tahapan 3 tersebut, kemudian jalankan perintah php artisan migrate pada terminal VSCode untuk melakukan migrasi</p>  <pre>PS C:\laragon\www\PWL_POS> php artisan migrate INFO Preparing database. Creating migration table 12ms DONE INFO Running migrations. 2014_10_12_000000_create_users_table 22ms DONE 2014_10_12_100000_create_password_reset_tokens_table 7ms DONE 2019_08_19_000000_create_failed_jobs_table 22ms DONE 2019_12_14_000001_create_personal_access_tokens_table 29ms DONE 2024_03_06_050723_create_m_level_table 20ms DONE PS C:\laragon\www\PWL_POS></pre>																																																												
5.	<p>Kemudian kita cek di phpMyAdmin apakah table sudah ter-generate atau belum</p>  <table border="1"> <thead> <tr> <th>#</th> <th>Name</th> <th>Type</th> <th>Collation</th> <th>Attributes</th> <th>Null</th> <th>Default</th> <th>Comments</th> <th>Extra</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>level_id</td> <td>bigint</td> <td></td> <td>UNSIGNED</td> <td>No</td> <td>None</td> <td></td> <td>AUTO_INCREMENT</td> <td> Change Drop More</td> </tr> <tr> <td>2</td> <td>level_kode</td> <td>varchar(10)</td> <td>utf8mb4_unicode_ci</td> <td></td> <td>No</td> <td>None</td> <td></td> <td></td> <td> Change Drop More</td> </tr> <tr> <td>3</td> <td>level_nama</td> <td>varchar(100)</td> <td>utf8mb4_unicode_ci</td> <td></td> <td>No</td> <td>None</td> <td></td> <td></td> <td> Change Drop More</td> </tr> <tr> <td>4</td> <td>created_at</td> <td>timestamp</td> <td></td> <td></td> <td>Yes</td> <td>NULL</td> <td></td> <td></td> <td> Change Drop More</td> </tr> <tr> <td>5</td> <td>updated_at</td> <td>timestamp</td> <td></td> <td></td> <td>Yes</td> <td>NULL</td> <td></td> <td></td> <td> Change Drop More</td> </tr> </tbody> </table>	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	1	level_id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More	2	level_kode	varchar(10)	utf8mb4_unicode_ci		No	None			Change Drop More	3	level_nama	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More	4	created_at	timestamp			Yes	NULL			Change Drop More	5	updated_at	timestamp			Yes	NULL			Change Drop More
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action																																																				
1	level_id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More																																																				
2	level_kode	varchar(10)	utf8mb4_unicode_ci		No	None			Change Drop More																																																				
3	level_nama	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More																																																				
4	created_at	timestamp			Yes	NULL			Change Drop More																																																				
5	updated_at	timestamp			Yes	NULL			Change Drop More																																																				
6.	<p>Buat table database dengan migration untuk table m_kategori yang sama-sama tidak memiliki foreign key</p>																																																												

	<pre>PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_kategori_table --create=m_level [INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_03_06_051125_create_m_kategori_table.php] created successfully. PS C:\laragon\www\PWL_POS></pre>																																																								
7.	<p>Pada file migrasi yang kita buat, kita akan modifikasi sesuai desain database yang sudah ada</p>  <pre>2024_03_06_074932_create_m_kategori_table.php U database > migrations > 2024_03_06_074932_create_m_kategori_table.php > class > ↗ down 6 7 return new class extends Migration 8 { 9 /** 10 * Run the migrations. 11 */ 12 public function up(): void 13 { 14 Schema::create('m_kategori', function (Blueprint \$table) { 15 \$table->id('kategori_id'); 16 \$table->string('kategori.kode', 10)->unique(); 17 \$table->string('kategori.nama', 100); 18 \$table->timestamps(); 19 }); 20 } 21 22 /** 23 * Reverse the migrations. 24 */ 25 public function down(): void 26 { 27 Schema::dropIfExists('m_kategori'); 28 } 29 }; 30 </pre>																																																								
8.	<p>Kemudian jalankan perintah php artisan migrate pada terminal VSCode untuk melakukan migrasi</p>  <pre>PS C:\laragon\www\PWL_POS> php artisan migrate [INFO] Running migrations. 2024_03_06_052748_create_m_kategori_table 23ms DONE PS C:\laragon\www\PWL_POS></pre>																																																								
9.	<p>Kemudian kita cek di phpMyAdmin apakah table sudah ter-generate atau belum</p>  <table border="1"> <thead> <tr> <th>#</th> <th>Name</th> <th>Type</th> <th>Collation</th> <th>Attributes</th> <th>Null</th> <th>Default</th> <th>Comments</th> <th>Extra</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>kategori_id</td> <td>bigint</td> <td></td> <td>UNSIGNED</td> <td>No</td> <td>None</td> <td></td> <td>AUTO_INCREMENT</td> <td> More</td> </tr> <tr> <td>2</td> <td>kategori_kode</td> <td>varchar(10)</td> <td>utf8mb4_unicode_ci</td> <td></td> <td>No</td> <td>None</td> <td></td> <td> More</td> </tr> <tr> <td>3</td> <td>kategori_nama</td> <td>varchar(100)</td> <td>utf8mb4_unicode_ci</td> <td></td> <td>No</td> <td>None</td> <td></td> <td> More</td> </tr> <tr> <td>4</td> <td>created_at</td> <td>timestamp</td> <td></td> <td></td> <td>Yes</td> <td>NULL</td> <td></td> <td> More</td> </tr> <tr> <td>5</td> <td>updated_at</td> <td>timestamp</td> <td></td> <td></td> <td>Yes</td> <td>NULL</td> <td></td> <td> More</td> </tr> </tbody> </table>	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	1	kategori_id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More	2	kategori_kode	varchar(10)	utf8mb4_unicode_ci		No	None		More	3	kategori_nama	varchar(100)	utf8mb4_unicode_ci		No	None		More	4	created_at	timestamp			Yes	NULL		More	5	updated_at	timestamp			Yes	NULL		More
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action																																																
1	kategori_id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More																																																
2	kategori_kode	varchar(10)	utf8mb4_unicode_ci		No	None		More																																																	
3	kategori_nama	varchar(100)	utf8mb4_unicode_ci		No	None		More																																																	
4	created_at	timestamp			Yes	NULL		More																																																	
5	updated_at	timestamp			Yes	NULL		More																																																	
10.	Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.																																																								



- Praktikum 2.2 - Pembuatan file migrasi dengan relasi

No.	Keterangan dan Gambar
1.	Buka terminal VSCode kalian, dan buat file migrasi untuk table m_user  <pre>PS 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_03_06_063132_create_m_user_table.php] created successfully. PS C:\laragon\www\PWL_POS></pre>
2.	Buka file migrasi untuk table m_user, dan modifikasi  <pre>public function up(): void { Schema::create('m_user', function (Blueprint \$table) { \$table->id('user_id'); \$table->unsignedInteger('level_id')->index(); \$table->string('username', 20)->unique(); \$table->string('name', 100); \$table->string('password'); \$table->timestamps(); //Mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level \$table->foreign('level_id')->references('level_id')->on('m_level'); }); } /** * Reverse the migrations. */ public function down(): void { Schema::dropIfExists('m_user'); }</pre>
3.	Simpan kode program Langkah 2, dan jalankan perintah php artisan migrate. Amati apa yang terjadi pada database  <pre>PS C:\laragon\www\PWL_POS> php artisan migrate [INFO] Running migrations. 2024_03_06_065713_create_m_user_table 89ms DONE PS C:\laragon\www\PWL_POS></pre>

Maka setelah dijalankan perintah php artisan migrate, yang pada database PWL_POS akan dibuat sebuah tabel m_user yang berisi field yang sesuai dengan yang telah ditentukan.	<p>4. Buat table database dengan migration untuk table-tabel yang memiliki foreign key</p> <p>a. m_barang</p> <pre>PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_barang_table --table=m_barang [INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_03_06_070542_create_m_barang_table.php] created successfully. PS C:\laragon\www\PWL_POS> / public function up(): void { Schema::create('m_barang', function (Blueprint \$table) { \$table->id('barang_id'); \$table->string('barang.kode', 10)->unique(); \$table->unsignedBigInteger('kategori_id')->index(); \$table->string('barang.nama', 100); \$table->integer('harga_beli'); \$table->integer('harga_jual'); \$table->timestamps(); \$table->foreign('kategori_id')->references('kategori_id')->on('m_kategori'); }); } /** * Reverse the migrations. */ public function down(): void { Schema::dropIfExists('m_barang'); } ; PS C:\laragon\www\PWL_POS> php artisan migrate [INFO] Running migrations. 2024_03_06_075323_create_m_barang_table 63ms DONE PS C:\laragon\www\PWL_POS></pre> <p>b. t_penjualan</p>

```

PS C:\laragon\www\PWL_POS> php artisan make:migration create_t_penjualan_table --table=t_penjualan
[INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_03_06_081220_create_t_penjualan_table.php] created successfully.

PS C:\laragon\www\PWL_POS>
public function up(): void
{
    Schema::create('t_penjualan', function (Blueprint $table) {
        $table->id('penjualan_id');
        $table->unsignedBigInteger('user_id')->index();
        $table->string('pembeli', 50);
        $table->string('penjualan_kode', 20);
        $table->dateTime('penjualan_tanggal');
        $table->timestamps();

        $table->foreign('user_id')->references('user_id')->on('m_user');
    });
}

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

PS C:\laragon\www\PWL_POS> php artisan migrate
[INFO] Running migrations.

2024_03_06_081220_create_t_penjualan_table ..... 66ms DONE
PS C:\laragon\www\PWL_POS>

```

c. t_stok

```

PS C:\laragon\www\PWL_POS> php artisan make:migration create_t_stok_table --table=t_stok
[INFO] Migration [C:\laragon\www\PWL_POS\database\migrations\2024_03_06_082002_create_t_stok_table.php] created successfully.

PS C:\laragon\www\PWL_POS>
public function up(): void
{
    Schema::create('t_stok', function (Blueprint $table) {
        $table->id('stok_id');
        $table->unsignedBigInteger('barang_id')->index();
        $table->unsignedBigInteger('user_id')->index();
        $table->dateTime('stok_tanggal');
        $table->integer('stok_jumlah');
        $table->timestamps();

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

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

PS C:\laragon\www\PWL_POS> php artisan migrate
[INFO] Running migrations.

2024_03_06_082002_create_t_stok_table ..... 90ms DONE
PS C:\laragon\www\PWL_POS>

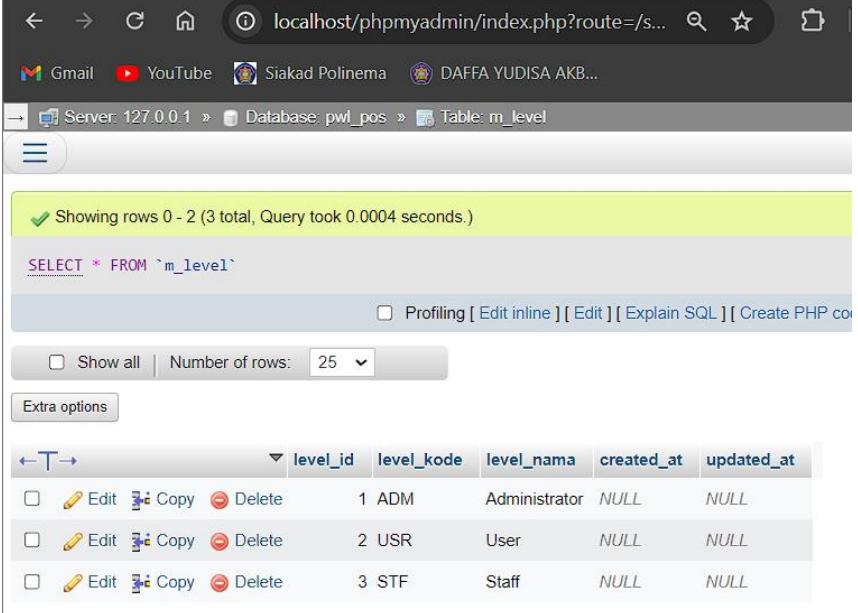
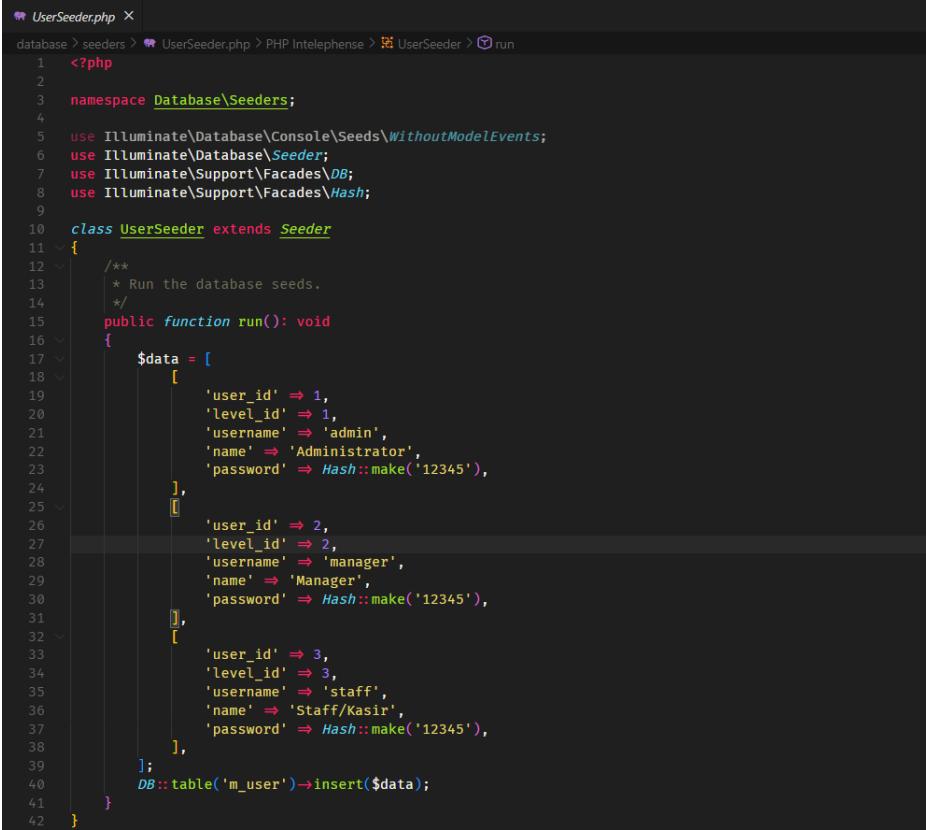
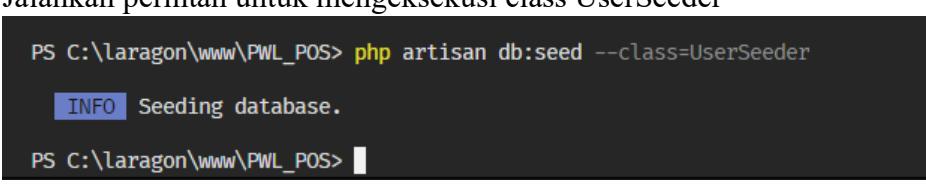
```

d. t_penjualan_detail

	<pre> INFO Migration [c:\laragon\www\PWL_POS\database\migrations\2024_03_06_082401_create_t_penjualan_detail_table.php] created successfully. PS C:\laragon\www\PWL_POS> */ public function up(): void { Schema::create('t_penjualan_detail', function (Blueprint \$table) { \$table->id('detail_id'); \$table->unsignedBigInteger('penjualan_id')->index(); \$table->unsignedBigInteger('barang_id')->index(); \$table->integer('harga'); \$table->integer('jumlah'); \$table->foreign('penjualan_id')->references('penjualan_id')->on('t_per'); \$table->foreign('barang_id')->references('barang_id')->on('m_barang'); }); } /** * Reverse the migrations. */ public function down(): void { Schema::dropIfExists('t_penjualan_detail'); } }; PS C:\laragon\www\PWL_POS> php artisan migrate INFO Running migrations. 2024_03_06_082401_create_t_penjualan_detail_table 235ms DONE PS C:\laragon\www\PWL_POS> </pre>
5.	<p>Jika semua file migrasi sudah di buat dan dijalankan maka bisa kita lihat tampilan designer pada phpMyAdmin</p>
6.	<p>Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git</p>

C. Praktikum 3 – Membuat File Seeder

No.	Keterangan dan Gambar
1.	<p>Kita akan membuat file seeder untuk table m_level dengan mengetikkan perintah</p> <pre>PS C:\laragon\www\PWL_POS> php artisan make:seeder LevelSeeder [INFO] Seeder [C:\laragon\www\PWL_POS\database\seeders\LevelSeeder.php] created successfully. PS C:\laragon\www\PWL_POS></pre>
2.	<p>Selanjutnya, untuk memasukkan data awal, kita modifikasi file tersebut di dalam function run()</p> <pre>database > seeders > LevelSeeder.php > PHP Intelephense > 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' => 'USR', 'level_nama' => 'User'], 19 ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff'] 20]; 21 22 DB::table('m_level')->insert(\$data); 23 } 24 } 25</pre>
3.	<p>Selanjutnya, kita jalankan file seeder untuk table m_level pada terminal</p> <pre>PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE powershell + × PS C:\laragon\www\PWL_POS> php artisan make:seeder LevelSeeder [INFO] Seeder [C:\laragon\www\PWL_POS\database\seeders\LevelSeeder.php] created successfully. PS C:\laragon\www\PWL_POS> php artisan db:seed --class=LevelSeeder [INFO] Seeding database. PS C:\laragon\www\PWL_POS></pre>
4.	<p>Ketika seeder berhasil dijalankan maka akan tampil data pada table m_level</p>

	
5.	<p>Sekarang kita buat file seeder untuk table m_user yang me-refer ke table m_level serta melakukan modifikasi file class UserSeeder</p>  <pre> UserSeeder.php < /> database > seeders > UserSeeder.php > PHP Intelephense > UserSeeder > run 1 <?php 2 3 namespace Database\Seeders; 4 5 use Illuminate\Database\Console\Seeds\WithoutModelEvents; 6 use Illuminate\Database\Seeder; 7 use Illuminate\Support\Facades\DB; 8 use Illuminate\Support\Facades\Hash; 9 10 class UserSeeder extends Seeder 11 { 12 /** 13 * Run the database seeds. 14 */ 15 public function run(): void 16 { 17 \$data = [18 [19 'user_id' => 1, 20 'level_id' => 1, 21 'username' => 'admin', 22 'name' => 'Administrator', 23 'password' => Hash::make('12345'), 24], 25 [26 'user_id' => 2, 27 'level_id' => 2, 28 'username' => 'manager', 29 'name' => 'Manager', 30 'password' => Hash::make('12345'), 31], 32 [33 'user_id' => 3, 34 'level_id' => 3, 35 'username' => 'staff', 36 'name' => 'Staff/Kasir', 37 'password' => Hash::make('12345'), 38], 39]; 40 DB::table('m_user')->insert(\$data); 41 } 42 } </pre>
6.	<p>Jalankan perintah untuk mengeksekusi class UserSeeder</p>  <pre> PS C:\laragon\www\PWL_POS> php artisan db:seed --class=UserSeeder INFO Seeding database. PS C:\laragon\www\PWL_POS> </pre>

7.

Perhatikan hasil seeder pada table m_user

```

SELECT * FROM `m_user`

+-----+-----+-----+-----+-----+-----+-----+
| user_id | level_id | username | name | password | created_at | updated_at |
+-----+-----+-----+-----+-----+-----+-----+
| 1       | 1       | admin    | Administrator | $2y$12$VnmytrIKeoxJvx4yWibdu7Losqru1RfYv2XoGU/T... | NULL      | NULL      |
| 2       | 2       | manager  | Manager     | $2y$12$SJAdJCpCCMd95004T4BcJEeKLX1cSOTG.78qnUlVFYz... | NULL      | NULL      |
| 3       | 3       | staff    | Staff/Kasir | $2y$12$RIuamUkJ6nrxCU/Pg6updOQ2IDhxux6lIn1J4FURzQ... | NULL      | NULL      |
+-----+-----+-----+-----+-----+-----+-----+
  
```

8.

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

No	Nama Tabel	Jumlah Data	Keterangan
1	m_kategori	5	5 kategori barang
2	m_barang	10	10 barang yang berbeda
3	t_stok	10	Stok untuk 10 barang
4	t_penjualan	10	10 transaksi penjualan
5	t_penjualan_detail	30	3 barang untuk setiap transaksi penjualan

- m_kategori

```

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

```

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class KategoriSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data=[
            ['kategori_id' => 1, 'katoegori_kode' => 'KAT001', 'kategori_nama' => 'Makanan'],
            ['kategori_id' => 2, 'katoegori_kode' => 'KAT002', 'kategori_nama' => 'Minuman'],
            ['kategori_id' => 3, 'katoegori_kode' => 'KAT003', 'kategori_nama' => 'Pakaiian'],
            ['kategori_id' => 4, 'katoegori_kode' => 'KAT004', 'kategori_nama' => 'Elektronik'],
            ['kategori_id' => 5, 'katoegori_kode' => 'KAT005', 'kategori_nama' => 'Alat Rumah Tangga']
        ];
    }
}
  
```

```

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

```

INFO Seeding database.
  
```

```

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

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Server: 127.0.0.1 Database: pwl_pos Table: m_kategori

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

```
SELECT * FROM `m_kategori`
```

Profiling [Edit inline]

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
Edit Copy Delete	1	KAT001	Makanan	NULL	NULL
Edit Copy Delete	2	KAT002	Minuman	NULL	NULL
Edit Copy Delete	3	KAT003	Pakaian	NULL	NULL
Edit Copy Delete	4	KAT004	Elektronik	NULL	NULL
Edit Copy Delete	5	KAT005	Alat Rumah Tangga	NULL	NULL

- 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.

PS C:\laragon\www\PWL_POS>

```

namespace Database\Seeders;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;

class BarangSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            ['barang_id' => 1, 'barang.kode' => 'B0001', 'kategori_id' => '1', 'barang.nama' => 'Salad Buah', 'harga_beli' => 15000, 'harga_jual' => 20000],
            ['barang_id' => 2, 'barang.kode' => 'B0002', 'kategori_id' => '1', 'barang.nama' => 'Cheese Cake', 'harga_beli' => 20000, 'harga_jual' => 30000],
            ['barang_id' => 3, 'barang.kode' => 'B0003', 'kategori_id' => '2', 'barang.nama' => 'Jus Apel', 'harga_beli' => 8000, 'harga_jual' => 10000],
            ['barang_id' => 4, 'barang.kode' => 'B0004', 'kategori_id' => '2', 'barang.nama' => 'Milkshake', 'harga_beli' => 10000, 'harga_jual' => 12000],
            ['barang_id' => 5, 'barang.kode' => 'B0005', 'kategori_id' => '3', 'barang.nama' => 'Kaos Polo', 'harga_beli' => 70000, 'harga_jual' => 80000],
            ['barang_id' => 6, 'barang.kode' => 'B0006', 'kategori_id' => '3', 'barang.nama' => 'Kemeja Flannel', 'harga_beli' => 80000, 'harga_jual' => 150000],
            ['barang_id' => 7, 'barang.kode' => 'B0007', 'kategori_id' => '4', 'barang.nama' => 'Mouse Gaming', 'harga_beli' => 300000, 'harga_jual' => 450000],
            ['barang_id' => 8, 'barang.kode' => 'B0008', 'kategori_id' => '4', 'barang.nama' => 'Lampu Neon Flex', 'harga_beli' => 30000, 'harga_jual' => 50000],
            ['barang_id' => 9, 'barang.kode' => 'B0009', 'kategori_id' => '5', 'barang.nama' => 'Sapu Dua Macan', 'harga_beli' => 10000, 'harga_jual' => 15000],
            ['barang_id' => 10, 'barang.kode' => 'B0010', 'kategori_id' => '5', 'barang.nama' => 'Kemoceng Dua Ayam', 'harga_beli' => 10000, 'harga_jual' => 15000]
        ];
        DB::table('m_barang')->insert($data);
    }
}

```

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

INFO Seeding database.

PS C:\laragon\www\PWL_POS>

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Server: 127.0.0.1 Database: pwl_pos Table: m_barang

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 9 (10 total, Query took 0.0003 seconds.)

```
SELECT * FROM `m_barang`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create]

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

Extra options

	barang_id	barang.kode	kategori_id	barang.nama	harga_beli	harga_jual	created_at	updated_at
Edit Copy Delete	1	B0001	1	Salad Buah	15000	20000	NULL	NULL
Edit Copy Delete	2	B0002	1	Cheese Cake	20000	30000	NULL	NULL
Edit Copy Delete	3	B0003	2	Jus Apel	8000	10000	NULL	NULL
Edit Copy Delete	4	B0004	2	Milkshake	10000	12000	NULL	NULL
Edit Copy Delete	5	B0005	3	Kaos Polo	70000	80000	NULL	NULL
Edit Copy Delete	6	B0006	3	Kemeja Flannel	80000	150000	NULL	NULL
Edit Copy Delete	7	B0007	4	Mouse Gaming	300000	450000	NULL	NULL
Edit Copy Delete	8	B0008	4	Lampu Neon Flex	30000	50000	NULL	NULL
Edit Copy Delete	9	B0009	5	Sapu Dua Macan	10000	15000	NULL	NULL
Edit Copy Delete	10	B0010	5	Kemoceng Dua Ayam	10000	15000	NULL	NULL

● t_stok

```

KategoriSeeder.php U  BarangSeeder.php U  StokSeeder.php U
database > seeders > StokSeeder.php > PHP Inteliphense > StokSeeder.php run
use Illuminate\Support\Facades\DB;
class StokSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data=[
            ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 20],
            ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 11],
            ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 32],
            ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 45],
            ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 12],
            ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 55],
            ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 6],
            ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 72],
            ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 14],
            ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 3, 'stok_tanggal' => NOW(), 'stok_jumlah' => 9]
        ];
        DB::table('t_stok')->insert($data);
    }
}

```

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

INFO Seeding database.

PS C:\laragon\www\PWL_POS>

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Server: 127.0.0.1 > Database: pwl_pos > Table: t_stok

	stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	1	3 2024-03-08 03:40:33	20	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	2	3 2024-03-08 03:40:33	11	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	3	3 2024-03-08 03:40:33	32	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	4	3 2024-03-08 03:40:33	45	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	5	3 2024-03-08 03:40:33	12	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	6	3 2024-03-08 03:40:33	55	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	7	3 2024-03-08 03:40:33	6	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	8	3 2024-03-08 03:40:33	72	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	9	9	3 2024-03-08 03:40:33	14	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	10	10	3 2024-03-08 03:40:33	9	NULL	NULL

● t_penjualan

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

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

PS C:\laragon\www\PWL_POS>

Database > seeders > **PenjualanSeeder.php** U

```

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
10 class PenjualanSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14     */
15     public function run(): void
16     {
17         $data=[
18             ['penjualan_id' => 1, 'user_id' => 3, 'pembeli' => 'Daffa Yudisa A', 'penjualan_kode' => 'PJ0001', 'penjualan_tanggal' => NOW()],
19             ['penjualan_id' => 2, 'user_id' => 3, 'pembeli' => 'Jihan Karunia P', 'penjualan_kode' => 'PJ0002', 'penjualan_tanggal' => NOW()],
20             ['penjualan_id' => 3, 'user_id' => 3, 'pembeli' => 'Bagus Indrawa', 'penjualan_kode' => 'PJ0003', 'penjualan_tanggal' => NOW()],
21             ['penjualan_id' => 4, 'user_id' => 3, 'pembeli' => 'Afrizal Dwi S', 'penjualan_kode' => 'PJ0004', 'penjualan_tanggal' => NOW()],
22             ['penjualan_id' => 5, 'user_id' => 3, 'pembeli' => 'Fedi Riansyah K', 'penjualan_kode' => 'PJ0005', 'penjualan_tanggal' => NOW()],
23             ['penjualan_id' => 6, 'user_id' => 3, 'pembeli' => 'Thoriq Fathurozzi', 'penjualan_kode' => 'PJ0006', 'penjualan_tanggal' => NOW()],
24             ['penjualan_id' => 7, 'user_id' => 3, 'pembeli' => 'Avicenna Mumtaza', 'penjualan_kode' => 'PJ0007', 'penjualan_tanggal' => NOW()],
25             ['penjualan_id' => 8, 'user_id' => 3, 'pembeli' => 'Ana Bellatus M', 'penjualan_kode' => 'PJ0008', 'penjualan_tanggal' => NOW()],
26             ['penjualan_id' => 9, 'user_id' => 3, 'pembeli' => 'Dea Putri N', 'penjualan_kode' => 'PJ0009', 'penjualan_tanggal' => NOW()],
27             ['penjualan_id' => 10, 'user_id' => 3, 'pembeli' => 'Denny Malik Ibrahim', 'penjualan_kode' => 'PJ0010', 'penjualan_tanggal' => NOW()]
28         ];
29         DB::table('t_penjualan')->insert($data);
30     }
31 }

```

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

INFO Seeding database.

PS C:\laragon\www\PWL_POS>

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Server: 127.0.0.1 > Database: pwl_pos > Table: t_penjualan

	Browse	Structure	SQL	Search	Insert	Export	Import	Privileges	Operations	Triggers
Showing rows 0 - 9 (10 total, Query took 0.0003 seconds.)										
SELECT * FROM `t_penjualan`										
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL]										
<input type="checkbox"/> Show all Number of rows: 100		Filter rows: <input type="text" value="Search this table"/>		Sort by key: <input type="text" value="None"/>						
<input type="checkbox"/> Extra options										
	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at			
<input type="checkbox"/>	Edit	Copy	Delete	1	3 Daffa Yudisa A	PJ0001	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	2	3 Jihan Karunia P	PJ0002	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	3	3 Bagus Indrawa	PJ0003	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	4	3 Afrizal Dwi S	PJ0004	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	5	3 Fedi Riansyah K	PJ0005	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	6	3 Thoriq Fathurozzi	PJ0006	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	7	3 Avicenna Mumtaza	PJ0007	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	8	3 Ana Bellatus M	PJ0008	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	9	3 Dea Putri N	PJ0009	2024-03-08 04:07:59	NULL	NULL	
<input type="checkbox"/>	Edit	Copy	Delete	10	3 Denny Malik Ibrahim	PJ0010	2024-03-08 04:07:59	NULL	NULL	

- **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.

PS C:\laragon\www\PWL_POS>

```

  PenjualanDetailSeeder.php •
database > seeders > PenjualanDetailSeeder.php > PHP Inteliphense > PenjualanDetailSeeder
  9  class PenjualanDetailSeeder extends Seeder
 10 {
 11     public function run(): void
 12     {
 13         $data = [
 14             ['detail_id' => 1, 'penjualan_id' => 1, 'barang_id' => 1, 'harga' => 20000, 'jumlah' => 2],
 15             ['detail_id' => 2, 'penjualan_id' => 1, 'barang_id' => 7, 'harga' => 450000, 'jumlah' => 1],
 16             ['detail_id' => 3, 'penjualan_id' => 1, 'barang_id' => 5, 'harga' => 80000, 'jumlah' => 3],
 17
 18             ['detail_id' => 4, 'penjualan_id' => 2, 'barang_id' => 2, 'harga' => 30000, 'jumlah' => 1],
 19             ['detail_id' => 5, 'penjualan_id' => 2, 'barang_id' => 3, 'harga' => 10000, 'jumlah' => 1],
 20             ['detail_id' => 6, 'penjualan_id' => 2, 'barang_id' => 8, 'harga' => 50000, 'jumlah' => 2],
 21
 22             ['detail_id' => 7, 'penjualan_id' => 3, 'barang_id' => 4, 'harga' => 12000, 'jumlah' => 1],
 23             ['detail_id' => 8, 'penjualan_id' => 3, 'barang_id' => 7, 'harga' => 450000, 'jumlah' => 1],
 24             ['detail_id' => 9, 'penjualan_id' => 3, 'barang_id' => 9, 'harga' => 15000, 'jumlah' => 2],
 25
 26             ['detail_id' => 10, 'penjualan_id' => 4, 'barang_id' => 10, 'harga' => 15000, 'jumlah' => 1],
 27             ['detail_id' => 11, 'penjualan_id' => 4, 'barang_id' => 6, 'harga' => 150000, 'jumlah' => 1],
 28             ['detail_id' => 12, 'penjualan_id' => 4, 'barang_id' => 3, 'harga' => 10000, 'jumlah' => 2],
 29
 30             ['detail_id' => 13, 'penjualan_id' => 5, 'barang_id' => 8, 'harga' => 50000, 'jumlah' => 4],
 31             ['detail_id' => 14, 'penjualan_id' => 5, 'barang_id' => 5, 'harga' => 80000, 'jumlah' => 1],
 32             ['detail_id' => 15, 'penjualan_id' => 5, 'barang_id' => 2, 'harga' => 30000, 'jumlah' => 2],
 33
 34             ['detail_id' => 16, 'penjualan_id' => 6, 'barang_id' => 5, 'harga' => 80000, 'jumlah' => 1],
 35             ['detail_id' => 17, 'penjualan_id' => 6, 'barang_id' => 7, 'harga' => 450000, 'jumlah' => 1],
 36             ['detail_id' => 18, 'penjualan_id' => 6, 'barang_id' => 1, 'harga' => 20000, 'jumlah' => 2],
 37
 38             ['detail_id' => 19, 'penjualan_id' => 7, 'barang_id' => 2, 'harga' => 30000, 'jumlah' => 1],
 39             ['detail_id' => 20, 'penjualan_id' => 7, 'barang_id' => 1, 'harga' => 20000, 'jumlah' => 3],
 40             ['detail_id' => 21, 'penjualan_id' => 7, 'barang_id' => 4, 'harga' => 12000, 'jumlah' => 3],
 41
 42             ['detail_id' => 22, 'penjualan_id' => 8, 'barang_id' => 8, 'harga' => 50000, 'jumlah' => 1],
 43             ['detail_id' => 23, 'penjualan_id' => 8, 'barang_id' => 4, 'harga' => 12000, 'jumlah' => 2],
 44             ['detail_id' => 24, 'penjualan_id' => 8, 'barang_id' => 3, 'harga' => 10000, 'jumlah' => 1],
 45
 46             ['detail_id' => 25, 'penjualan_id' => 9, 'barang_id' => 9, 'harga' => 15000, 'jumlah' => 1],
 47             ['detail_id' => 26, 'penjualan_id' => 9, 'barang_id' => 10, 'harga' => 15000, 'jumlah' => 1],
 48             ['detail_id' => 27, 'penjualan_id' => 9, 'barang_id' => 2, 'harga' => 30000, 'jumlah' => 2],
 49
 50             ['detail_id' => 28, 'penjualan_id' => 10, 'barang_id' => 10, 'harga' => 15000, 'jumlah' => 2],
 51             ['detail_id' => 29, 'penjualan_id' => 10, 'barang_id' => 9, 'harga' => 15000, 'jumlah' => 1],
 52             ['detail_id' => 30, 'penjualan_id' => 10, 'barang_id' => 8, 'harga' => 50000, 'jumlah' => 2]
 53         ];
 54     }
 55     DB::table('t_penjualan_detail')->insert($data);
 56 }

```

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

INFO Seeding database.

PS C:\laragon\www\PWL_POS>

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

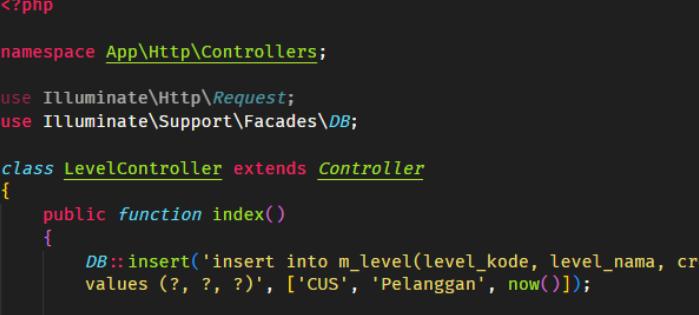
- Server:** 127.0.0.1
- Database:** pwl_pos
- Table:** t_penjualan_detail
- Rows:** 30 (total)
- Query took:** 0.00002 seconds
- Table Structure:**

	detail_id	penjualan_id	barang_id	harga	jumlah
1	1	1	1	20000	2
2	2	1	7	450000	1
3	3	1	5	80000	3
4	4	2	2	30000	1
5	5	2	3	10000	1
6	6	2	8	50000	2
7	7	3	4	12000	1
8	8	3	7	450000	1
9	9	3	9	15000	2
10	10	4	10	15000	1
11	11	4	6	150000	1
12	12	4	3	10000	2
13	13	5	8	50000	4
14	14	5	5	80000	1
15	15	5	2	30000	2
16	16	6	5	80000	1
17	17	6	7	450000	1
18	18	6	1	20000	2
19	19	7	2	30000	1
20	20	7	1	20000	3
21	21	7	4	12000	3
22	22	8	6	50000	1
23	23	8	4	12000	2
24	24	8	3	10000	1
25	25	9	9	15000	1
26	26	9	10	15000	1
27	27	9	2	30000	2
28	28	10	10	15000	2
29	29	10	9	15000	1
30	30	10	8	50000	2

9.	Laporkan hasil Praktikum-3 ini dan commit perubahan pada git
Commit	
<p>feat: Praktikum 3 (Membuat file seeder)</p> <p>main</p> <p>daffayudisaa committed 18 hours ago</p> <p>1 parent a0556e3 commit 8fcc72a</p> <p>Showing 8 changed files with 246 additions and 0 deletions.</p> <p>Whitespace Ignore whitespace Split Unified</p>	

D. Praktikum 4 – Implementasi DB Façade

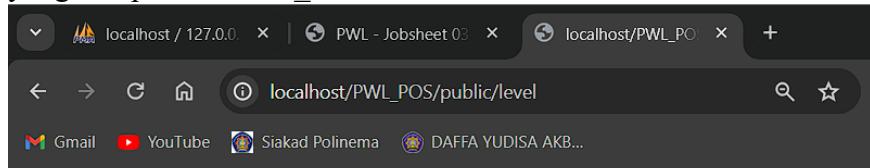
No.	Keterangan dan Gambar
1.	<p>Kita buat controller dahulu untuk mengelola data pada table m_level</p> <pre>PS C:\laragon\www\PWL_POS> php artisan make:controller LevelController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\LevelController.php] created successfully.</pre> <p>PS C:\laragon\www\PWL_POS></p>
2.	<p>Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php</p> <pre><?php use Illuminate\Support\Facades\Route; use App\Http\Controllers\LevelController; /* Web Routes Here is where you can register web routes for your application. These routes are loaded by the RouteServiceProvider and all of them will be assigned to the "web" middleware group. Make something like this: * > Route::get('/', function () { return view('welcome'); }); Route::get('/level', [LevelController::class, 'index']);</pre>
3.	Selanjutnya, kita modifikasi file LevelController untuk menambahkan 1 data ke table m_level



The screenshot shows a code editor with two tabs at the top: 'web.php M' and 'LevelController.php U'. The current file is 'LevelController.php'. The code is as follows:

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

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



Insert Data Baru Berhasil!

localhost / PWL - Job | localhost/| +

localhost/phpmyadmin/index.php?r... 🔍 ☆ 📁 | 🌐

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Server: 127.0.0.1 » Database: pwl_pos » Table: m_level

Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

```
SELECT * FROM `m_level`
```

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

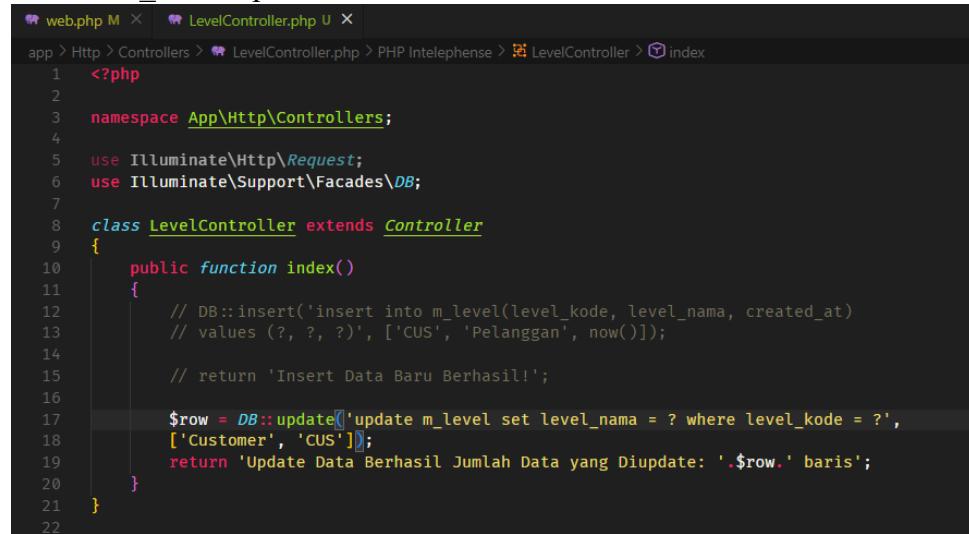
Show all | Number of rows: 25

Extra options

			level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	ADM	Administrator	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	USR	User	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	STF	Staff	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	CUS	Pelanggan	2024-03-08 07:53:05

* Jika diamati maka setelah menjalankan url localhost/PWL_POS/public/level maka akan ditambah sebuah data yang sesuai dengan pada LevelController, hal ini karena adanya fungsi insert yang berfungsi untuk memasukkan data pada database tersebut.

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

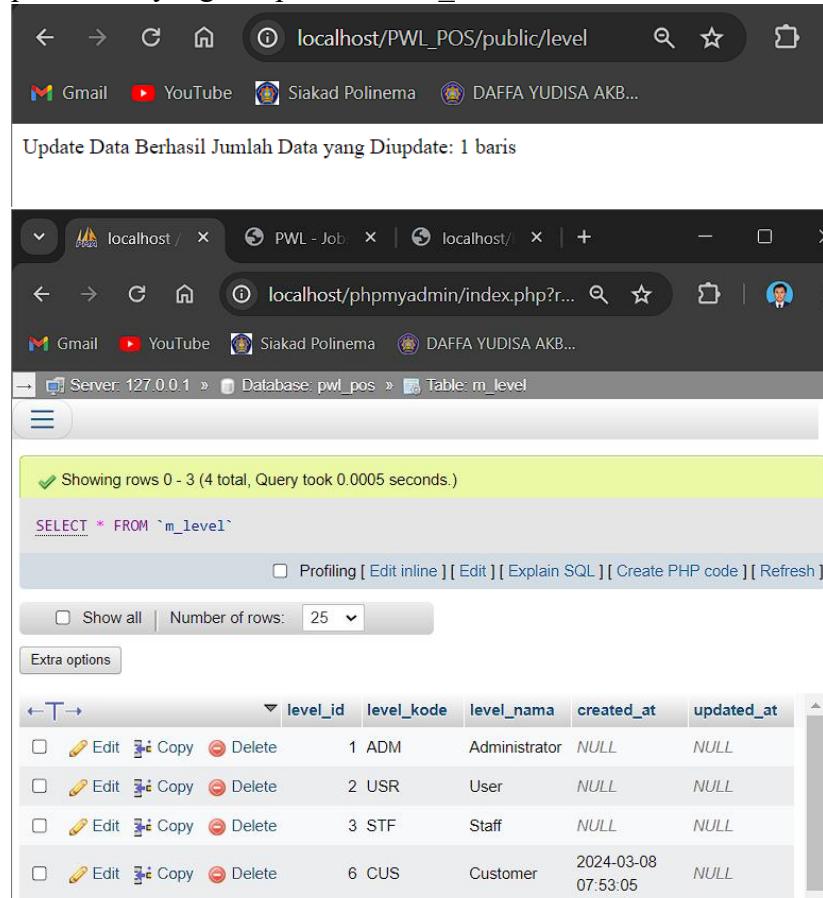


```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         // DB::insert('insert into m_level(level_kode, level_nama, created_at)
13         // values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
14
15         // return 'Insert Data Baru Berhasil!';
16
17         $row = DB::update(['update m_level set level_nama = ? where level_kode = ?'],
18                         ['Customer', 'CUS']);
19
20         return 'Update Data Berhasil Jumlah Data yang Diupdate: ' . $row . ' baris';
21     }
22 }

```

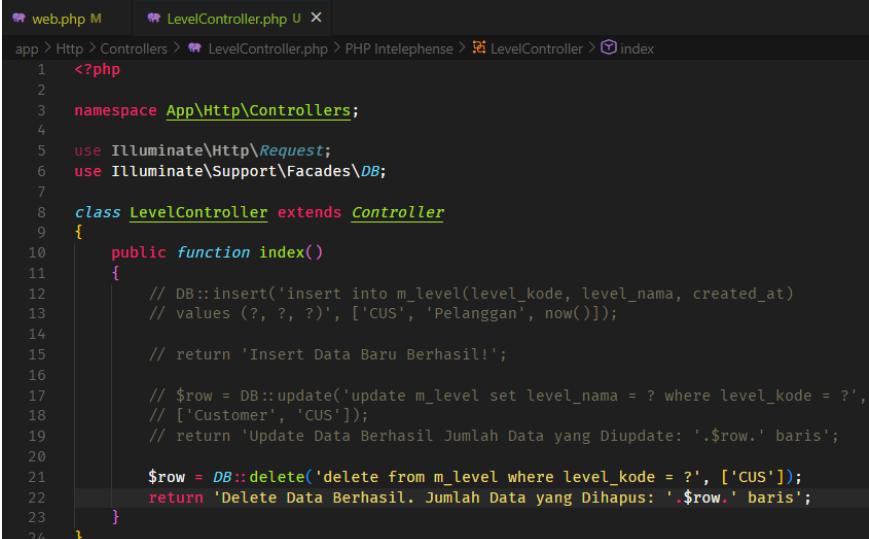
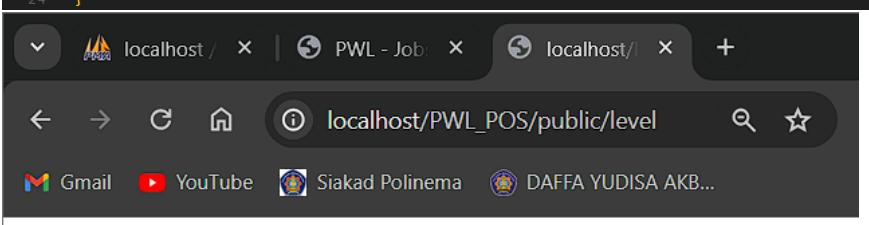
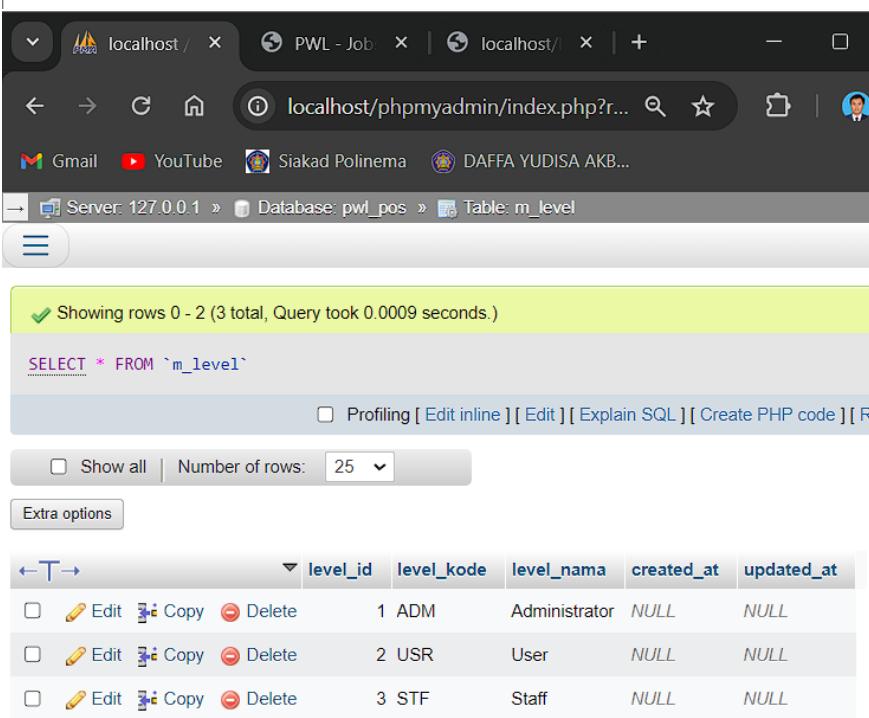
6. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level lagi dan amati apa yang terjadi pada table m_level di database, screenshot perubahan yang ada pada table m_level



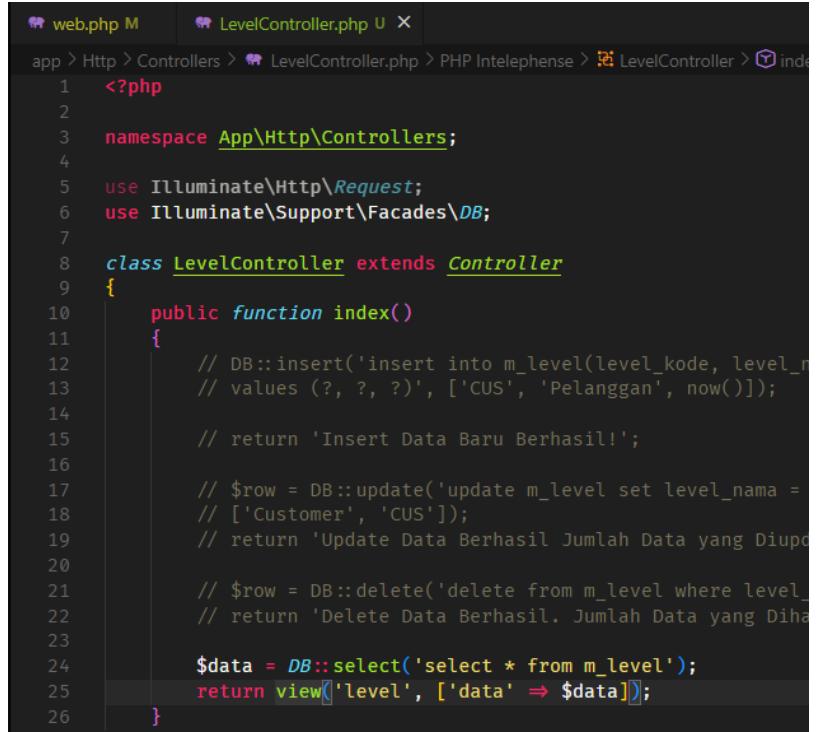
Update Data Berhasil Jumlah Data yang Diupdate: 1 baris

level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	NULL	NULL
2	USR	User	NULL	NULL
3	STF	Staff	NULL	NULL
6	CUS	Customer	2024-03-08 07:53:05	NULL

* Jika dilihat maka level nama dari level kode CUS berubah, yakni yang awalnya bernilai ‘Pelanggan’ sekarang berubah menjadi ‘Customer’. Hal ini karena adanya fungsi update yang dilakukan pada LevelController, yang akan

	<p>melakukan update dari nilai yang telah ditentukan (dimana pada contoh ini pada level_kode = CUS, level-nama nya diubah menjadi ‘Customer’).</p>
7.	<p>Kita coba modifikasi lagi file LevelController untuk melakukan proses hapus data</p>   

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

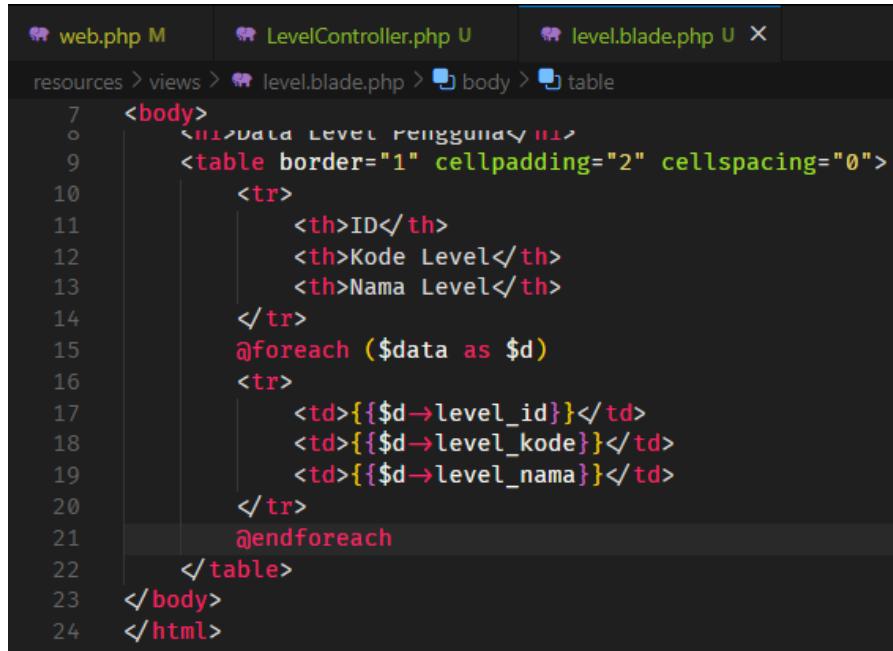


```

web.php M LevelController.php U
app > Http > Controllers > LevelController.php > PHP Intelephense > LevelController > index()
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class LevelController extends Controller
9 {
10     public function index()
11     {
12         // DB::insert('insert into m_level(level_kode, level_nama)
13         // values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
14
15         // return 'Insert Data Baru Berhasil!';
16
17         // $row = DB::update('update m_level set level_nama =
18         // ['Customer', 'CUS']);
19         // return 'Update Data Berhasil. Jumlah Data yang Diupd
20
21         // $row = DB::delete('delete from m_level where level_
22         // return 'Delete Data Berhasil. Jumlah Data yang Diha
23
24         $data = DB::select('select * from m_level');
25         return view('level', ['data' => $data]);
26     }
}

```

9. Coba kita perhatikan kode yang diberi tanda kotak merah, berhubung kode tersebut memanggil view('level'), maka kita buat file view pada VSCode di PWL_POS/resources/view/level.blade.php dan amati apa yang terjadi



```

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

```

	<p>Data Level Pengguna</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Kode Level</th> <th>Nama Level</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ADM</td> <td>Administrator</td> </tr> <tr> <td>2</td> <td>USR</td> <td>User</td> </tr> <tr> <td>3</td> <td>STF</td> <td>Staff</td> </tr> </tbody> </table>	ID	Kode Level	Nama Level	1	ADM	Administrator	2	USR	User	3	STF	Staff
ID	Kode Level	Nama Level											
1	ADM	Administrator											
2	USR	User											
3	STF	Staff											
10.	<p>*Setelah dijalankan maka akan ditampilkan berupa view seluruh data level pengguna karena pada LevelController ada return view level yang mengambil \$data, dimana \$data tersebut berisikan query yang akan menampilkan seluruh data pada tabel m_level (select * from m_level).</p> <p>Laporkan hasil Praktikum-4 ini dan commit perubahan pada git.</p> <p>Commit</p> <pre>feat: Praktikum 4 (Implementasi DB Facade) main daffayudisaa committed 18 hours ago 1 parent 02f6503 commit 5fb37d6</pre> <p>Showing 3 changed files with 55 additions and 0 deletions.</p> <p>Whitespace Ignore whitespace Split Unified</p>												

E. Praktikum 5 – Implementasi Query Builder

No.	Keterangan dan Gambar
1.	<p>Kita buat controller dahulu untuk mengelola data pada table m_kategori</p> <pre>PS C:\laragon\www\PWL_POS> php artisan make:controller KategoriController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\KategoriController.php] created successfull y. PS C:\laragon\www\PWL_POS></pre>
2.	Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php

```

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

```

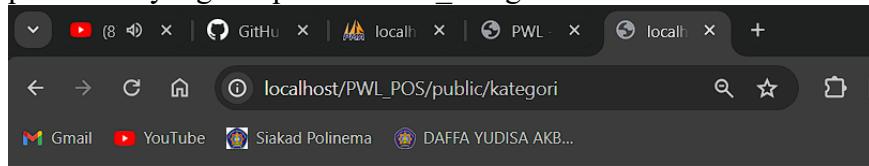
3. Selanjutnya, kita modifikasi file KategoriController untuk menambahkan 1 data ke table m_kategori

```

app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class KategoriController extends Controller
9  {
10     public function index()
11     {
12         $data=[
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack / Makanan Ringan',
15             'created_at' => now()
16         ];
17         DB::table('m_kategori')->insert($data);
18         return 'Insert Data Baru Berhasil!';
19     }
20 }
21

```

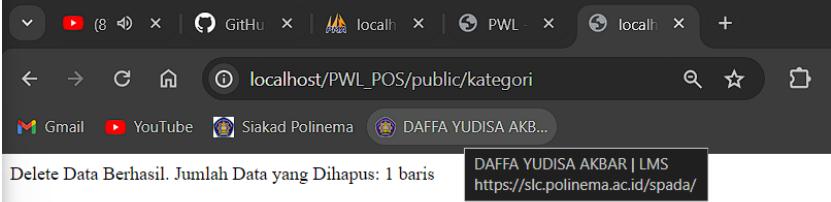
4. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori dan amati apa yang terjadi pada table m_kategori di database, screenshot perubahan yang ada pada table m_kategori

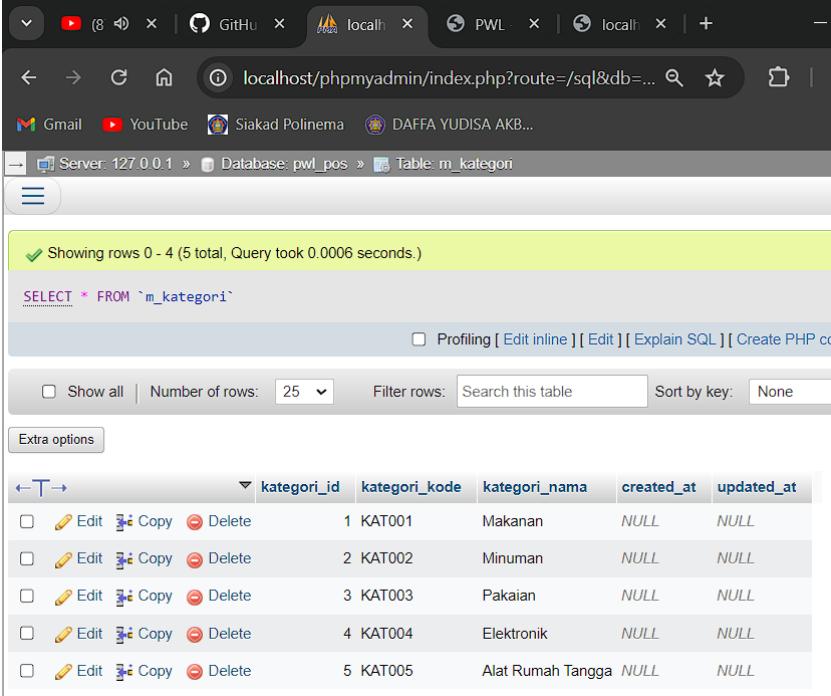


Insert Data Baru Berhasil!

	<p>* setelah URL dijalankan maka akan ditambahkan baris baru yang berisi mengenai suatu kategori yakni Snack. Hal ini bisa dilakukan karena adanya query builder insert yang dilakukan pada KategoriController untuk menambahkan data pada tabel m_kategori database ketika URL dijalankan.</p>
5.	<p>Selanjutnya, kita modifikasi lagi file KategoriController untuk meng-update data di table m_kategori seperti berikut</p> <pre> web.php M KategoriController.php U app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > index 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use Illuminate\Support\Facades\DB; 7 8 class KategoriController extends Controller 9 { 10 public function index() 11 { 12 // \$data=[13 // 'kategori.kode' => 'SNK', 14 // 'kategori.nama' => 'Snack / Makanan Ringan', 15 // 'created_at' => now() 16 //]; 17 // DB::table('m_kategori')->insert(\$data); 18 // return 'Insert Data Baru Berhasil!'; 19 20 \$row = DB::table('m_kategori')->where('kategori.kode', 'SNK')->update(['kategori.nama' => 'Camilan']); 21 return 'Update Data Berhasil. Jumlah Data yang Diupdate: ' . \$row . ' baris'; 22 } 23 } </pre>
6.	<p>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</p> <p>Update Data Berhasil. Jumlah Data yang Diupdate: 1 baris</p>

	<pre> SELECT * FROM `m_kategori`</pre> <p>* Pada bagian ini dilakukan update suatu data yang ada pada tabel m_kategori, pada update data ini dilakukan menggunakan query builder yang ada pada KategoriController, dimana kategori_nama yang awalnya bernilai 'Snack/Makanan RIngan' diupdate nilainya berupa 'Camilan'.</p>
7.	<p>Kita coba modifikasi lagi file KategoriController untuk melakukan proses hapus data</p> <pre> 1 > 2 3 <space> App\HTTP\Controllers; 4 5 Illuminate\HTTP\Request; 6 Illuminate\Support\Facades\DB; 7 8 <space> 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([' 21 // return 'Update Data Berhasil. Jumlah Data yang Diupdate: '.\$row. ' baris'; 22 23 \$row = DB::table('m_kategori')->where('kategori.kode', 'SNK')->delete(); 24 return 'Delete Data Berhasil. Jumlah Data yang Dihapus: '.\$row. ' baris'; 25] 26</pre>





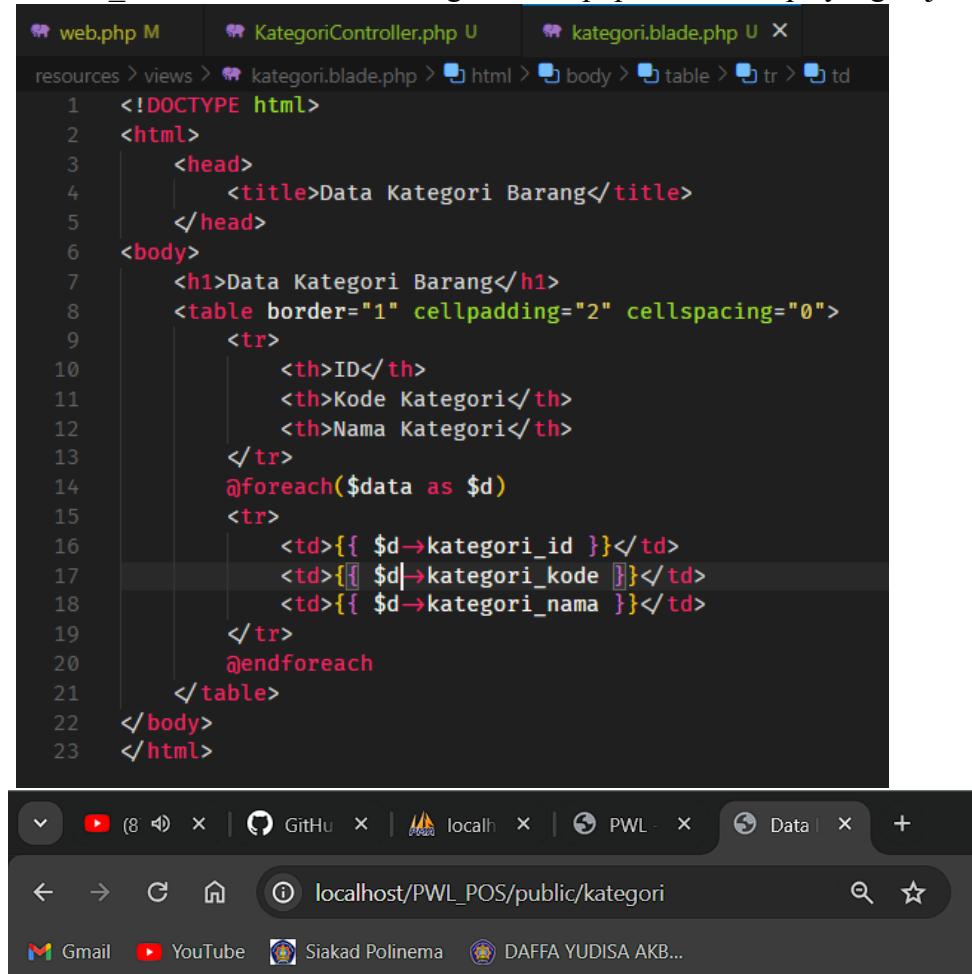
8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m_kategori. Kita modifikasi file KategoriController seperti berikut

```

web.php M KategoriController.php U kategori.blade.php U
app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController
3   namespace App\Http\Controllers;
4
5   use Illuminate\Http\Request;
6   use Illuminate\Support\Facades\DB;
7
8   class KategoriController extends Controller
9   {
10     public function index()
11     {
12         // $data=[
13         //     'kategori_kode' => 'SNK',
14         //     'kategori_nama' => 'Snack / Makanan Ringan'
15         //     'created_at' => now()
16         // ];
17         // DB::table('m_kategori')->insert($data);
18         // return 'Insert Data Baru Berhasil!';
19
20         // $row = DB::table('m_kategori')->where('kategori');
21         // return 'Update Data Berhasil. Jumlah Data yang';
22
23         // $row = DB::table('m_kategori')->where('kategori');
24         // return 'Delete Data Berhasil. Jumlah Data yang';
25
26         $data = DB::table('m_kategori')->get();
27         return view('kategori', ['data' => $data]);
28     }
29 }
30

```

9. Coba kita perhatikan kode yang diberi tanda kotak merah, berhubung kode tersebut memanggil view('kategori'), maka kita buat file view pada VSCode di PWL_POS/resources/view/kategori.blade.php dan amati apa yang terjadi



```

web.php M KategoriController.php U kategori.blade.php U X
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>

```

The screenshot shows a code editor with three tabs: web.php, KategoriController.php, and kategori.blade.php. The kategori.blade.php tab is active. The code is a Blade template for displaying category data. Red boxes highlight several parts of the code: the entire template structure from <!DOCTYPE html> to </html>, the table structure from <table> to </table>, and specific sections of the table rows like <td>{{ \$d->kategori_id }}</td>. Below the code editor is a browser window showing the URL localhost/PWL_POS/public/kategori. The browser title bar also lists other open tabs: YouTube, GitHub, localh, PWL, Data, Gmail, YouTube, Siakad Polinema, and DAFFA YUDISA AKB...

Data Kategori Barang

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

* Setelah URL dijalankan, maka akan ditampilkan tabel m_kategori yang tampilannya disesuaikan pada view, dimana pada tabel ini akan menampilkan seluruh data yang ada pada tabel m_kategori. Hal ini dikarenakan pada kategori controller terdapat kode program \$data = DB::table('m_kategori')->get() yang memiliki arti akan mengambil seluruh nilai pada tabel m_kategori.

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

Commit

feat: Praktikum 5 (Implementasi Query Builder)

main

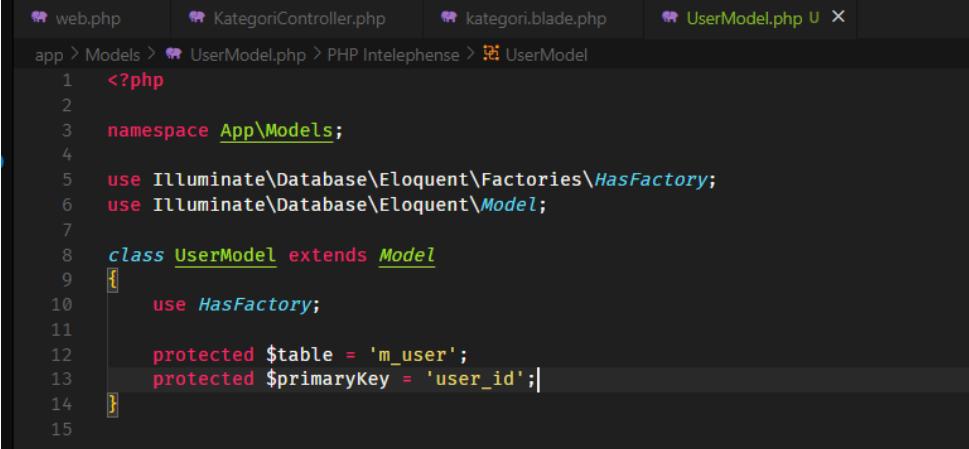
daffayudisaa committed 11 hours ago

1 parent 5fb37d6 commit d297c43

Showing 3 changed files with 54 additions and 0 deletions.

[Whitespace](#) [Ignore whitespace](#) [Split](#) [Unified](#)

F. Praktikum 6 – Implementasi Eloquent ORM

No .	Keterangan dan Gambar
1.	<p>Kita buat file model untuk tabel m_user dengan mengetikkan perintah</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>
2.	<p>Kita buka file UserModel.php dan modifikasi seperti berikut</p>  <pre> 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 } 15 </pre>
3.	<p>Kita modifikasi route web.php untuk mencoba routing ke controller UserController</p> <p>INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\UserController.php] created successfully.</p>

```

web.php M UserController.php U UserModel.php U
routes > web.php > ...
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\LevelController;
5 use App\Http\Controllers\KategoriController;
6 use App\Http\Controllers\UserController;
7
8 /*
9 | Web Routes
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "web" middleware group. Make something great!
14 |
15 */
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
21 Route::get('/level', [LevelController::class, 'index']);
22 Route::get('/kategori', [KategoriController::class, 'index']);
23 Route::get('/user', [UserController::class, 'index']);
24
25
26
27

```

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

```

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

```

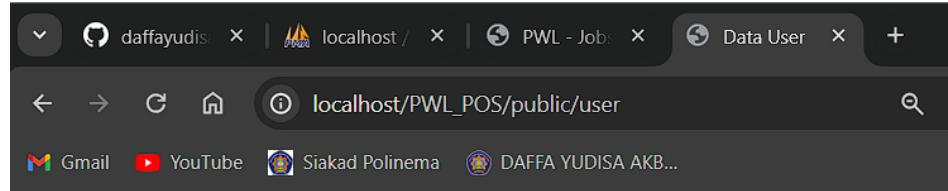
5. Kemudian kita buat view user.blade.php

```

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

```

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



Data User

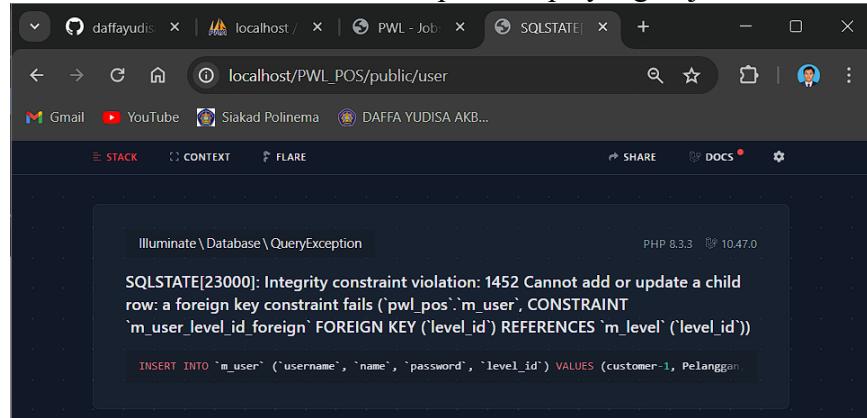
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3

* Setelah dijalankan maka akan muncul output seperti gambar di atas, yakni seluruh data dari user akan ditampilkan dalam bentuk tabel yang telah di atur pada view. Dalam praktikum ini dilakukan pendefinisian tabel dan primary key yang akan digunakan oleh model yang telah kita buat, yang kemudian pada UserController kita akan mengakses UserModel dan mengambil semua data dari tabel m_user yang telah kita definisikan di awal, dan menampilkannya pada view yang telah ditentukan.

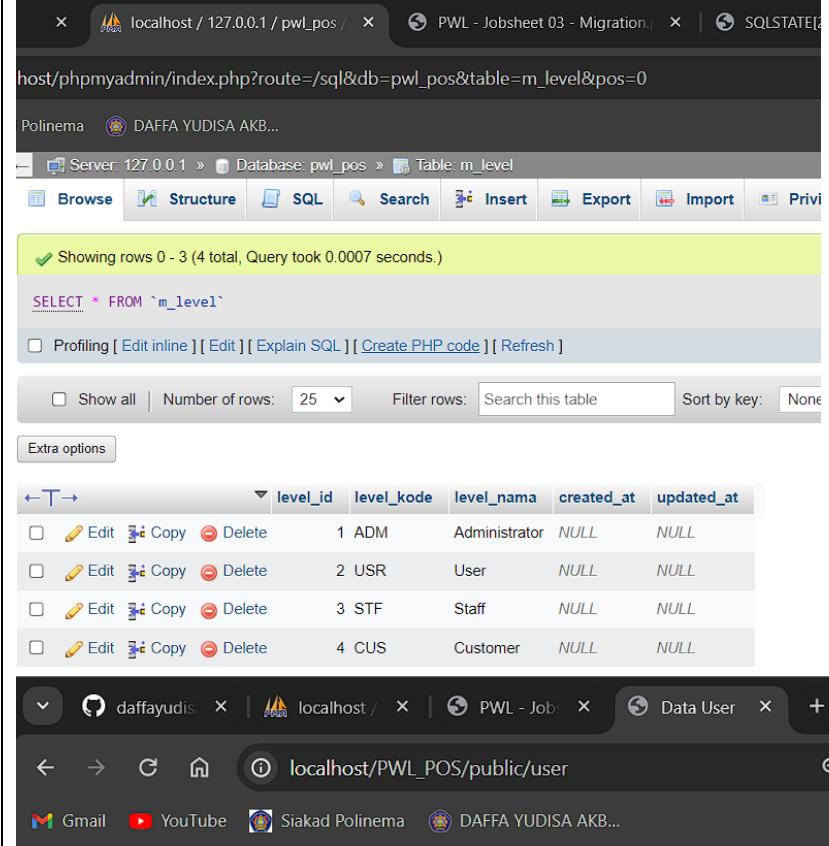
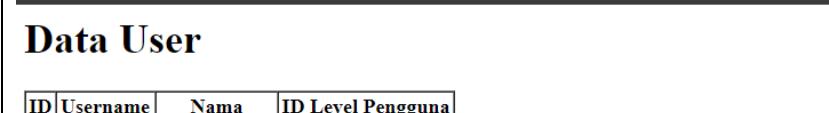
7. Setelah itu, kita modifikasi lagi file UserController

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

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



*Maka akan terjadi error, karena sistem tidak bisa menemukan level_id = 4. Agar sistem bisa dijalankan dengan baik maka perlu diinsertkan data level_id =4.

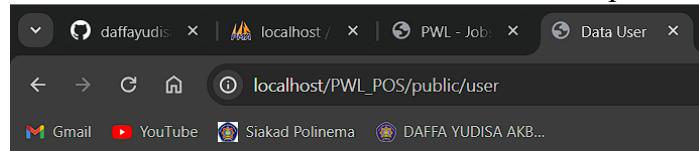
	 <p>The screenshot shows the PHPMyAdmin interface for the 'm_level' table. The table has columns: level_id, level_kode, level_nama, created_at, and updated_at. The data is as follows:</p> <table border="1"> <thead> <tr> <th>level_id</th> <th>level_kode</th> <th>level_nama</th> <th>created_at</th> <th>updated_at</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ADM</td> <td>Administrator</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>USR</td> <td>User</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>STF</td> <td>Staff</td> <td>NULL</td> <td>NULL</td> </tr> <tr> <td>4</td> <td>CUS</td> <td>Customer</td> <td>NULL</td> <td>NULL</td> </tr> </tbody> </table>  <p>The screenshot shows a web browser displaying a table of user data. The columns are: ID, Username, Nama, and ID Level Pengguna. The data is as follows:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Username</th> <th>Nama</th> <th>ID Level Pengguna</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>admin</td> <td>Administrator</td> <td>1</td> </tr> <tr> <td>2</td> <td>manager</td> <td>Manager</td> <td>2</td> </tr> <tr> <td>3</td> <td>staff</td> <td>Staff/Kasir</td> <td>3</td> </tr> <tr> <td>11</td> <td>customer-1</td> <td>Pelanggan</td> <td>4</td> </tr> </tbody> </table>	level_id	level_kode	level_nama	created_at	updated_at	1	ADM	Administrator	NULL	NULL	2	USR	User	NULL	NULL	3	STF	Staff	NULL	NULL	4	CUS	Customer	NULL	NULL	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	2	manager	Manager	2	3	staff	Staff/Kasir	3	11	customer-1	Pelanggan	4
level_id	level_kode	level_nama	created_at	updated_at																																										
1	ADM	Administrator	NULL	NULL																																										
2	USR	User	NULL	NULL																																										
3	STF	Staff	NULL	NULL																																										
4	CUS	Customer	NULL	NULL																																										
ID	Username	Nama	ID Level Pengguna																																											
1	admin	Administrator	1																																											
2	manager	Manager	2																																											
3	staff	Staff/Kasir	3																																											
11	customer-1	Pelanggan	4																																											
9.	<p>*Setelah ditambah data dengan level_id = 4, maka sistem akan menampilkan secara normal data yang ada pada tabel m_user. Dimana pada no. ini dilakukan penambahan data yang awalnya user sebanyak 3 sekarang bertambah menjadi 4, penambahan tersebut dilakukan diUserController dengan username = customer-1, nama = pelanggan, password = 12345 (hash), dan level_id = 4 menggunakan Eloquent Model.</p> <p>Kita modifikasi lagi file UserController menjadi seperti berikut</p>																																													

```

web.php M UserController.php U user.blade.php U UserModel.php U
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         // $data = [
14         //     'username' => 'customer-1',
15         //     'name' => 'Pelanggan',
16         //     'password' => Hash::make('12345'),
17         //     'level_id' => 4
18         // ];
19         // UserModel::insert($data);
20
21         $data = [
22             'name' => 'Pelanggan Pertama',
23         ];
24         UserModel::where('username', 'customer-1')->update($data);
25
26         $user = UserModel::all();
27         return view('user', ['data' => $user]);
28     }
29 }

```

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



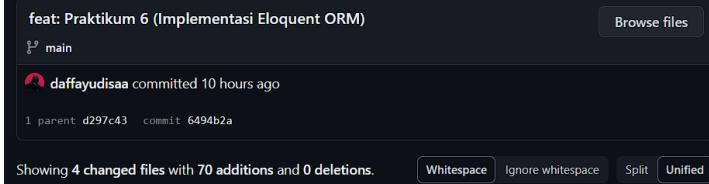
Data User

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

*Pada no. ini dilakukan update data pada data yang memiliki username = customer-1. Update yang dilakukan yakni name yang awalnya bernilai ‘Pelanggan’ diupdate menjadi ‘Pelanggan Pertama’ menggunakan Eloquent Model pada UserController.

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

Commit



G. Penutup

1. Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada file setting .env Laravel?

Jawab:

Fungsi utama dari APP_KEY adalah pada bidang keamanan, laravel menggunakan APP_KEY untuk mengenkripsi data sensitif seperti password user atau kunci API sebelum menyimpannya pada database atau mengirimkannya melalui jaringan. Dengan begini akan memastikan jika seseorang mendapatkan akses menuju database atau mengintercepts lalu lintas jaringan, mereka tidak akan bisa untuk membaca data sensitif tersebut tanpa menggunakan APP_KEY.

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

Jawab:

Untuk men-generate nilai APP_KEY, kita perlu memasukkan perintah php artisan key:generate pada terminal. Maka dengan melakukan perintah tersebut akan tercipta string 32 karakter baru dan akan memperbarui APP_KEY pada file .env.

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

Jawab:

Ada 4 file migrasi default Laravel yakni :

- a. 2014_10_12_000000_create_users_table.php, bertujuan untuk membuat tabel user yang digunakan untuk menyimpan informasi user seperti nama, alamat email, dan kata sandi.
 - b. 2014_10_12_100000_create_password_resets_table.php, bertujuan untuk membuat tabel password_resets yang digunakan untuk menyimpan token reset kata sandi dan informasi terkait pengguna. Tabel ini berguna dalam proses reset kata sandi pengguna.
 - c. 2019_08_19_000000_create_failed_jobs_table.php. bertujuan untuk membuat tabel failed_jobs yang digunakan untuk menyimpan informasi tentang pekerjaan yang gagal dalam antrian pekerjaan Laravel. Tabel ini membantu dalam penanganan pekerjaan yang gagal dan pemantauan antrian pekerjaan.
 - d. 2019_12_14_000001_create_personal_access_tokens_table.php, bertujuan untuk membuat tabel personal_access_tokens yang digunakan untuk menyimpan token akses pribadi yang digunakan untuk otentikasi API pengguna. Tabel ini digunakan saat menggunakan Laravel Passport atau autentikasi API lainnya untuk menyediakan otentikasi token.
4. Secara default, file migrasi terdapat kode \$table->timestamps();, apa tujuan/output dari fungsi tersebut?

Jawab:

Tujuan dari kode tersebut adalah untuk menambahkan dua kolom pada suatu tabel secara otomatis, yakni kolom created_at dan updated_at. Dimana kolom created_at akan menyimpan waktu ketika sebuah record dalam tabel pertama kali ditambahkan dan kolom updated_ad akan menyimpan waktu ketika sebuah record dalam tabel diperbarui.

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

Jawab:

Tipe data yang dihasilkan dari fungsi tersebut adalah bigint (big integer) unsigned yang secara otomatis juga akan menjadi primary key untuk tabel tersebut dan nilainya secara otomatis akan bertambah setiap kali sebuah baris baru ditambahkan ke dalam tabel (auto increment).

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

Jawab:

Perbedaannya adalah dalam penamaan, jika menggunakan `$table->id();` secara default nama kolomnya akan menjadi 'id', sedangkan `$table->id('level_id');` nama kolom akan menjadi 'level_id'. Keduanya memiliki tipe data yang sama yakni bigint unsigned dan auto increment serta menjadi primary key pada tabel.

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

Jawab:

Fungsi tersebut digunakan untuk menetapkan keunikan pada kolom tertentu dalam tabel database. Ketika menambahkan `->unique()` ke kolom dalam file migrasi, kita memberi tahu Laravel bahwa setiap nilai dalam kolom tersebut harus unik di antara semua baris dalam tabel. Dengan begitu jika kita akan menyimpan data yang sama akan eror (untuk mencegah adanya duplikasi).

8. Pada Praktikum 2.2 - Tahap 2, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$table->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$table->id('level_id')` ?

Jawab:

Pada tabel `m_user`, kolom `level_id` didefinisikan menggunakan `unsignedBigInteger ('level_id')`, hal ini menunjukkan bahwa kolom tersebut adalah foreign key yang merujuk pada kolom primary key tabel lain serta penggunaan `unsignedBigInteger()` mengindikasikan bahwa kolom `level_id` memiliki tipe data bigint unsigned, yang umumnya digunakan foreign key. Di sisi lain, dalam tabel `m_level`, kolom `level_id` didefinisikan menggunakan `id('level_id')`, yang menunjukkan bahwa kolom ini adalah primary key (kunci utama) dari tabel tersebut.

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

Jawab:

Class Hash bertujuan untuk meningkatkan keamanan data dalam aplikasi. Dengan menyediakan metode untuk mengenkripsi dan memverifikasi data, terutama kata sandi pengguna, Hash membantu melindungi informasi sensitif dari akses yang tidak sah. Maksud dari `Hash::make('1234');` adalah mengenkripsi kata sandi '12345' menggunakan algoritma hash yang telah ditentukan. Jadi, `Hash::make('12345')` akan menghasilkan hash dari kata sandi '12345', yang kemudian akan disimpan ke dalam basis data.

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

Jawab:

Kegunaan tanda tanya (?) yang ada dalam query builder tersebut berperan sebagai parameter. Dalam konteks penggunaan ini, tanda tanya (?) menunjukkan tempat di mana nilai akan disisipkan ke dalam query.

Contoh: DB::insert('insert into m_level(level_kode, level_nama, created_at) values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);

Nilai 'CUS', 'Pelanggan', dan now() akan digunakan untuk mengisi tanda tanya (?) sesuai urutan yang ditentukan dalam query. Misalnya, 'CUS' akan menggantikan tanda tanya pertama, 'Pelanggan' akan menggantikan tanda tanya kedua, dan now() akan menggantikan tanda tanya ketiga.

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

Jawab:

Penulisan kode tersebut bertujuan dalam menentukan tabel mana yang akan digunakan oleh model serta kolom mana yang menjadi primary key dari tabel tersebut. Dari kode program tersebut maka tabel yang digunakan adalah m_user dan primary keynya adalah user_id.

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

Jawab:

Menurut tingkat pemahaman saya, lebih mudah menggunakan Query Builder. Hal tersebut dikarenakan query builder memungkinkan penulisan query dengan sintaks yang mudah dipahami dan disesuaikan dengan kebutuhan serta pada query builder perintah SQL ini diakses menggunakan method. Jadi, kita tidak menulis perintah SQL secara langsung. melainkan cukup memanggil method-method yang ada di query builder. Selain itu query builder bisa digunakan untuk mengakses berbagai jenis database seperti MySQL, MariaDB, PostgreSQL, SQL Server, dan lain-lain. Bukan hanya itu salah satu keunggulan utama Query Builder adalah kemampuannya untuk menggunakan chaining method. Dimana pengguna dapat dengan mudah menambahkan kondisi tambahan, pengurutan, dan limit pada query dengan menambahkan metode ke dalam rangkaian.