

PRAKTIKUM PEMROGRAMAN WEB LANJUT

TUGAS PERTEMUAN 3

Migration

DISUSUN OLEH :

GILANG PURNOMO

NIM:2341720042



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

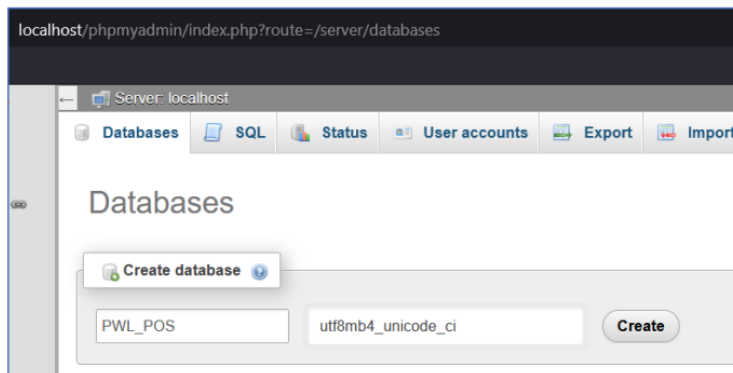
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

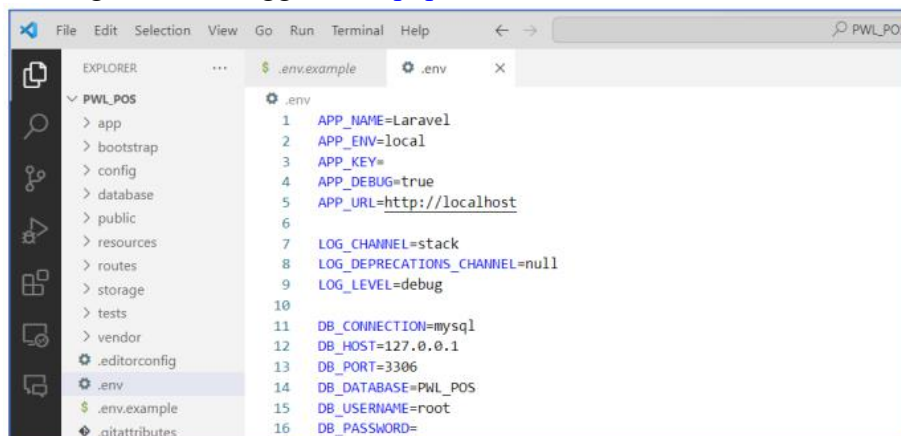
FEBRUARI 2025

Praktikum 1 - pengaturan database:

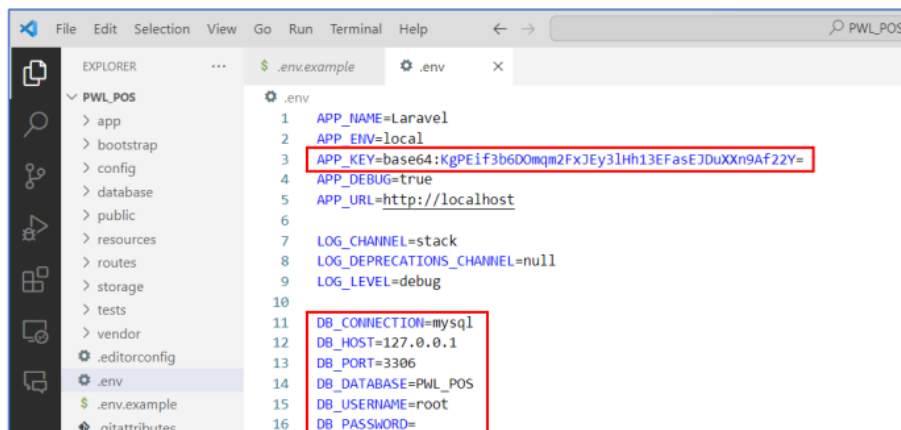
1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama **PWL_POS**



2. Buka aplikasi VSCode dan buka folder project **PWL_POS** yang sudah kita buat.
3. Copy file **.env.example** menjadi **.env**
4. Buka file **.env**, dan pastikan konfigurasi **APP_KEY** bernilai. Jika belum bernilai silahkan kalian generate menggunakan **php artisan**.



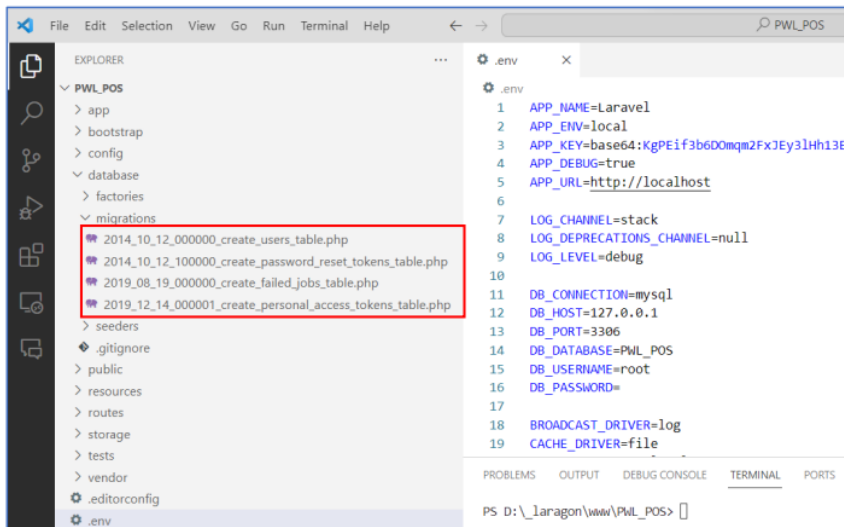
5. Edit file **.env** dan sesuaikan dengan database yang telah dibuat



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

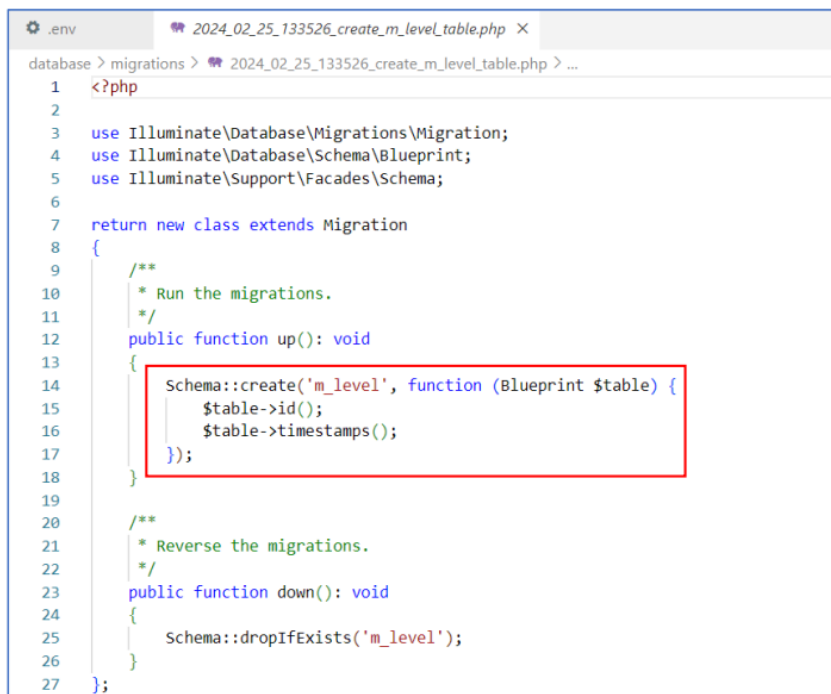
Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

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



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

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



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

```

7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('m_level', function (Blueprint $table) {
15             $table->id('level_id');
16             $table->string('level_kode', 10)->unique();
17             $table->string('level_nama', 100);
18             $table->timestamps();
19         });
20     }
21
22     /**
23     * Reverse the migrations.
24     */
25     public function down(): void
26     {
27         Schema::dropIfExists('m_level');
28     }
29 };

```

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

```

php artisan migrate

```

```

PS D:\laragon\www\PMW_POS> php artisan migrate

[INFO] Preparing database.

Creating migration table ..... 12ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 16ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 6ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 42ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 15ms DONE
2024_02_25_133526_create_m_level_table ..... 13ms DONE
PS D:\laragon\www\PMW_POS>

```

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

Table	Action	Rows
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	5
<input type="checkbox"/> m_level	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> password_reset_tokens	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0

7. Ok, table sudah dibuat di database
8. Buat table database dengan migration untuk table **m_kategori** yang sama-sama tidak memiliki foreign key

Jawab :

```

PS C:\laragon\www\Web_Lanjut\Winggu 3\PMW_POS> php artisan make:migration create_m_kategori_table --create=m_kategori

[INFO] Migration [C:\laragon\www\Web_Lanjut\Winggu 3\PMW_POS\database\Migrations\2025_02_24_100241_create_m_kategori_table.php] created successfully.

```

```

return new class extends Migration
{
    Codeium: Refactor | Explain | Generate Function Comment | X
    public function up(): void
    {
        Schema::create('m_kategori', function (Blueprint $table) {
            $table->id('kategori_id');
            $table->string('kategori_kode', 10)->unique();
            $table->string('kategori_nama', 100);
            $table->timestamps();
        });
    }

    Codeium: Refactor | Explain | Generate Function Comment | X
    public function down(): void
    {
        Schema::dropIfExists('m_kategori');
    }
};

```

PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan migrate

INFO Running migrations.

2025_02_24_100241_create_m_kategori_table 253ms DONE

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> m_kategori	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> m_level	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> password_reset_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-

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

Jawab :

Minggu 3/PWL_POS

Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

6 minutes ago

Praktikum 2.2 - Pembuatan file migrasi dengan relasi

1. Buka terminal VSCode kalian, dan buat file migrasi untuk table **m_user**

```
php artisan make:migration create_m_user_table --table=m_user
```

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

```

7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('m_user', function (Blueprint $table) {
15               $table->id('user_id');
16               $table->unsignedBigInteger('level_id')->index(); // indexing untuk ForeignKey
17               $table->string('username', 20)->unique(); // unique untuk memastikan tidak ada username yang sama
18               $table->string('nama', 100);
19               $table->string('password');
20               $table->timestamps();
21
22               // Mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
23               $table->foreign('level_id')->references('level_id')->on('m_level');
24           });
25       }
26
27       /**
28       * Reverse the migrations.
29       */
30       public function down(): void
31       {
32           Schema::dropIfExists('m_user');
33       }
34 };

```

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

Jawab :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan migrate
INFO: Running migrations.
2025_02_24_103755_create_m_user_table ..... 403ms DONE
```

pwl_pos.m_kategori	
kategori_id	bigint unsigned
kategori_kode	varchar(10)
kategori_nama	varchar(100)
created_at	timestamp
updated_at	timestamp

pwl_pos.m_level	
level_id	bigint unsigned
level_kode	varchar(10)
level_nama	varchar(100)
created_at	timestamp
updated_at	timestamp

pwl_pos.m_user	
user_id	bigint unsigned
level_id	bigint unsigned
username	varchar(20)
nama	varchar(100)
password	varchar(255)
created_at	timestamp
updated_at	timestamp

Laravel mengeksekusi file migrasi yang belum dijalankan dan memastikan/membuat struktur database sesuai dengan yang ada dalam kode.

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

m_barang
t_penjualan
t_stok
t_penjualan_detail

Jawab :

Tabel m_barang

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:migration create_m_barang_table --create=m_barang
INFO: Migration [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\migrations\2025_02_24_135949_create_m_barang_table.php] created successfully.
```

```
public function up(): void
{
    Schema::create('m_barang', function (Blueprint $table) {
        $table->id('barang_id');
        $table->unsignedBigInteger('kategori_id')->index();
        $table->string('barang_kode', 10)->unique();
        $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');
    });
}
```

Tabel t_penjualan

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:migration create_t_penjualan_table --create=t_penjualan
INFO Migration [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\Migrations\2025_02_24_140024_create_t_penjualan_table.php] created successfully.
```

```
public function up(): void
{
    Schema::create('t_penjualan', function (Blueprint $table) {
        $table->id('penjualan_id');
        $table->unsignedBigInteger('user_id')->index();
        $table->string('pembeli', 50)->unique();
        $table->string('penjualan_kode', 20);
        $table->dateTime('tanggal_penjualan');
        $table->timestamps();

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

Tabel t_stok

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:migration create_t_stok_table --create=t_stok
INFO Migration [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\Migrations\2025_02_24_140129_create_t_stok_table.php] created successfully.
```

```
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();

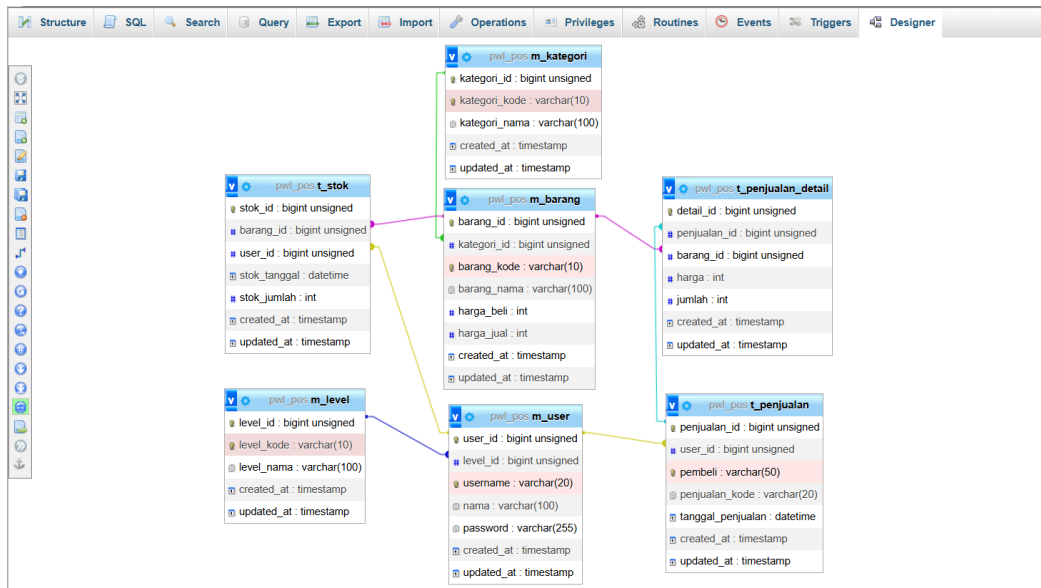
        $table->foreign('barang_id')->references('barang_id')->on('m_barang');
        $table->foreign('user_id')->references('user_id')->on('m_user');
    });
}
```

Tabel t_penjualan_detail

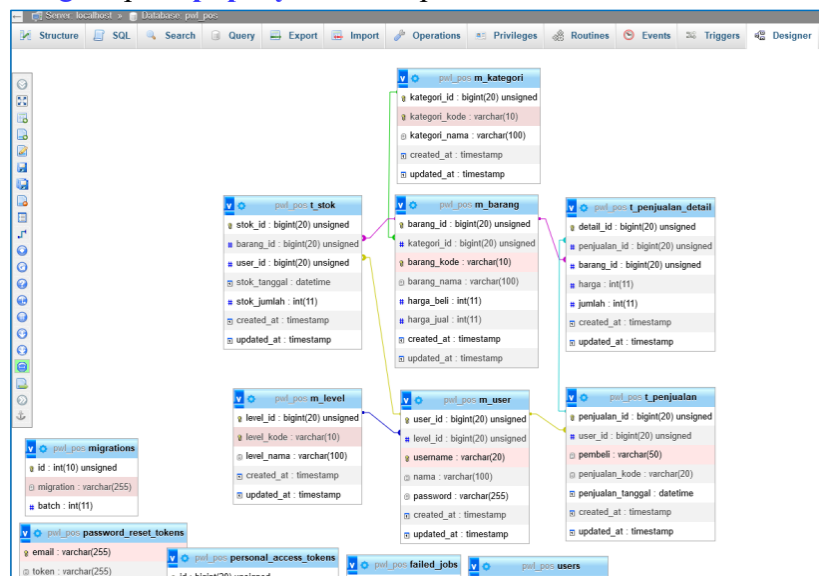
```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:migration create_t_penjualan_detail_table --create=t_penjualan_detail
INFO Migration [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\Migrations\2025_02_24_140221_create_t_penjualan_detail_table.php] created successfully.
```

```
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->timestamps();

        $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan');
        $table->foreign('barang_id')->references('barang_id')->on('m_barang');
    });
}
```



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



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

Minggu 3/PWL_POS

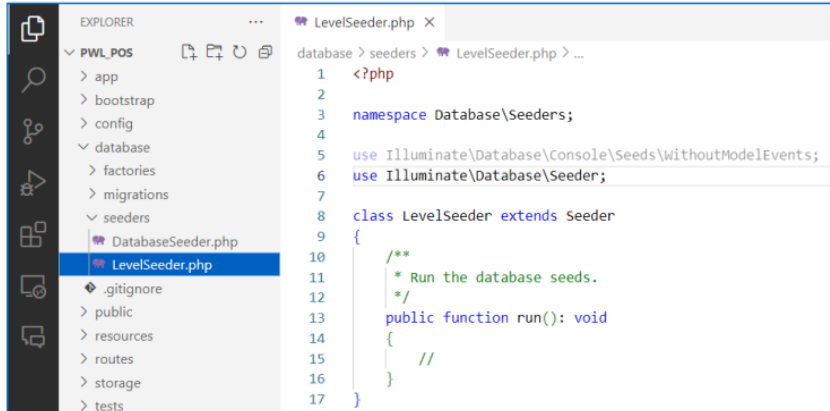
Praktikum 2.2 - Pembuatan file migrasi dengan relasi

15 minutes ago

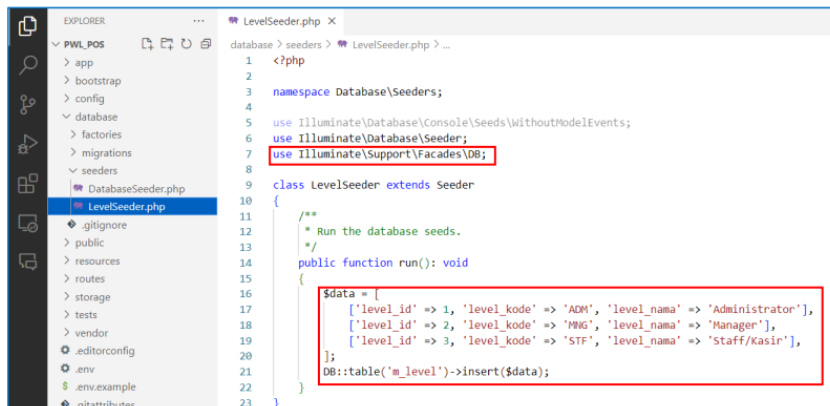
Praktikum 3 – Membuat file seeder

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

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



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



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

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



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

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/> Check all With selected: Edit Copy Delete Export						

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

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

6. Modifikasi file `class UserSeeder` seperti berikut

```

9  class UserSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             [
15                 'user_id' => 1,
16                 'level_id' => 1,
17                 'username' => 'admin',
18                 'nama' => 'Administrator',
19                 'password' => Hash::make('12345'), // class untuk mengenkripsi/hash password
20             ],
21             [
22                 'user_id' => 2,
23                 'level_id' => 2,
24                 'username' => 'manager',
25                 'nama' => 'Managen',
26                 'password' => Hash::make('12345'),
27             ],
28             [
29                 'user_id' => 3,
30                 'level_id' => 3,
31                 'username' => 'staff',
32                 'nama' => 'Staff/Kasir',
33                 'password' => Hash::make('12345'),
34             ],
35         ];
36         DB::table('m_user')->insert($data);
37     }
38 }
```

7. Jalankan perintah untuk mengeksekusi class `UserSeeder`

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

8. Perhatikan hasil seeder pada table `m_user`

		user_id	level_id	username	nama	password
<input type="checkbox"/>	Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$Tevu4dDO1CUAQpeM6H.Vp.LySwY.4oAKU7FzwS6DX...
<input type="checkbox"/>	Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$Ajfns20/FdPteUgghz31muEhIFaruLxkh5wvZ9NGRpu...
<input type="checkbox"/>	Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$Gi23TqGdIW5pYeR0VL4o5OxPwb30sk99VMY/BHnbJ9W...
<input type="checkbox"/> Check all With selected: Edit Copy Delete Export						

9. Ok, data seeder berhasil di masukkan ke database.
 10. Sekarang coba kalian masukkan data seeder untuk table yang lain, dengan ketentuan seperti berikut

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

Jawab :

Kategori :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:seeder KategoriSeeder
```

```
INFO Seeder [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\seeders\KategoriSeeder.php] created successfully.
```

```
public function run(): void
{
    $data = [
        [
            'kategori_id' => 1,
            'kategori_kode' => 'K001',
            'kategori_nama' => 'Bayi & Anak-anak',
        ],
        [
            'kategori_id' => 2,
            'kategori_kode' => 'K002',
            'kategori_nama' => 'Kecantikan & Kesehatan',
        ],
        [
            'kategori_id' => 3,
            'kategori_kode' => 'K003',
            'kategori_nama' => 'Makanan & Minuman',
        ],
        [
            'kategori_id' => 4,
            'kategori_kode' => 'K004',
            'kategori_nama' => 'Perawatan Rumah',
        ],
        [
            'kategori_id' => 5,
            'kategori_kode' => 'K005',
            'kategori_nama' => 'Elektronik',
        ],
    ];
    DB::table('m_kategori')->insert($data);
}
```

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan db:seed --class=KategoriSeeder
```

```
INFO Seeding database.
```

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	K001	Bayi & Anak-anak	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	K002	Kecantikan & Kesehatan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	K003	Makanan & Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	K004	Perawatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	K005	Elektronik	NULL	NULL

Barang :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:seeder BarangSeeder
```

```
INFO Seeder [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\seeders\BarangSeeder.php] created successfully.
```

```

public function run(): void
{
    $data = [
        // Kategori 1: Bayi & Anak-anak
        [
            'kategori_id' => 1,
            'barang_kode' => 'B001',
            'barang_nama' => 'Popok Bayi',
            'harga_beli' => 30000,
            'harga_jual' => 40000,
        ],
        [
            'kategori_id' => 1,
            'barang_kode' => 'B002',
            'barang_nama' => 'Susu Formula',
            'harga_beli' => 70000,
            'harga_jual' => 85000,
        ],
        // Kategori 2: Kecantikan & Kesehatan
        [
            'kategori_id' => 2,
            'barang_kode' => 'B003',
            'barang_nama' => 'Sabun Herbal',
            'harga_beli' => 15000,
            'harga_jual' => 20000,
        ],
        [
            'kategori_id' => 2,
            'barang_kode' => 'B004',
            'barang_nama' => 'Shampoo Anti Ketombe',
            'harga_beli' => 25000,
            'harga_jual' => 35000,
        ],
        // Kategori 3: Makanan & Minuman
        [
            'kategori_id' => 3,
            'barang_kode' => 'B005',
            'barang_nama' => 'Biskuit Gandum',
            'harga_beli' => 10000,
            'harga_jual' => 15000,
        ],
        [
            'kategori_id' => 3,
            'barang_kode' => 'B006',
            'barang_nama' => 'Minuman Kopi Golda',
            'harga_beli' => 3500,
            'harga_jual' => 5000,
        ],
        // Kategori 4: Perawatan Rumah
        [
            'kategori_id' => 4,
            'barang_kode' => 'B007',
            'barang_nama' => 'Deterjen Cair',
            'harga_beli' => 25000,
            'harga_jual' => 35000,
        ],
        [
            'kategori_id' => 4,
            'barang_kode' => 'B008',
            'barang_nama' => 'Pembersih Lantai',
            'harga_beli' => 15000,
            'harga_jual' => 17000,
        ],
        // Kategori 5: Elektronik
        [
            'kategori_id' => 5,
            'barang_kode' => 'B009',
            'barang_nama' => 'Rice Cooker',
            'harga_beli' => 300000,
            'harga_jual' => 350000,
        ],
        [
            'kategori_id' => 5,
            'barang_kode' => 'B010',
            'barang_nama' => 'Kipas Angin',
            'harga_beli' => 160000,
            'harga_jual' => 200000,
        ],
    ];
    DB::table('m_barang')->insert($data);
}

```

PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan db:seed --class=BarangSeeder

INFO Seeding database.

		barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/>	 Edit  Copy  Delete	1	1	B001	Popok Bayi	30000	40000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	2	1	B002	Susu Formula	70000	85000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	3	2	B003	Sabun Herbal	15000	20000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	4	2	B004	Shampoo Anti Ketombe	25000	35000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	5	3	B005	Biskuit Gandum	10000	15000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	6	3	B006	Minuman Kopi Golda	3500	5000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	7	4	B007	Deterjen Cair	25000	35000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	8	4	B008	Pembersih Lantai	15000	17000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	9	5	B009	Rice Cooker	300000	350000	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	10	5	B010	Kipas Angin	160000	200000	NULL	NULL

Stok :

PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:seeder StokSeeder

INFO Seeder [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\seeders\StokSeeder.php] created successfully.

```
public function run(): void
{
    $data = [
        [
            'barang_id' => 1,
            'user_id' => 1,
            'stok_tanggal' => now(),
            'stok_jumlah' => 50,
        ],
        [
            'barang_id' => 2,
            'user_id' => 2,
            'stok_tanggal' => now(),
            'stok_jumlah' => 40,
        ],
        [
            'barang_id' => 3,
            'user_id' => 3,
            'stok_tanggal' => now(),
            'stok_jumlah' => 30,
        ],
        [
            'barang_id' => 4,
            'user_id' => 1,
            'stok_tanggal' => now(),
            'stok_jumlah' => 25,
        ],
        [
            'barang_id' => 5,
            'user_id' => 2,
            'stok_tanggal' => now(),
            'stok_jumlah' => 60,
        ],
    ];
    DB::table('t_stok')->insert($data);
}

PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan db:seed --class=StokSeeder
INFO Seeding database.
```

Penjualan :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:seeder PenjualanSeeder
INFO Seeder [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\seeders\PenjualanSeeder.php] created successfully.

public function run(): void
{
    $data = [
        [
            'user_id' => 1,
            'pembeli' => 'Gilang Purnomo',
            'penjualan_kode' => 'TRX001',
            'tanggal_penjualan' => date('Y-m-d H:i:s', strtotime('2025-02-26')),
        ],
        [
            'user_id' => 2,
            'pembeli' => 'Alpin Rahman',
            'penjualan_kode' => 'TRX002',
            'tanggal_penjualan' => date('Y-m-d H:i:s', strtotime('2025-02-26')),
        ],
        [
            'user_id' => 3,
            'pembeli' => 'Satrio Nugroho',
            'penjualan_kode' => 'TRX003',
            'tanggal_penjualan' => date('Y-m-d H:i:s', strtotime('2025-02-27')),
        ],
        [
            'user_id' => 1,
            'pembeli' => 'Rizky Pratama',
            'penjualan_kode' => 'TRX004',
            'tanggal_penjualan' => date('Y-m-d H:i:s', strtotime('2025-02-27')),
        ],
        [
            'user_id' => 2,
            'pembeli' => 'Fajar Aditya',
            'penjualan_kode' => 'TRX005',
            'tanggal_penjualan' => date('Y-m-d H:i:s', strtotime('2025-02-28')),
        ],
    ];
    DB::table('t_penjualan')->insert($data);
}

PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan db:seed --class=PenjualanSeeder
INFO Seeding database.
```

←T→		penjualan_id	user_id	pembeli	penjualan_kode	tanggal_penjualan	created_at	updated_at
<input type="checkbox"/>	 Edit  Copy  Delete	1	1	Gilang Purnomo	TRX001	2025-02-26 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	2	2	Alpin Rahman	TRX002	2025-02-26 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	3	3	Satrio Nugroho	TRX003	2025-02-27 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	4	1	Rizky Pratama	TRX004	2025-02-27 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	5	2	Fajar Aditya	TRX005	2025-02-28 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	6	3	Farhan Putra	TRX006	2025-02-28 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	7	1	Citra Ayu	TRX007	2025-03-01 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	8	2	Andi Kurniawan	TRX008	2025-03-01 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	9	3	Fadillah Rahma	TRX009	2025-03-01 00:00:00	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	10	1	Fikri Ramadhan	TRX010	2025-03-02 00:00:00	NULL	NULL

Penjualan Detail :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan make:seeder PenjualanDetailSeeder
•
INFO Seeder [C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.

public function run(): void
{
    $data = [
        // Transaksi 1
        ['detail_id' => 1, 'penjualan_id' => 1, 'barang_id' => 1, 'harga' => 40000, 'jumlah' => 2],
        ['detail_id' => 2, 'penjualan_id' => 1, 'barang_id' => 2, 'harga' => 85000, 'jumlah' => 1],
        ['detail_id' => 3, 'penjualan_id' => 1, 'barang_id' => 3, 'harga' => 20000, 'jumlah' => 3],

        // Transaksi 2
        ['detail_id' => 4, 'penjualan_id' => 2, 'barang_id' => 4, 'harga' => 35000, 'jumlah' => 1],
        ['detail_id' => 5, 'penjualan_id' => 2, 'barang_id' => 5, 'harga' => 15000, 'jumlah' => 4],
        ['detail_id' => 6, 'penjualan_id' => 2, 'barang_id' => 6, 'harga' => 5000, 'jumlah' => 2],

        // Transaksi 3
        ['detail_id' => 7, 'penjualan_id' => 3, 'barang_id' => 7, 'harga' => 35000, 'jumlah' => 3],
        ['detail_id' => 8, 'penjualan_id' => 3, 'barang_id' => 8, 'harga' => 17000, 'jumlah' => 1],
        ['detail_id' => 9, 'penjualan_id' => 3, 'barang_id' => 9, 'harga' => 350000, 'jumlah' => 2],

        // Transaksi 4
        ['detail_id' => 10, 'penjualan_id' => 4, 'barang_id' => 10, 'harga' => 200000, 'jumlah' => 1],
        ['detail_id' => 11, 'penjualan_id' => 4, 'barang_id' => 1, 'harga' => 40000, 'jumlah' => 2],
        ['detail_id' => 12, 'penjualan_id' => 4, 'barang_id' => 2, 'harga' => 85000, 'jumlah' => 3],

        // Transaksi 5
        ['detail_id' => 13, 'penjualan_id' => 5, 'barang_id' => 3, 'harga' => 20000, 'jumlah' => 1],
        ['detail_id' => 14, 'penjualan_id' => 5, 'barang_id' => 4, 'harga' => 35000, 'jumlah' => 2],
        ['detail_id' => 15, 'penjualan_id' => 5, 'barang_id' => 5, 'harga' => 15000, 'jumlah' => 4],

        // Transaksi 6
        ['detail_id' => 16, 'penjualan_id' => 6, 'barang_id' => 6, 'harga' => 5000, 'jumlah' => 2],
        ['detail_id' => 17, 'penjualan_id' => 6, 'barang_id' => 7, 'harga' => 35000, 'jumlah' => 1],
        ['detail_id' => 18, 'penjualan_id' => 6, 'barang_id' => 8, 'harga' => 17000, 'jumlah' => 3],

        // Transaksi 7
        ['detail_id' => 19, 'penjualan_id' => 7, 'barang_id' => 9, 'harga' => 350000, 'jumlah' => 1],
        ['detail_id' => 20, 'penjualan_id' => 7, 'barang_id' => 10, 'harga' => 200000, 'jumlah' => 2],
        ['detail_id' => 21, 'penjualan_id' => 7, 'barang_id' => 1, 'harga' => 40000, 'jumlah' => 3],

        // Transaksi 8
        ['detail_id' => 22, 'penjualan_id' => 8, 'barang_id' => 2, 'harga' => 85000, 'jumlah' => 1],
        ['detail_id' => 23, 'penjualan_id' => 8, 'barang_id' => 3, 'harga' => 20000, 'jumlah' => 2],
        ['detail_id' => 24, 'penjualan_id' => 8, 'barang_id' => 4, 'harga' => 35000, 'jumlah' => 3],

        // Transaksi 9
        ['detail_id' => 25, 'penjualan_id' => 9, 'barang_id' => 5, 'harga' => 15000, 'jumlah' => 1],
        ['detail_id' => 26, 'penjualan_id' => 9, 'barang_id' => 6, 'harga' => 5000, 'jumlah' => 2],
        ['detail_id' => 27, 'penjualan_id' => 9, 'barang_id' => 7, 'harga' => 35000, 'jumlah' => 3],

        // Transaksi 10
        ['detail_id' => 28, 'penjualan_id' => 10, 'barang_id' => 8, 'harga' => 17000, 'jumlah' => 1],
        ['detail_id' => 29, 'penjualan_id' => 10, 'barang_id' => 9, 'harga' => 350000, 'jumlah' => 2],
        ['detail_id' => 30, 'penjualan_id' => 10, 'barang_id' => 10, 'harga' => 200000, 'jumlah' => 3],
    ];
    DB::table('t_penjualan_detail')->insert($data);
}
```

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan db:seed --class=PenjualanDetailSeeder
INFO Seeding database.
```

		detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	1	1	40000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	1	2	85000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	1	3	20000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	2	4	35000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	2	5	15000	4	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	2	6	5000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	3	7	35000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	3	8	17000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	9	3	9	350000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	10	4	10	200000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	11	4	1	40000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	12	4	2	85000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	13	5	3	20000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	14	5	4	35000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	15	5	5	15000	4	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	16	6	6	5000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	17	6	7	35000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	18	6	8	17000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	19	7	9	350000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	20	7	10	200000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	21	7	1	40000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	22	8	2	85000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	23	8	3	20000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	24	8	4	35000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	25	9	5	15000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	26	9	6	5000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	27	9	7	35000	3	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	28	10	8	17000	1	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	29	10	9	350000	2	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	30	10	10	200000	3	NULL	NULL

11. Jika sudah, laporkan hasil Praktikum-3 ini dan commit perubahan pada git
Jawab :

Minggu 3/PWL_POS Praktikum 3 – Membuat file seeder 13 hours ago

Praktikum 4 – Implementasi DB Façade

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

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



2. Kita modifikasi dulu untuk routing-nya, ada di `PWL_POS/routes/web.php`

```
LevelController.php  web.php X
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\LevelController;
4  use Illuminate\Support\Facades\Route;
5
6
7  Route::get('/', function () {
8      return view('welcome');
9  });
10
11 Route::get('/level', [LevelController::class, 'index']);
```

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

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













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

				level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	CUS	Pelanggan	2024-02-26 08:20:00	NULL

Jawab :



Insert data baru berhasil

				level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	CUS	Pelanggan	2025-02-27 08:23:16	NULL

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

```

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

```

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](#)

Jawab :



Update data berhasil. Jumlah data yang diupdate: 1 baris

<div><div><div>←</div><div>→</div></div></div>			level_id	level_kode	level_nama	created_at	updated_at	
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	1	ADM	Administrator	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	2	MNG	Manager	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	3	STF	Staff/Kasir	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	4	CUS	Customer	2025-02-27 08:23:16	NULL

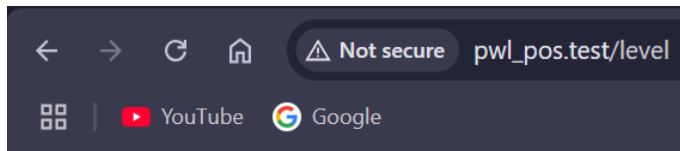
7. Kita coba modifikasi lagi file [LevelController](#) untuk melakukan proses hapus data

```

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

```

Jawab :



Delete data berhasil. Jumlah data yang dihapus: 1 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	Staff/Kasir	NULL	NULL

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

```

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

```

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

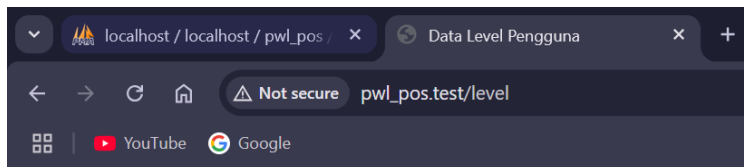
```

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

```

10. Silahkan dicoba pada browser dan amati apa yang terjadi

Jawab :



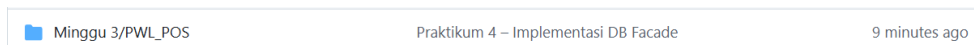
Data Level Pengguna

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

Jadi, ketika mengakses url /level Laravel mengeksekusi method index() di class LevelController. Controller ini mengambil data dari tabel m_level dengan menggunakan DB::select(), lalu diteruskan ke view(level.blade.php). Di vie, data ditampilkan dengan tabel menggunakan loop.

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

Jawab :



Praktikum 5 – Implementasi Query Builder

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

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

2. Kita modifikasi dulu untuk routing-nya, ada di `PWL_POS/routes/web.php`

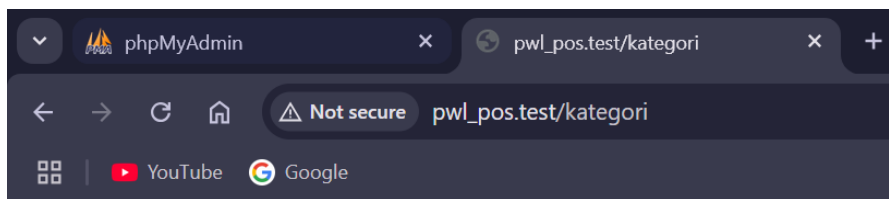
```
LevelController.php KategoriController.php level.blade.php web.php X
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\KategoriController;
4  use App\Http\Controllers\LevelController;
5  use Illuminate\Support\Facades\Route;
6
7
8  Route::get('/', function () {
9      return view('welcome');
10 });
11
12 Route::get('/level', [LevelController::class, 'index']);
13 Route::get('/kategori', [KategoriController::class, 'index']);
```

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

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

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`

Jawab :



Insert data baru berhasil

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	K001	Bayi & Anak-anak	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	K002	Kecantikan & Kesehatan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	K003	Makanan & Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	K004	Perawatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	K005	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Snack/Makanan Ringan	2025-02-27 17:37:18	NULL

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

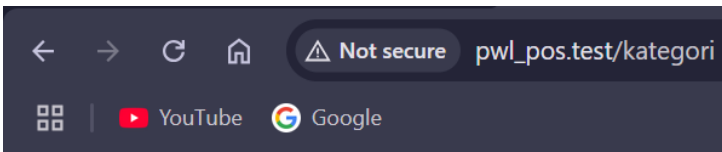
```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class 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 }

```

6. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori lagi dan amati apa yang terjadi pada table [m_kategori](#) di database, screenshot perubahan yang ada pada table [m_kategori](#)

Jawab :



Update data berhasil. Jumlah data yang diupdate: 1 baris

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	K001	Bayi & Anak-anak	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	K002	Kecantikan & Kesehatan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	K003	Makanan & Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	K004	Perawatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	K005	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Camilan	2025-02-27 17:37:18	NULL

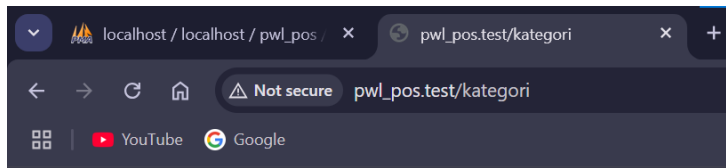
7. Kita coba modifikasi lagi file [KategoriController](#) untuk melakukan proses hapus data

```

10     public function index()
11     {
12         /* $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17         DB::table('m_kategori')->insert($data);
18         return 'Insert data baru berhasil'; */
19
20         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
21         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
22
23         $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
24         return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
25     }

```

Jawab :



Delete data berhasil. Jumlah data yang dihapus: 1 baris

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	K001	Bayi & Anak-anak	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	K002	Kecantikan & Kesehatan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	K003	Makanan & Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	K004	Perawatan Rumah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	K005	Elektronik	NULL	NULL

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

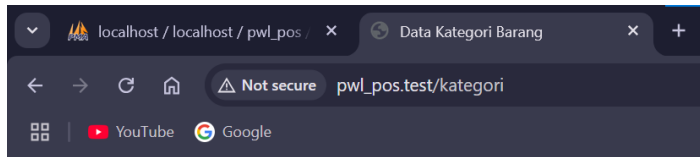
```
10 public function index()
11 {
12     /* $data = [
13         'kategori_kode' => 'SNK',
14         'kategori_nama' => 'Snack/Makanan Ringan',
15         'created_at' => now()
16     ];
17     DB::table('m_kategori')->insert($data);
18     return 'Insert data baru berhasil'; */
19
20     // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
21     // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
22
23     // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
24     // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
25
26     $data = DB::table('m_kategori')->get();
27     return view('kategori', ['data' => $data]);
28 }
```

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

```
resources > views > kategori.blade.php > html > body > table > tr > td
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Data Kategori Barang</title>
5 </head>
6 <body>
7 <h1>Data Kategori Barang</h1>
8 <table border="1" cellpadding="2" cellspacing="0">
9 <tr>
10 <th>ID</th>
11 <th>Kode Kategori</th>
12 <th>Nama Kategori</th>
13 </tr>
14 @foreach ($data as $d)
15 <tr>
16 <td>{{ $d->kategori_id }}</td>
17 <td>{{ $d->kategori_kode }}</td>
18 <td>{{ $d->kategori_nama }}</td>
19 </tr>
20 @endforeach
21 </table>
22 </body>
23 </html>
```

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

Jawab :



Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	K001	Bayi & Anak-anak
2	K002	Kecantikan & Kesehatan
3	K003	Makanan & Minuman
4	K004	Perawatan Rumah
5	K005	Elektronik

Jadi, saat kita mengakses url /kategori Laravel akan mengeksekusi class KategoriController. Controller akan memanggil method index() dan mengambil semua data dari tabel m_kategori dengan menggunakan DB::table('m_kategori')->get(), lalu data dikirim ke view kategori.blade.php untuk menampilkan datanya.

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

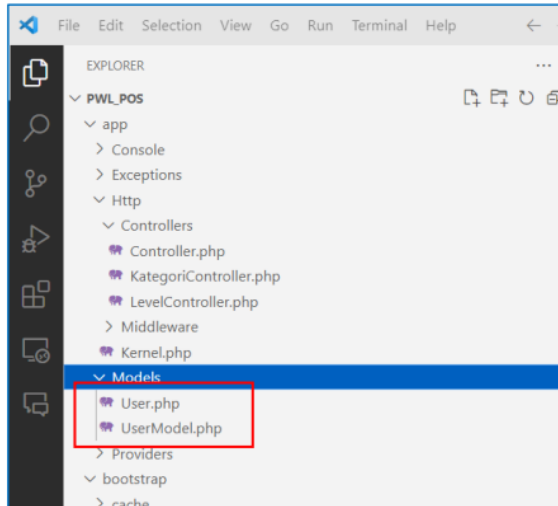
Jawab :

Minggu 3/PWL_POS	Praktikum 5 – Implementasi Query Builder	now
------------------	--	-----

Praktikum 6 – Implementasi Eloquent ORM

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

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



2. Setelah berhasil generate model, terdapat 2 file pada folder `model` yaitu file `User.php` bawaan dari laravel dan file `UserModel.php` yang telah kita buat. Kali ini kita akan menggunakan file `UserModel.php`
3. Kita buka file `UserModel.php` dan modifikasi seperti berikut

```
app > Models > UserModel.php > UserModel
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user'; // Mendefinisikan nama tabel yang digunakan oleh model ini
13     protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan
14 }
15
```

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

```
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\KategoriController;
4  use App\Http\Controllers\LevelController;
5  use App\Http\Controllers\UserController;
6  use Illuminate\Support\Facades\Route;
7
8
9  Route::get('/', function () {
10     return view('welcome');
11 });
12
13 Route::get('/level', [LevelController::class, 'index']);
14 Route::get('/kategori', [KategoriController::class, 'index']);
15 Route::get('/user', [UserController::class, 'index']);
16
```

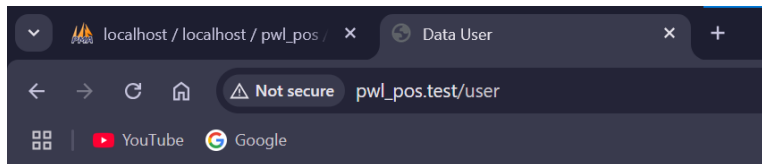

5. Sekarang, kita buat file controller **UserController** dan memodifikasinya seperti berikut

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

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

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

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



Data User

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

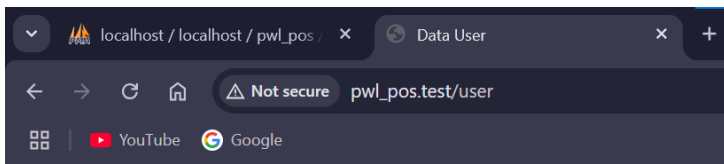
Menampilkan semua data pada tabel m_user

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

```
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data user dengan Eloquent Model
14         $data = [
15             'username' => 'customer-1',
16             'nama' => 'Pelanggan',
17             'password' => Hash::make('12345'),
18             'level_id' => 4
19         ];
20         UserModel::insert($data); // tambahkan data ke tabel m_user
21
22         // coba akses model UserModel
23         $user = UserModel::all(); // ambil semua data dari tabel m_user
24         return view('user', ['data' => $user]);
25     }
26 }
```

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

Jawab :



Data User

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

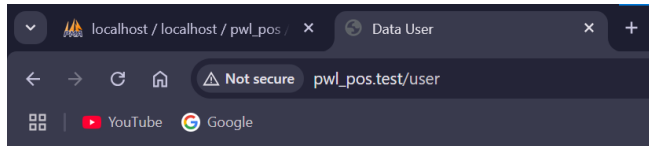
Menampilkan is dari tabel m_user yang sudah melakukan penambahan data.

10. Kita modifikasi lagi file [UserController](#) menjadi seperti berikut

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

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

Jawab :

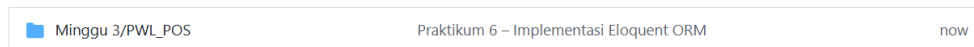


Data User

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

Menampilkan semua data yang ada di tabel m_user dengan data Nama yang sudah diupdate terhadap customer-1

12. Jika sudah, laporkan hasil Praktikum-6 ini dan commit perubahan pada git
- Jawab :



Penutup

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

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

Jawab : `APP_KEY` berfungsi sebagai kunci enkripsi yang sangat penting dalam keamanan aplikasi. Jika tidak diatur, ada beberapa fitur laravel yang bergantung pada enkripsi tidak dapat bekerja dengan baik.

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

Jawab : menggunakan perintah :

```
PS C:\laragon\www\Web_Lanjut\Minggu 3\PWL_POS> php artisan key:generate
```

Perintah tersebut secara otomatis akan menghasilkan kunci enkripsi baru dan memperbarui nilai `APP_KEY` di 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 3 file migrasi yang secara default ditambah saat proyek dibuat, yaitu

- `create_users_table.php` : ini untuk membuat tabel users, yang dapat menyimpan data pengguna.
- `create_password_reset_table.php` : untuk menyimpan token reset password bagi pengguna.
- `create_failed_jobs_table.php` : untuk mencatat jobs yang gagal.

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

Jawab : untuk menambah dua kolom secara otomatis yaitu

- `created_at` : untuk menyimpan waktu kapan data dibuat.
- `updated_at` : untuk menyimpan waktu ketika kapan data terakhir diperbarui.

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

Jawab : fungsi tersebut menghasilkan kolom dengan tipe data :

- `BIGINT (20) UNSIGNED` yaitu bilangan bulat yang bisa menyimpan angka dalam rentang yang sangat besar dan hanya bisa menyimpan angka positif.
- Auto Increment yaitu nilainya secara otomatis bertambah jika data ditambahkan.
- Primary key dan nama kolomnya `id`

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

Jawab : ini sama seperti diatas(no 5), tetapi dengan nama kolom `level_id`.

7. Pada migration, Fungsi `->unique()` digunakan untuk apa?
Jawab : untuk memastikan bahwa nilai dalam kolom tidak ada yang sama didalam tabel.
8. Pada **Praktikum 2.2 - Tahap 2**, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$tabel->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$tabel->id('level_id')` ?
Jawab : karena pada tabel `m_level`, `level_id` adalah sebagai primary key. Sedangkan pada tabel `m_user`, `level_id` ini sebagai foreign key yang mengacu pada tabel `m_level`, sehingga harus menggunakan `$tabel->unsignedBigInteger('level_id')`; agar tipe datanya cocok.
9. Pada **Praktikum 3 - Tahap 6**, apa tujuan dari Class `Hash`? dan apa maksud dari kode program `Hash::make('1234')`?
Jawab : class `Hash` digunakan untuk mengamankan data sensitif, terutama password agar lebih aman. Kode `Hash::make('1234')`; digunakan untuk mengubah string “12345” menjadi betuk hash bcrypt. Jadi hasil hash ini akan beda setiap kali dijalankan, walaupun inputnya sama.
10. Pada **Praktikum 4 - Tahap 3/5/7**, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
Jawab : jadi, tanda ? berfungsi sebagai placeholder sebagai pengganti nilai dinamis dalam query, sehingga lebih aman dari SQL Injection dan lebih fleksibel pengelolaan datanya.
11. Pada **Praktikum 6 - Tahap 3**, apa tujuan penulisan kode `protected $table = 'm_user'`; dan `protected $primaryKey = 'user_id'`; ?
Jawab : tujuan kode tersebut digunakan agar model Eloquent bisa bekerja dengan tabel `m_user` dan menggunakan `user_id` sebagai primary key, karena secara default Laravel mengasumsikan nama tabel dan primary key yang berbeda.
12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan
Jawab :
- DB Façade
ini sangat cepat karena langsung mengeksekusi SQL, tetapi kurang fleksibel karena harus menulis query manual.
 - Query Builder
Ini lebih dinamis tanpa harus menulis SQL secara langsung. Metode ini lebih mudah digunakan dibandingkan raw query dan mendukung berbagai database tanpa harus mengubah sintaksnya.
 - Eloquent ORM
Ini menerapkan pendekatan berbasis objek yang di mana setiap tabel dipresentasikan sebagai model. Oleh karena itu kode lebih rapi, mudah dibaca, dan lebih aman karena dapat mencegah SQL injection secara otomatis.

Jadi menurut saya dari ketiga database itu, metode Eloquent ORM yang paling direkomendasikan untuk operasi CRUD karena pendekatannya lebih sederhana, berbasis model, dan mudah dipahami.