

# **Laporan Pemrograman Web Lanjut**

**Jobsheet 3 : Migration, Seeder, Db Façade,**

**Query Builder, dan Eloquent Orm**

**Dosen Pengampu : Dimas Wahyu Wibowo, ST.,MT.**



**Disusun Oleh:**

**Queenadhynar Azarine Dwipa A.**

**2341760109**

**SIB 2B**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

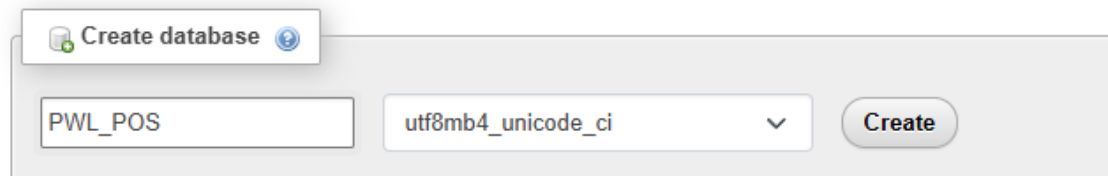
**2023/2024**

## A. PENGATURAN DATABASE

### Praktikum 1 – Pengaturan database :

1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL\_POS

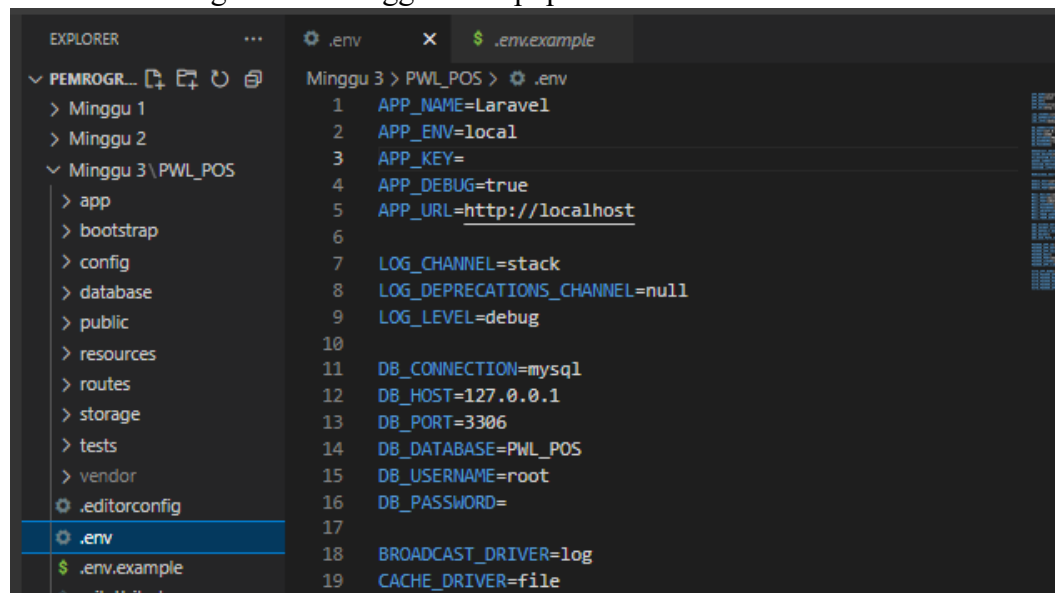
## Databases



Create database

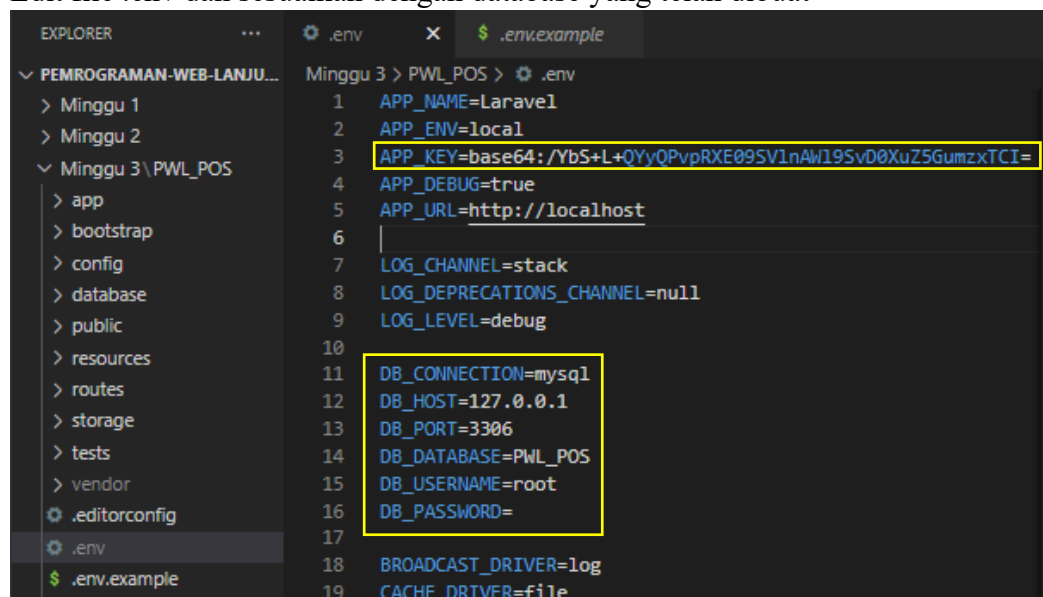
PWL\_POS utf8mb4\_unicode\_ci Create

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.



```
Minggu 3 > PWL_POS > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=PWL_POS
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
```

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



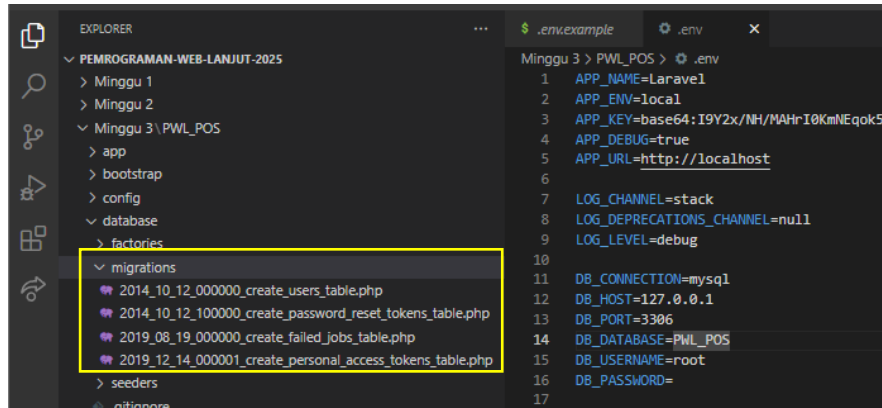
```
Minggu 3 > PWL_POS > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:/YbS+L+QYyQPvpRXE09SV1nAw19SvD0XuZ5GumzxTCI=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=PWL_POS
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
```

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

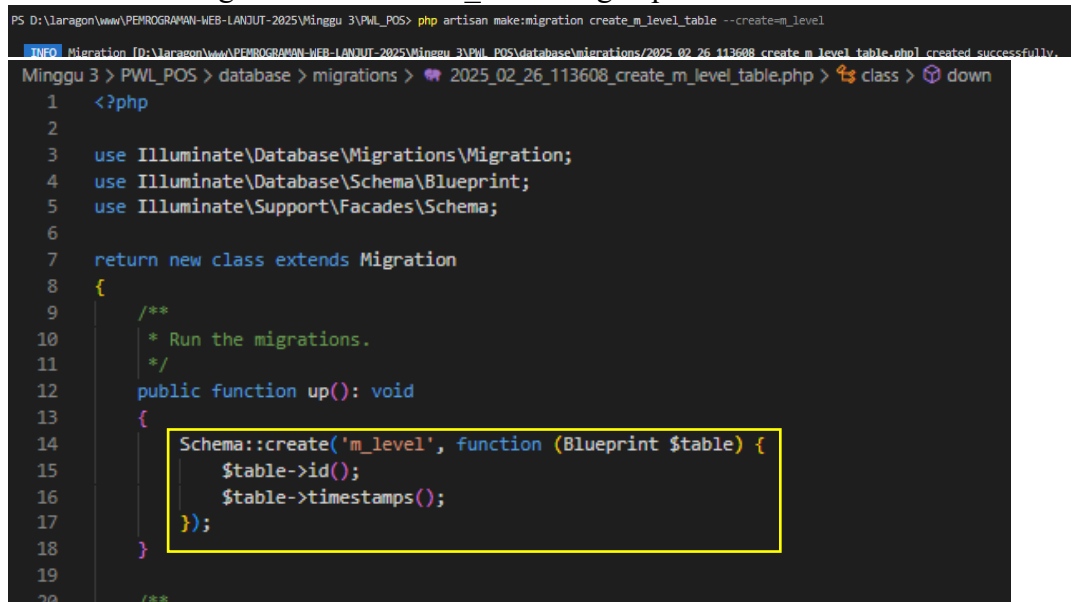
## B. MIGRATION

### Praktikum 2.1 - Pembuatan file migrasi tanpa relasi

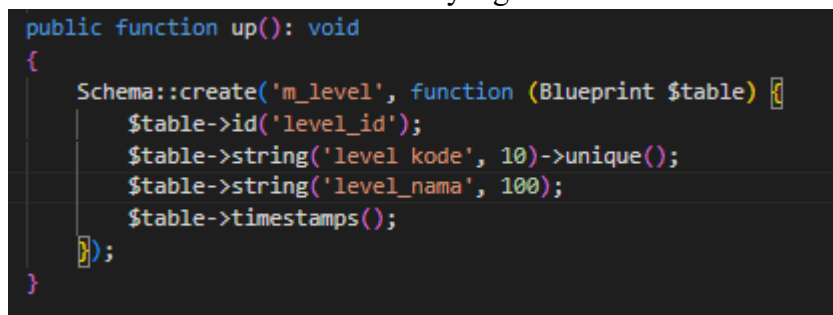
1. Buka terminal VSCode kalian, untuk yang di kotak merah adalah default dari Laravel



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



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



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

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan migrate

[WARN] The database 'PWL_POS' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [no]
> yes

[INFO] Preparing database.

Creating migration table ..... 113ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 104ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 28ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 76ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 84ms DONE
2025_02_26_113608_create_m_level_table ..... 67ms DONE

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_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
6 tables	Sum	5

7. Ok, table sudah dibuat di database

8. Buat table database dengan migration untuk table m\_kategori yang sama-sama tidak memiliki foreign key

- Membuat file migrasi untuk table m\_kategori dengan perintah

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:migration create_m_kategori_table --create=m_kategori

[INFO] Migration [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\database\migrations\2025_03_03_040939_create_m_kategori_table.php] created successfully.

```

- Menjalankan perintah dibawah ini untuk melakukan migrasi

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan migrate

[INFO] Running migrations.

2025_03_03_040939_create_m_kategori_table ..... 240ms DONE

```

- Tampilan di phpMyAdmin

<input type="checkbox"/> m_kategori	★ Browse Structure Search Insert Empty Drop	0 InnoDB utf8mb4_
-------------------------------------	---	-------------------

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

## Praktikum 2.2 - Pembuatan file migrasi dengan relasi

1. Buka terminal VSCode kalian, dan buat file migrasi untuk table m\_user

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:migration create_m_user_table --create=m_user

[INFO] Migration [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\database\migrations\2025_03_03_042450_create_m_user_table.php] created successfully.

```

2. Buka file migrasi untuk table m\_user, dan modifikasi seperti berikut

```

public function up(): void
{
    Schema::create('m_user', function (Blueprint $table) {
        $table->id('user_id');
        $table->unsignedBigInteger('level_id')->index(); // indexing untuk ForeignKey
        $table->string('username', 20)->unique(); // unique untuk memastikan tidak ada username yang sama
        $table->string('nama', 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');
    });
}

```

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



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

m_barang
t_penjualan
t_stok
t_penjualan_detail

- m\_barang

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    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();

            // Menambahkan foreign key constraint
            $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori')->onDelete('cascade');
        });
    }

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

```

- t\_penjualan

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('t_penjualan', function (Blueprint $table) {
            $table->id('penjualan_id');
            $table->unsignedBigInteger('user_id')->index();
            $table->string('penjualan_kode', 50)->unique();
            $table->integer('harga_jual');
            $table->timestamps();

            // Menambahkan foreign key constraint
            $table->foreign('user_id')->references('user_id')->on('m_user')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('t_penjualan', function (Blueprint $table) {
            //
        });
    }
};

```

- t\_stok

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    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();

            // Menambahkan foreign key constraint
            $table->foreign('user_id')->references('user_id')->on('m_user')->onDelete('cascade');
            $table->foreign('barang_id')->references('barang_id')->on('m_barang')->onDelete('cascade');
        });
    }
};

```

- t\_penjualan\_detail

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

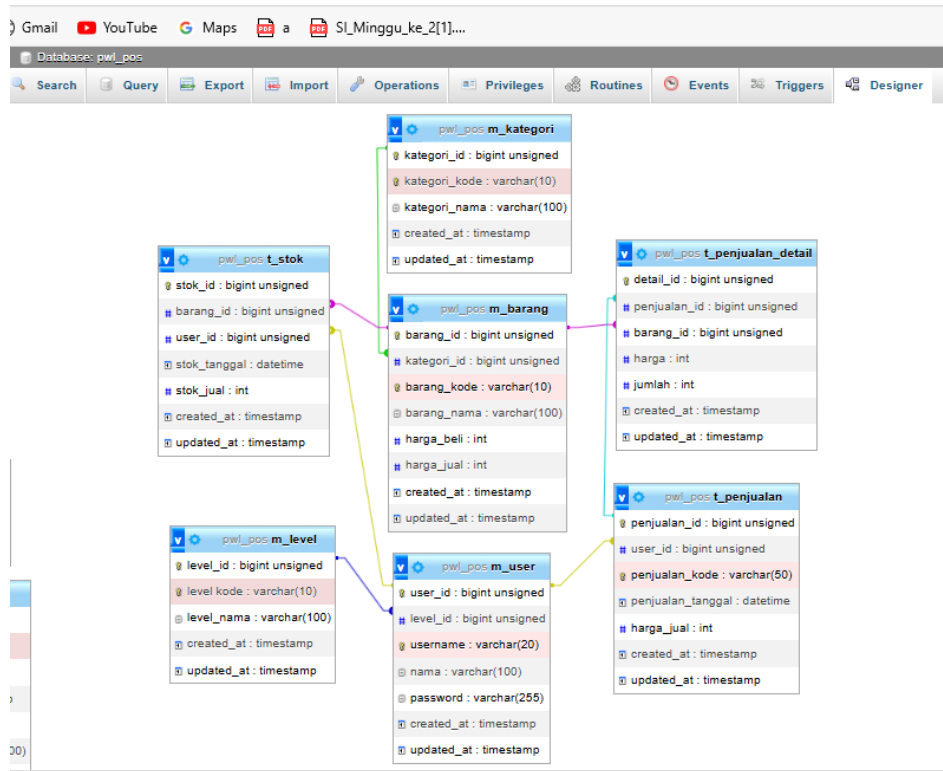
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('t_penjualan_detail', function (Blueprint $table) {
            $table->id('detail_id');
            $table->unsignedBigInteger('penjualan_id')->index();
            $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');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('t_penjualan_detail', function (Blueprint $table) {
            //
        });
    }
};
```

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

localhost/phpmyadmin/index.php?route=/database/designer&db=pwl\_pos



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

## C. SEEDER

### Praktikum 3 – Membuat file seeder

1. Kita akan membuat file seeder untuk table m\_level dengan mengetikkan perintah

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:seeder LevelSeeder

INFO Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\database\seeders\LevelSeeder.php] created successfully.
```

The screenshot shows a terminal window with the command `php artisan make:seeder LevelSeeder` executed successfully. Below the terminal, a file explorer window shows the 'seeders' directory containing two files: 'DatabaseSeeder.php' and 'LevelSeeder.php'.



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

```
<?php

namespace Database\Seeders;

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

class LevelSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            ['level_id' => 1, 'level kode' => 'ADM', 'level_nama' => 'Administrator'],
            ['level_id' => 2, 'level kode' => 'MNG', 'level_nama' => 'Manager'],
            ['level_id' => 3, 'level kode' => 'STF', 'level_nama' => 'Staff/Kasir'],
        ];
        DB::table('m_level')->insert($data);
    }
}
```

3. Selanjutnya, kita jalankan file seeder untuk table m\_level pada terminal

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan db:seed --class=LevelSeeder
>>

INFO Seeding database.
```

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

☐ 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

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:seeder UserSeeder
```

6. Modifikasi file class UserSeeder seperti berikut

```

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            [
                'user_id' => 1,
                'level_id' => 1,
                'username' => 'admin',
                'nama' => 'Administrator',
                'password' => Hash::make('12345'),
            ],
            [
                'user_id' => 2,
                'level_id' => 2,
                'username' => 'manager',
                'nama' => 'Manager',
                'password' => Hash::make('12345'),
            ],
            [
                'user_id' => 3,
                'level_id' => 3,
                'username' => 'staff',
                'nama' => 'Staff/Kasir',
                'password' => Hash::make('12345'),
            ],
        ];

        DB::table('m_user')->insert($data);
    }
}

```

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan db:seed --class=UserSeeder

INFO Seeding database.

```

8. Perhatikan hasil seeder pada table m\_user

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$GbatlXYqAMlnGJAhVGKUTOR02PhIZvMXt52aFcVSzqv..
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$L0n3SRRVxqvrL8HkLi.4teLFAOVf1SilfgzdC4GAWE9...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$v5365NjKLxaAjZrBLUBqH.yShSjbl81R.7rZ4vR5Acl...

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	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

- KategoriSeeder

```

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:seeder KategoriSeeder

INFO Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\database\seeders\KategoriSeeder.php] created successfully.

```

```

class KategoriSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $kategori = [
            ['kategori_id' => 1, 'kategori_kode' => 'SKC', 'kategori_nama' => 'Skincare'],
            ['kategori_id' => 2, 'kategori_kode' => 'PKN', 'kategori_nama' => 'Pakaian'],
            ['kategori_id' => 3, 'kategori_kode' => 'ATK', 'kategori_nama' => 'Alat Tulis Kantor'],
            ['kategori_id' => 4, 'kategori_kode' => 'AOR', 'kategori_nama' => 'Alat Olahraga'],
            ['kategori_id' => 5, 'kategori_kode' => 'FNB', 'kategori_nama' => 'Food and Beverage'],
        ];

        DB::table('m_kategori')->insert($kategori);
    }
}

```

				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	SKC	Skincare	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	PKN	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	ATK	Alat Tulis Kantor	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	AOR	Alat Olahraga	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	FNB	Food and Beverage	NULL	NULL

## ➤ BarangSeeder

**INFO** Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL\_POS\database\seeds\BarangSeeder.php] created successfully.  
 PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL\_POS> php artisan db:seed --class=BarangSeeder

**INFO** Seeding database.

```

class BarangSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $barang = [
            ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'BNG001', 'barang_nama' => 'Toner', 'harga_beli' => 150000, 'harga_jual' => 185000],
            ['barang_id' => 2, 'kategori_id' => 1, 'barang_kode' => 'BNG002', 'barang_nama' => 'Serum', 'harga_beli' => 250000, 'harga_jual' => 315000],
            ['barang_id' => 3, 'kategori_id' => 2, 'barang_kode' => 'BNG003', 'barang_nama' => 'Rok Plisket', 'harga_beli' => 50000, 'harga_jual' => 65000],
            ['barang_id' => 4, 'kategori_id' => 2, 'barang_kode' => 'BNG004', 'barang_nama' => 'Kemeja Bunga', 'harga_beli' => 65000, 'harga_jual' => 75000],
            ['barang_id' => 5, 'kategori_id' => 3, 'barang_kode' => 'BNG005', 'barang_nama' => 'Penggaris', 'harga_beli' => 2500, 'harga_jual' => 4000],
            ['barang_id' => 6, 'kategori_id' => 3, 'barang_kode' => 'BNG006', 'barang_nama' => 'Kalkulator', 'harga_beli' => 100000, 'harga_jual' => 115000],
            ['barang_id' => 7, 'kategori_id' => 4, 'barang_kode' => 'BNG007', 'barang_nama' => 'Matras Yoga', 'harga_beli' => 80000, 'harga_jual' => 120000],
            ['barang_id' => 8, 'kategori_id' => 5, 'barang_kode' => 'BNG008', 'barang_nama' => 'Waffer Coklat', 'harga_beli' => 15000, 'harga_jual' => 18500],
            ['barang_id' => 9, 'kategori_id' => 5, 'barang_kode' => 'BNG009', 'barang_nama' => 'Yogurt', 'harga_beli' => 10000, 'harga_jual' => 10500],
            ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'BNG010', 'barang_nama' => 'Kopi Susu', 'harga_beli' => 10000, 'harga_jual' => 15000],
        ];

        DB::table('m_barang')->insert($barang);
    }
}

```

				barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	1	BNG001	Toner	150000	185000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	1	BNG002	Serum	250000	315000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	2	BNG003	Rok Plisket	50000	65000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	2	BNG004	Kemeja Bunga	65000	75000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	3	BNG005	Penggaris	2500	4000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	3	BNG006	Kalkulator	100000	115000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	7	4	BNG007	Matras Yoga	80000	120000	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	8	5	BNG008	Waffer Coklat	15000	18500	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	9	5	BNG009	Yogurt	10000	10500	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	10	5	BNG010	Kopi Susu	10000	15000	NULL	NULL

## ➤ StokSeeder

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PMW_POS> php artisan make:seeder StokSeeder
```

```
INFO Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PMW_POS\database\seeders\StokSeeder.php] created successfully.
```

```
class StokSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $stok = [
            ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 5, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 15, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 9, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 4, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 17, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 35, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 20, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 2, 'stok_tanggal' => now(), 'stok_jumlah' => 12, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 3, 'stok_tanggal' => now(), 'stok_jumlah' => 27, 'created_at' => now(), 'updated_at' => now()],
            ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 10, 'created_at' => now(), 'updated_at' => now()],
        ];

        DB::table('t_stok')->insert($stok);
    }
}
```

		stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/>	Edit	1	1	1	2025-03-03 14:46:51	5	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	2	2	2	2025-03-03 14:46:51	15	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	3	3	3	2025-03-03 14:46:51	9	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	4	4	1	2025-03-03 14:46:51	4	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	5	5	2	2025-03-03 14:46:51	17	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	6	6	3	2025-03-03 14:46:51	35	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	7	7	1	2025-03-03 14:46:51	20	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	8	8	2	2025-03-03 14:46:51	12	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	9	9	3	2025-03-03 14:46:51	27	2025-03-03 14:46:51	2025-03-03 14:46:51
<input type="checkbox"/>	Edit	10	10	1	2025-03-03 14:46:51	10	2025-03-03 14:46:51	2025-03-03 14:46:51

## ➤ PenjualanSeeder

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PMW_POS> php artisan make:seeder PenjualanSeeder
```

```
INFO Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PMW_POS\database\seeders\PenjualanSeeder.php] created successfully.
```

```
class PenjualanSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $penjualan = [
            ['penjualan_id' => 1, 'user_id' => 1, 'pembeli' => 'Dhynar', 'penjualan_kode' => 'PJ001', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 2, 'user_id' => 2, 'pembeli' => 'Aqueena', 'penjualan_kode' => 'PJ002', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 3, 'user_id' => 3, 'pembeli' => 'Bagas', 'penjualan_kode' => 'PJ003', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 4, 'user_id' => 1, 'pembeli' => 'Putera', 'penjualan_kode' => 'PJ004', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 5, 'user_id' => 2, 'pembeli' => 'Daffa', 'penjualan_kode' => 'PJ005', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 6, 'user_id' => 3, 'pembeli' => 'Veve', 'penjualan_kode' => 'PJ006', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 7, 'user_id' => 1, 'pembeli' => 'Bebby', 'penjualan_kode' => 'PJ007', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 8, 'user_id' => 2, 'pembeli' => 'Eva', 'penjualan_kode' => 'PJ008', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 9, 'user_id' => 3, 'pembeli' => 'Nimas', 'penjualan_kode' => 'PJ009', 'penjualan_tanggal' => now()],
            ['penjualan_id' => 10, 'user_id' => 1, 'pembeli' => 'Rizky', 'penjualan_kode' => 'PJ010', 'penjualan_tanggal' => now()],
        ];

        DB::table('t_penjualan')->insert($penjualan);
    }
}
```

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	Dhynar	PJ001	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	2	Aqueena	PJ002	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	3	Bagas	PJ003	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	1	Putera	PJ004	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	2	Daffa	PJ005	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	6	3	Veve	PJ006	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	1	Bebby	PJ007	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	8	2	Eva	PJ008	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	9	3	Nimas	PJ009	2025-03-03 15:37:37	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	10	1	Rizky	PJ010	2025-03-03 15:37:37	NULL	NULL

## ➤ PenjualanDetailSeeder

PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL\_POS> php artisan make:seeder PenjualanDetailSeeder

INFO Seeder [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL\_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.

```
class PenjualanDetailSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $detailpjl = [];
        for ($i = 1; $i <= 10; $i++) { // 10 transaksi penjualan
            for ($j = 1; $j <= 3; $j++) { // 3 barang per transaksi
                $barang_id = rand(1, 10); // Pilih barang secara acak
                $harga = DB::table('m_barang')->where('barang_id', $barang_id)->value('harga_jual') ?? 10000; // Default harga 10000 jika null

                $detailpjl[] = [
                    'penjualan_id' => $i,
                    'barang_id' => $barang_id,
                    'harga' => $harga,
                    'jumlah' => rand(1, 5),
                    'created_at' => now(),
                    'updated_at' => now()
                ];
            }
        }

        // Insert batch untuk efisiensi
        DB::table('t_penjualan_detail')->insert($detailpjl);
    }
}
```

	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	1	185000	1	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	2	1	3	65000	5	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	3	1	8	18500	4	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	4	2	5	4000	4	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	5	2	2	315000	1	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	6	2	3	65000	2	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	7	3	1	185000	4	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	8	3	6	115000	5	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	9	3	6	115000	3	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	10	4	3	65000	2	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	11	4	3	65000	1	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	12	4	2	315000	3	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	13	5	10	15000	3	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	14	5	4	75000	2	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	15	5	10	15000	1	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	16	6	3	65000	2	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	17	6	5	4000	4	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	18	6	9	10500	1	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	19	7	9	10500	4	2025-03-03 16:13:53	2025-03-03 16:13:53
<input type="checkbox"/> Edit Copy Delete	20	7	4	75000	3	2025-03-03 16:13:53	2025-03-03 16:13:53

oute=/database/structure&db=information\_s...

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

## D. DB FACEDE

### Praktikum 4 – Implementasi DB Façade

1. Kita buat controller dahulu untuk mengelola data pada table m\_level

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:controller LevelController
INFO Controller [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\app\Http\Controllers\LevelController.php] created successfully.
```

2. Kita modifikasi dulu untuk routing-nya, ada di PWL\_POS/routes/web.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 great.
|
*/

Route::get('/', function () {
    return view('welcome');
});

Route::get('/', function(){
    return view('welcome');
});

Route::get('/level', [LevelController::class, 'index']);
```

3. Selanjutnya, kita modifikasi file LevelController untuk menambahkan 1 data ke table m\_level

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class LevelController extends Controller
{
    public function index(){
        DB::insert('insert into m_level (level_kode, level_nama, created_at, updated_at) values (?, ?, ?, ?)',
        ['CUS', 'Pelanggan', now(), now()]);

        return 'Insert data baru berhasil';
    }
}
```

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<title>Data level Pengguna</title>
</head>
<body>
<h1>Data Level Pengguna</h1>
<table border="1" cellpadding="2" cellspacing="0">
<tr>
<th>ID</th>
<th>Kode Level</th>
<th>Nama Level</th>
</tr>

@foreach ($data as $d)
<tr>
<td>{{ $d->level_id }}</td>
<td>{{ $d->level_kode }}</td>
<td>{{ $d->level_nama }}</td>
</tr>
@endforeach
</table>
</body>
</html>
```

10. Silahkan dicoba pada browser dan amati apa yang terjadi

## Data Level Pengguna

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

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



## E. QUERY BUILDER

### Praktikum 5 – Implementasi Query Builder

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

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:controller KategoriController

INFO Controller [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.
```

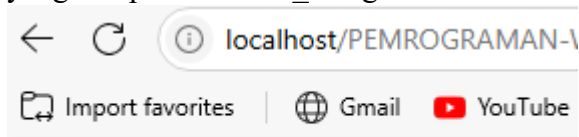
2. Kita modifikasi dulu untuk routing-nya, ada di `PWL_POS/routes/web.php`

```
Route::get('/level', [LevelController::class, 'index']);
Route::get('/kategori', [KategoriController::class, 'index']);
```

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

```
class KategoriController extends Controller
{
    public function index()
    {
        $data = [
            'kategori_kode' => 'SNK',
            'kategori_nama' => 'Snack/Makanan Ringan',
            'created_at' => now()
        ];
        DB::table('m_kategori')->insert($data);
        return 'Insert data baru berhasil';
    }
}
```

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

				kategori_id	kategori_kode	kategori_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	SKC	Skincare	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	PKN	Pakaian	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	ATK	Alat Tulis Kantor	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	AOR	Alat Olahraga	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	FNB	Food and Beverage	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Snack/Makanan Ringan	2025-03-03 17:46:18

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

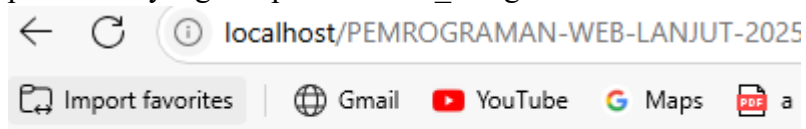
```

class KategoriController extends Controller
{
    public function index()
    {
        // $data = [
        //     'kategori_kode' => 'SNK',
        //     'kategori_nama' => 'Snack/Makanan Ringan',
        //     'created_at' => now()
        // ];
        // DB::table('m_kategori')->insert($data);
        // return 'Insert data baru berhasil';

        $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
        return 'update data berhasil. Jumlah data yang di update :'.$row.' baris';
    }
}

```

6. Kita coba jalankan di browser dengan url localhost/PWL\_POS/public/kategori lagi dan amati apa yang terjadi pada table m\_kategori di database, screenshot perubahan yang ada pada table m\_kategori



update data berhasil. Jumlah data yang di update :1 baris

<input type="checkbox"/>	Edit	Copy	Delete	4 AOR	Alat Olahraga	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5 FNB	Food and Beverage	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6 SNK	Camilan	2025-03-03 17:46:18

7. Kita coba modifikasi lagi file KategoriController untuk melakukan proses hapus data

```

$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
return 'Delete berhasil. Jumlah data yang di hapus '.$row.' Baris';

```

Delete berhasil. Jumlah data yang di hapus 1 Bari

				kategori_id	kategori_kode	kategori_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	SKC	Skincare	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	PKN	Pakaian	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	ATK	Alat Tulis Kantor	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	AOR	Alat Olahraga	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	FNB	Food and Beverage	NULL

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

```
$data = DB::table('m_kategori')->get();  
return view('kategori', ['data'=>$data]);
```

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

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

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

ID	Kode Kategori	Nama Kategori
1	SKC	Skincare
2	PKN	Pakaian
3	ATK	Alat Tulis Kantor
4	AOR	Alat Olahraga
5	FNB	Food and Beverage

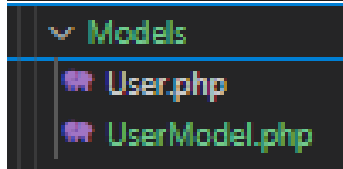
11. Laporkan hasil Praktikum-5 ini dan commit perubahan pada git

## F. ELOQUENT ORM

### Praktikum 6 – Implementasi Eloquent ORM

1. Kita buat file model untuk tabel m\_user dengan mengetikkan perintah

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS> php artisan make:model UserModel  
  
INFO Model [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 3\PWL_POS\app\Models\UserModel.php]  
reated successfully.
```



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

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class UserModel extends Model  
{  
    use HasFactory;  
  
    protected $table = 'm_user'; // mendefinisikan nama tabel yang digunakan  
    protected $primaryKey = 'user_id'; // mendefinisikan primary key dari tabel  
}
```

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

```
Route::get('/level', [LevelController::class, 'index']);  
Route::get('/kategori', [KategoriController::class, 'index']);  
Route::get('/user', [UserController::class, 'index']);
```

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

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\UserModel;  
use Illuminate\Http\Request;  
  
class UserController extends Controller  
{  
    public function index() {  
        //coba akses model UserModel  
        $user = UserModel::all(); // ambil semua data dari table m_user  
        return view('user', ['data' => $user]);  
    }  
}
```

6. Kemudian kita buat view user.blade.php

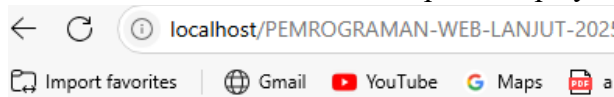
```

<!DOCTYPE html>
<html>
<head>
    <title>Data User</title>
</head>
<body>
    <h1>Data User</h1>
    <table border="1" cellpadding="2">
        <tr>
            <th>ID</th>
            <th>Username</th>
            <th>Nama</th>
            <th>IDPengguna</th>
        </tr>

        @foreach ($data as $d)
            <tr>
                <td>{{ $d->user_id }}</td>
                <td>{{ $d->username }}</td>
                <td>{{ $d->nama }}</td>
                <td>{{ $d->level_id }}</td>
            </tr>
        @endforeach
    </table>
</body>
</html>

```

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



## Data User

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

8. Setelah itu, kita modifikasi lagi file UserController

```

<?php

namespace App\Http\Controllers;

use App\Models\UserModel;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class UserController extends Controller
{
    public function index()
    {
        // tambah data user dengan Eloquent Model
        $data = [
            'username' => 'customer-1',
            'nama' => 'Pelanggan',
            'password' => Hash::make('12345'),
            'level_id' => 5
        ];
        UserModel::insert($data); // tambahkan data ke tabel m_user

        // coba akses model UserModel
        $user = UserModel::all(); // ambil semua data dari tabel m_user
        return view('user', ['data' => $user]);
    }
}

```

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

## Data User

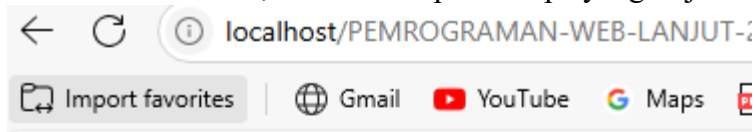
ID	Username	Nama	IDPengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
15	customer-1	Pelanggan	5

10. Kita modifikasi lagi file UserController menjadi seperti berikut

```
class UserController extends Controller
{
    public function index()
    {
        // tambah data user dengan Eloquent Model
        $data = [
            'nama' => 'Pelanggan Pertama',];

        UserModel::where('username', 'customer-1')->update($data);
        $user = UserModel::all();
        return view('user', ['data'=>$user]);
    }
}
```

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



## Data User

ID	Username	Nama	IDPengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
15	customer-1	Pelanggan Pertama	5

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

## G. PENUTUP

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

1. Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP\_KEY pada file setting .env Laravel?
  - APP\_KEY digunakan untuk mengenkripsi data dalam aplikasi Laravel, seperti session dan password. Tanpa APP\_KEY maka fitur keamanan tidak akan bekerja optimal
2. Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP\_KEY?
  - Dengan perintah `php artisan key:generate`
3. Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?
  - Secara default, Laravel memiliki 3 migrasi
    - `create_user_table.php` > untuk membuat table user menyimpan data pengguna
    - `create_password_resets_table.php` > untuk membuat table yang digunakan untuk menyimpan token reset password
    - `create_personal_access_tokens_table.php` > untuk membuat tabel yang digunakan oleh Laravel Sanctum untuk otentikasi berbasis token.
4. Secara default, file migrasi terdapat kode `$table->timestamps();`, apa tujuan/output dari fungsi tersebut?
  - Kode tersebut secara otomatis menambahkan dua kolom, yaitu *created\_at* dan *updated\_at* yang digunakan untuk melacak waktu pembuatan dan pembaruan suatu record table
5. Pada File Migrasi, terdapat fungsi `$table->id();` Tipe data apa yang dihasilkan dari fungsi tersebut?
  - Fungsi tersebut menghasilkan tipe data *unsigned big integer* yang secara otomatis bertambah nilai. Digunakan untuk menentukan kolom primary key
6. Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');` ?
  - Secara default jika `$table->id` akan menghasilkan kolom dengan nama 'id'. Jika ada tambahan `$table->id('level_id')` maka akan menghasilkan kolom dengan nama 'level\_id'
7. Pada migration, Fungsi `->unique()` digunakan untuk apa?
  - Fungsi `->unique()` untuk memberi aturan bahwa value dari kolom tersebut harus unique/tidak boleh ada yang sama
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')` ?
  - Kolom `level_id` pada tabel `m_user` menggunakan `$table->unsignedBigInteger('level_id')` karena sebagai primary key
  - kolom `level_id` pada tabel `m_level` menggunakan `$table->id('level_id')` karena berfungsi sebagai foreign key yang merujuk ke `level_id` di tabel `m_level`.
9. Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program `Hash::make('1234');`;
  - Hash memiliki tujuan untuk mengenkripsi suatu data tertentu seperti pada password

10. Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
  - Tanda (?) digunakan sebagai placeholder untuk binding parameter, agar lebih aman dari SQL Injection.
11. Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode `protected $table = 'm_user';` dan `protected $primaryKey = 'user_id';` ?
  - `protected $table = 'm_user';` untuk mendefinisikan nama tabel database yang digunakan oleh *Model Eloquent*.
  - `protected $primaryKey = 'user_id';` untuk menentukan kolom yang digunakan sebagai primary key pada tabel tersebut.
12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan
  - Menggunakan CRUD lebih mudah digunakan pada Eloquent ORM karena berbasis model dan OOP sehingga kode program lebih mudah dipahami