

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT (PWL)
JOB SHEET 3: MIGRATION, SEEDER, DB FAÇADE, QUERY BUILDER,
dan ELOQUENT ORM



OLEH:

LOVELYTA SEKARAYU KRISDIYANTI

KELAS 2B SIB / 11 (2341760081)

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

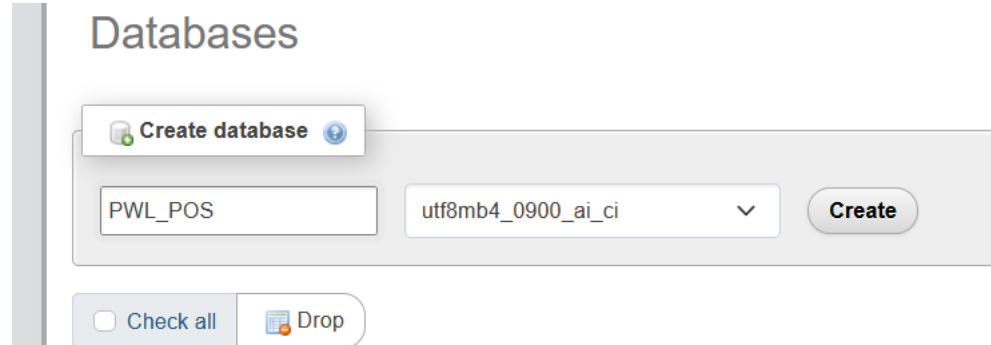
Jl. Soekarno Hatta No. 9 Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

65141

A. PENGATURAN DATABASE

❖ Praktikum 1

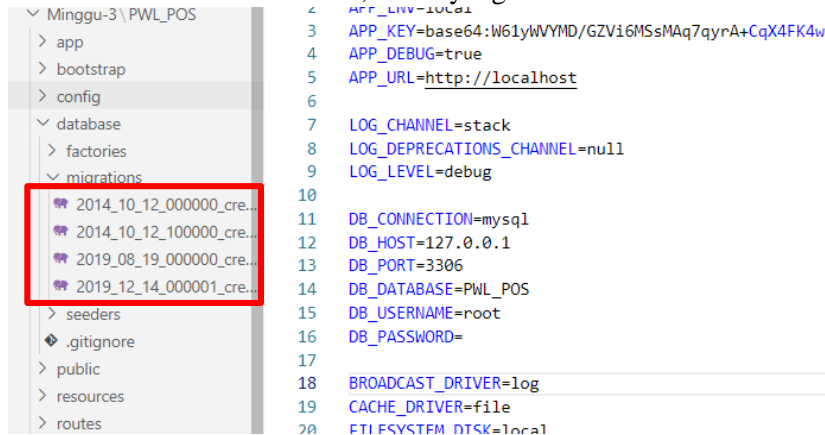
1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL_POS



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.

❖ Praktikum 2.1

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

```
INFO Migration [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS\database\migrations\2025_02_27_133543_create_m_level_table.php] created successfully.
```

Minggu-3 > PWL_POS > database > migrations > 2025_02_27_133543_create_m_level_table.php

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create('m_level', function (Blueprint $table) {
15              $table->id();
16              $table->timestamps();
17          });
18      }
19
20      /**
21       * Reverse the migrations.
22       */
23      public function down(): void
24      {
25          Schema::dropIfExists('m_level');
26      }
27  };
28
```

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

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_level', function (Blueprint $table) {
            $table->id('level_id');
            $table->string('level_kode', 10)->unique();
            $table->string('level_nama', 100);
            $table->timestamps();
        });
    }

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

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

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan migrate
```

```
WARN The database 'laravel' does not exist on the 'mysql' connection.
```

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

```
> yes
```

```
INFO Preparing database.
```

```
INFO Running migrations.
```

```
2014_10_12_000000_create_users_table ..... 62ms DONE
2014_10_12_100000_create_password_reset_tokens_table 14ms DONE
2019_08_19_000000_create_failed_jobs_table ... 101ms DONE
2019_12_14_000001_create_personal_access_tokens_table 550ms DONE
2025_02_27_133543_create_m_level_table ..... 388ms DONE
```

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

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan make:migration create_m_kategori
```

```
INFO Migration [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS\database\migrations\2025_03_01_142346_create_m_kategori.php] created successfully.
```

```
return new class extends Migration
```

```
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_kategori', function (Blueprint $table) {
            $table->unsignedBigInteger('kategori_id', 20);
            $table->string('kategori_kode', 10);
            $table->string('kategori_nama', 100);
            $table->timestamps();
        });
    }

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

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

❖ Praktikum 2.2

- Buka terminal VSCode kalian, dan buat file migrasi untuk table m_user

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan make:migration create_m_user_table --table=m_user
```

```
INFO Migration [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS\database\migrations\2025_02_27_141228_create_m_user_table.php] created successfully.
```

- Buka file migrasi untuk table m_user, dan modifikasi seperti berikut

```
12 public function up(): void
13 {
14     Schema::table('m_user', function (Blueprint $table) {
15         $table->id('user_id');
16         $table->unsignedBigInteger('level_id')->index();
17         $table->string('username', 20)->unique();
18         $table->string('nama', 100);
19         $table->string('password');
20         $table->timestamps();
21
22         $table->foreign('level_id')->references('level_id')->on('m_level');
23     });
24 }
25
26 /**
27  * Reverse the migrations.
28  */
29 public function down(): void
30 {
31     Schema::table('m_user');
32 }
33
```

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

Table	Action	Rows	Ty
<input type="checkbox"/> failed_jobs	★ Browse Structure Search Insert Empty Drop	0	Inr
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	6	Inr
<input type="checkbox"/> m_level	★ Browse Structure Search Insert Empty Drop	0	Inr
<input type="checkbox"/> m_user	★ Browse Structure Search Insert Empty Drop	0	Inr
<input type="checkbox"/> password_reset_tokens	★ Browse Structure Search Insert Empty Drop	0	Inr
<input type="checkbox"/> personal_access_tokens	★ Browse Structure Search Insert Empty Drop	0	Inr
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	0	Inr
7 tables	Sum	6	Inr

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

m_barang
t_penjualan
t_stok
t_penjualan_detail

- M_barang

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_barang', function (Blueprint $table) {
            $table->id('barang_id'); // Primary Key & Auto Increment
            $table->unsignedBigInteger('kategori_id')->index();
            $table->string('barang_kode', 10);
            $table->string('barang_nama', 100);
            $table->integer('harga_beli');
            $table->integer('harga_jual');
            $table->timestamps();

            // Foreign Key
            $table->foreign('kategori_id')->references('kategori_id')->on('m_kategori');
        });
    }

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

```

- t_penjualan

```

7   return new class extends Migration
8   {
9       public function up(): void
10      {
11          Schema::create('t_penjualan', function (Blueprint $table) {
12              $table->id('penjualan_id');
13              $table->unsignedBigInteger('user_id')->index();
14              $table->string('pembeli');
15              $table->string('penjualan_kode');
16              $table->dateTime('penjualan_tanggal');
17              $table->timestamps();

18              // Foreign Key
19              $table->foreign('user_id')->references('user_id')->on('m_user');
20          });
21      }
22  }
23
24  public function down(): void
25  {
26      Schema::dropIfExists('t_penjualan');
27  }
28  };

```

- t_stok

```

7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('t_stok', function (Blueprint $table) {
15             $table->id('stok_id');
16             $table->unsignedBigInteger('barang_id')->index();
17             $table->unsignedBigInteger('user_id')->index();
18             $table->dateTime('stok_tanggal');
19             $table->integer('stok_jumlah');
20             $table->timestamps();
21
22             // Foreign Key
23             $table->foreign('barang_id')->references('barang_id')->on('m_barang');
24             $table->foreign('user_id')->references('user_id')->on('m_user');
25         });
26     }
27
28     /**
29     * Reverse the migrations.
30     */
31     public function down(): void
32     {
33         Schema::dropIfExists('t_stok');
34     }
35 };

```

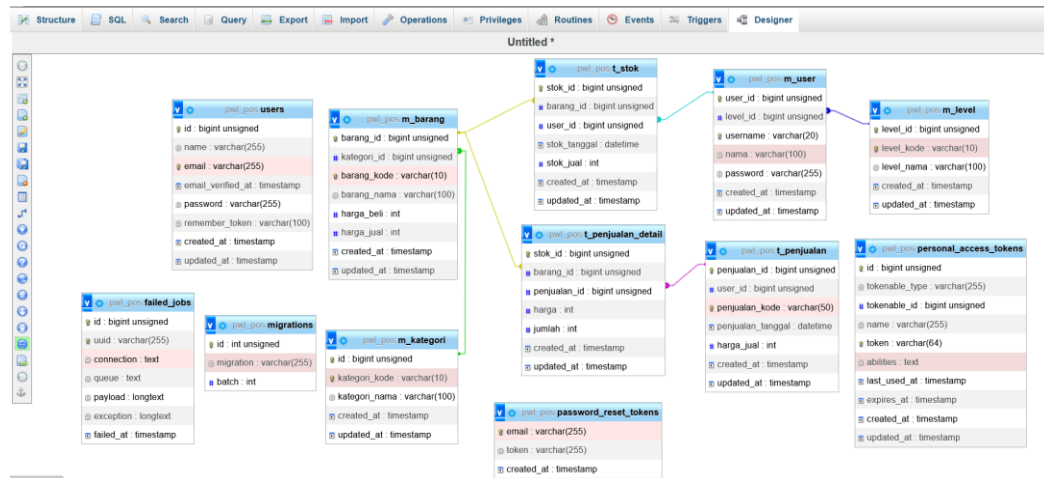
- **t_penjualan_detail**

```

7  return new class extends Migration
8  {
9      public function up(): void
10     {
11         Schema::create('t_stok', function (Blueprint $table) {
12             $table->id('stok_id');
13             $table->unsignedBigInteger('barang_id')->index();
14             $table->unsignedBigInteger('user_id')->index();
15             $table->dateTime('stok_tanggal');
16             $table->integer('stok_jumlah');
17             $table->timestamps();
18
19             // Foreign Key
20             $table->foreign('barang_id')->references('barang_id')->on('m_barang');
21             $table->foreign('user_id')->references('user_id')->on('m_user');
22         });
23     }
24     public function down(): void
25     {
26         Schema::dropIfExists('t_stok');
27     }
28 };

```

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.

B. SEEDER

❖ Praktikum 3

1. Perintah umum dalam membuat file seeder adalah seperti berikut

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan make :seeder LevelSeeder
```

INFO Seeder [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS\database\seeders\LevelSeeder.php] created successfully.

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

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

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


```
PS C:\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

Extra options						
		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

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

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan make:seeder UserSeeder
```

INFO Seeder [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS\database\seeders\UserSeeder.php] created successfully.

6. Modifikasi file class UserSeeder seperti berikut

```
class UserSeeder extends Seeder
{
    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 C:\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

Extra options

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$u1zI0FYp4gZ9mNEmW7vRguC
<input type="checkbox"/> Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$zDbSC/KtrSIIBsb/PkP9HeCh8Z
<input type="checkbox"/> Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$.byBH0xJETVymrKDPyDPeOdl

↑ 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	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

```
9 class KategoriSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [
17             ['kategori_id' => 1, 'kategori_kode' => 'KAT001', 'kategori_nama' => 'Makanan'],
18             ['kategori_id' => 2, 'kategori_kode' => 'KAT002', 'kategori_nama' => 'Elektronik'],
19             ['kategori_id' => 3, 'kategori_kode' => 'KAT003', 'kategori_nama' => 'Meubel'],
20             ['kategori_id' => 4, 'kategori_kode' => 'KAT004', 'kategori_nama' => 'Minuman'],
21             ['kategori_id' => 5, 'kategori_kode' => 'KAT005', 'kategori_nama' => 'Perabotan Pecah Belah'],
22         ];
23
24         // Insert ke tabel `m_kategori`
25         DB::table('m_kategori')->insert($data);
26     }
27 }
```

	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	KAT001	Makanan	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	KAT003	Meubel	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	KAT005	Perabotan Pecah Belah	NULL	NULL

- BarangSeeder

```

9 class BarangSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             ['barang_id' => 1, 'barang_kode' => 'B001', 'barang_nama' => 'Laptop', 'harga_beli' => 5000000, 'harga_jual' => 6000000, 'kategori_id' => 1],
15             ['barang_id' => 2, 'barang_kode' => 'B002', 'barang_nama' => 'Handphone', 'harga_beli' => 3000000, 'harga_jual' => 3500000, 'kategori_id' => 1],
16             ['barang_id' => 3, 'barang_kode' => 'B003', 'barang_nama' => 'Polo', 'harga_beli' => 50000, 'harga_jual' => 75000, 'kategori_id' => 2],
17             ['barang_id' => 4, 'barang_kode' => 'B004', 'barang_nama' => 'Pants', 'harga_beli' => 150000, 'harga_jual' => 200000, 'kategori_id' => 2],
18             ['barang_id' => 5, 'barang_kode' => 'B005', 'barang_nama' => 'Nasi Bungkus', 'harga_beli' => 25000, 'harga_jual' => 35000, 'kategori_id' => 3],
19             ['barang_id' => 6, 'barang_kode' => 'B006', 'barang_nama' => 'Air AQUA', 'harga_beli' => 5000, 'harga_jual' => 10000, 'kategori_id' => 4],
20             ['barang_id' => 7, 'barang_kode' => 'B007', 'barang_nama' => 'Kipas Angin', 'harga_beli' => 200000, 'harga_jual' => 300000, 'kategori_id' => 5],
21             ['barang_id' => 8, 'barang_kode' => 'B008', 'barang_nama' => 'Rice Cooker', 'harga_beli' => 400000, 'harga_jual' => 500000, 'kategori_id' => 5],
22             ['barang_id' => 9, 'barang_kode' => 'B009', 'barang_nama' => 'Speaker Bluetooth', 'harga_beli' => 100000, 'harga_jual' => 150000, 'kategori_id' => 1],
23             ['barang_id' => 10, 'barang_kode' => 'B010', 'barang_nama' => 'Teh Botol', 'harga_beli' => 7000, 'harga_jual' => 12000, 'kategori_id' => 4],
24         ];
25         DB::table('m_barang')->insert($data);
26     }
27 }

```

	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	B001	Laptop	5000000	6000000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	B002	Handphone	3000000	3500000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	2	B003	Polo	50000	75000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	2	B004	Pants	150000	200000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	3	B005	Nasi Bungkus	25000	35000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	4	B006	Air AQUA	5000	10000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	5	B007	Kipas Angin	200000	300000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	5	B008	Rice Cooker	400000	500000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	1	B009	Speaker Bluetooth	100000	150000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	4	B010	Teh Botol	7000	12000	NULL	NULL

- StokSeeder

```

9 class StokSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [];
14         for ($i = 1; $i <= 10; $i++) {
15             $data[] = [
16                 'stok_id' => $i,
17                 'barang_id' => rand(1, 10),
18                 'user_id' => rand(1, 2),
19                 'stok_tanggal' => now(),
20                 'stok_jumlah' => rand(10, 50),
21             ];
22         }
23         DB::table('t_stok')->insert($data);
24     }
25 }

```

	stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	6	1	2025-03-03 07:40:03	48	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	7	2	2025-03-03 07:40:03	39	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	2	2025-03-03 07:40:03	46	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	9	2	2025-03-03 07:40:03	22	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	1	2025-03-03 07:40:03	21	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	9	1	2025-03-03 07:40:03	34	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	10	1	2025-03-03 07:40:03	28	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	5	1	2025-03-03 07:40:03	27	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	1	1	2025-03-03 07:40:03	26	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	2	2	2025-03-03 07:40:03	37	NULL	NULL

- PenjualanSeeder

```

9 class PenjualanSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $data = [
14             ['penjualan_id' => 1, 'user_id' => 1, 'pembeli' => 'Bobi', 'penjualan_kode' => 'TRX001', 'penjualan_tanggal' => '2023-01-02 10:15:00'],
15             ['penjualan_id' => 2, 'user_id' => 1, 'pembeli' => 'Budi', 'penjualan_kode' => 'TRX002', 'penjualan_tanggal' => '2023-01-05 14:30:00'],
16             ['penjualan_id' => 3, 'user_id' => 1, 'pembeli' => 'Andi', 'penjualan_kode' => 'TRX003', 'penjualan_tanggal' => '2023-01-07 09:45:00'],
17             ['penjualan_id' => 4, 'user_id' => 1, 'pembeli' => 'Sinta', 'penjualan_kode' => 'TRX004', 'penjualan_tanggal' => '2023-01-10 16:00:00'],
18             ['penjualan_id' => 5, 'user_id' => 1, 'pembeli' => 'Dwi', 'penjualan_kode' => 'TRX005', 'penjualan_tanggal' => '2023-01-12 11:20:00'],
19             ['penjualan_id' => 6, 'user_id' => 1, 'pembeli' => 'Fajar', 'penjualan_kode' => 'TRX006', 'penjualan_tanggal' => '2023-01-15 17:10:00'],
20             ['penjualan_id' => 7, 'user_id' => 1, 'pembeli' => 'Vina', 'penjualan_kode' => 'TRX007', 'penjualan_tanggal' => '2023-01-18 13:40:00'],
21             ['penjualan_id' => 8, 'user_id' => 1, 'pembeli' => 'Edo', 'penjualan_kode' => 'TRX008', 'penjualan_tanggal' => '2023-01-20 08:50:00'],
22             ['penjualan_id' => 9, 'user_id' => 1, 'pembeli' => 'Putri', 'penjualan_kode' => 'TRX009', 'penjualan_tanggal' => '2023-01-25 15:05:00'],
23             ['penjualan_id' => 10, 'user_id' => 1, 'pembeli' => 'Eko', 'penjualan_kode' => 'TRX010', 'penjualan_tanggal' => '2023-01-28 12:25:00'],
24         ];
25
26         // Insert data ke tabel `t_penjualan`
27         DB::table('t_penjualan')->insert($data);
28     }
29 }
30

```

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Bobi	TRX001	2023-01-02 10:15:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	Budi	TRX002	2023-01-05 14:30:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	Andi	TRX003	2023-01-07 09:45:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	1	Sinta	TRX004	2023-01-10 16:00:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	1	Dwi	TRX005	2023-01-12 11:20:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	1	Fajar	TRX006	2023-01-15 17:10:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	1	Vina	TRX007	2023-01-18 13:40:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	1	Edo	TRX008	2023-01-20 08:50:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	1	Putri	TRX009	2023-01-25 15:05:00	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	1	Eko	TRX010	2023-01-28 12:25:00	NULL	NULL

- PenjualanDetailSeeder

```

9 class PenjualanDetailSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         $penjualan_ids = DB::table('t_penjualan')->pluck('penjualan_id'); // Ambil semua ID penjualan
14         $barang_ids = DB::table('m_barang')->pluck('barang_id'); // Ambil semua ID barang
15
16         $data = [];
17         $detail_id = 1; // Mulai dari 1 untuk detail_id
18
19         foreach ($penjualan_ids as $penjualan_id) {
20             // Ambil 3 barang secara acak untuk setiap transaksi
21             $barang_acak = $barang_ids->shuffle()->take(3);
22
23             foreach ($barang_acak as $barang_id) {
24                 $harga = rand(10000, 50000); // Harga random antara 10.000 - 50.000
25                 $jumlah = rand(1, 5); // Jumlah barang antara 1 - 5
26
27                 $data[] = [
28                     'detail_id' => $detail_id++, // Auto-increment manual
29                     'penjualan_id' => $penjualan_id,
30                     'barang_id' => $barang_id,
31                     'harga' => $harga,
32                     'jumlah' => $jumlah,
33                 ];
34             }
35         }
36
37         // Insert data ke tabel `t_penjualan_detail`
38         DB::table('t_penjualan_detail')->insert($data);
39     }
40 }

```

	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	3	30619	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	1	4	11983	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	1	2	33526	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	2	7	24218	1	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	2	3	22051	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	6	2	1	39122	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	3	3	49832	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	8	3	6	16899	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	9	3	10	46825	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	10	4	6	40284	1	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	11	4	10	27303	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	12	4	8	31588	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	13	5	6	26554	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	14	5	1	48357	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	15	5	9	13554	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	16	6	10	34340	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	17	6	4	48241	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	18	6	2	15722	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	19	7	2	21199	5	NULL	NULL

C. DB FAÇADE

❖ Praktikum 4

1. Kita buat controller dahulu untuk mengelola data pada table m_level
2. Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php

```

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

```

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

```

8  class LevelController extends Controller
9  {
10     public function index()
11     {
12         DB::insert('insert into m_level (level_kode, level_nama, created_at)
13             values (?, ?, ?)', [1, 'CUS', 'Pelanggan', now()]);
14
15         return 'Insert data baru berhasil';
16     }
17 }
18

```

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

← → ↻ localhost/PEMROGRAMAN_WEB_LANJUT_2025/Minggu-3/PWL_POS/public/level

Insert data baru berhasil

← T →	level_id	level_kode	level_name	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	2025-03-03 08:17:39	NULL

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

Minggu-3 > PWL_POS > 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_name, created_at) values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         $row = DB::update('update m_level set level_name = ? 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

← → ↻ localhost/PEMROGRAMAN_WEB_LANJUT_2025/Minggu-3/PWL_POS/public/level

Update data berhasil. Jumlah data yang diupdate: 1 baris

← T →	level_id	level_kode	level_name	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	Customer	2025-03-03 08:17:39	NULL

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

```
Minggu-3 > PWL_POS > 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_name, created_at) values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         // $row = DB::update('update m_level set level_name = ? 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 }
22
```

localhost/PEMROGRAMAN_WEB_LANJUT_2025/Minggu-3/PWL_POS/public/level

Delete data berhasil. Jumlah data yang dihapus: 1 baris

	level_id	level_kode	level_name	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

```
Minggu-3 > PWL_POS > 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_name, created_at) values (?, ?, ?)', ['CUS', 'Pelanggan', now()]);
13         // return 'Insert data baru berhasil';
14
15         // $row = DB::update('update m_level set level_name = ? 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`

```

Minggu-3 > PWL_POS > resources > views > level.blade.php > html > body > table
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Data Level Pengguna</title>
8  </head>
9  <body>
10     <h1>Data Level Pengguna</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Kode Level</th>
15             <th>Nama Level</th>
16         </tr>
17         @foreach ($data as $d)
18             <tr>
19                 <td>{{ $d->level_id }}</td>
20                 <td>{{ $d->level_kode }}</td>
21                 <td>{{ $d->level_nama }}</td>
22             </tr>
23         @endforeach
24     </table>
25 </body>
26 </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.

D. QUERY BUILDER

❖ Praktikum 5

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

- PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan make:controller KategoriController

INFO Controller [C:\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

Minggu-3 > PWL_POS > 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
9  Route::get('/', function () {
10     return view('welcome');
11 });
12
13 Route::get('/level', [LevelController::class, 'index']);
14 Route::get('/kategori', [KategoriController::class, 'index']);
```

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

Minggu-3 > PWL_POS > app > Http > Controllers > KategoriController.php > ...

```
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         $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_name' => 'Snack/Makanan Ringan',
15         ];
16         DB::table('m_kategori')->insert($data);
17         return 'Insert data baru berhasil';
18     }
19 }
```

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

localhost/PWL_POS/public/kategori

Insert data baru berhasil

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	KAT001	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KAT003	Meubel	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KAT005	Perabotan Pecah Belah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	SNK	Snack/Makanan Ringan	2025-03-03 09:16:55	NULL

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

Minggu-3 > PWL_POS > 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         $row = DB::table('m_kategori')
21             ->where('kategori_kode', 'SNK')
22             ->update(['kategori_nama' => 'Camilan']);
23         return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
24     }
25 }

```

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

localhost/PEMROGRAMAN_WEB_LANJUT_2025/Minggu-3/PWL_POS/public/kategori

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	KAT001	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KAT002	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KAT003	Meubel	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KAT004	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KAT005	Perabotan Pecah Belah	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	SNK	Camilan	2025-03-03 09:16:55	NULL

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

```
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')
21         //     ->where('kategori_kode', 'SNK')
22         //     ->update(['kategori_nama' => 'Camilan']);
23         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
24
25         $row = DB::table('m_kategori')
26             ->where('kategori_kode', 'SNK')
27             ->delete();
28         return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
29     }
30 }
```

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

Minggu-3 > PWL_POS > 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         // $row = DB::table('m_kategori')
21         //     ->where('kategori_kode', 'SNK')
22         //     ->update(['kategori_nama' => 'Camilan']);
23         // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row . ' baris';
24
25         // $row = DB::table('m_kategori')
26         //     ->where('kategori_kode', 'SNK')
27         //     ->delete();
28         // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row . ' baris';
29
30         $data = DB::table('m_kategori')->get();
31         return view('kategori', ['data' => $data]);
32     }
33 }
```

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 > body > table

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Data Kategori Barang</title>
8  </head>
9  <body>
10     <h1>Data Kategori Barang</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Kode Kategori</th>
15             <th>Nama Kategori</th>
16         </tr>
17         @foreach ($data as $d)
18             <tr>
19                 <td>{{ $d->kategori_id }}</td>
20                 <td>{{ $d->kategori_kode }}</td>
21                 <td>{{ $d->kategori_nama }}</td>
22             </tr>
23         @endforeach
24     </table>
25 </body>
26 </html>
```

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

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	KAT001	Makanan
2	KAT002	Elektronik
3	KAT003	Meubel
4	KAT004	Minuman
5	KAT005	Perabotan Pecah Belah
8	SNK	Camilan

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

E. ELOQUENT ORM




❖ Praktikum 6

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

```
PS C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS> php artisan  
make:model UserModel
```

INFO Model [C:\laragon\www\PEMROGRAMAN_WEB_LANJUT_2025\Minggu-3\PWL_POS
\app\Models\UserModel.php] created 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

```
Minggu-3 > PWL_POS > 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 = 'users';  
13     protected $primaryKey = 'user_id';  
14 }  
15
```

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

Minggu-3 > PWL_POS > 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  use App\Http\Controllers\UserController;
7
8
9
10 Route::get('/', function () {
11     return view('welcome');
12 });
13
14 Route::get('/level', [LevelController::class, 'index']);
15 Route::get('/kategori', [KategoriController::class, 'index']);
16 Route::get('/user', [UserController::class, 'index']);
```

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

Minggu-3 > PWL_POS > app > Http > Controllers > UserController.php > ..

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

6. Kemudian kita buat view user.blade.php

Minggu-3 > PWL_POS > resources > views > user.blade.php > html > head

```
1 <!DOCTYPE html>
2 <html lang="en">
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->unama }}</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

Data User

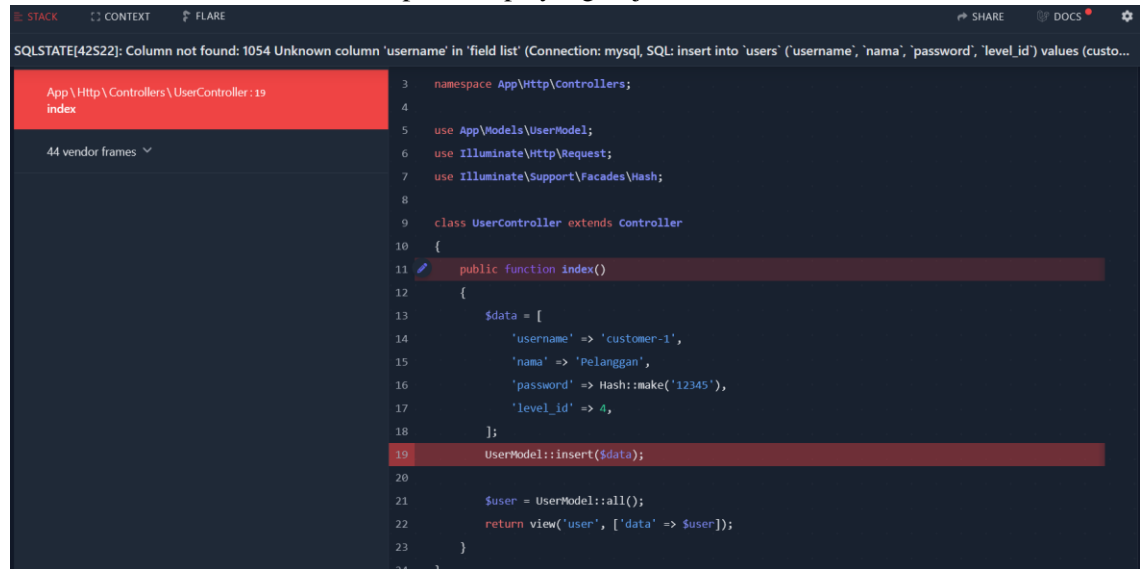
ID	Username	Nama	ID Level Pengguna
----	----------	------	-------------------

8. Setelah itu, kita modifikasi lagi file UserController

Minggu-3 > PWL_POS > 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         $data = [
14             'username' => 'customer-1',
15             'nama' => 'Pelanggan',
16             'password' => Hash::make('12345'),
17             'level_id' => 4,
18         ];
19         UserModel::insert($data);
20
21         $user = UserModel::all();
22         return view('user', ['data' => $user]);
23     }
24 }
```

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



10. Kita modifikasi lagi file UserController menjadi seperti berikut


```

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

```

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

Data User

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

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

F. PENUTUP

- Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada file setting .env Laravel?
 - APP_KEY merupakan kunci enkripsi yang digunakan untuk beberapa fungsi keamanan
- Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP_KEY?
 - Tahapan men-generate nilai untuk APP_KEY:
 - Buka terminal
 - Jalankan perintah php artisan key:generate
 - Setelah dijalankan, Laravel akan menghasilkan APP_KEY baru dan otomatis menyimpannya di file .env
- Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?
 - Secara default, Laravel memiliki 4 file migrasi yaitu:
 - create_users_table.php (untuk menyimpan data pengguna)
 - create_password_reset_tokens_table.php (untuk menyimpan token reset password)

- create_failed_jobs_table.php (menyimpan antrian pekerjaan (jobs) yang gagal dieksekusi.)
 - create_personal_access_tokens_table.php (untuk API Authentication dengan Laravel Sanctum.)
4. Secara default, file migrasi terdapat kode `$table->timestamps();`, apa tujuan/output dari fungsi tersebut?
 - `$table->timestamps();` berfungsi untuk otomatis menambahkan dua kolom ke dalam tabel database
 - created_at : Menyimpan waktu saat pertama kali data dibuat
 - updated_at: Menyimpan waktu saat terakhir data diperbarui.
 5. Pada File Migrasi, terdapat fungsi `$table->id();` Tipe data apa yang dihasilkan dari fungsi tersebut?
 - Tipe data BIGINT (unsigned big integer) yang bersifat AUTO INCREMENT dan digunakan sebagai PRIMARY KEY
 6. Apa bedanya hasil migrasi pada table m_level, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');` ?
 - `$table->id();` digunakan jika ingin nama primary key default (id)
 - `$table->id('level_id');` digunakan jika ingin mencustom nama kolom menjadi 'level_id'
 7. Pada migration, Fungsi `->unique()` digunakan untuk apa?
 - Fungsi `->unique()` digunakan untuk memastikan bahwa nilai dalam suatu kolom tidak ada duplikat di dalam tabel database.
 8. Pada Praktikum 2.2 - Tahap 2, kenapa kolom level_id pada tabel m_user menggunakan `$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 berfungsi 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 digunakan untuk mengenkripsi data agar lebih aman
 - `Hash::make('1234');` berfungsi untuk menghasilkan hash dari password '1234'
 10. Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
 - digunakan sebagai parameter binding untuk menghindari 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';` bertujuan untuk menentukan model tersebut berhubungan dengan tabel m_user
 - `protected $primaryKey = 'user_id';` bertujuan untuk mengubah primary key default (id) menjadi user_id
 12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan
 - Menurut saya operasi CRUD lebih mudah digunakan pada Query Builder karena penulisan sitaksnya hampir sama dengan penulisan query pada database itu sendiri