

**NIM** : 2241720095  
**Nama** : Ana Bellatus Mustaqfiro  
**Kelas** : D4 TI - 2H  
**No. Urut** : 04  
**Mata Kuliah** : Pemrograman Web Lanjut Jobsheet 3 (Migration)

---

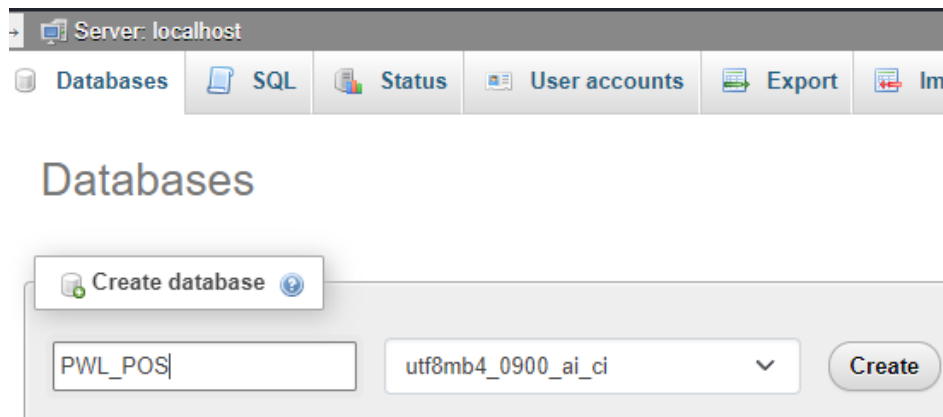
Sebelum kita masuk materi, kita buat dulu project baru yang akan kita gunakan untuk membangun aplikasi sederhana dengan topik Point of Sales (PoS), sesuai dengan Studi Kasus PWL.pdf. Jadi kita bikin project Laravel 10 dengan nama **PWL\_POS**.

```
Ana Bella > www > 105ms > composer create-project laravel/laravel="10.3.*" PWL_POS
Creating a "laravel/laravel=10.3.*" project at "./PWL_POS"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\PWL_POS
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
```

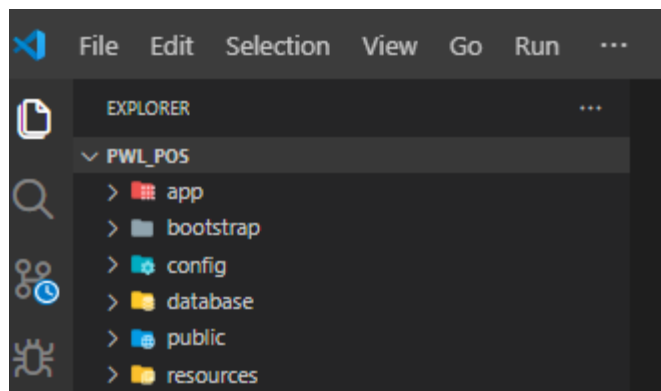
## Praktikum 1. pengaturan database

### A. Pengaturan Database

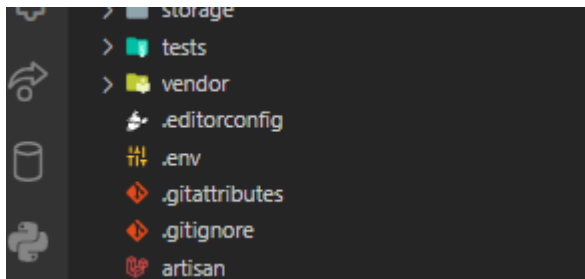
1. Buka aplikasi phpMyAdmin, dan buat database baru dengan nama PWL\_POS



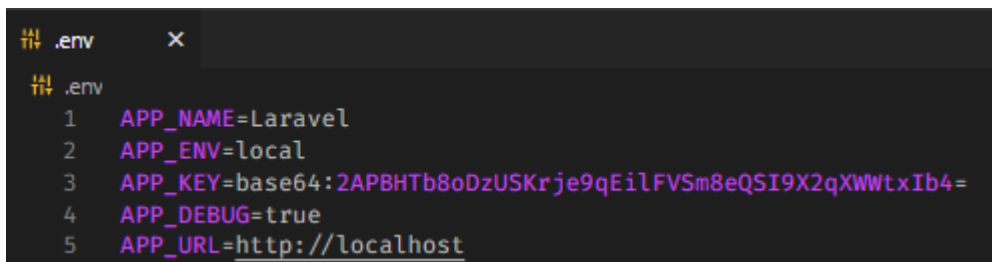
2. Buka aplikasi VSCode dan buka folder project **PWL\_POS** yang sudah kita buat



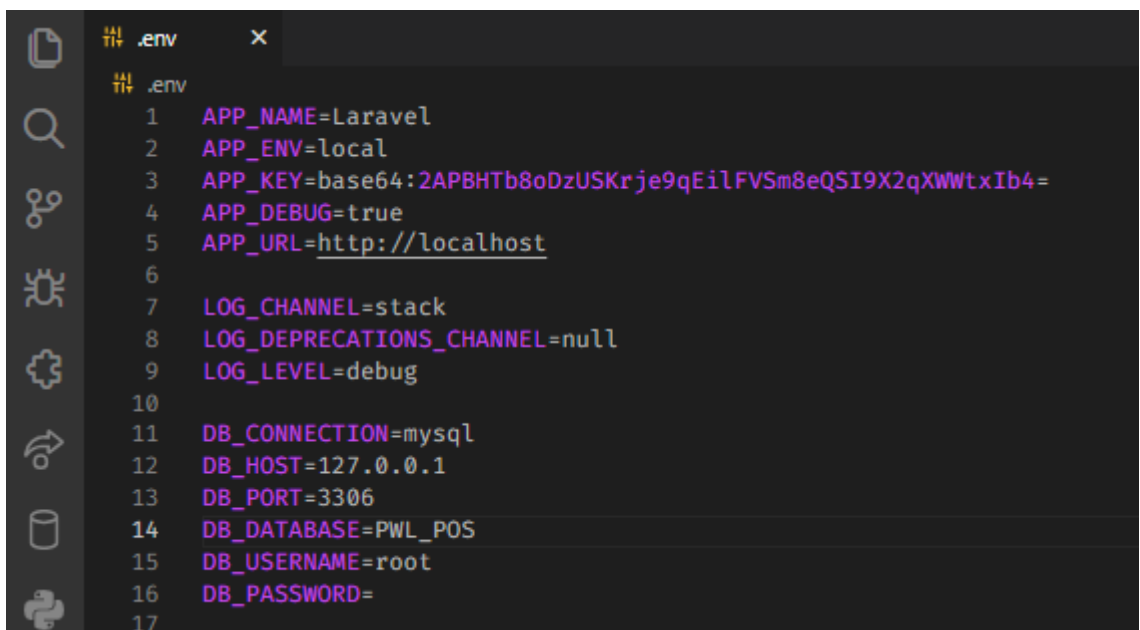
3. Copy file `.env.example` menjadi `.env`



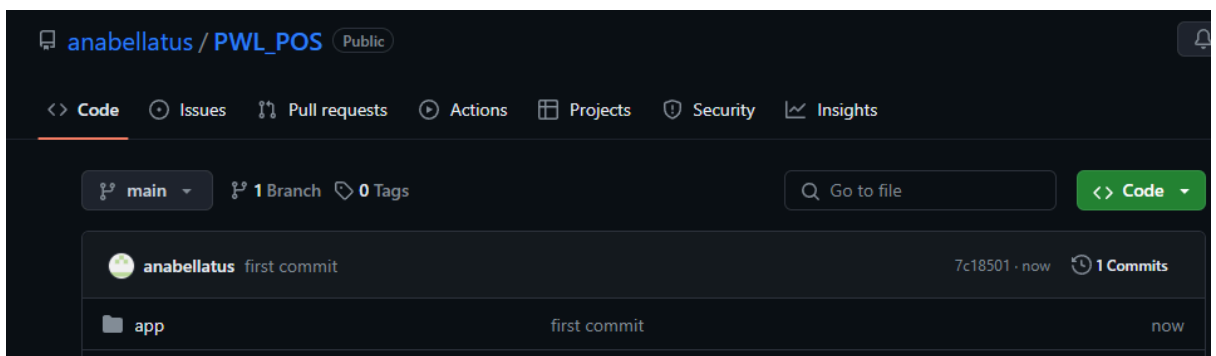
4. Buka file `.env`, dan pastikan konfigurasi `APP_KEY` bernilai. Jika belum bernilai silahkan kalian generate menggunakan `php artisan`.



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



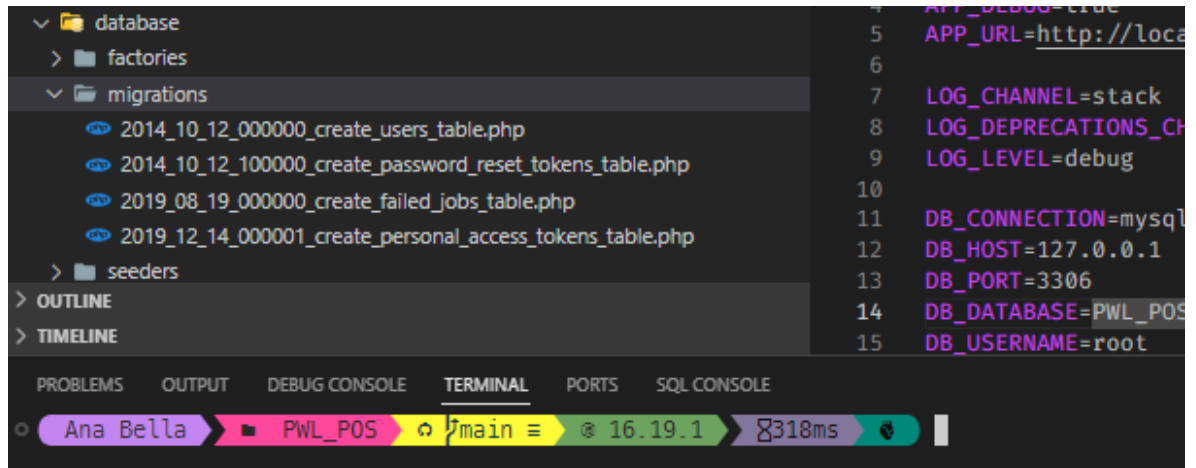
6. Laporkan hasil Praktikum-1 ini dan commit perubahan pada git.



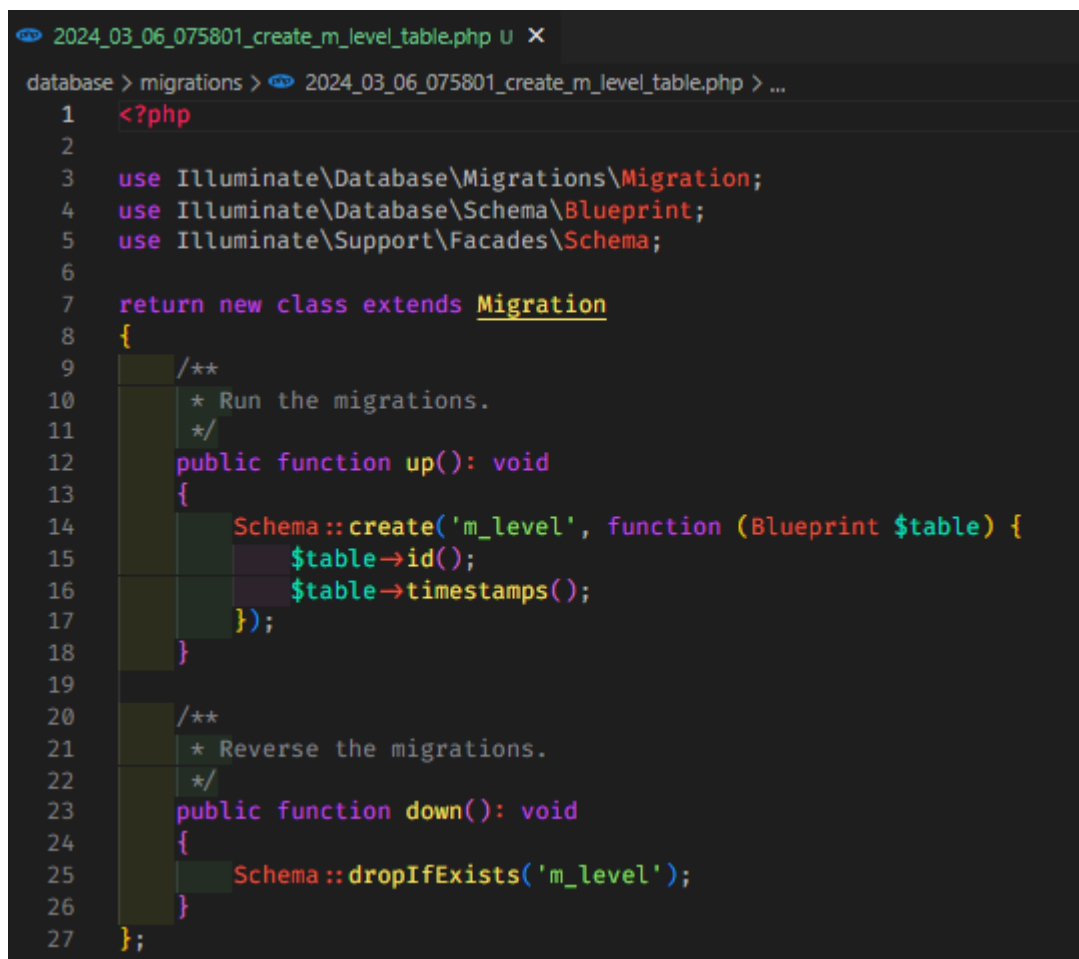
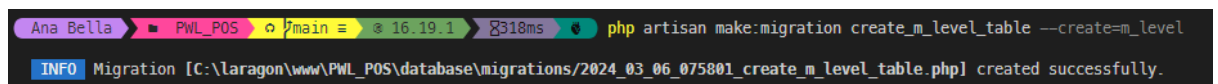
## B. Migration

### 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

```
2024_03_06_075801_create_m_level_table.php U X
database > migrations > 2024_03_06_075801_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('level_id');
16             $table->string('level_kode', 10)->unique();
17             $table->string('level_nama', 100);
18             $table->timestamps();
19         });
20     }
}
```

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

```
Ana Bella  PwL_P05  /main = 71  16.19.1  86.218s  php artisan migrate
INFO: Preparing database.
Creating migration table ..... 244ms DONE
INFO: Running migrations.
2014_10_12_000000_create_users_table ..... 277ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 43ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 154ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 120ms DONE
2024_03_06_075801_create_m_level_table ..... 83ms DONE
```

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

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

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

```
Ana Bella > PWL_POS > /main = 71 > 16.19.1 > 82.966s > php artisan make:migration create_m_kategori_table --create=m_kategori
INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_03_06_083137_create_m_kategori_table.php] created successfully.
```

```
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('m_kategori', function (Blueprint $table) {
15             $table->id('kategori_id');
16             $table->string('kategori_kode', 10)->unique();
17             $table->string('kategori_nama', 100);
18             $table->timestamps();
19         });
20     }
21 }
```

```
Ana Bella > PWL_POS > /main = 72 > 16.19.1 > 82.821s > php artisan migrate
INFO Running migrations.
2024_03_06_083137_create_m_kategori_table ..... 429ms DONE
```

	Table	Action
<input type="checkbox"/>	failed_jobs	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/>	migrations	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/>	m_kategori	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/>	m_level	★ Browse Structure Search Insert Empty Drop

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

anabellatus / PWL\_POS Public

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

Go to file

Code

anabellatus Add migrations for m\_level and m\_kategori tables 1f4e48b · 1 minute ago 1 Commits

## Praktikum 2.2 Pembuatan file migrasi dengan relasi

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

```
Ana Bella > PWL_POS > /main = 72 > 16.19.1 > 82.066s > php artisan make:migration create_m_user_table --table=m_user
INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_03_06_084958_create_m_user_table.php] created successfully.
```

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

```
2024_03_06_084958_create_m_user_table.php U X
database > migrations > 2024_03_06_084958_create_m_user_table.php > class > down
7   return new class extends Migration
8   {
9       /**
10        * Run the migrations.
11        */
12        public function up(): void
13        {
14            Schema::table('m_user', function (Blueprint $table) {
15                $table->id('user_id');
16                $table->unsignedBigInteger('level_id')->index();
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::dropIfExists('m_user');
32        }
33    };
```

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

Ana Bella | PWL\_POS | main = 71 | 16.19.1 | 80ms | php artisan migrate

INFO Running migrations.

2024\_03\_06\_084958\_create\_m\_user\_table ..... 544ms DONE

<input type="checkbox"/>	failed_jobs	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	migrations	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	m_kategori	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	m_level	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	m_user	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	password_reset_tokens	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	personal_access_tokens	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop
<input type="checkbox"/>	users	★	📄 Browse	📊 Structure	🔍 Search	➕ Insert	🖨 Empty	🚫 Drop

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

m_barang
t_penjualan
t_stok
t_penjualan_detail

```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_barang_table --table=m_barang
```

**INFO** Migration [C:\laragon\www\PWL\_POS\database\Migrations\2024\_03\_12\_202840\_create\_m\_barang\_table.php] created successfully.

```
return new class extends Migration
{
    public function up(): void
    {
        Schema::create('m_barang', function (Blueprint $table) {
            $table->id('barang_id');
            $table->unsignedBigInteger('kategori_id')->index();
            $table->string('barang_kode', 10)->unique();
            $table->string('barang_nama', 100);
            $table->integer('harga_beli');
            $table->integer('harga_jual');
            $table->timestamps();

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

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

```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_t_penjualan_table --table=t_penjualan
```

**INFO** Migration [C:\laragon\www\PWL\_POS\database\Migrations\2024\_03\_12\_205158\_create\_t\_penjualan\_table.php] created successfully.

```

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('t_penjualan', function (Blueprint $table) {
            $table->id('penjualan_id');
            $table->unsignedBigInteger('user_id')->index();
            $table->string('pembeli', 50);
            $table->string('penjualan_kode', 20);
            $table->dateTime('penjualan_tanggal');
            $table->timestamps();

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

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

```

PS C:\laragon\www\PNL\_POS> php artisan make:migration create\_t\_stok\_table --table=t\_stok

**INFO** Migration [C:\laragon\www\PNL\_POS\database\migrations\2024\_03\_12\_205908\_create\_t\_stok\_table.php] created successfully.

```

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('t_stok', function (Blueprint $table) {
            $table->id('stok_id');
            $table->unsignedBigInteger('barang_id')->index();
            $table->unsignedBigInteger('user_id')->index();
            $table->dateTime('stok_tanggal');
            $table->integer('stok_jumlah');
            $table->timestamps();

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

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

```



```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_t_penjualan_detail_table --table=t_penjualan_detail
```

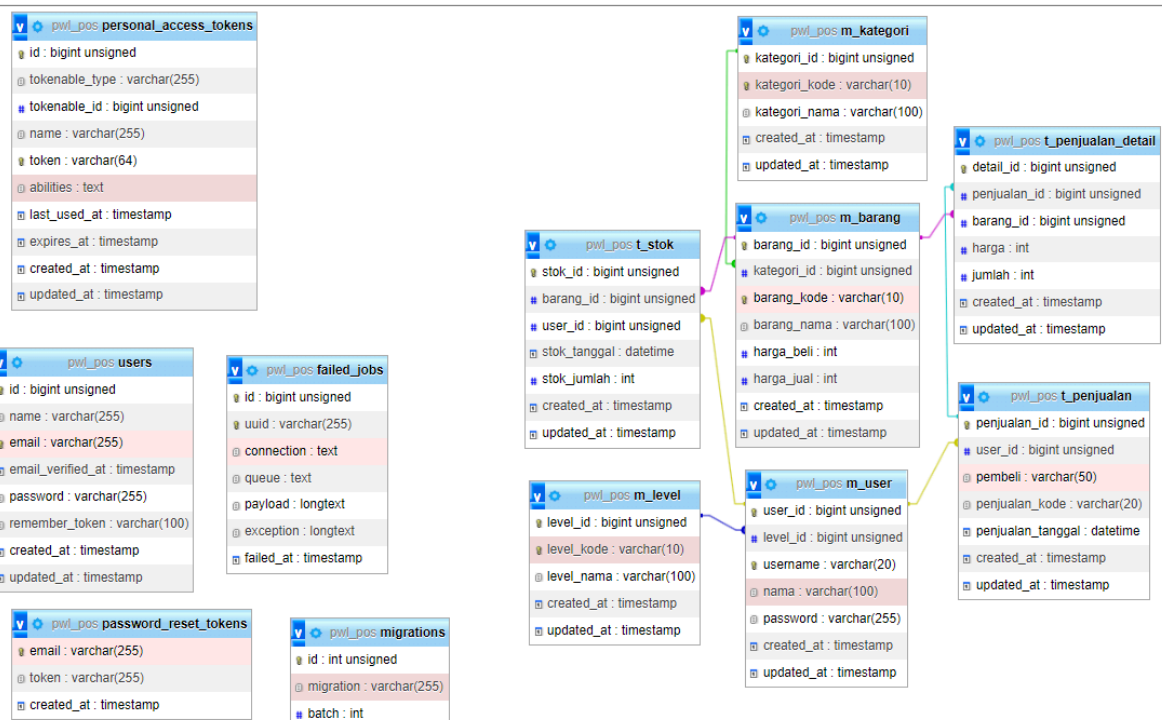
INFO Migration [C:\laragon\www\PWL\_POS\database\migrations\2024\_03\_12\_210514\_create\_t\_penjualan\_detail\_table.php] created successfully.

```
return new class extends Migration
{
    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();

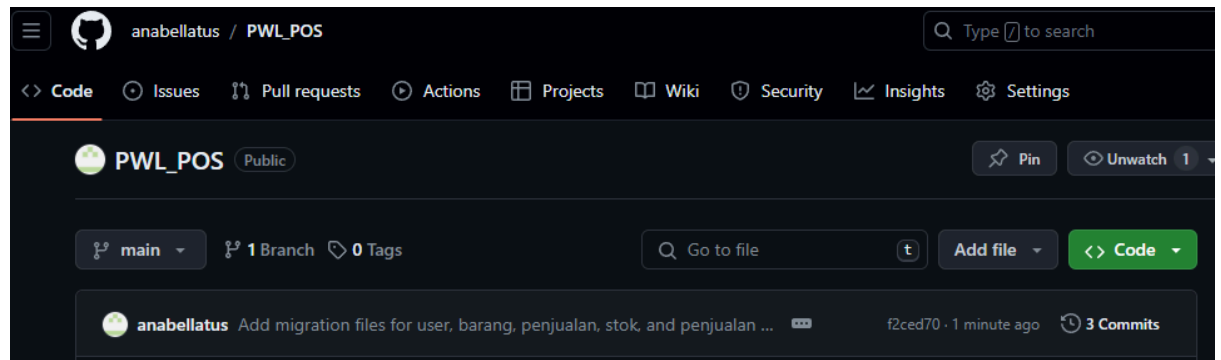
            $table->foreign('penjualan_id')->references('penjualan_id')->on('t_penjualan');
            $table->foreign('barang_id')->references('barang_id')->on('m_barang');
        });
    }

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

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



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



### C. Seeder

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

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

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

```
LevelSeeder.php U X
database > seeders > LevelSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7
8  class LevelSeeder extends Seeder
9  {
10     /**
11      * Run the database seeds.
12      */
13     public function run(): void
14     {
15         //
16     }
17 }
```

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

```

LevelSeeder.php U X
database > seeders > LevelSeeder.php > ...

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     }

```

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

```

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

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

```

6. Modifikasi file class UserSeeder seperti berikut

```
UserSeeder.php U X
database > seeders > UserSeeder.php > UserSeeder > run

10 class UserSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      */
15     public function run(): void
16     {
17         $data = [
18             [
19                 'user_id' => 1,
20                 'level_id' => 1,
21                 'username' => 'admin',
22                 'nama' => 'Administrator',
23                 'password' => Hash::make('12345'),
24             ],
25             [
26                 'user_id' => 2,
27                 'level_id' => 2,
28                 'username' => 'manager',
29                 'nama' => 'Manager',
30                 'password' => Hash::make('12345'),
31             ],
32             [
33                 'user_id' => 3,
34                 'level_id' => 3,
35                 'username' => 'staff',
36                 'nama' => 'Staff/Kasir',
37                 'password' => Hash::make('12345'),
38             ],
39         ];
40         DB::table('m_user')->insert($data);
```

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```
PS C:\laragon\www\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 Copy Delete	1	1	admin	Administrator	\$2y\$12\$6G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/M/...
<input type="checkbox"/> Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$qYFZLkc3OVBM0en63p0.tRs0uzIQN1KgZB6C.NsXt...
<input type="checkbox"/> Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$87W6e6ZhT1WaulXGzRS0juEnd8WG44OCX0o9evRGkD3...
↑ <input type="checkbox"/> Check all With selected: Edit Copy Delete Export					

9. Ok, data seeder berhasil dimasukkan 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

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

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

```
class KategoriSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            ['kategori_id' => 1, 'kategori_kode' => 'MKN', 'kategori_nama' => 'Makanan'],
            ['kategori_id' => 2, 'kategori_kode' => 'MNM', 'kategori_nama' => 'Minuman'],
            ['kategori_id' => 3, 'kategori_kode' => 'PKN', 'kategori_nama' => 'PAKAIAN'],
            ['kategori_id' => 4, 'kategori_kode' => 'OBT', 'kategori_nama' => 'OBAT'],
            ['kategori_id' => 5, 'kategori_kode' => 'ELK', 'kategori_nama' => 'ELEKTRONIK'],
        ];

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

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

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

```
database > seeders > BarangSeeder.php > BarangSeeder
9 class BarangSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [
17             ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'MKN001', 'barang_nama' => 'Sari Roti', 'harga_beli' => 3000, 'harga_jual' => 5000],
18             ['barang_id' => 2, 'kategori_id' => 1, 'barang_kode' => 'MKN002', 'barang_nama' => 'Indomie', 'harga_beli' => 4000, 'harga_jual' => 6000],
19             ['barang_id' => 3, 'kategori_id' => 2, 'barang_kode' => 'MNM001', 'barang_nama' => 'Aqua', 'harga_beli' => 3000, 'harga_jual' => 4000],
20             ['barang_id' => 4, 'kategori_id' => 2, 'barang_kode' => 'MNM002', 'barang_nama' => 'Teh Botol', 'harga_beli' => 4500, 'harga_jual' => 7000],
21             ['barang_id' => 5, 'kategori_id' => 3, 'barang_kode' => 'PKN001', 'barang_nama' => 'Kaos Oblong', 'harga_beli' => 35000, 'harga_jual' => 55000],
22             ['barang_id' => 6, 'kategori_id' => 3, 'barang_kode' => 'PKN002', 'barang_nama' => 'Celana Jeans', 'harga_beli' => 50000, 'harga_jual' => 70000],
23             ['barang_id' => 7, 'kategori_id' => 4, 'barang_kode' => 'OBT001', 'barang_nama' => 'Paracetamol', 'harga_beli' => 8000, 'harga_jual' => 10000],
24             ['barang_id' => 8, 'kategori_id' => 4, 'barang_kode' => 'OBT002', 'barang_nama' => 'Antangin', 'harga_beli' => 4000, 'harga_jual' => 5000],
25             ['barang_id' => 9, 'kategori_id' => 5, 'barang_kode' => 'ELK001', 'barang_nama' => 'Kulkas', 'harga_beli' => 1000000, 'harga_jual' => 1500000],
26             ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'ELK002', 'barang_nama' => 'Laptop', 'harga_beli' => 4000000, 'harga_jual' => 5000000],
27         ];
28         DB::table('m_barang')->insert($data);
29     }
}
```

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

**INFO** Seeder [C:\laragon\www\PWL\_POS\database\seeders\StokSeeder.php] created successfully.

```
database > seeders > StokSeeder.php > StokSeeder > run
```

```
9 class StokSeeder extends Seeder
12     /**
13      * Run the database seeds.
14      */
15     public function run(): void
16     {
17         $data = [
18             ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 20],
19             ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 50],
20             ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 100],
21             ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 100],
22             ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 30],
23             ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 20],
24             ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 50],
25             ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 50],
26             ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 5],
27             ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 1, 'stok_tanggal' => '2024-03-06', 'stok_jumlah' => 10],
28         ];
29         DB::table('t_stok')->insert($data);
```

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

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

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

```
9 class PenjualanSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [
17             ['penjualan_id' => 1, 'user_id' => 3, 'pembeli' => 'Ana', 'penjualan_kode' => 'PJL001', 'penjualan_tanggal' => '2024-03-06'],
18             ['penjualan_id' => 2, 'user_id' => 3, 'pembeli' => 'Bella', 'penjualan_kode' => 'PJL002', 'penjualan_tanggal' => '2024-03-06'],
19             ['penjualan_id' => 3, 'user_id' => 3, 'pembeli' => 'Melati', 'penjualan_kode' => 'PJL003', 'penjualan_tanggal' => '2024-03-07'],
20             ['penjualan_id' => 4, 'user_id' => 3, 'pembeli' => 'Jihan', 'penjualan_kode' => 'PJL004', 'penjualan_tanggal' => '2024-03-07'],
21             ['penjualan_id' => 5, 'user_id' => 3, 'pembeli' => 'Dea', 'penjualan_kode' => 'PJL005', 'penjualan_tanggal' => '2024-03-08'],
22             ['penjualan_id' => 6, 'user_id' => 3, 'pembeli' => 'Fanessa', 'penjualan_kode' => 'PJL006', 'penjualan_tanggal' => '2024-03-08'],
23             ['penjualan_id' => 7, 'user_id' => 3, 'pembeli' => 'Elva', 'penjualan_kode' => 'PJL007', 'penjualan_tanggal' => '2024-03-08'],
24             ['penjualan_id' => 8, 'user_id' => 3, 'pembeli' => 'Nadilla', 'penjualan_kode' => 'PJL008', 'penjualan_tanggal' => '2024-03-09'],
25             ['penjualan_id' => 9, 'user_id' => 3, 'pembeli' => 'Putri', 'penjualan_kode' => 'PJL009', 'penjualan_tanggal' => '2024-03-09'],
26             ['penjualan_id' => 10, 'user_id' => 3, 'pembeli' => 'Octa', 'penjualan_kode' => 'PJL010', 'penjualan_tanggal' => '2024-03-10'],
27         ];
28         DB::table('t_penjualan')->insert($data);
```

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

**INFO** Seeder [C:\laragon\www\PWL\_POS\database\seeders\PenjualanDetailSeeder.php] created successfully.

```

database > seeders > PenjualanDetailSeeder.php > PenjualanDetailSeeder > run
9  class PenjualanDetailSeeder extends Seeder
13  */
14  public function run(): void
15  {
16      $data = [
17          ['detail_id' => 1, 'penjualan_id' => 1, 'barang_id' => 1, 'harga' => 5000, 'jumlah' => 3],
18          ['detail_id' => 2, 'penjualan_id' => 1, 'barang_id' => 2, 'harga' => 6000, 'jumlah' => 2],
19          ['detail_id' => 3, 'penjualan_id' => 1, 'barang_id' => 3, 'harga' => 4000, 'jumlah' => 5],
20          ['detail_id' => 4, 'penjualan_id' => 2, 'barang_id' => 4, 'harga' => 7000, 'jumlah' => 3],
21          ['detail_id' => 5, 'penjualan_id' => 2, 'barang_id' => 5, 'harga' => 55000, 'jumlah' => 2],
22          ['detail_id' => 6, 'penjualan_id' => 2, 'barang_id' => 6, 'harga' => 70000, 'jumlah' => 1],
23          ['detail_id' => 7, 'penjualan_id' => 3, 'barang_id' => 7, 'harga' => 10000, 'jumlah' => 4],
24          ['detail_id' => 8, 'penjualan_id' => 3, 'barang_id' => 8, 'harga' => 5000, 'jumlah' => 3],
25          ['detail_id' => 9, 'penjualan_id' => 3, 'barang_id' => 9, 'harga' => 1500000, 'jumlah' => 1],
26          ['detail_id' => 10, 'penjualan_id' => 4, 'barang_id' => 10, 'harga' => 5000000, 'jumlah' => 1],
27          ['detail_id' => 11, 'penjualan_id' => 4, 'barang_id' => 1, 'harga' => 5000, 'jumlah' => 2],
28          ['detail_id' => 12, 'penjualan_id' => 4, 'barang_id' => 2, 'harga' => 6000, 'jumlah' => 3],
29          ['detail_id' => 13, 'penjualan_id' => 5, 'barang_id' => 3, 'harga' => 4000, 'jumlah' => 4],
30          ['detail_id' => 14, 'penjualan_id' => 5, 'barang_id' => 4, 'harga' => 7000, 'jumlah' => 2],
31          ['detail_id' => 15, 'penjualan_id' => 5, 'barang_id' => 5, 'harga' => 55000, 'jumlah' => 1],
32          ['detail_id' => 16, 'penjualan_id' => 6, 'barang_id' => 6, 'harga' => 70000, 'jumlah' => 3],
33          ['detail_id' => 17, 'penjualan_id' => 6, 'barang_id' => 7, 'harga' => 10000, 'jumlah' => 2],
34          ['detail_id' => 18, 'penjualan_id' => 6, 'barang_id' => 8, 'harga' => 5000, 'jumlah' => 1],
35          ['detail_id' => 19, 'penjualan_id' => 7, 'barang_id' => 9, 'harga' => 1500000, 'jumlah' => 2],
36          ['detail_id' => 20, 'penjualan_id' => 7, 'barang_id' => 10, 'harga' => 5000000, 'jumlah' => 1],
37          ['detail_id' => 21, 'penjualan_id' => 7, 'barang_id' => 1, 'harga' => 5000, 'jumlah' => 3],
38          ['detail_id' => 22, 'penjualan_id' => 8, 'barang_id' => 2, 'harga' => 6000, 'jumlah' => 4],
39          ['detail_id' => 23, 'penjualan_id' => 8, 'barang_id' => 3, 'harga' => 4000, 'jumlah' => 3],
40          ['detail_id' => 24, 'penjualan_id' => 8, 'barang_id' => 4, 'harga' => 7000, 'jumlah' => 2],
41          ['detail_id' => 25, 'penjualan_id' => 9, 'barang_id' => 5, 'harga' => 55000, 'jumlah' => 1],
42          ['detail_id' => 26, 'penjualan_id' => 9, 'barang_id' => 6, 'harga' => 70000, 'jumlah' => 2],
43          ['detail_id' => 27, 'penjualan_id' => 9, 'barang_id' => 7, 'harga' => 10000, 'jumlah' => 3],
44          ['detail_id' => 28, 'penjualan_id' => 10, 'barang_id' => 8, 'harga' => 5000, 'jumlah' => 4],
45          ['detail_id' => 29, 'penjualan_id' => 10, 'barang_id' => 9, 'harga' => 1500000, 'jumlah' => 2],
46          ['detail_id' => 30, 'penjualan_id' => 10, 'barang_id' => 10, 'harga' => 5000000, 'jumlah' => 1],
47      ];
48      DB::table('t_penjualan_detail')->insert($data);
49  }
50

```

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

```
INFO Seeding database.
```

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

```
INFO Seeding database.
```

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

```
INFO Seeding database.
```

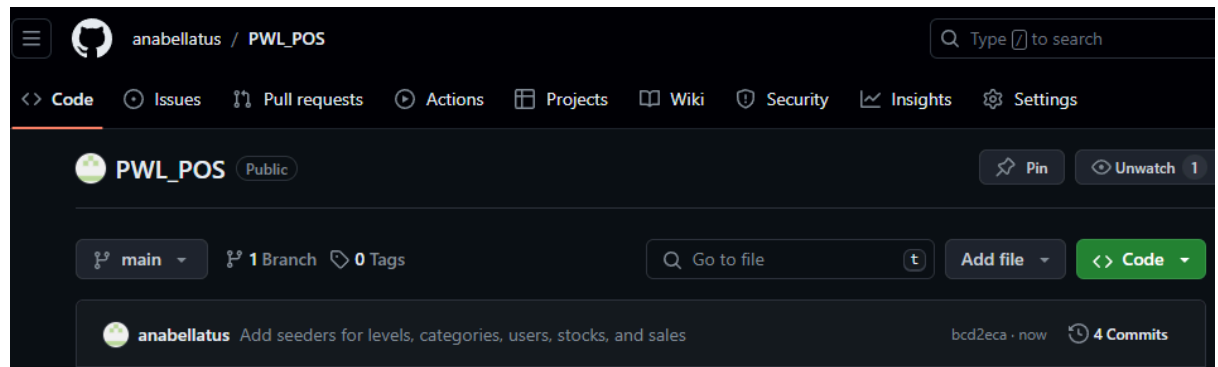
```
PS C:\laragon\www\PWL_POS> php artisan db:seed --class=PenjualanSeeder
```

```
INFO Seeding database.
```

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

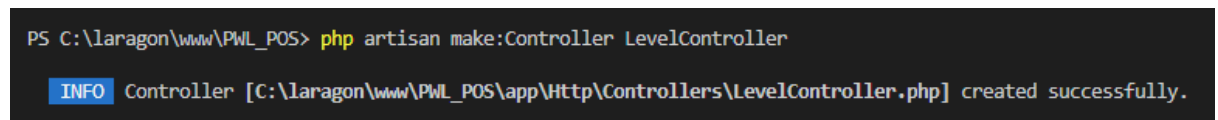
```
INFO Seeding database.
```

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

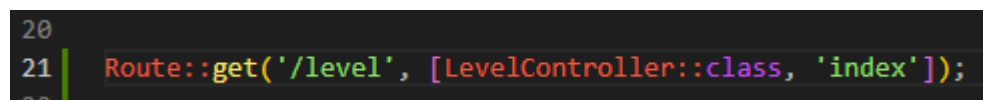


#### D. DB Facade

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



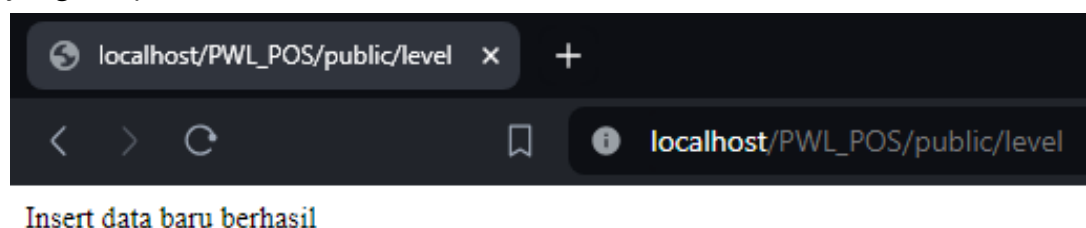
2. Kita modifikasi dulu untuk routing-nya, ada di PWL\_POS/routes/web.php















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



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





<div><div><div>←</div><div>T</div><div>→</div></div></div>				level_id	level_kode	level_nama	created_at
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	ADM	Administrator	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	MNG	Manager	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	STF	Staff/Kasir	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	CUS	Pelanggan	2024-03-12 23:39:23

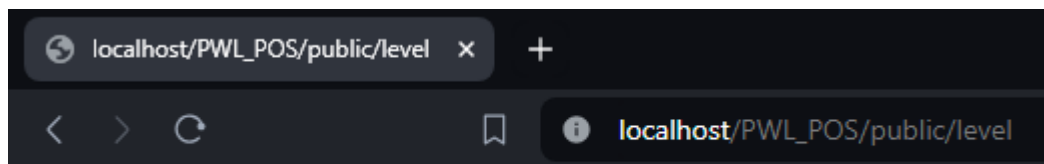
5. Selanjutnya, kita modifikasi lagi file LevelController untuk mengupdate data di table m\_level seperti berikut

```

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

```

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

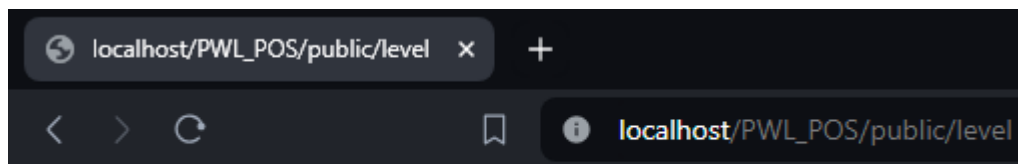


Update data berhasil. Jumlah data yang diupdate: 1 baris

				level_id	level_kode	level_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	ADM	Administrator	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNG	Manager	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	STF	Staff/Kasir	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	CUS	Customer	2024-03-12 23:39:23

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

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



Delete data berhasil. Jumlah data yang dihapus: 1 baris

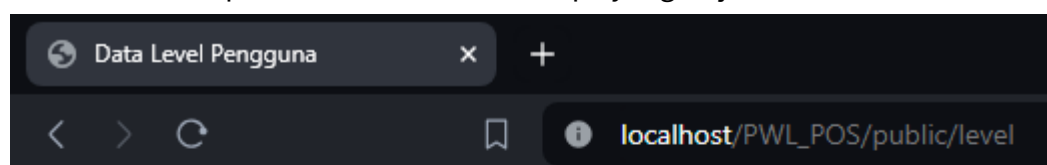
8. Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m\_level. Kita modifikasi file LevelController seperti berikut

```
LevelController.php U X
app > Http > Controllers > LevelController.php > LevelController > index
8  class LevelController extends Controller
10     public function index()
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

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

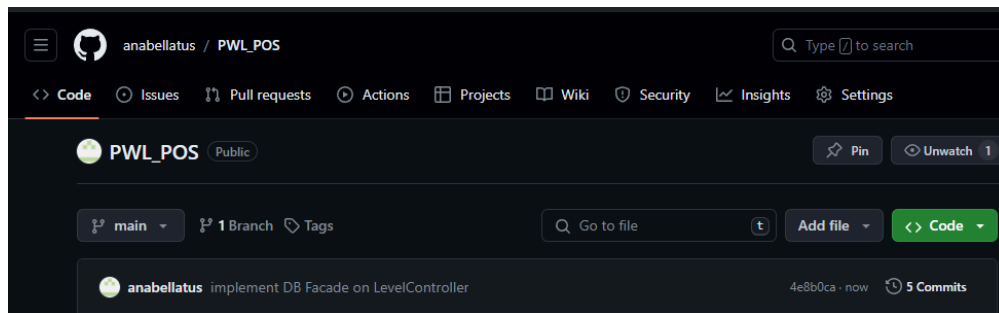
10. Silahkan dicoba pada browser dan amati apa yang terjadi



## 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

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

```
PS C:\laragon\www\PWL_POS> php artisan make:controller KategoriController  
  
INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.
```

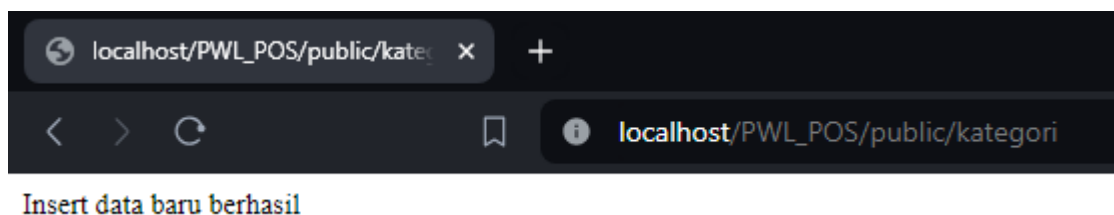
2. Kita modifikasi dulu untuk routing-nya, ada di PWL\_POS/routes/web.php

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

3. Selanjutnya, kita modifikasi file KategoriController untuk menambahkan 1 data ke table m\_kategori

```
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     }  
}
```

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



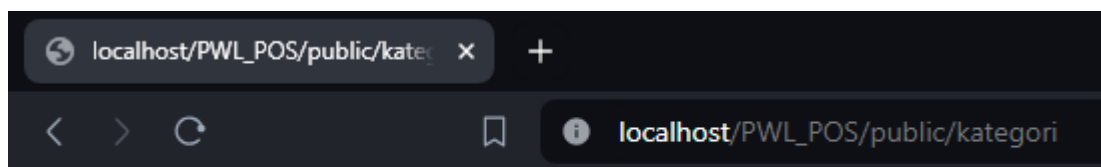
				kategori_id	kategori_kode	kategori_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	MKN	Makanan	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNM	Minuman	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	PKN	PAKAIAN	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	OBT	OBAT	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	ELK	ELEKTRONIK	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Snack/Makanan Ringan	2024-03-13 00:41:30

5. Selanjutnya, kita modifikasi lagi file KategoriController untuk mengupdate 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';

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

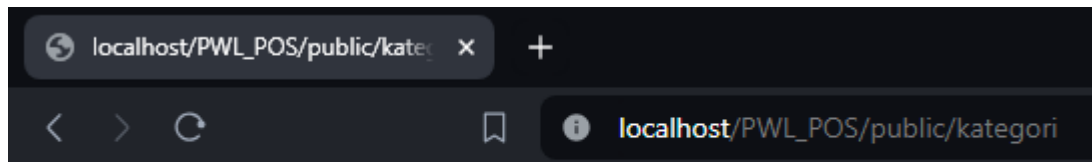


Update data berhasil. Jumlah data yang diupdate: 1 baris

				kategori_id	kategori_kode	kategori_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	MKN	Makanan	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNM	Minuman	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	PKN	PAKAIAN	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	OBT	OBAT	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	ELK	ELEKTRONIK	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Camilan	2024-03-13 00:41:30

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

```
KategoriController.php U X
app > Http > Controllers > KategoriController.php > ...
8   class KategoriController extends Controller
10      public function index()
23      {
24          $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
25          return 'Delete data berhasil. Jumlah data yang dihapus: '.$row.' baris';
26      }
27  }
```



				kategori_id	kategori_kode	kategori_nama	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	MKN	Makanan	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	MNM	Minuman	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	PKN	PAKAIAN	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	OBT	OBAT	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	ELK	ELEKTRONIK	NULL

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

```
KategoriController.php U X
app > Http > Controllers > KategoriController.php > KategoriController > index
8   class KategoriController extends Controller
10      public function index()
26      {
27          $data = DB::table('m_kategori')->get();
28          return view('kategori', ['data' => $data]);
29      }
30  }
```

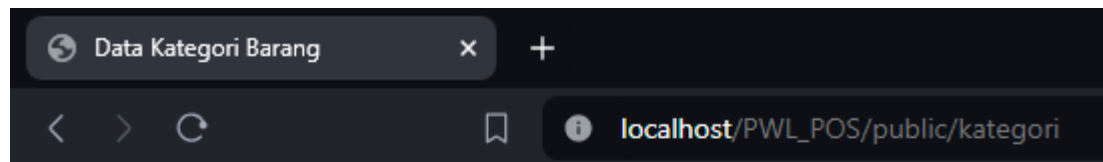
9. Coba kita perhatikan kode yang diberi tanda kotak merah, berhubung kode tersebut memanggil view('kategori'), maka kita buat file view pada VSCode di PWL\_POS/resources/view/kategori.blade.php

```

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

```

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



## Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	MKN	Makanan
2	MNM	Minuman
3	PNM	PAKAIAN
4	OBT	OBAT
5	ELK	ELEKTRONIK

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

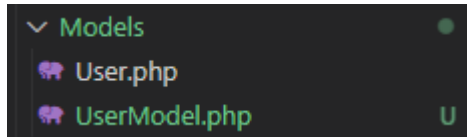
## F. Eloquent ORM

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

```
PS C:\laragon\www\PWL_POS> php artisan make:model UserModel
```

```
INFO Model [C:\laragon\www\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

```
UserModel.php U x
app > Models > UserModel.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user';
13     protected $primaryKey = 'user_id';
```

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

```
25 | Route::get('/user', [UserController::class, 'index']);
```

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

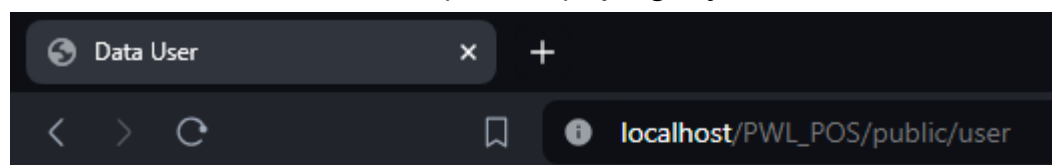
```
UserController.php U x
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function index()
11     {
12         $user = UserModel::all();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



6. Kemudian kita buat view user.blade.php

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

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



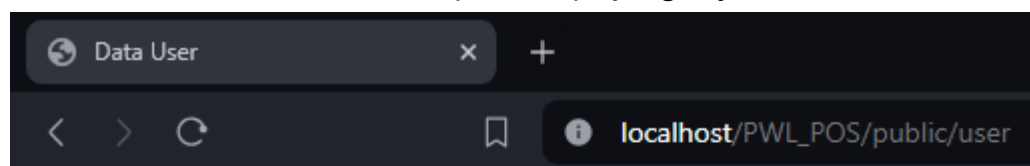
## Data User

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

8. Setelah itu, kita modifikasi lagi file UserController

```
UserController.php U X
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
14         $data = [
15             'username' => 'customer-1',
16             'nama' => 'Pelanggan',
17             'password' => Hash::make('12345'),
18             'level_id' => 4
19         ];
20
21         UserModel::insert($data);
22
23         $user = UserModel::all();
24         return view('user', ['data' => $user]);
25     }
26 }
```

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



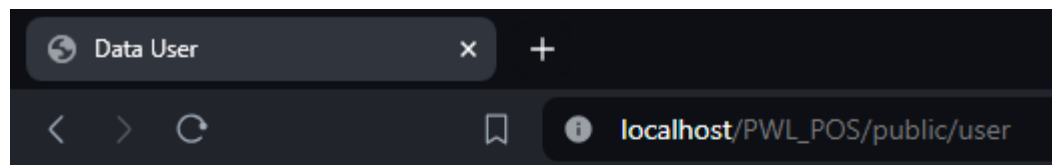
## Data User

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

10. Kita modifikasi lagi file UserController menjadi seperti berikut

```
UserController.php U X
app > Http > Controllers > UserController.php > UserController > index
9  class UserController extends Controller
11     public function index()
20
21         $data = [
22             'nama' => 'Pelanggan Pertama'
23         ];
24
25         UserModel::where('username', 'customer-1')->update($data);
26
27         $user = UserModel::all();
28         return view('user', ['data' => $user]);
29     }
30 }
```

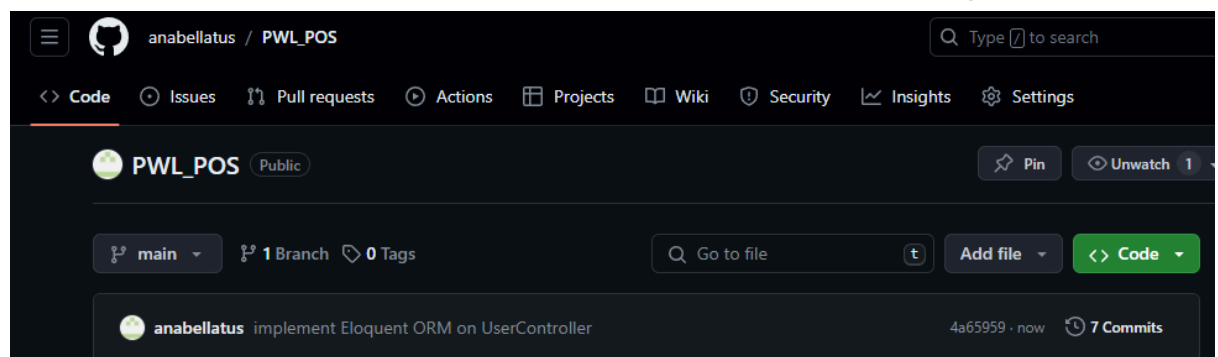
11. Jalankan di browser, amati dan laporkan apa yang terjadi



## Data User

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

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?  
→ fungsi APP\_KEY pada laravel adalah digunakan untuk enkripsi data yang tersimpan di dalam program laravel, sehingga meningkatkan keamanan program.
2. Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP\_KEY?  
→ menggunakan perintah `php artisan`
3. Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?  
→ iya, file migrasi tersebut gunanya sebagai kerangka kerja untuk membuat struktur database yang digunakan dalam laravel.
4. Secara default, file migrasi terdapat kode `$table->timestamps();`, apa tujuan/output dari fungsi tersebut?  
→ tujuan atau output dari `timestamps()` adalah untuk menambahkan kolom ``created at`` dan ``updated at`` secara otomatis, dengan format `datetime`. Ketika data dibuat/diubah laravel secara otomatis akan mencatatnya, sehingga tidak perlu dilakukan secara manual.
5. Pada File Migrasi, terdapat fungsi `$table->id();` Tipe data apa yang dihasilkan dari fungsi tersebut?  
→ tipe data yang dihasilkan dari fungsi tersebut adalah `Unsigned BigInt AUTO_INCREMENT Primary Key`
6. Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id();` dengan menggunakan `$table->id('level_id');`?  
→ jika menggunakan **`$table->id();`** maka laravel secara otomatis akan membuat primary key dengan nama kolom `'id'`, namun jika menggunakan **`$table->('level_id');`** laravel akan membuat kolom primary key dengan nama `'level_id'`.
7. Pada migration, Fungsi `->unique()` digunakan untuk apa?  
→ fungsi tersebut digunakan untuk menetapkan bahwa nilai kolom tersebut harus unik di seluruh baris dalam tabel, sehingga dapat mencegah duplikasi data.

8. Pada Praktikum 2.2 - Tahap 2, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$tabel->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$tabel->id('level_id')` ?  
→ karena pada tabel `m_user` `level_id` digunakan sebagai foreign key, sedangkan pada tabel `m_level` `level_id` digunakan sebagai primary key dimana untuk set primary key menggunakan `$tabel->('nama_id')`;
9. Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program `Hash::make('1234')` ;?  
→ tujuannya adalah untuk membuat enkripsi nilai dari '1234'
10. Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?  
→ digunakan sebagai placeholder dari parameter yang akan diinputkan
11. Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode `protected $table = 'm_user'`; dan `protected $primaryKey = 'user_id'` ; ?  
→ untuk mendefinisikan nama tabel dan primary key
12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan  
→ menurut saya lebih mudah menggunakan Query Builder karena tidak perlu menuliskan query nya.