KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
https://jti.polinema.ac.id

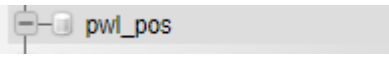
Mata Kuliah	:	Pemrograman Web Lanjut (PWL)
Program Studi	:	D4 – Sistem Informasi Bisnis
Semester	:	5

Kelas	:	SIB
NIM	:	2241760097
Nama	:	Ahmad Iqbal Firmansyah
Jobsheet Ke-	:	3

Link Github : https://github.com/ahmadbale/PWL_POS

Laporan Jobsheet

Praktikum Ke-1

Langkah	Jawaban/Deskripsi
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

	.env APP_NAME=Laravel APP_ENV=local APP_KEY=base64:YUdQ/z//BYSfqnCEEz4LM1SKY5GvDNSbxM7B2K7Qlog= APP_DEBUG=true APP_URL=http://localhost LOG_CHANNEL=stack LOG_DEPRECATIONS_CHANNEL=null LOG_LEVEL=debug DB_CONNECTION=mysql DB_HOST=127.0.0.1 DB_PORT=3306 DB_DATABASE=PWL_POS DB_USERNAME=root DB_PASSWORD= BROADCAST_DRIVER=log CACHE_DRIVER=file FILESYSTEM_DISK=local QUEUE_CONNECTION=sync SESSION_DRIVER=file SESSION_LIFETIME=120
5.	Edit file .env dan sesuaikan dengan database yang telah dibuat
6.	Laporkan hasil Praktikum-1 ini dan commit perubahan pada git.

Praktikum Ke-2.1

Langkah	Jawaban/Deskripsi
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 PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_level_table --create=m_level INFO Migration [C:\laragon\www\PWL_POS\database\migrations/2024_09_13_001506_create_m_level_table.php] created successfully.
4.	Kita perhatikan bagian yang di kotak merah, bagian tersebut yang akan kita modifikasi sesuai desain database yang sudah ada

```php
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();
    });
}
```

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

```
PS C:\laragon\www\PWL_POS> php artisan migrate

  INFO  Preparing database.

  Creating migration table ........................ 143ms DONE

  INFO  Running migrations.

  2014_10_12_000000_create_users_table ........... 298ms DONE
  2014_10_12_100000_create_password_reset_tokens_table  50ms DO
NE
  2019_08_19_000000_create_failed_jobs_table ..... 127ms DONE
  2019_12_14_000001_create_personal_access_tokens_table  192ms
DONE
  2024_09_13_001506_create_m_level_table ......... 220ms DONE
```

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

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
6 tables	Sum	5	InnoDB	utf8mb4_unicode_ci	96.0 KiB	0 B

7. Ok, table sudah dibuat di database

8. Buat table database dengan migration untuk table m_kategori dan m_supplier yang sama-sama tidak memiliki foreign key

```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_kategori_table

  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_09_13_002308_create_m_kategori_table.php] created
successfully.
```

```
PS C:\laragon\www\PWL_POS> php artisan make:migration create_m_supplier_table

  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_09_13_002355_create_m_supplier_table.php] created
successfully.
```

```
C:\laragon\www\PWL_POS> php artisan migrate

  INFO  Running migrations.

  2024_09_13_002308_create_m_kategori_table ..................................................... 74ms DONE
  2024_09_13_002355_create_m_supplier_table ..................................................... 50ms DONE
```

Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_kategori	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_level	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
m_supplier	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
8 tables	Sum	7	InnoDB	utf8mb4_unicode_ci	128.0 KiB	0 B

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

Praktikum Ke-2.2

Langkah	Jawaban/Deskripsi
1.	Buka terminal VSCode kalian, dan buat file migrasi untuk table m_use

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

  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_09_13_003855_create_m_user_table.php] created successfully.
```

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

```php
public function up(): void
{
    Schema::create('m_user', function (Blueprint $table) {
        $table->id('user_id');
        $table->unsignedBigInteger('level_id')->index(); //indexing untuk ForeignKey
        $table->string('username',20)->unique(); //unique untuk memastikan tidak ada username yang sama
        $table->string('nama',100);
        $table->string('password');
        $table->timestamps();

        //Mendefinisikan Foreign Key pada kolom level_id mengacu pada kolom level_id di tabel m_level
        $table->foreign('level_id',)->references('level_id')->on('m_level');
    });
}

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

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

```
PS C:\laragon\www\PWL_POS> php artisan migrate


  INFO  Running migrations.

  2024_09_13_003855_create_m_user_table ........ 407ms DONE
```

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

- Table m_barang

```php
public function up(): void
{
    Schema::create('m_barang', function (Blueprint $table) {
        $table->id('barang_id')->autoIncrement();
        $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();

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

- Tabel t_penjualan

```php
public function up(): void
{
    Schema::create('t_penjualan', function (Blueprint $table) {
        $table->id('penjualan_id')->autoIncrement();
        $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');
    });
}
```

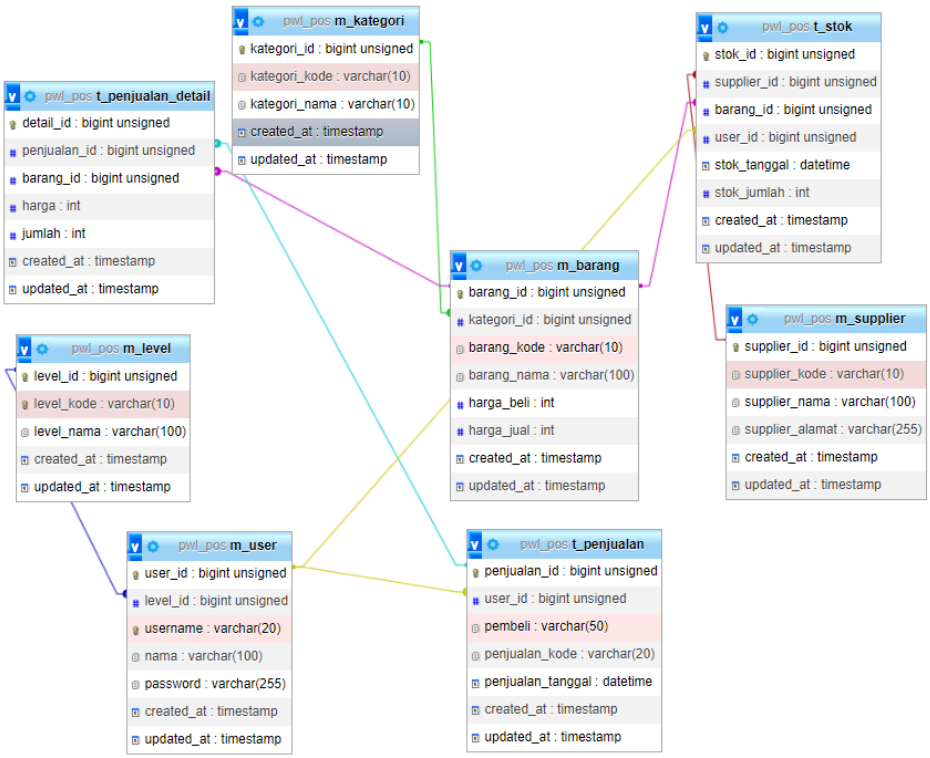
- Tabel t_stok

```php
public function up(): void
{
    Schema::create('t_stok', function (Blueprint $table) {
        $table->id('stok_id')->autoIncrement();
        $table->unsignedBigInteger('supplier_id')->index();
        $table->unsignedBigInteger('barang_id')->index();
        $table->unsignedBigInteger('user_id')->index();
        $table->datetime('stok_tanggal');
        $table->integer('stok_jumlah');
        $table->timestamps();

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

- Tabel t_penjualan_detail

```php
public function up(): void
{
    Schema::create('t_penjualan_detail', function (Blueprint $table) {
        $table->id('detail_id');
        $table->unsignedBigInteger('penjualan_id')->index();
        $table->unsignedBigInteger('barang_id')->index();
        $table->integer('harga');
        $table->integer('jumlah');
        $table->timestamps();

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

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

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

Praktikum Ke-3

Langkah	Jawaban/Deskripsi

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.`

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

```php
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facedes\DB;

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

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

```
PS 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

←T→	level_id	level_kode	level_nama	created_at	updated_at
Edit Copy Delete	1	ADM	Administrator	NULL	NULL
Edit Copy Delete	2	MNG	Manager	NULL	NULL
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\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

```
class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            [
                'user_id' => 1,
                'level_id' => 1,
                'username' => 'admin',
                'nama' => 'Administrator',
                'password' => Hash::make('12345'),//class untuk mengenskrispsi hash password
            ],
            [
                'user_id' => 2,
                'level_id' => 2,
                'username' => 'manager',
                'nama' => 'Manager',
                'password' => Hash::make('12345'),//class untuk mengenskrispsi hash password
            ],
            [
                'user_id' => 3,
                'level_id' => 3,
                'username' => 'staff',
                'nama' => 'Staff/Kasir',
                'password' => Hash::make('12345'),//class untuk mengenskrispsi hash password
            ],
        ];
        DB::table('m_user')->insert($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	created_at	updated_at
Edit Copy Delete	1	1	admin	Administrator	$2y$12$VrkLMZSGeez6ETzVhF7s5uXu3gbdXkn8xtnWfsoCWD....	NULL	NULL
Edit Copy Delete	2	2	manager	Manager	$2y$12$Wj2ktoadyfZcVHQOTsyASOxy5nC7b..NMreBBKgSSo0...	NULL	NULL
Edit Copy Delete	3	3	staff	Staff/Kasir	$2y$12$GSO5StRfyQdtWwzzoU6mxebZwBlAleb/BdGUipWiDbY...	NULL	NULL

9. Ok, data seeder berhasil di masukkan ke database.

10. Sekarang coba kalian masukkan data seeder untuk table yang lain, dengan ketentuan

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

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

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

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

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

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

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

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

PS C:\laragon\www                                    seeder PenjualanSeeder
                                Open file in editor (ctrl + click)
  INFO  Seeder [C:\laragon\www\PWL_POS\database\seeders\PenjualanSeeder.php] created successfully.

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

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

- m_kategori

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	ELK	Elektronik	NULL	NULL
2	PKN	Pakaian	NULL	NULL
3	ATK	Alat Tulis	NULL	NULL
4	PRT	Perabotan	NULL	NULL
5	SNK	Snack	NULL	NULL

- m_supplier

supplier_id	supplier_kode	supplier_nama	supplier_alamat	created_at	updated_at
1	SUP1	Bandung Clothing	Jl. Bandung No.3	NULL	NULL
2	SUP2	Hartono Elektronik	Jl. Soekarno Hatta	NULL	NULL
3	SUP3	PT Sinar Kencaya Agung	Jl. Kyai Tamin No.115	NULL	NULL

- m_barang

barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
1	1	E01	Lampu Gantung	100000	120000	NULL	NULL
2	1	E02	Kulkas	5000000	5500000	NULL	NULL
3	1	E03	Mesin Cuci	2000000	2500000	NULL	NULL
4	1	E04	Mouse	80000	100000	NULL	NULL
5	1	E05	Keyboard	90000	110000	NULL	NULL
6	2	P01	Baju Laki-Laki	100000	120000	NULL	NULL
7	2	P02	Baju Perempuan	200000	250000	NULL	NULL
8	2	P03	Baju Anak-Anak	50000	80000	NULL	NULL
9	2	P04	Baju Bayi	30000	40000	NULL	NULL
10	2	P05	Topi	50000	65000	NULL	NULL
11	3	A01	Pensil	5000	10000	NULL	NULL
12	3	A02	Penghapus	3000	5000	NULL	NULL
13	3	A03	Tipe-X	6000	8000	NULL	NULL
14	3	A04	Kotak Pensil	20000	25000	NULL	NULL
15	3	A05	Pensil Warna	25000	30000	NULL	NULL

- t_stok

stok_id	supplier_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
1	1	1	1	2024-09-14 00:00:00	50	NULL	NULL
2	1	2	1	2024-09-14 00:00:00	25	NULL	NULL
3	1	3	1	2024-09-14 00:00:00	15	NULL	NULL
4	1	4	1	2024-09-14 00:00:00	22	NULL	NULL
5	1	5	2	2024-09-14 00:00:00	5	NULL	NULL
6	1	6	2	2024-09-14 00:00:00	20	NULL	NULL
7	1	7	2	2024-09-14 00:00:00	40	NULL	NULL
8	1	8	2	2024-09-14 00:00:00	10	NULL	NULL
9	1	9	2	2024-09-14 00:00:00	10	NULL	NULL
10	1	10	2	2024-09-14 00:00:00	10	NULL	NULL
11	1	11	3	2024-09-14 00:00:00	5	NULL	NULL
12	1	12	3	2024-09-14 00:00:00	10	NULL	NULL
13	1	13	3	2024-09-14 00:00:00	15	NULL	NULL
14	1	14	3	2024-09-14 00:00:00	25	NULL	NULL
15	1	15	3	2024-09-14 00:00:00	10	NULL	NULL

- t_penjualan

penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
1	1	Cortez Pagac	PJ01	2024-09-14 00:00:00	NULL	NULL
2	1	Dr. Jeramie Vandervort PhD	PJ02	2024-09-14 00:00:00	NULL	NULL
3	1	Mrs. Adaline Daniel Jr.	PJ03	2024-09-14 00:00:00	NULL	NULL
4	1	Ms. Carolyne Raynor DVM	PJ04	2024-09-14 00:00:00	NULL	NULL
5	1	Ms. Joyce Williamson	PJ05	2024-09-14 00:00:00	NULL	NULL
6	2	Rick Wisozk	PJ06	2024-09-14 00:00:00	NULL	NULL
7	2	Kurtis Kirlin	PJ07	2024-09-14 00:00:00	NULL	NULL
8	3	Erica Jacobson	PJ08	2024-09-14 00:00:00	NULL	NULL
9	3	Emie Collier	PJ09	2024-09-14 00:00:00	NULL	NULL
10	3	Ms. Tierra Weber III	PJ10	2024-09-14 00:00:00	NULL	NULL

- t_penjualan_detail

detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
1	1	1	120000	2	NULL	NULL
2	1	2	5500000	2	NULL	NULL
3	1	3	2500000	3	NULL	NULL
4	2	2	5500000	1	NULL	NULL
5	2	4	100000	2	NULL	NULL
6	2	6	120000	2	NULL	NULL
7	3	2	5500000	3	NULL	NULL
8	3	9	20000	9	NULL	NULL
9	3	12	40000	1	NULL	NULL
10	4	2	5500000	4	NULL	NULL
11	4	10	65000	2	NULL	NULL
12	4	14	25000	4	NULL	NULL
13	5	2	5500000	4	NULL	NULL
14	5	2	5500000	1	NULL	NULL
15	5	10	65000	1	NULL	NULL
16	6	11	10000	3	NULL	NULL
17	6	15	30000	1	NULL	NULL
18	6	13	8000	2	NULL	NULL

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

Praktikum Ke-4

Langkah	Jawaban/Deskripsi

1.	Kita buat controller dahulu untuk mengelola data pada table m_level
	`PS C:\laragon\www\PWL_POS> php artisan make:controller LevelController`
	`INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\Level`

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

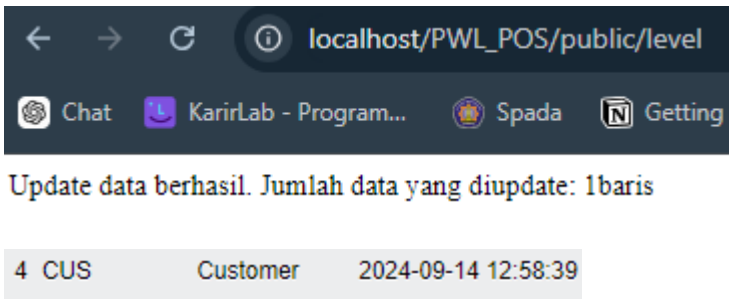
	`Route::get('/level', [LevelController::class, 'index']);`

3.	Selanjutnya, kita modifikasi file LevelController untuk menambahkan 1 data ke table m_level
	`<?php`
	`namespace App\Http\Controllers;`
	`use Illuminate\Http\Request;`
	`class LevelController extends Controller`
	`{`
	`public function index()`
	`{`
	`DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?,?,?)',['CUS', 'Pelanggan', now()]);`
	`return 'Insert data baru berhasil';`
	`}`
	`}`

4.	Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level dan amati apa yang terjadi pada table m_level di database, screenshot perubahan yang ada pada table m_level localhost/PWL_POS/public/level Chat KarirLab - Program... Spada Getting S Insert data baru berhasil 4 CUS Pelanggan 2024-09-14 12:58:39

5.	Selanjutnya, kita modifikasi lagi file LevelController untuk meng-update data di table m_level seperti berikut
	`//update`
	`$row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);`
	`return 'Update data berhasil. Jumlah data yang diupdate: ' .$row. 'baris';`

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

<table>
<tr><td></td><td>


</td></tr>
<tr><td>7.</td><td>
Kita coba modifikasi lagi file LevelController untuk melakukan proses hapus data

```
// delete
$row=DB::delete('delete from m_level where level_kode = ?', ['CUS']);
return 'Delete data berhasil. Jumlah data yang dihapus: '. $row . ' baris';
```

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

```
$data = DB::select('select * from m_level');
return view('level', ['data' => $data]);
```
</td></tr>
<tr><td>9.</td><td>
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

<!
</td></tr>
</table>

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Data level pengguna</title>
</head>
<body>
    <h1>Data level pengguna</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <th>ID</th>
            <th>Kode level</th>
            <th>Nama level</th>
        </tr>
        @foreach ($data as $d)
        <tr>
            <td> {{$d->level_id}} </td>
            <td> {{$d->level_kode}} </td>
            <td> {{$d->level_nama}} </td>
        </tr>
        @endforeach
    </table>
</body>
</html>
```

10. Silahkan dicoba pada browser dan amati apa yang terjadi

Data level pengguna

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

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

Praktikum Ke-5

Langkah	Jawaban/Deskripsi

1.	Kita buat controller dahuku 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

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

3.	Selanjutnya, kita modifikasi file KategoriController untuk menambahkan 1 data ke table m_kategori
	
	`<?php`
	
	`namespace App\Http\Controllers;`
	
	`use Illuminate\Http\Request;`
	`use Illuminate\Support\Facades\DB;`
	
	`class KategoriController extends Controller`
	`{`
	`  public function index(){`
	`  $data = [`
	`  'kategori_kode' => 'SNK',`
	`  'kategori_nama' => 'Snack/Makanan Ringan',`
	`  'created_at' => now()`
	`  ];`
	`  DB::table('m_kategori')->insert($data);`
	`  return 'Insert data baru berhasil';`
	`  }`
	`}`

4.	Kita coba jalankan di browser dengan url localhost/PWL_POS/public/kategori dan amati apa yang terjadi pada table m_kategori di database, screenshot perubahan yang ada pada table m_kategori localhost/PWL_POS/public/kategori Chat KarirLab - Program... Spada Getting Sta Insert data baru berhasil 6 SNK Snack/Makanan Ringan 2024-09-14 13:29:53

5.	Selanjutnya, kita modifikasi lagi file KategoriController untuk meng-update data di table m_kategori seperti berikut
	
	`//update`
	`  $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: 6 SNK Camilan 2024-09-14 13:29:53

7.	Kita coba modifikasi lagi file KategoriController untuk melakukan proses hapus data
	`//delete`
	`$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();`
	`return 'Delete data berhasil. Jumlah data yang dihapus:' .$row . ' baris';`
	localhost/PWL_POS/public/kategori
	Chat KarirLab - Program... Spada Getting Start
	Delete data berhasil. Jumlah data yang dihapus:1 baris

8.	Method terakhir yang kita coba adalah untuk menampilkan data yang ada di table m_kategori. Kita modifikasi file KategoriController seperti berikut
	`// view`
	`$data = DB::table('m_kategori') -> get();`
	`return view('kategori', ['data' => $data]);`

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

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Data kategori barang</title>
</head>
<body>
    <h1>Data kategori barang</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <th>ID</th>
            <th>Kode kategori</th>
            <th>Nama kategori</th>
        </tr>
        @foreach ($data as $d)
            <tr>
                <td> {{$d->kategori_id}} </td>
                <td> {{$d->kategori_kode}} </td>
                <td> {{$d->kategori_nama}} </td>
            </tr>
        @endforeach
    </table>
</body>
</html>
```

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

Data kategori barang

ID	Kode kategori	Nama kategori
1	ELK	Elektronik
2	PKN	Pakaian
3	ATK	Alat Tulis
4	PRT	Perabotan
5	SNCK	Camilan

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

Praktikum Ke-6

Langkah	Jawaban/Deskripsi
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
	`<?php`
	`namespace App\Models;`
	`use Illuminate\Database\Eloquent\Factories\HasFactory;`
	`use Illuminate\Database\Eloquent\Model;`
	`class UserModel extends Model`
	`{`
	`  use HasFactory;`
	`  protected $table = 'm_user';`
	`  protected $primaryKey = 'user_id';`
	`}`

4.	Kita modifikasi route web.php untuk mencoba routing ke controller UserController `Route::get('/user', [UserController::class, 'index']);`

5.	Sekarang, kita buat file controller UserController dan memodifikasinya seperti berikut
	`// coba akses model UserModel`
	`$user = UserModel::all(); //ambil semua data m_user`
	`return view('user', ['data' => $user]);`

6.	Kemudian kita buat view user.blade.php

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Data user</title>
</head>
<body>
    <h1>Data user</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <td>ID</td>
            <td>Username</td>
            <td>Nama</td>
            <td>ID Level pengguna</td>
        </tr>
        @foreach ($data as $d)
            <tr>
                <td> {{$d->user_id}} </td>
                <td> {{$d->username}} </td>
                <td> {{$d->nama}} </td>
                <td> {{$d->level_id}} </td>
            </tr>
        @endforeach
    </table>
</body>
</html>
```

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

Data user

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

8. Setelah itu, kita modifikasi lagi file UserController

```php
//tambah data
$data=[
    'username' => 'customer 1',
    'nama' => 'pelanggan',
    'password' => Hash::make('12345'),
    'level_id' => 5
];
UserModel::insert($data);
```

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

<table>
<tr><td></td><td>

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	5

</td></tr>
<tr><td>10.</td><td>Kita modifikasi lagi file UserController menjadi seperti berikut

```
//tambah data dengan Eloquent
$data = [
    'nama' => 'Pelanggan pertama',
];
UserModel::where('username', 'customer 1')->update($data); //update data user
```

</td></tr>
<tr><td>11.</td><td>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	5

</td></tr>
<tr><td>12.</td><td>Jika sudah, laporkan hasil Praktikum-6 ini dan commit perubahan pada git</td></tr>
</table>

Soal Praktikum

Langkah	Jawaban/Deskripsi
1.	Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada file setting .env Laravel? **Jawaban :** Banyak fungsi diantaranya dapat mengenkripsi data session, menghasilkan kode otentikasi, membuat hash.
2.	Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP_KEY? **Jawaban :** Menggunakan kode “php artisan key:generate” pada terminal VS Code
3.	Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut? **Jawaban :** • Create_users – menyimpan data pengguna • Create_password_reset_token – menyimpan token reset kata sandi • Create_failed_jobs – menyimpan informasi tentang pekerjaan yang gagal diproses • Create_personal_accses_token – menyimpan token otentikasi API untuk pengguna.
4.	Secara default, file migrasi terdapat kode $table->timestamps();, apa tujuan/output dari fungsi tersebut? **Jawaban :** • created_at yang memiliki format datetime, yang akan diisikan dengan waktu saat kita membuat table tersebut • updated_at yang memiliki format datetime, yang akan diisikan dengan waktu saat kita mengupdate table tersebut
5.	Pada File Migrasi, terdapat fungsi $table->id(); Tipe data apa yang dihasilkan dari fungsi tersebut?

	Jawaban : Bigint
6.	Apa bedanya hasil migrasi pada table m_level, antara menggunakan $table->id(); dengan menggunakan $table->id('level_id'); ? **Jawaban :** • $table->id(); memiliki tipe data bingint yang sudah autoincrement dan otomatis menjadi primary key • $table->id('level_id'); nama kolomnya akan menjadi level_id
7.	Pada migration, Fungsi ->unique() digunakan untuk apa? **Jawaban :** Untuk memebedakan antara data satu dengan data yang lainnya, sehingga tidak terjadi 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') ? **Jawaban** : • M_user : primary key dan auto-increment • M_level: foreign key yang merujuk ke primary key di tabel m_level dan tidak memerlukan auto-increment
9.	Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program Hash::make('1234');? **Jawaban :** Hash berfungsi untuk menyamarkan kata sandi yang asli yaitu ‘1234’ menjadi kode hashing yang sulit dibaca oleh manusia berguna untuk membuat password menjadi lebih aman.
10.	Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut? **Jawaban :**

	Tanda tanya akan menggantikan data yang ada pada array kedu
11.	Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode protected $table = 'm_user'; dan protected $primaryKey = 'user_id'; ? **Jawaban :** Untuk menyesuaikan model dengan struktur tabel database yang ada. Dengan mendefinisikan $table dan $primaryKey, Laravel akan mengetahui tabel mana yang harus digunakan dan kolom mana yang menjadi primary key saat berinteraksi dengan model tersebut.
12.	Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan **Jawaban :** Lebih mudah untuk menggunakan DB Façade karena lebih praktis.