

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

```
.env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:1C/1K4ak1o0W3aGhVVubX3G93TWjdRtZ06yCid4Z0ZU=
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=
```

Praktikum 2

1. Kita buat file migrasi untuk table m_level dengan perintah

```
branch main set up to track origin/main .
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_m_level_table --create=m_level

INFO Migration [D:\Laravel Kuliah\PWL_POS\database\Migrations\2024_03_06_072124_create_m_level_table.php] created successfully.
```

2. modifikasi sesuai desain database yang sudah ada

```
*/
public function up(): void
{
    Schema::create('m_level', function (Blueprint $table) {
        $table->id('level_id');
        $table->string('level_kode', 10)->unique();
        $table->string('level_name', 100);
        $table->timestamps();
    });
}
```

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

```
INFO Running migrations.

2014_10_12_000000_create_users_table ..... 65ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 20ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 52ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 74ms DONE
2024_03_06_072124_create_m_level_table ..... 42ms DONE
```

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



5. Buat table database dengan migration untuk table m_kategori yang sama-sama tidak memiliki foreign key

```
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_m_kategori_table --create=m_kategori
```

```
INFO Migration [D:\Laravel Kuliah\PWL_POS\database\migrations\2024_03_06_072710_create_m_kategori_table.php] created successfully.
```

```
public function up(): void
{
    Schema::create('m_kategori', function (Blueprint $table) {
        $table->id('level_id');
        $table->integer('kategori_id')->unique();
        $table->string('kategori_kode',10);
        $table->string('kategori_nama',100);
    });
}
```

```
PS D:\Laravel Kuliah\PWL_POS> php artisan migrate
```

```
INFO Running migrations.
```

```
2024_03_06_073431_create_m_kategori_table ..... 64ms DONE
```

```
PS D:\Laravel Kuliah\PWL_POS>
```

Praktikum 2.2

1. Buka terminal VSCode kalian, dan buat file migrasi untuk table m_user
2. Buka file migrasi untuk table m_user, dan modifikasi seperti berikut

```
Schema::create('m_user', function (Blueprint $table) {
    $table->id("user_id");
    $table->unsignedBigInteger('level_id')->index();
    $table->string('username',20)->unique();
    $table->string('nama',100);
    $table->string('password');
    $table->timestamps();

    $table->foreign('level_id')->references('level_id')->on('m_level');
});
```

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan migrate
```

```
INFO Running migrations.
```

```
2024_03_06_073858_create_m_user_table ..... 238ms DONE
```

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_m_barang_table --create=m_barang
INFO Migration [D:\Laravel Kuliah\PWL_POS\database\migrations\2024_03_06_075217_create_m_barang_table.php] created successfully.
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_t_penjualan_table --create=t_penjualan
INFO Migration [D:\Laravel Kuliah\PWL_POS\database\migrations\2024_03_06_075242_create_t_penjualan_table.php] created successfully.
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_t_stok_table --create=t_stok
INFO Migration [D:\Laravel Kuliah\PWL_POS\database\migrations\2024_03_06_075306_create_t_stok_table.php] created successfully.
PS D:\Laravel Kuliah\PWL_POS> php artisan make:migration create_t_penjualan_detail_table --create=t_penjualan_detail
INFO Migration [D:\Laravel Kuliah\PWL_POS\database\migrations\2024_03_06_075335_create_t_penjualan_detail_table.php] created successfully.
```

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

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

```
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('penjual_code',20);
        $table->dateTime('penjualan_tanggal');
        $table->timestamps();

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

```
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');
    });
}
```

```
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');
    });
}
```

```

PS D:\Laravel Kuliah\PWL_POS> php artisan migrate

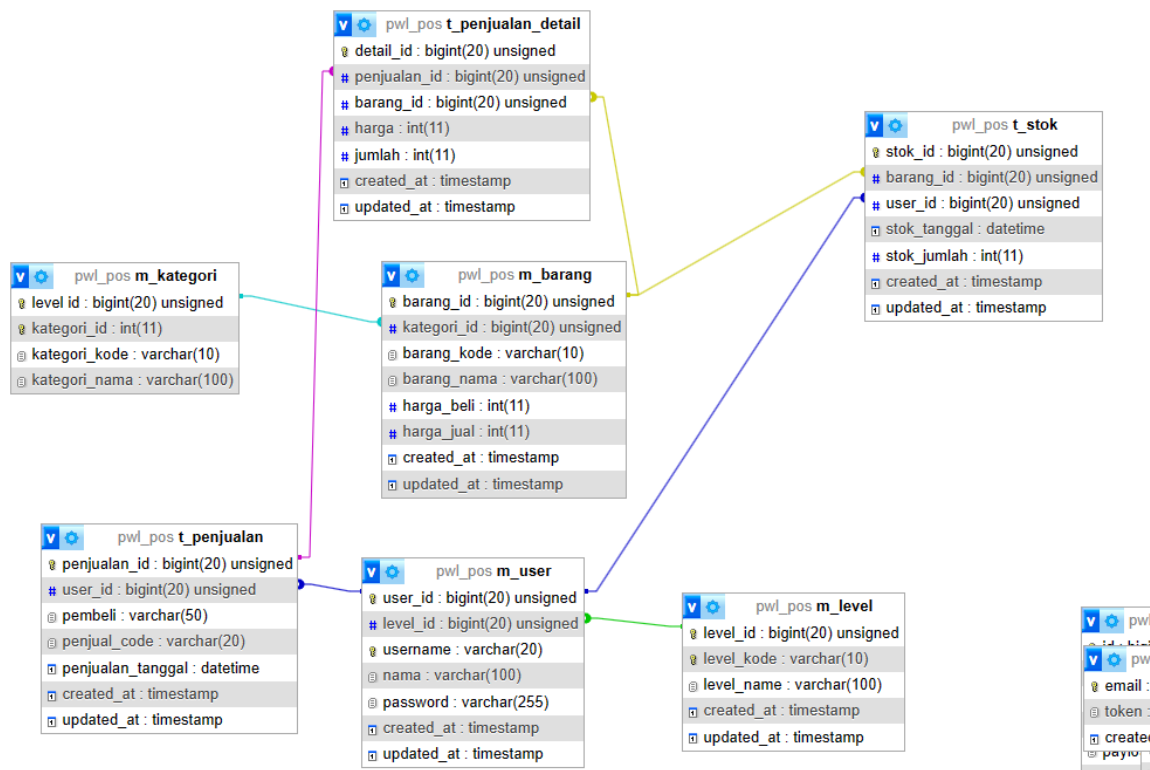
INFO Running migrations.

2024_03_06_075217_create_m_barang_table ..... 73ms DONE
2024_03_06_075242_create_t_penjualan_table ..... 101ms DONE
2024_03_06_075306_create_t_stok_table ..... 142ms DONE
2024_03_06_075335_create_t_penjualan_detail_table ..... 134ms DONE

PS D:\Laravel Kuliah\PWL_POS>

```

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



Praktikum 3

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

```

create mode 100644 database/migrations/2024_03_06_075335_create_t_penjualan_detail_table.php
PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder LevelSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\LevelSeeder.php] created successfully.

```

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

```

public function run(): void
{
    $data = [
        ['level_id' => 1, 'level_kode' => 'ADM', 'level_name' => 'Administrator'],
        ['level_id' => 2, 'level_kode' => 'MNG', 'level_name' => 'Manager'],
        ['level_id' => 3, 'level_kode' => 'STF', 'level_name' => 'Staff/Kasir'],
    ];
    DB::table('m_level')->insert($data);
}

```

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

```
PS D:\Laravel Kuliah\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_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

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder UserSeeder
```

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\UserSeeder.php] created successfully.

6. Modifikasi file class UserSeeder seperti berikut

```
0 references | 0 overrides
public function run(): void
{
    $data = [
        ['user_id' => 1, 'level_id' => 1, 'username' => 'admin', 'nama' => 'Administrator', 'password' => Hash::make('12345')],
        ['user_id' => 2, 'level_id' => 2, 'username' => 'manager', 'nama' => 'Manager', 'password' => Hash::make('12345')],
        ['user_id' => 3, 'level_id' => 3, 'username' => 'staff', 'nama' => 'Staff/Kasir', 'password' => Hash::make('12345')],
    ];
    DB::table('m_user')->insert($data);
}
```

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```
PS D:\Laravel Kuliah\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
<input type="checkbox"/>	Edit	Copy	Delete	1	1	admin	Administrator	\$2y\$12\$52MpM3n2yCourrGGoGSNuWJfbkrLJlJPPnOVp0rB1...	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	2	manager	Manager	\$2y\$12\$JMIBR4IRhPOH5s/2GVOb0u8OAD1MKP7PpvO7paEdFg...	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	3	staff	Staff/Kasir	\$2y\$12\$DHcIJ2QXDP3WW6ZlVu0WZesptTl6/01z5vMNZ9pm2xq...	NULL	NULL

9. Sekarang coba kalian masukkan data seeder untuk table yang lain, dengan ketentuan seperti berikut

```
PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder KategoriSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\KategoriSeeder.php] created successfully.

PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder BarangSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\BarangSeeder.php] created successfully.

PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder StokSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\StokSeeder.php] created successfully.

PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder PenjualanSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\PenjualanSeeder.php] created successfully.

PS D:\Laravel Kuliah\PWL_POS> php artisan make:seeder Penjualan_detailSeeder

INFO Seeder [D:\Laravel Kuliah\PWL_POS\database\seeders\Penjualan_detailSeeder.php] created successfully.
```

```
0 references | 0 overrides
public function run(): void
{
    $data = [
        ['kategori_id' => 1, 'kategori_kode' => 'K01', 'kategori_nama' => 'Baju'],
        ['kategori_id' => 2, 'kategori_kode' => 'K02', 'kategori_nama' => 'Celana'],
        ['kategori_id' => 3, 'kategori_kode' => 'K03', 'kategori_nama' => 'Topi'],
        ['kategori_id' => 4, 'kategori_kode' => 'K04', 'kategori_nama' => 'Sepatu'],
        ['kategori_id' => 5, 'kategori_kode' => 'K05', 'kategori_nama' => 'Tas']
    ];

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan db:seed --class=KategoriSeeder

INFO Seeding database.
```

```
0 references | 0 overrides
public function run(): void
{
    $data = [
        ['barang_id' => 1, 'kategori_id' => 1, 'barang_kode' => 'B01', 'barang_nama' => 'Uniqlo', 'harga_beli' => 10000, 'harga_jual' => 12000],
        ['barang_id' => 2, 'kategori_id' => 2, 'barang_kode' => 'C01', 'barang_nama' => 'Levi's Jeans', 'harga_beli' => 150000, 'harga_jual' => 180000],
        ['barang_id' => 3, 'kategori_id' => 3, 'barang_kode' => 'T01', 'barang_nama' => 'Topi New Era', 'harga_beli' => 50000, 'harga_jual' => 70000],
        ['barang_id' => 4, 'kategori_id' => 4, 'barang_kode' => 'S01', 'barang_nama' => 'Sepatu Nike Air Force 1', 'harga_beli' => 1000000, 'harga_jual' => 1200000],
        ['barang_id' => 5, 'kategori_id' => 5, 'barang_kode' => 'TA01', 'barang_nama' => 'Tas Gucci', 'harga_beli' => 20000000, 'harga_jual' => 25000000],
        ['barang_id' => 6, 'kategori_id' => 1, 'barang_kode' => 'B02', 'barang_nama' => 'KCM Dress', 'harga_beli' => 80000, 'harga_jual' => 100000],
        ['barang_id' => 7, 'kategori_id' => 2, 'barang_kode' => 'C02', 'barang_nama' => 'Adidas Jogger Pants', 'harga_beli' => 120000, 'harga_jual' => 150000],
        ['barang_id' => 8, 'kategori_id' => 3, 'barang_kode' => 'T02', 'barang_nama' => 'Topi Snapback', 'harga_beli' => 30000, 'harga_jual' => 40000],
        ['barang_id' => 9, 'kategori_id' => 4, 'barang_kode' => 'S02', 'barang_nama' => 'Sepatu Converse Chuck Taylor', 'harga_beli' => 600000, 'harga_jual' => 750000],
        ['barang_id' => 10, 'kategori_id' => 5, 'barang_kode' => 'TA02', 'barang_nama' => 'Tas Jansport', 'harga_beli' => 300000, 'harga_jual' => 400000]
    ];

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan db:seed --class=BarangSeeder

INFO Seeding database.
```

```
0 references | 0 overrides
public function run(): void
{
    $data = [
        ['stok_id' => 1, 'barang_id' => 1, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 100],
        ['stok_id' => 2, 'barang_id' => 2, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 50],
        ['stok_id' => 3, 'barang_id' => 3, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 25],
        ['stok_id' => 4, 'barang_id' => 4, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 20],
        ['stok_id' => 5, 'barang_id' => 5, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 10],
        ['stok_id' => 6, 'barang_id' => 6, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 75],
        ['stok_id' => 7, 'barang_id' => 7, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 100],
        ['stok_id' => 8, 'barang_id' => 8, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 50],
        ['stok_id' => 9, 'barang_id' => 9, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 75],
        ['stok_id' => 10, 'barang_id' => 10, 'user_id' => 1, 'stok_tanggal' => now(), 'stok_jumlah' => 50],
    ];

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

```
PS D:\Laravel Kuliah\PWL_POS> php artisan db:seed --class=StokSeeder
```

INFO Seeding database.

```
public function run(): void
{
    $data = [
        ['penjualan_id' => 1, 'user_id' => 3, 'pembeli' => 'Dena', 'penjual_code' => 'PJL001', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 2, 'user_id' => 3, 'pembeli' => 'Budi', 'penjual_code' => 'PJL002', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 3, 'user_id' => 3, 'pembeli' => 'Ani', 'penjual_code' => 'PJL003', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 4, 'user_id' => 3, 'pembeli' => 'Citra', 'penjual_code' => 'PJL004', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 5, 'user_id' => 3, 'pembeli' => 'Doni', 'penjual_code' => 'PJL005', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 6, 'user_id' => 3, 'pembeli' => 'Eko', 'penjual_code' => 'PJL006', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 7, 'user_id' => 3, 'pembeli' => 'Fitri', 'penjual_code' => 'PJL007', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 8, 'user_id' => 3, 'pembeli' => 'Gina', 'penjual_code' => 'PJL008', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 9, 'user_id' => 3, 'pembeli' => 'Hendra', 'penjual_code' => 'PJL009', 'penjualan_tanggal' => now()],
        ['penjualan_id' => 10, 'user_id' => 3, 'pembeli' => 'Ira', 'penjual_code' => 'PJL010', 'penjualan_tanggal' => now()],
    ];
    DB::table('t_penjualan')->insert($data);
}
```

```
PS D:\Laravel Kuliah\PWL_POS> php artisan db:seed --class=PenjualanSeeder
```

INFO Seeding database.

```
0 references (0 overrides)
public function run(): void
{
    $data = [
        ['detail_id' => 1, 'penjualan_id' => 1, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 1],
        ['detail_id' => 2, 'penjualan_id' => 1, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 3, 'penjualan_id' => 1, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 1],
        ['detail_id' => 4, 'penjualan_id' => 2, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 2],
        ['detail_id' => 5, 'penjualan_id' => 2, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 6, 'penjualan_id' => 2, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 2],
        ['detail_id' => 7, 'penjualan_id' => 3, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 3],
        ['detail_id' => 8, 'penjualan_id' => 3, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 2],
        ['detail_id' => 9, 'penjualan_id' => 3, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 3],
        ['detail_id' => 10, 'penjualan_id' => 4, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 1],
        ['detail_id' => 11, 'penjualan_id' => 4, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 12, 'penjualan_id' => 4, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 1],
        ['detail_id' => 13, 'penjualan_id' => 5, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 2],
        ['detail_id' => 14, 'penjualan_id' => 5, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 15, 'penjualan_id' => 5, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 2],
        ['detail_id' => 16, 'penjualan_id' => 6, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 3],
        ['detail_id' => 17, 'penjualan_id' => 6, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 2],
        ['detail_id' => 18, 'penjualan_id' => 6, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 3],
        ['detail_id' => 19, 'penjualan_id' => 7, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 1],
        ['detail_id' => 20, 'penjualan_id' => 7, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 21, 'penjualan_id' => 7, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 1],
        ['detail_id' => 22, 'penjualan_id' => 7, 'barang_id' => 4, 'harga' => 1000000, 'jumlah' => 1],
        ['detail_id' => 23, 'penjualan_id' => 8, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 2],
        ['detail_id' => 24, 'penjualan_id' => 8, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
        ['detail_id' => 25, 'penjualan_id' => 8, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 2],
        ['detail_id' => 26, 'penjualan_id' => 9, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 3],
        ['detail_id' => 27, 'penjualan_id' => 9, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 2],
        ['detail_id' => 28, 'penjualan_id' => 9, 'barang_id' => 3, 'harga' => 50000, 'jumlah' => 3],
        ['detail_id' => 29, 'penjualan_id' => 10, 'barang_id' => 1, 'harga' => 10000, 'jumlah' => 1],
        ['detail_id' => 30, 'penjualan_id' => 10, 'barang_id' => 2, 'harga' => 150000, 'jumlah' => 1],
    ];
    DB::table('t_penjualan_detail')->insert($data);
}
```

```
PS D:\Laravel Kuliah\PWL_POS> php artisan db:seed --class=Penjualan_detailSeeder
```

INFO Seeding database.

Praktikum 4

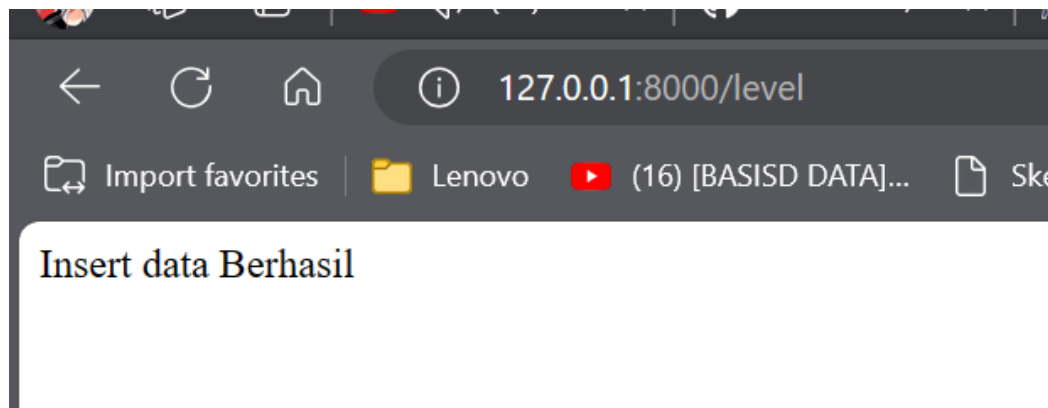
1. Kita buat controller dahulu untuk mengelola data pada table m_level
2. Kita modifikasi dulu untuk routing-nya, ada di PWL_POS/routes/web.php

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

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

```
public function index(){  
    DB::insert('insert into m_level(level_kode,level_name,created_at)  
    values (?, ?,?) ', ['CUS', 'Pelanggan', now()]);  
    return 'Insert data Berhasil';  
}
```

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



		level_id	level_kode	level_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	2024-03-07 13:34:32	NULL

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

```
$row = DB::update('update m_level set level_name = ? 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

127.0.0.1:8000/level

Update data berhasil. Jumlah data yang diupdate :1baris

	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	2024-03-07 13:34:32	NULL

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

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

Hasil :

127.0.0.1:8000/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

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

←

→

↺

🏠

127.0.0.1:8000/level

🔖 Import favorites

📁 Lenovo

▶️ (16) [BASISD DATA]...

📄 Skema Sertif

Data Level Pengguna

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

Praktikum 5

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

```
public function index(){
    $data =[
        'kategori_kode' => 'SNK',
        'kategori_nama' => 'Snak/Makanan Ringan',
        'created_at' => now()
    ];

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

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

←

127.0.0.1:8000/kat...

A

☆

⚙️

🔗

⋮

🔖 Import favorites

📁 Lenovo

▶️ (16) [BASISD DATA]...

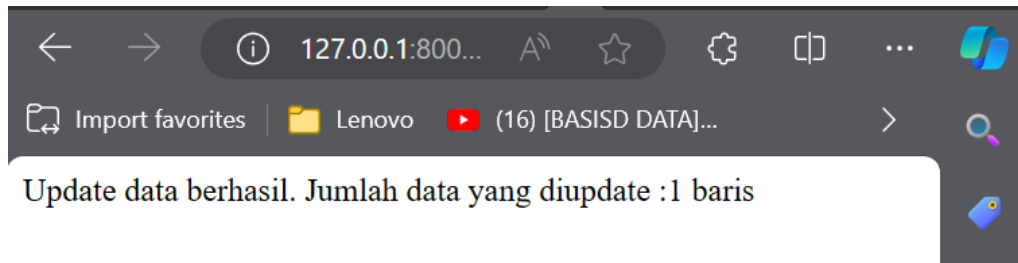
➤

Insert data baru berhasil

← T →		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	K01	Baju	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	K02	Celana	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	K03	Topi	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	K04	Sepatu	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	K05	Tas	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Snak/Makanan Ringan	2024-03-07 14:08:36	NULL

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

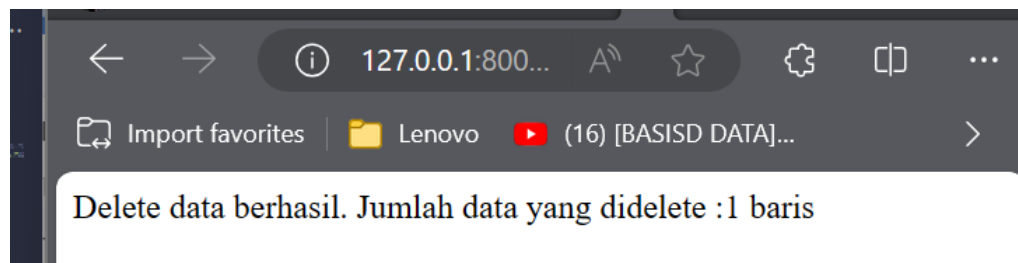
```
$row = DB::table('m_kategori') -> where('kategori_kode', 'SNK') ->
update(['kategori_nama'=>'Camilan']);
return 'Update data berhasil. Jumlah data yang diupdate :'. $row.' baris';
```



				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	K01	Baju	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	K02	Celana	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	K03	Topi	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	K04	Sepatu	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	K05	Tas	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	SNK	Camilan	2024-03-07 14:08:36	NULL

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

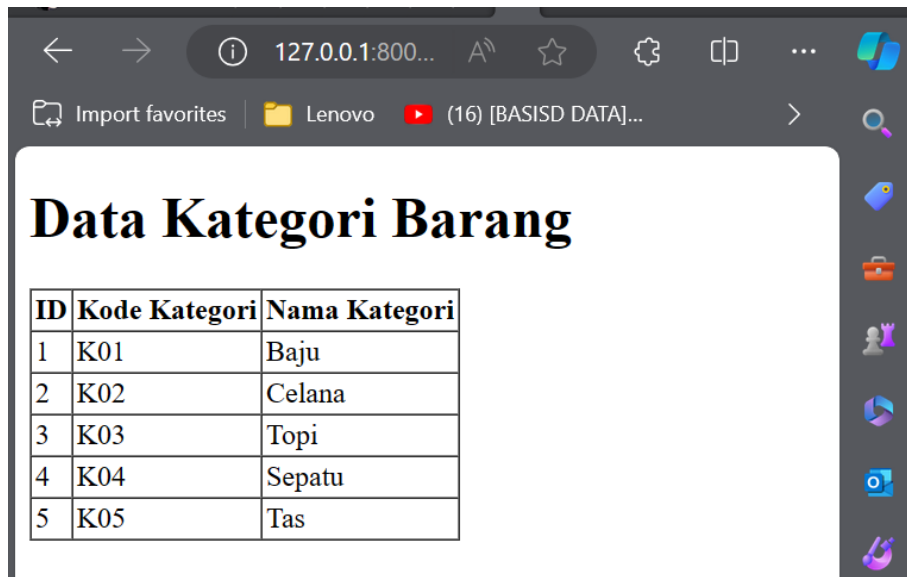
```
$row = DB::table('m_kategori') -> where('kategori_kode', 'SNK')-> delete();
return 'Delete data berhasil. Jumlah data yang didelete :'. $row.' baris';
```



				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	K01	Baju	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	K02	Celana	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	K03	Topi	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	K04	Sepatu	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	K05	Tas	NULL	NULL

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

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



A screenshot of a web browser displaying a table titled "Data Kategori Barang". The browser's address bar shows "127.0.0.1:8000...". The table has three columns: "ID", "Kode Kategori", and "Nama Kategori". It contains five rows of data representing different product categories.

ID	Kode Kategori	Nama Kategori
1	K01	Baju
2	K02	Celana
3	K03	Topi
4	K04	Sepatu
5	K05	Tas

Praktikum 6

1. 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
2. Kita buka file UserModel.php dan modifikasi seperti berikut

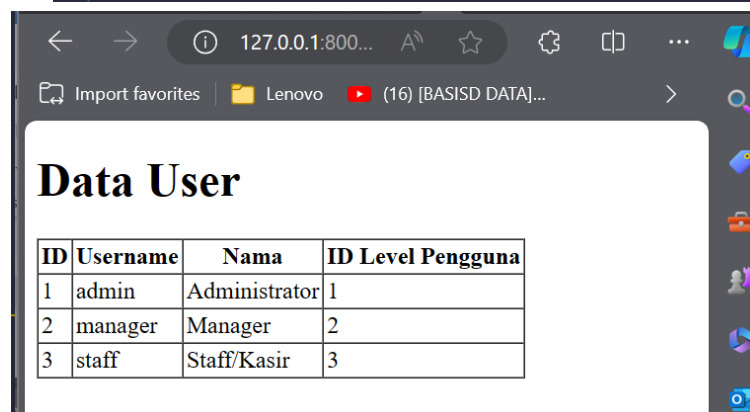
```
class UserModel extends Model
{
    use HasFactory;

    0 references
    protected $table = 'm_user';
    0 references
    protected $primaryKey = 'user_id';
}
```

3. Kita modifikasi route web.php untuk mencoba routing ke controller UserController
4. Sekarang, kita buat file controller UserController dan memodifikasinya seperti berikut

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

```
1 reference | 0 overrides
public function index(){
    $user = UserModel::all();
    return view('user',['data'=>$user]);
}
```

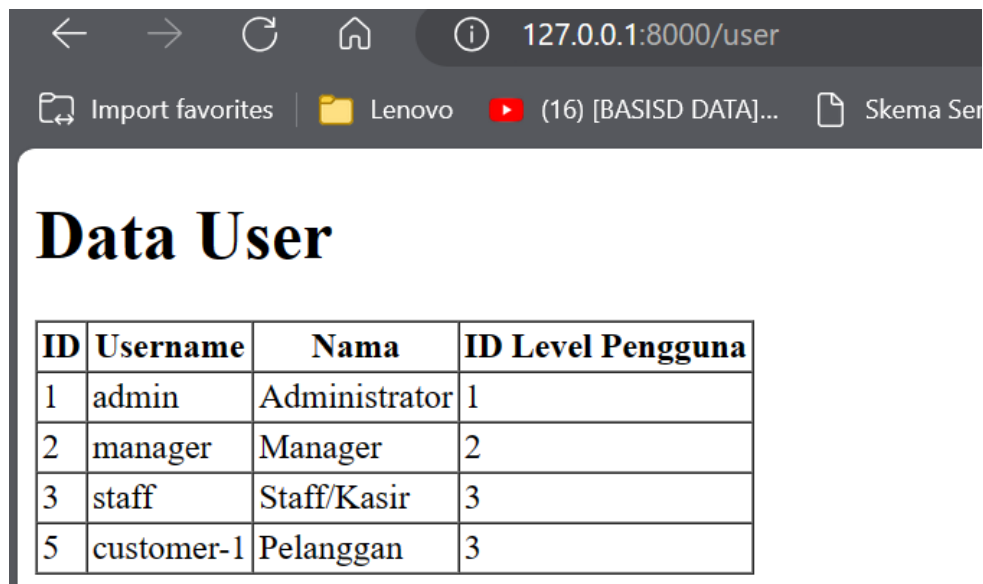


A screenshot of a web browser displaying a table titled "Data User". The browser's address bar shows "127.0.0.1:8000...". The table has four columns: "ID", "Username", "Nama", and "ID Level Pengguna". It contains three rows of data representing different user roles.

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

5. Setelah itu, kita modifikasi lagi file UserController

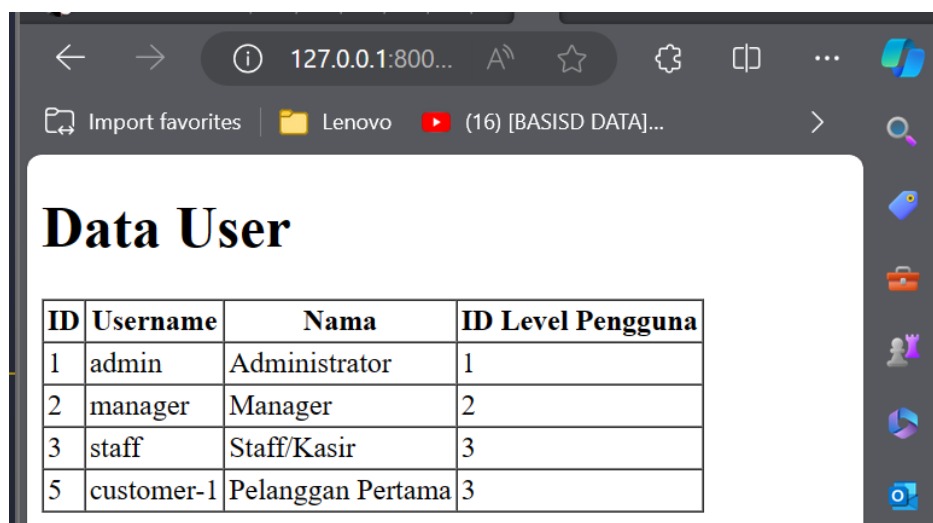
```
$data = [  
    'username' => 'customer-1',  
    'nama' => 'Pelanggan',  
    'password' => Hash::make('12345'),  
    'level_id' => 3  
];  
  
UserModel::insert($data);
```



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

6. Kita modifikasi lagi file UserController menjadi seperti berikut

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



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

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

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

Jawaban : APP_KEY adalah salah satu variabel lingkungan yang sangat penting di dalam file .env. Fungsi utamanya adalah sebagai kunci enkripsi yang digunakan oleh Laravel untuk enkripsi data sensitif seperti session dan cookies. Ketika kita membuat aplikasi Laravel baru, biasanya Anda akan melihat bahwa nilai dari APP_KEY diset dengan nilai acak.

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

Jawaban : Dengan menjalankan php artisan key:generate

3. Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?

Jawaban:

- create_users_table.php: File migrasi ini bertanggung jawab untuk membuat tabel users di dalam database. Tabel users adalah tabel standar yang digunakan untuk menyimpan informasi pengguna, seperti nama pengguna, alamat email, dan sandi terenkripsi.
- create_password_resets_table.php: File migrasi ini bertanggung jawab untuk membuat tabel password_resets. Tabel ini digunakan untuk menyimpan token pengaturan ulang kata sandi yang dikirimkan kepada pengguna yang lupa kata sandi mereka. Dengan menggunakan token ini, pengguna dapat mengatur ulang kata sandi mereka sendiri.

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

Jawaban : Kode \$table->timestamps(); yang ada dalam file migrasi Laravel bertujuan untuk secara otomatis menambahkan dua kolom timestamp ke tabel yang sedang dibuat, yaitu created_at dan updated_at

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

Jawaban : Fungsi \$table->id() dalam file migrasi Laravel bertujuan untuk membuat kolom utama (primary key) dengan tipe data yang sesuai untuk digunakan sebagai identifikasi unik untuk setiap baris data dalam tabel. Tipe data yang dihasilkan dari fungsi ini adalah BIGINT UNSIGNED dengan atribut AUTO_INCREMENT (atau UNSIGNED BIGINT AUTO_INCREMENT, tergantung pada versi MySQL yang digunakan).

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

Jawaban: perbedaan utama antara kedua pernyataan tersebut adalah nama kolom primary key yang dihasilkan. Dalam kebanyakan kasus, penggunaan \$table->id() lebih umum karena Laravel secara otomatis mengikuti konvensi penamaan dengan menggunakan id sebagai nama kolom primary key. Namun, jika kita ingin memberikan nama kolom primary key secara eksplisit, Anda dapat menggunakan \$table->id('nama_kolom');

7. Pada migration, Fungsi ->unique() digunakan untuk apa?

Jawaban : Fungsi `->unique()` dalam file migrasi Laravel digunakan untuk menetapkan batasan unik (unique constraint) pada kolom tertentu di dalam tabel database. Ketika Anda menambahkan fungsi `->unique()` pada sebuah kolom, Laravel akan memastikan bahwa setiap nilai di dalam kolom tersebut harus unik, artinya tidak boleh ada dua baris dalam tabel dengan nilai yang sama untuk kolom yang ditetapkan sebagai unique.

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')` ?

Jawaban :

- `$table->unsignedBigInteger('level_id')` pada tabel `m_user`: Penggunaan `unsignedBigInteger()` bertujuan untuk membuat kolom `level_id` sebagai sebuah kolom yang memuat nilai bilangan bulat (integer) yang besar (big integer) yang tidak boleh negatif (unsigned)
- `$table->id('level_id')` pada tabel `m_level`: Di sini, kita menggunakan `id('level_id')` untuk memberikan nama kolom primary key secara eksplisit. Ini menandakan bahwa kita ingin menggunakan kolom `level_id` sebagai primary key di tabel `m_level`. Dalam konteks ini, penggunaan `id('level_id')`

9. Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program `Hash::make('1234');`?

Jawaban : Dengan menggunakan `Hash::make()`, kita dapat mengenkripsi kata sandi tersebut sehingga hanya bisa dibaca oleh sistem dan tidak bisa diketahui dengan mudah oleh orang lain, termasuk administrator database.

10. Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?

Jawaban : Tanda tanya (?) pada query builder Laravel adalah placeholder atau parameterized query. Ini digunakan untuk mengganti nilai variabel dalam query dengan nilai aktual saat query dijalankan. Penggunaan tanda tanya ini sangat berguna untuk mencegah serangan SQL Injection dan mempermudah penggunaan parameter dalam query.

11. Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode `protected $table = 'm_user';` dan `protected $primaryKey = 'user_id';` ?

Jawaban : tujuan dari penulisan kode tersebut adalah untuk memberikan fleksibilitas dalam menentukan nama tabel dan kunci utama yang digunakan oleh model Eloquent, terutama jika tabel atau kolom tidak mengikuti konvensi penamaan default Laravel. Dengan menentukan secara eksplisit properti `$table` dan `$primaryKey`, kita dapat dengan mudah menyesuaikan model dengan struktur tabel yang ada dalam database

12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan

Jawaban : menurut saya Query Builder karena menyediakan antarmuka yang lebih ekspresif daripada DB Facade untuk membangun query SQL menggunakan metode berantai. Query Builder memungkinkan kita untuk membangun query dengan lebih mudah menggunakan metode PHP daripada menulis query SQL