

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT
MINGGU KE III : MIGRATION, SEEDER, DB FAÇADE, QUERY BUILDER,
DAN ELOQUENT ORM



OLEH :
FAIZA ANATHASYA EKA FALEN
2341760105 / 12

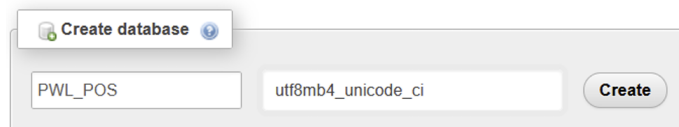
PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur
65141

TAHUN 2025

Praktikum 1 - pengaturan database:

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

Databases



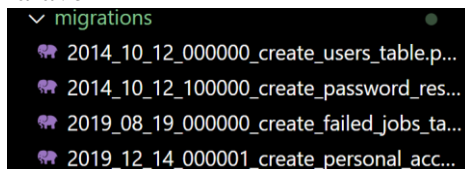
2. Buka aplikasi VSCode dan buka folder project PWL_POS yang sudah kita buat
3. Copy file .env.example menjadi .env
4. Buka file .env, dan pastikan konfigurasi APP_KEY bernilai. Jika belum bernilai silahkan kalian generate menggunakan php artisan.
5. Edit file .env dan sesuaikan dengan database yang telah dibuat

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

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

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

```
database > migrations > 2025_03_17_185306_create_m_level_table.php > class > up > Closure
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     }
21 }
```

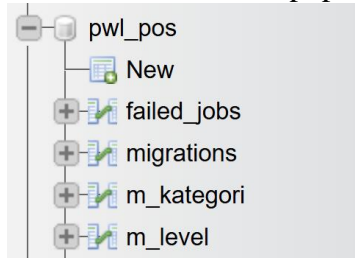
5. Simpan kode pada tahapan 4 tersebut, kemudian jalankan perintah ini pada terminal VSCode untuk melakukan migrasi

```
PS C:\laragon\www\PWL_2025\POS> php artisan migrate

INFO Preparing database.

2014_10_12_000000_create_users_table ..... 36ms DONE
2014_10_12_000000_create_users_table ..... 36ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 17ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 35ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 51ms DONE
2025_03_17_185306_create_m_level_table ..... 36ms DONE
```

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



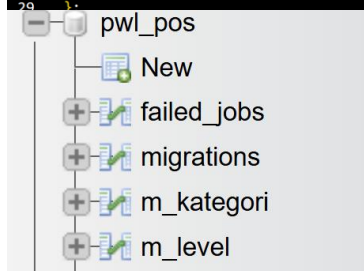
7. Ok, table sudah dibuat di database
8. Buat table database dengan migration untuk table m_kategori yang sama-sama tidak memiliki foreign key

```
PS C:\laragon\www\PWL_2025\POS> php artisan make:migration create_m_kategori_table --create=m_kategori

INFO Migration [C:\laragon\www\PWL_2025\POS\database\migrations\2025_03_17_185838_create_m_kategori_table.php] created successfully.

database > migrations > 2025_03_17_185838_create_m_kategori_table.php > class > up > Closure

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_kategori', function (Blueprint $table) {
15               $table->unsignedBigInteger('kategori_id', 20) -> autoIncrement();
16               $table->string('kategori_kode', 10)->unique();
17               $table->string('kategori_nama', 100);
18               $table->timestamps();
19           });
20       }
21
22       /**
23        * Reverse the migrations.
24        */
25       public function down(): void
26       {
27           Schema::dropIfExists('m_kategori');
28       }
29   }
```



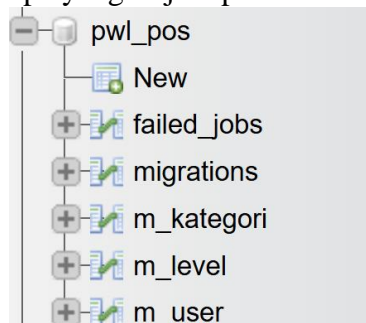
9. Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.

Praktikum 2.2 - Pembuatan file migrasi dengan relasi

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

```
database > migrations > 2025_03_17_190713_create_m_user_table.php > class > up > Closure
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_user', function (Blueprint $table) {
15              $table->id('user_id');
16              $table->unsignedBigInteger('level_id')->index(); // indexing u
17              $table->string('username', 20)->unique(); // unique untuk mema
18              $table->string('nama', 100);
19              $table->string('password');
20              $table->timestamps();
21
22              // mendefinisikan foreign key pada kolom level_id mengacu pada
23              $table->foreign('level_id')->references('level_id')->on('m_lev
24          });
25      }
26
27      /**
28       * Reverse the migrations.
29       */
30      public function down(): void
31      {
32          Schema::dropIfExists('m_user');
```

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

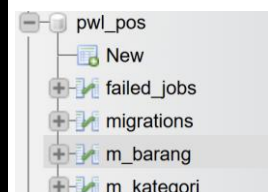


4. Buat table database dengan migration untuk table-tabel yang memiliki foreign key

- **M_barang**

```
12  public function up(): void
13  {
14      Schema::create('m_barang', function (Blueprint $table) {
15          $table->id('barang_id')->autoIncrement();
16          $table->unsignedBigInteger('kategori_id')->index();
17          // foreign key kategori_id -> m_kategori.kategori_id
18          $table->foreign('kategori_id')->references('kategori_id')->on('
19          $table->string('barang_kode', 10);
20          $table->string('barang_nama', 100);
21          $table->integer('harga_beli');
22          $table->integer('harga_jual');
23          $table->timestamps();
24      });
25  }
26
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS  + v
INFO Migration [C:\laragon\www\PWL_2025\POS\database\migrations\2025_03_17_
223736_create_m_barang_table.php] created successfully.
PS C:\laragon\www\PWL_2025\POS> php artisan migrate
INFO Running migrations.
2025_03_17_223736_create_m_barang_table ..... 216ms DONE
PS C:\laragon\www\PWL_2025\POS>
```



- **T_penjualan**

```

2025_03_17_224622_create_t_penjualan_table.php > PHP Intelephense > CreateTPenjualanTable > up >
6
7 class CreateTPenjualanTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('t_penjualan', function (Blueprint $table) {
17             $table->bigIncrements('penjualan_id'); // Primary key + auto i
18             $table->unsignedBigInteger('user_id'); // Foreign key
19             $table->string('pembeli', 50);
20             $table->string('penjualan_kode', 20);
21             $table->dateTime('penjualan_tanggal');
22             $table->timestamps();
23
24             // Optional: tambahkan foreign key constraint kalau tabel `use
25             $table->foreign('user_id')->references('id')->on('users')->onD
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      *
32      * @return void
33     }
34 }

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS + v ...

PS C:\laragon\www\PWL_2025\POS> php artisan migrate

INFO Running migrations.

INFO Running migrations.

2025_03_17_224622_create_t_penjualan_table 80ms **DONE**

- **T_stok**

```

database > migrations > 2025_03_17_231937_create_t_stok_table.php > class > up > Closure
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('t_stok', function (Blueprint $table) {
15             $table->id('stok_id');
16             $table->unsignedBigInteger('barang_id')->index();
17             $table->foreign('barang_id')->references('barang_id')->on('m_b
18             $table->unsignedBigInteger('user_id')->index();
19             $table->foreign('user_id')->references('id')->on('users');
20             $table->dateTime('stok_tanggal');
21             $table->integer('stok_jumlah');
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28     }
29 }

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS + v

--create=t_stok

INFO Migration [C:\laragon\www\PWL_2025\POS\database\migrations\2025_03_17_231937_create_t_stok_table.php] created successfully.

PS C:\laragon\www\PWL_2025\POS> php artisan migrate

INFO Running migrations.

2025_03_17_231937_create_t_stok_table 128ms **DONE**

- **T_penjualan detail**

```

database > migrations > 2025_03_17_232315_create_t_penjualan_detail_table.php > class > up > Closure
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('t_penjualan_detail', function (Blueprint $table) {
15             $table->id('detail_id');
16             $table->unsignedBigInteger('penjualan_id')->index();
17             $table->foreign('penjualan_id')->references('penjualan_id')->on('m_b
18             $table->unsignedBigInteger('barang_id')->index();
19             $table->foreign('barang_id')->references('barang_id')->on('m_b
20             $table->integer('harga');
21             $table->integer('jumlah');
22             $table->timestamps();
23         });
24     }
25 }

```

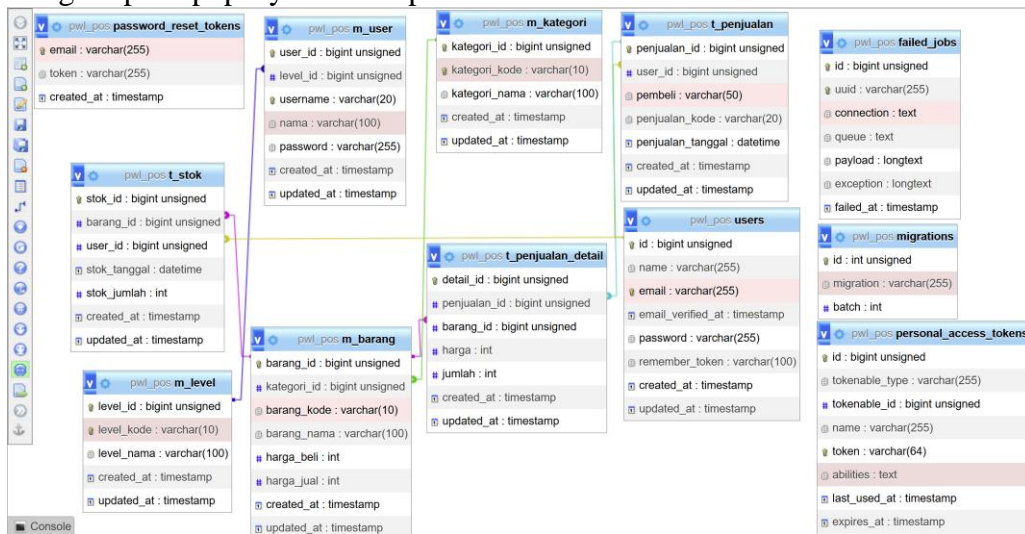
232315_create_t_penjualan_detail_table.php] created successfully.

PS C:\laragon\www\PWL_2025\POS> php artisan migrate

INFO Running migrations.

2025_03_17_232315_create_t_penjualan_detail_table 126ms DONE

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 3 – Membuat file seeder

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

```
PS C:\laragon\www\PWL_2025\POS> php artisan make:seeder LevelSeeder
```

INFO Seeder [C:\laragon\www\PWL_2025\POS\database\seeders\LevelSeeder.php] created successfully.

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

```
class LevelSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [
            ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Admin'],
            ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manag'],
            ['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staf']
        ];

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

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

```
PS C:\laragon\www\PWL_2025\POS> php artisan make:seeder LevelSeeder

INFO Seeder [C:\laragon\www\PWL_2025\POS\database\seeders\LevelSeeder.php]
created successfully.

PS C:\laragon\www\PWL_2025\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 <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> 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
6. Modifikasi file class UserSeeder seperti berikut

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

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

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```
PS C:\laragon\www\PWL_2025\POS> php artisan make:seeder UserSeeder

INFO Seeder [C:\laragon\www\PWL_2025\POS\database\seeders\UserSeeder.php]
reated successfully.

PS C:\laragon\www\PWL_2025\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\$QAga2on8UQlsUJON/61MgOXPEFG/GnxgtsPzKsllcL...	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	2	2	manager	Manager	\$2y\$12\$xq90HMnYdyPtnFPGFLe1e4TFPi9NSrib74jPjw5...	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	3	3	staff	Staff/Kasir	\$2y\$12\$KY28mYO4nEO6cxGvOKdFuGYPC/6dyZFNPtQo64JwR5...	NULL	NULL

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

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

No	Nama Tabel	Jumlah Data	Keterangan
1	m_kategori	5	5 kategori barang
2	m_barang	10	10 barang yang berbeda
3	t_stok	10	Stok untuk 10 barang
4	t_penjualan	10	10 transaksi penjualan
5	t_penjualan_detail	30	3 barang untuk setiap transaksi penjualan

- M_kategori

```
database > seeders > KategoriSeeder.php > PHP Intepense > KategoriSeeder > run
9 class KategoriSeeder extends Seeder
14 {
15     public function run(): void
16     {
17         $data = [
18             [
19                 'kategori_id' => 1,
20                 'kategori_kode' => 'KAT001',
21                 'kategori_nama' => 'Makanan',
22             ],
23             [
24                 'kategori_id' => 2,
25                 'kategori_kode' => 'KAT002',
26                 'kategori_nama' => 'Minuman',
27             ],
28             [
29                 'kategori_id' => 3,
30                 'kategori_kode' => 'KAT003',
31                 'kategori_nama' => 'Obat',
32             ],
33             [
34                 'kategori_id' => 4,
35                 'kategori_kode' => 'KAT004',
36                 'kategori_nama' => 'Elektronik',
37             ],
38             [
39                 'kategori_id' => 5,
40                 'kategori_kode' => 'KAT005',
41                 'kategori_nama' => 'Meubel',
42             ],
43         ];
44     }
45 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS +v ... ^ :
INFO Seeder [C:\laragon\www\PWL_2025\POS\database\seeders\KategoriSeeder.php] created successfully.
PS C:\laragon\www\PWL_2025\POS> php artisan db:seed --class=KategoriSeeder
INFO Seeding database.

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

- M_barang

database > seeders > BarangSeeder.php > PHP Intelephense > BarangSeeder > run

```

9  class BarangSeeder extends Seeder
14 public function run(): void
69     'harga_beli' => 3000,
70     'harga_jual' => 5000,
71     'kategori_id' => 3,
72 },
73 [
74     'barang_id' => '8',
75     'barang_kode' => 'BRG008',
76     'barang_nama' => 'Neozep',
77     'harga_beli' => 2500,
78     'harga_jual' => 4000,
79     'kategori_id' => 3,
80 ],
81 [
82     'barang_id' => '9',
83     'barang_kode' => 'BRG009',
84     'barang_nama' => 'Earphone',
85     'harga_beli' => 125000,
86     'harga_jual' => 128000,
87     'kategori_id' => 4,
88 ],
89 [
90     'barang_id' => '10',
91     'barang_kode' => 'BRG010',
92     'barang_nama' => 'Iphone 15',
93     'harga_beli' => 15000000,
94     'harga_jual' => 16000000,
95     'kategori_id' => 4,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS + v ... ^ x

PS C:\laragon\www\PWL_2025\POS> php artisan db:seed --class=BarangSeeder

INFO Seeding database.

	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	5	BRG001	Rak Buku	25000	30000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	5	BRG002	Container Barang	47500	50000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1	BRG003	Indomie Goreng	2500	3000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	2	BRG004	Cimory Yougurt	5500	8000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	2	BRG005	Milku Susu Strobery	4500	6000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	2	BRG006	Kopi Kapal Api	5000	7000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	3	BRG007	Tolak Angin	3000	5000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	3	BRG008	Neozep	2500	4000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	4	BRG009	Earphone	125000	128000	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	4	BRG010	Iphone 15	15000000	16000000	NULL	NULL

- T_stok

database > seeders > StokSeeder.php > PHP Intelephense > StokSeeder > run

```

1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class StokSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $data = [];
17         for ($i = 1; $i <= 10; $i++) {
18             $data[] = [
19                 'stok_id' => $i,
20                 'barang_id' => rand(1, 10),
21                 'user_id' => rand(1, 3),
22                 'stok_tanggal' => now(),
23                 'stok_jumlah' => rand(10, 50),
24             ];
25         }
26         DB::table('t_stok')->insert($data);
27     }
28 }
29

```

	stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	6	3	2025-03-18 22:45:10	18	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	5	3	2025-03-18 22:45:10	35	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	9	1	2025-03-18 22:45:10	16	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	7	2	2025-03-18 22:45:10	25	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	8	2	2025-03-18 22:45:10	18	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	6	4	1	2025-03-18 22:45:10	25	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	5	3	2025-03-18 22:45:10	31	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	8	1	3	2025-03-18 22:45:10	39	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	9	2	1	2025-03-18 22:45:10	44	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	10	4	2	2025-03-18 22:45:10	16	NULL	NULL

- T_penjualan

```

database > seeders > PenjualanSeeder.php > PHP Intelephense > PenjualanSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Seeder;
6  use Illuminate\Support\Facades\DB;
7
8  class PenjualanSeeder extends Seeder
9  {
10     public function run(): void
11     {
12         $data = [
13             [
14                 'penjualan_id' => 1,
15                 'user_id' => 1,
16                 'pembeli' => 'Faiza',
17                 'penjualan_kode' => 'TRX001',
18                 'penjualan_tanggal' => '2023-01-03 10:15:00'
19             ],
20             [
21                 'penjualan_id' => 2,
22                 'user_id' => 1,
23                 'pembeli' => 'Ana',
24                 'penjualan_kode' => 'TRX002',
25                 'penjualan_tanggal' => '2023-01-05 14:30:00'
26             ],
27             [
28                 'penjualan_id' => 3,
29                 'user_id' => 1,
30                 'pembeli' => 'Thasya',
31                 'penjualan_kode' => 'TRX003',
32                 'penjualan_tanggal' => '2023-01-07 09:45:00'
33             ],
34             [

```

	penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	Faiza	TRX001	2023-01-03 10:15:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	1	Ana	TRX002	2023-01-05 14:30:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	1	Thasya	TRX003	2023-01-07 09:45:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	1	Eka	TRX004	2023-01-10 16:00:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	1	Falen	TRX005	2023-01-12 11:20:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	6	1	Izung	TRX006	2023-01-15 17:10:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	1	Azul	TRX007	2023-01-18 13:40:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	8	1	Elang	TRX008	2023-01-20 08:50:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	9	1	Doni	TRX009	2023-01-25 15:05:00	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	10	1	Agus	TRX010	2023-01-28 12:25:00	NULL	NULL

- T_penjualan_detail

```

database > seeders > PenjualanDetailSeeder.php > PHP Intelephense > PenjualanDetailSeeder > run
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8
9 class PenjualanDetailSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $penjualan_ids = DB::table('t_penjualan')->pluck('penjualan_id');
17         $barang_ids = DB::table('m_barang')->pluck('barang_id');
18
19         $data = [];
20         $detail_id = 1;
21
22         foreach ($penjualan_ids as $penjualan_id) {
23             $barang_acak = $barang_ids->shuffle()->take(3);
24
25             foreach ($barang_acak as $barang_id) {
26                 $harga = rand(10000, 50000);
27                 $jumlah = rand(1, 5);
28
29                 $data[] = [
30                     'detail_id' => $detail_id++,
31                     'penjualan_id' => $penjualan_id,
32                     'barang_id' => $barang_id,
33                     'harga' => $harga,
34                     'jumlah' => $jumlah,
35                 ];
36             }
37         }
38         DB::table('t_penjualan_detail')->insert($data);
39     }
40 }

```

	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1	2	16514	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	1	3	38951	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	1	6	26436	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	2	5	42921	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	2	8	37595	2	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	6	2	6	13499	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	3	8	12000	3	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	8	3	7	41706	1	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	9	3	1	18501	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	10	4	4	33189	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	11	4	6	39772	1	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	12	4	7	35245	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	13	5	6	28960	5	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	14	5	9	47911	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	15	5	7	39872	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	16	6	10	22328	1	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	17	6	7	20637	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	18	6	9	14526	4	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	19	7	9	25017	3	NULL	NULL

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

Praktikum 4 – Implementasi DB Façade

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

```

// Akses URL: '/', menjalankan function yang mengembalikan view
Route::get('/', function () {
    return view('welcome');
});

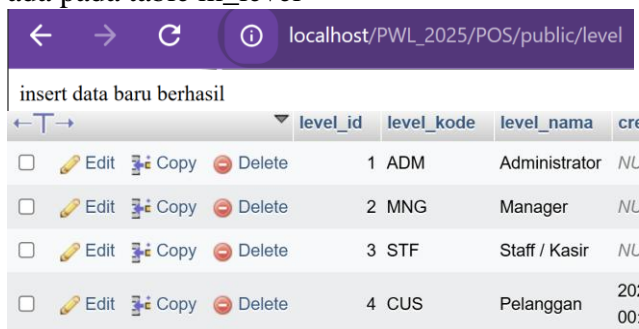
// Akses URL: '/level', menjalankan method 'index' di LevelController
Route::get('/level', [LevelController::class, 'index']);

```

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

```
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
13
14         return 'insert data baru berhasil';
15     }
16 }
```

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



	level_id	level_kode	level_nama	cre
<input type="checkbox"/> Edit Copy Delete	1	ADM	Administrator	NL
<input type="checkbox"/> Edit Copy Delete	2	MNG	Manager	NL
<input type="checkbox"/> Edit Copy Delete	3	STF	Staff / Kasir	NL
<input type="checkbox"/> Edit Copy Delete	4	CUS	Pelanggan	20:00:

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

```
namespace App\Http\Controllers;

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

class LevelController extends Controller
{
    public function index()
    {
        // DB::insert('insert into m_level(level_kode, level_nama, created
        // return 'insert data baru berhasil';

        $row = DB::update('UPDATE m_level SET level_nama = ? WHERE level_k
        return 'Update data berhasil. Jumlah data yang diupdate: ' . $row
    }
}
```

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



	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
<input type="checkbox"/> Edit Copy Delete	4	CUS	Customer	2025-03-19 00:06:02	NULL

Update data berhasil. Jumlah data yang diupdate: 1 baris

	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
<input type="checkbox"/> Edit Copy Delete	4	CUS	Customer	2025-03-19 00:06:02	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';
```

← → ↻ ⓘ localhost/PWL_2025/POS/public/level

Delete data berhasil. Jumlah data yang dihapus: 1 baris

	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

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

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

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

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <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

← → ↻ ⓘ localhost/PWL_2025/POS/public/level

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 5 – Implementasi Query Builder

1. Kita buat controller dahulu 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

```

app > Http > Controllers > KategoriController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class KategoriController extends Controller
9  {
10     public function index()
11     {
12         $data = [
13             'kategori_kode' => 'SNK',
14             'kategori_nama' => 'Snack/Makanan Ringan',
15             'created_at' => now()
16         ];
17         DB::table('m_kategori')->insert($data);
18         return 'Insert data kategori baru berhasil ditambahkan';
19     }

```

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

← → ↻ ⓘ localhost/PWL_2025/POS/public/kategori

Insert data kategori baru berhasil ditambahkan

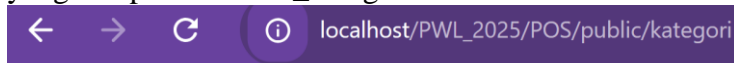
5. Selanjutnya, kita modifikasi lagi file KategoriController untuk meng-update data di table m_kategori seperti berikut

```

$row = DB::table('m_kategori')->where('kategori_kode', 'SNK')-
>update(['kategori_nama' => 'Camilan']);
return 'Update data kategori 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 kategori berhasil. Jumlah data yang diupdate: 1 baris

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	KAT001	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KAT002	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KAT003	Obat	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KAT004	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KAT005	Meubel	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	SNK	Camilan	2025-03-19 03:29:31	NULL

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

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

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

```
<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>  
    </foreach>  
  </table>  
</body>
```

Delete data kategori berhasil. Jumlah data yang dihapus: 1 baris

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

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

Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	KAT001	Makanan
2	KAT002	Minuman
3	KAT003	Obat
4	KAT004	Elektronik
5	KAT005	Meubel

10. Silahkan dicoba pada browser dan amati apa yang terjadi.
11. Laporkan hasil Praktikum-5 ini dan commit perubahan pada git

Praktikum 6 – Implementasi Eloquent ORM

1. Kita buat file model untuk tabel m_user dengan mengetikkan perintah
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 = 'users';
    protected $primaryKey = 'user_id';
}
```

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

```
<?php

use App\Http\Controllers\KategoriController;
use App\Http\Controllers\LevelController;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;

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

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

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

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

6. Kemudian kita buat view user.blade.php

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

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

Data User

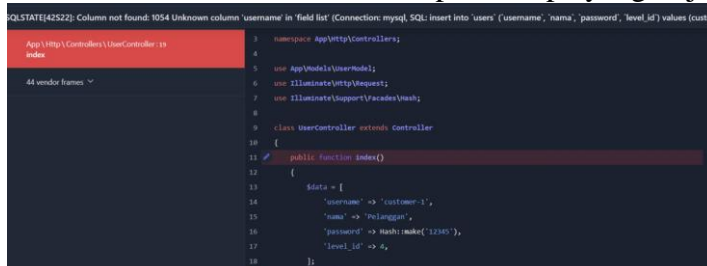
ID	Username	Nama	ID Level Pengguna
----	----------	------	-------------------

8. Setelah itu, kita modifikasi lagi file UserController

```
class UserController extends Controller
{
    public function index()
    {
        $data = [
            'username' => 'customer-1',
            'nama' => 'Pelanggan',
            'password' => Hash::make('12345'),
            'level_id' => 4,
        ];
        UserModel::insert($data);

        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}
```

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



10. Kita modifikasi lagi file UserController menjadi seperti berikut

```
public function index()
{
    $data = [
        'nama' => 'Pelanggan Pertama',
    ];
    UserModel::where('username', 'customer-1')->update($data);

    $user = UserModel::all();
    return view('user', ['data' => $user]);
}
```

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

Data User

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

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

PERTANYAAN

- Pada Praktikum 1 - Tahap 5, apakah fungsi dari APP_KEY pada file setting .env Laravel?**
 - APP_KEY merupakan kunci enkripsi yang digunakan untuk beberapa fungsi keamanan
- Pada Praktikum 1, bagaimana kita men-generate nilai untuk APP_KEY?**
 - Tahapan men-generate nilai untuk APP_KEY: •
 - Buka terminal
 - Jalankan perintah php artisan key:generate
 - Setelah dijalankan, Laravel akan menghasilkan APP_KEY baru dan otomatis menyimpannya di file .env
- Pada Praktikum 2.1 - Tahap 1, secara default Laravel memiliki berapa file migrasi? dan untuk apa saja file migrasi tersebut?**
 - Secara default, Laravel memiliki 4 file migrasi yaitu:
 - create_users_table.php (untuk menyimpan data pengguna)
 - create_password_reset_tokens_table.php (untuk menyimpan token reset password)
 - create_failed_jobs_table.php (menyimpan antrian pekerjaan (jobs) yang gagal dieksekusi.)
 - create_personal_access_tokens_table.php (untuk API Authentication dengan Laravel Sanctum.)
- Secara default, file migrasi terdapat kode \$table->timestamps();, apa tujuan/output dari fungsi tersebut?**
 - \$table->timestamps(); berfungsi untuk otomatis menambahkan dua kolom ke dalam tabel database

- `created_at` : Menyimpan waktu saat pertama kali data dibuat
- `updated_at`: Menyimpan waktu saat terakhir data diperbarui.

- Pada File Migrasi, terdapat fungsi `$table->id()`; Tipe data apa yang dihasilkan dari fungsi tersebut?**
 - Tipe data `BIGINT` (unsigned big integer) yang bersifat `AUTO INCREMENT` dan digunakan sebagai `PRIMARY KEY`
- Apa bedanya hasil migrasi pada table `m_level`, antara menggunakan `$table->id()`; dengan menggunakan `$table->id('level_id')`; ?**
 - `$table->id()`; digunakan jika ingin nama primary key default (`id`)
 - `$table->id('level_id')`; digunakan jika ingin mencustom nama kolom menjadi `'level_id'`
- Pada migration, Fungsi `->unique()` digunakan untuk apa?**
 - Fungsi `->unique()` digunakan untuk memastikan bahwa nilai dalam suatu kolom tidak ada duplikat di dalam tabel database.
- Pada Praktikum 2.2 - Tahap 2, kenapa kolom `level_id` pada tabel `m_user` menggunakan `$tabel->unsignedBigInteger('level_id')`, sedangkan kolom `level_id` pada tabel `m_level` menggunakan `$tabel->id('level_id')` ?**
 - kolom `level_id` pada tabel `m_user` menggunakan `$tabel->unsignedBigInteger('level_id')` karena berfungsi sebagai primary key
 - kolom `level_id` pada tabel `m_level` menggunakan `$tabel->id('level_id')` karena berfungsi sebagai foreign key yang merujuk ke `level_id` di tabel `m_level`.
- Pada Praktikum 3 - Tahap 6, apa tujuan dari Class Hash? dan apa maksud dari kode program `Hash::make('1234')`;**
 - Hash digunakan untuk mengenkripsi data agar lebih aman
 - `Hash::make('1234')`; berfungsi untuk menghasilkan hash dari password `'1234'`
- Pada Praktikum 4 - Tahap 3/5/7, pada query builder terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?**
 - digunakan sebagai parameter binding untuk menghindari SQL Injection.
- Pada Praktikum 6 - Tahap 3, apa tujuan penulisan kode `protected $table = 'm_user'`; dan `protected $primaryKey = 'user_id'`; ?**
 - `protected $table = 'm_user'`; bertujuan untuk menentukan model tersebut berhubungan dengan tabel `m_user`
 - `protected $primaryKey = 'user_id'`; bertujuan untuk mengubah primary key default (`id`) menjadi `user_id`
- Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (DB Façade / Query Builder / Eloquent ORM) ? jelaskan**

- Menurut saya operasi CRUD lebih mudah digunakan pada Query Builder karena penulisan sitaksnya hampir sama dengan penulisan query pada database itu sendiri