

POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420

http://www.polinema.ac.id

Mata Kuliah : Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 6 (enam)

Pertemuan ke- : 1 (satu)

Fallujah Ramadi Camshah – 09 SIB 2B

JOBSHEET 03

MIGRATION, SEEDER, DB FAÇADE, QUERY BUILDER, dan ELOQUENT ORM

Sebelumnya kita sudah membahas mengenai *Routing, Controller*, dan *View* yang ada di Laravel. Sebelum kita masuk pada pembuatan aplikasi berbasis website, alangkah baiknya kita perlu menyiapkan Basis data sebagai tempat menyimpan data-data pada aplikasi kita nanti. Selain itu, umumnya kita perlu menyiapkan juga data awal yang kita gunakan sebelum membuat aplikasi, seperti data user administrator, data pengaturan sistem, dll.

Untuk itu, kita memerlukan teknik untuk merancang/membuat table basis data sebelum membuat aplikasi. Laravel memiliki fitur dalam pengelolaan basis data seperti, migration, seeder, model, dll.

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

Jadi kita bikin project Laravel 10 dengan nama **PWL_POS**.

Project **PWL_POS** akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajarai

A. PENGATURAN DATABASE

Database atau basis data menjadi komponen penting dalam membangun sistem. Hal ini dikarenakan database menjadi tempat untuk menyimpan data-data transaksi yang ada pada sistem. Koneksi ke database perlu kita atur agar sesuai dengan database yang kita gunakan.

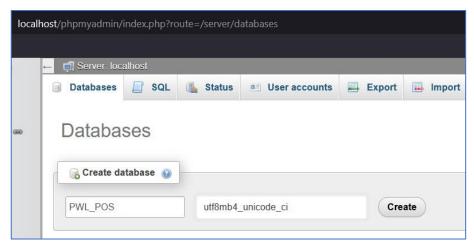
Praktikum 1 - pengaturan database:

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

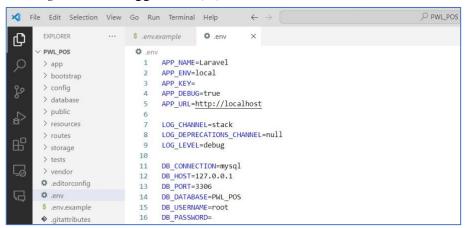


POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

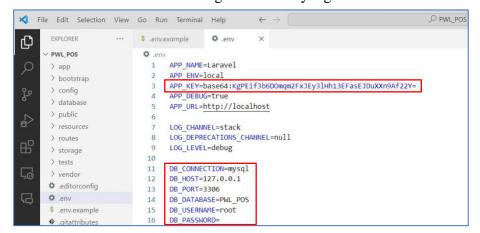
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id



- 2. Buka aplikasi VSCode dan buka folder project PWL_POS yang sudah kita buat
- 3. Copy file .env.example menjadi .env
- 4. Buka file .env, dan pastikan konfigurasi APP_KEY bernilai. Jika belum bernilai silahkan kalian *generate* menggunakan php artisan.



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



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

POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420

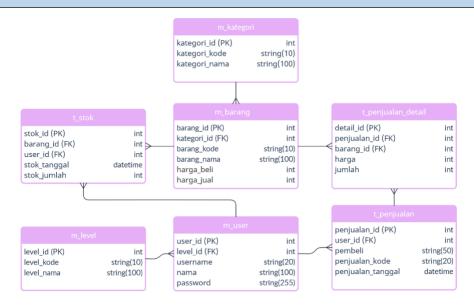
http://www.polinema.ac.id

B. MIGRATION

Migration pada Laravel merupakan sebuah fitur yang dapat membantu kita mengelola database secara efisien dengan menggunakan kode program. Migration membantu kita dalam membuat (create), mengubah (edit), dan menghapus (delete) struktur tabel dan kolom pada database yang sudah kita buat dengan cepat dan mudah. Dengan Migration, kita juga dapat melakukan perubahan pada struktur database tanpa harus menghapus data yang ada.

Salah satu keunggulan menggunakan migration adalah mempermudah proses instalasi aplikasi kita, Ketika aplikasi yang kita buat akan diimplementasikan di server/komputer lain.

Sesuai dengan topik pembelajaran kita untuk membangun sistem *Point of Sales (PoS)* sederhana, maka kita perlu membuat migration sesuai desain database yang sudah didefinisikan pada file Studi Kasus PWL.pdf



Dalam membuat file migration di Laravel, yang perlu kita perhatikan adalah struktur table yang ingin kita buat.

TIPS MIGRATION

Buatlah file migration untuk table yang tidak memiliki relasi (table yang tidak ada foreign key) dulu, dan dilanjutkan dengan membuat file migrasi yang memiliki relasi yang sedikit, dan dilanjut ke file migrasi dengan table yang memiliki relasi yang banyak.

Dari tips di atas, kita dapat melakukan cek untuk desain database yang sudah ada dengan mengetahui jumlah foreign key yang ada. Dan kita bisa menentukan table mana yang akan kita buat migrasinya terlebih dahulu.

POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

No Urut	Nama Tabel	Jumlah FK
1	m_level	0
2	m_kategori	0
3	m_user	1
4	m_barang	1
5	t_penjualan	1
6	t_stok	2
7	t_penjualan_detail	2

INFO

Secara default Laravel sudah ada table **users** untuk menyimpan data pengguna, tapi pada praktikum ini, kita gunakan table sesuai dari file **Studi Kasus PWL.pdf** yaitu **m_user.**

Pembuatan file migrasi bisa menggunakan 2 cara, yaitu

a. Menggunakan artisan untuk membuat file migration

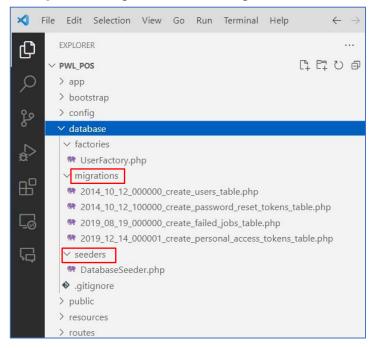
```
php artisan make:migration <nama-file-tabel> --create=<nama-tabel>
```

b. Menggunakan artisan untuk membuat file model + file migration

```
php artisan make:model <nama-model> -m
```

Perintah -m di atas adalah *shorthand* untuk opsi membuat file migrasi berdasarkan model yang dibuat.

Pada Laravel, file-file *migration* ataupun *seeder* berada pada folder PWL_POS/database



NEGERI PARTE S

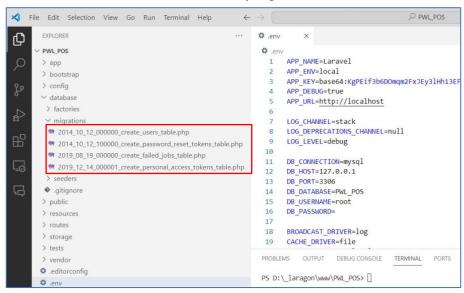
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

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

```
php artisan make:migration create_m_level_table --create=m_level
```

```
2024_02_25_133526_create_m_level_table.php ×
database > migrations > * 2024_02_25_133526_create_m_level_table.php > ...
  1
  2
      use Illuminate\Database\Migrations\Migration;
  3
  4
      use Illuminate\Database\Schema\Blueprint;
      use Illuminate\Support\Facades\Schema;
  6
       return new class extends Migration
  8
  9
           * Run the migrations.
 10
 11
           public function up(): void
 12
 13
 14
               Schema::create('m level', function (Blueprint $table) {
 15
                   $table->id();
 16
                   $table->timestamps();
 17
 18
 19
 20
            * Reverse the migrations.
 21
 22
 23
           public function down(): void
 24
 25
               Schema::dropIfExists('m_level');
 26
 27
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

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

```
return new class extends Migration
 8
9
10
          * Run the migrations.
11
         public function up(): void
12
13
14
              Schema::create('m_level', function (Blueprint $table) {
                 $table->id('level_id');
15
                  $table->string('level_kode', 10)->unique();
16
17
                  $table->string('level_nama', 100);
18
                  $table->timestamps();
19
              });
20
21
22
23
          * Reverse the migrations.
24
25
         public function down(): void
26
27
              Schema::dropIfExists('m level');
28
29
     };
```

INFO

Dalam fitur migration Laravel, terdapat berbagai macam function untuk membuat kolom di table database. Silahkan cek disini

https://laravel.com/docs/10.x/migrations#available-column-types

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. $(0341)\ 404424-404425,$ Fax $(0341)\ 404420$

http://www.polinema.ac.id

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*

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

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

```
php artisan make:migration create_m_user_table --table=m_user
```

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

```
return new class extends Migration
  8
  9
                                      * Run the migrations.
10
11
12
                                    public function up(): void
13
                                                   Schema::create('m_user', function (Blueprint $table) {
    $table->id('user_id');
}
14
15
                                                                    $table->unsignedBigInteger('level_id')->index(); // indexing untuk ForeignKey
16
17
                                                                   \theta = \pi (username', 20) - unique(); // unique untuk memastikan tidak ada username yang sama variatikan yang sama variatikan yang sama variatikan tidak ada username yang sama variatikan yang sama yang sama
                                                                  $table->string('nama', 100);
$table->string('password');
18
19
                                                                   $table->timestamps();
20
21
22
                                                                    // 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');
23
24
                                                   });
25
26
27
28
                                        * Reverse the migrations.
29
                                    public function down(): void
30
31
32
                                                    Schema::dropIfExists('m_user');
33
34
```



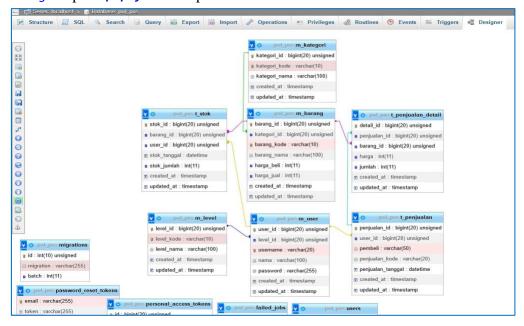
POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

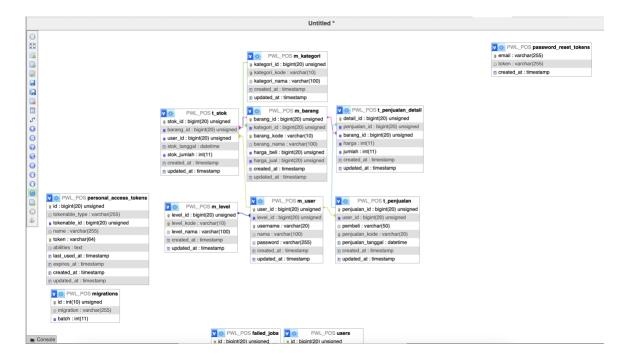
- 3. Simpan kode program Langkah 2, dan jalankan perintah php artisan migrate. Amati apa yang terjadi pada database.
- 4. Buat table database dengan migration untuk table-tabel yang memiliki foreign key

m_barang
t_penjualan
t_stok
t_penjualan_detail

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.



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420

http://www.polinema.ac.id

C. SEEDER

Seeder merupakan sebuah fitur yang memungkinkan kita untuk mengisi database kita dengan data awal atau data dummy yang telah ditentukan. Seeder memungkinkan kita untuk membuat data awal yang sama untuk setiap penggunaan dalam pembangunan aplikasi. Umumnya, data yang sering dibuat seeder adalah data penggunna karena data tersebut akan digunakan saat aplikasi pertama kali di jalankan dan membutuhkan aksi *login*.

1. Perintah umum dalam **membuat** file seeder adalah seperti berikut

```
php artisan make:seeder <nama-class-seeder>
```

Perintah tersebut akan men-generate file seeder pada folder PWL POS/database/seeders

2. Dan perintah untuk **menjalankan** file seeder seperti berikut

```
php artisan db:seed --class=<nama-class-seeder>
```

Dalam proses pengembangan suatu aplikasi, seringkali kita membutuhkan data awal tiruan atau dummy data untuk memudahkan pengujian dan pengembangan aplikasi kita. Sehingga fitur *seeder* bisa kita pakai dalam membuat sebuah aplikasi web.

Praktikum 3 – Membuat file *seeder*

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

```
php artisan make:seeder LevelSeeder
       EXPLORER
                                 M LevelSeeder.php X
     PWL POS
                   四日()日日
                                 database > seeders > ♥ LevelSeeder.php > ...
                                   1
                                       <?php
      > app
      > bootstrap
                                       namespace Database\Seeders;
      > confia

✓ database

                                   5
                                       use Illuminate\Database\Console\Seeds\WithoutModelEvents;
        > factories
                                   6
                                       use Illuminate\Database\Seeder:
        > migrations
                                   8 class LevelSeeder extends Seeder
        v seeders
品
                                   9
        natabaseSeeder.php
                                  10
        TevelSeeder.php
                                             * Run the database seeds.
                                  11
口
       · gitignore
                                  12
       > public
                                  13
                                            public function run(): void
品
       > resources
                                  14
                                                11
       > routes
                                  15
       > storage
                                  16
                                  17
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

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

```
** LevelSeeder.php X
PWL_POS
              [ ☐ ☐ database > seeders > M LevelSeeder.php >
> app
> bootstrap
                                      namespace Database\Seeders:
> config

√ database

                                      use Illuminate\Database\Console\Seeds\WithoutModelEvents;
 > factories
                                       use Illuminate\Database\Seeder;
 > migrations
                                      use Illuminate\Support\Facades\DB;
 ∨ seeders
 M DatabaseSeeder.php
                                 11
                                             * Run the database seeds.
> public
                                 13
14
> resources
                                            public function run(): void
                                 15
> routes
                                 16
> storage
                                 17
                                                     ['level_id' => 1, 'level_kode' => 'ADM', 'level_nama' => 'Administrator'],
> tests
                                                     ['level_id' => 2, 'level_kode' => 'MNG', 'level_nama' => 'Manager'],
['level_id' => 3, 'level_kode' => 'STF', 'level_nama' => 'Staff/kasir'],
                                 18
> vendor
                                 19
.editorconfig
.env
                                                DB::table('m_level')->insert($data);
                                 21
  .env.example
```

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



4. Ketika *seeder* berhasil dijalankan maka akan tampil data pada table m level



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

```
php artisan make:seeder UserSeeder
```

SERVICE NEGERIA DE LE SERVICE DE LA CONTROL DE LA CONTROL

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 http://www.polinema.ac.id

6. Modifikasi file class UserSeeder seperti berikut

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

7. Jalankan perintah untuk mengeksekusi class UserSeeder

```
php artisan db:seed --class=UserSeeder
```

8. Perhatikan hasil seeder pada table m user



- 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

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

kate	gori													
←_	Γ→				~	kategor	i_id	kategori	_kode	kategor	i_nama	creat	ed_at	updated_at
	<i>⊘</i> Ed	lit 🚁 (Сору	D	elete		1	ELEC		Elektron	ik	NULL		NULL
	Ø Ed	lit ≩≟ (Сору	D	elete		2	FASH		Fashion		NULL		NULL
	<i>⊘</i> Ec	it ₃∔i (Сору	D	elete		3	FOOD		Makanaı	n & Minuma	an <i>NULL</i>		NULL
	Ø Ec	- lit ₃∔i (Сору	D	elete		4	HOME		Peralata	n Rumah	NULL		NULL
	<i>⊘</i> Ec	it ₃≟(Сору	D	elete		5	AUTO		Otomotif		NULL		NULL
Bara		-												
←T	_			▼ bara	ang_id	kategori	_id ba	arang_kode	barang	nama h	narga_beli l	harga_jual	created_	at updated_at
	Ø Edit	с Сору	Dele	ete	1		1 T\	/001	Televisi 4	12 inch	3000000	3500000	NULL	NULL
	Ø Edit	с Сору	Dele	ete	2		1 H	P002	Smartph	one 6 inch	2500000	2800000	NULL	NULL
	Edit	с Сору	Dele	ete	3		2 Sh	HO003	Sepatu S	Sneakers	400000	500000	NULL	NULL
	Ø Edit	c Copy	Dele	ete	4		2 TS	SH004	Kaos Co	tton	150000	200000	NULL	NULL
	Edit	с Сору	Dele	ete	5		3 M	IL005	Susu UH	T 1L	25000	30000	NULL	NULL
	@ Edit	с Сору	Dele	ete	6		3 IN	S006	Mie Insta	เท	3000	5000	NULL	NULL
	Ø Edit	Сору	Dele	ete	7		4 Bl	_D007	Blender		350000	400000	NULL	NULL
	@ Edit	с Сору	Dele	ete	8		4 IR	N008	Setrika		150000		NULL	NULL
	Ø Edit	Е Сору	Dele	ete	9		5 OI	L009	Oli Motor	r 1L	75000	90000	NULL	NULL
		с Сору	Dele	ete	10		5 T	/R010	Ban Moto	or	250000	300000) NULL	NULL
Stok	ζ.													
← □	\rightarrow			~	stok_	id bara	ang_id	user_id	stok_t	anggal	stok_ju	ımlah c	reated_at	updated_at
	Edit	≩ € Cop	оу 🥥	Delete		1		1	1 2025-0	2-20 15:26	:49	20 N	ULL	NULL
	Edit	≩ € Cop	оу 🥥	Delete		2		2	2 2025-0	2-15 15:26	:49	30 N	ULL	NULL
	Edit	≩ € Cop	оу 🥥	Delete		3		3	1 2025-0	2-22 15:26	:49	25 N	ULL	NULL
	Edit	3 € Cop	оу 🥥	Delete		4		4	3 2025-0	2-25 15:26	:49	15 N	ULL	NULL
	Edit	≩ Cop	оу 🥥	Delete		5		5	2 2025-0	2-18 15:26	:49	50 N	ULL	NULL
	Edit	₹ Cop	оу 🥥	Delete		6		6	1 2025-0	2-12 15:26	:49	10 N	ULL	NULL
	Edit	≩ € Cop	оу 🥥	Delete		7		7	3 2025-0	2-10 15:26	:49	40 N	ULL	NULL
	Edit	₹ Cop	оу 🥥	Delete		8		8	1 2025-0	2-23 15:26	:49	35 N	ULL	NULL
	Edit	≩ € Cop	оу 🥥	Delete		9		9	2 2025-0	2-21 15:26	:49	28 N	ULL	NULL
	Edit	3 € Cop	оу 🥥	Delete		10	1	0	3 2025-0	2-16 15:26	:49	22 N	ULL	NULL
Pen	jualan													
$\leftarrow T$	→			▼ p	enjuala	n_id u	ser_id	pembeli	penjual	an_kode	penjualan_t	anggal	created_at	updated_at
	Edit	≩ € Copy	D	elete		1	1	I Andi	TRX000	1	2025-02-28	15:29:26	NULL	NULL
	Edit	≩ € Copy	(a) D	elete		2	2	2 Budi	TRX000	2	2025-02-26	15:29:26	NULL	NULL
	Edit	≩ € Copy	D	elete		3	3	3 Citra	TRX000	3	2025-02-24	15:29:26	NULL	NULL
	Edit	≩ сору	(a) D	elete		4	1	Dewi	TRX000	4	2025-03-01	15:29:26	NULL	NULL
	Edit	≩ € Copy	(a) D	elete		5	2	2 Eka	TRX000	5	2025-02-27	15:29:26	NULL	NULL
	Edit	≩ € Copy	(a) D	elete		6	3	3 Fajar	TRX000	6	2025-02-25	15:29:26	NULL	NULL
	Edit	≩ € Copy	D	elete		7	-	Gita	TRX000	7	2025-02-23	15:29:26	NULL	NULL
	Edit	3 € Copy	(a) D	elete		8	2	2 Hadi	TRX000	8	2025-02-21	15:29:26	NULL	NULL
	Edit	- Copy	D	elete		9	3	3 Indah	TRX000	9	2025-02-19	15:29:26	NULL	NULL
		_ Gopy	(a) D	elete		10		Joko	TRX001	0	2025-02-17	15:29:26	NULL	NULL



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

Penjualan detail

←Τ	→		~	detail_id	penjualan_id	barang_id	harga	jumlah	created_at	updated_at
	Edit	3 € Copy	Delete	1	1	1	50000	2	NULL	NULL
	Edit	3 € Сору	Delete	2	1	2	30000	1	NULL	NULL
	Edit	3 € Сору	Delete	3	1	3	40000	3	NULL	NULL
	Edit	3 € Сору	Delete	4	2	2	30000	2	NULL	NULL
	Edit	3 € Сору	Delete	5	2	4	25000	1	NULL	NULL
	Edit	3 € Copy	Delete	6	2	5	60000	-1	NULL	NULL
	Edit	3 - € Сору	Delete	7	3	6	45000	1	NULL	NULL
	Edit	3 - сору	Delete	8	3	7	70000	2	NULL	NULL
	Edit	3 € Copy	Delete	9	3	8	50000	2	NULL	NULL
	Edit	3 € Copy	Delete	10	4	9	60000	1	NULL	NULL
	Edit	3 € Copy	Delete	11	4	10	80000	3	NULL	NULL
	Edit	3 € Сору	Delete	12	4	1	50000	1	NULL	NULL
	Edit	3 € Copy	Delete	13	5	3	40000	2	NULL	NULL
	Edit	3 € Сору	Delete	14	5	5	60000	1	NULL	NULL
	Edit	3 € Copy	Delete	15	5	7	70000	1	NULL	NULL
	Edit	3 € Сору	Delete	16	6	2	30000	1	NULL	NULL
	Edit	3 € Copy	Delete	17	6	6	45000	2	NULL	NULL
	Edit	≩ сору	Delete	18	6	8	50000	2	NULL	NULL
	Edit	3 € Copy	Delete	19	7	1	50000	1	NULL	NULL
	Edit	≩ сору	Delete	20	7	3	40000	2	NULL	NULL
	Edit	≩ сору	Delete	21	7	9	60000	1	NULL	NULL
	Edit	≩ сору	Delete	22	8	4	25000	3	NULL	NULL
	Edit	≩ € Сору	Delete	23	8	5	60000	1	NULL	NULL
	Edit	≩ сору	Delete	24	8	10	80000	2	NULL	NULL
	Edit	≩ Copy	Delete	25	9	6	45000	1	NULL	NULL



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

D. DB FACADE

DB Façade merupakan fitur dari Laravel yang digunakan untuk melakukan *query* secara langsung dengan mengetikkan perinta SQL secara utuh (*raw query*). Disebut *raw query* (query mentah) karena penulisan query pada DB Façade langsung ditulis sebagaimana yang biasa dituliskan pada database, seperti "select * from m_user" atau "insert into m_user..." atau "update m_user set ... Where ..."

Raw query adalah cara paling dasar dan tradisional yang ada di Laravel. Raw query terasa familiar karena biasa kita pakai ketika melakukan query langsung ke database.

INFO

Dokumentasi penggunaan DB Façade bisa dicek di laman ini

https://laravel.com/docs/10.x/database#running-queries

Terdapat banyak method yang bisa digunakan pada DB Façade ini. Akan tetapi yang kita pelajari cukup 4 (empat) method yang umum dipakai, yaitu

a. DB::select()

Method ini digunakan untuk mengambil data dari database. Method ini mengembalikan (*return*) data hasil *query*. Contoh

```
DB::select('select * from m_user'); //Query semua data pada tabel m_user

DB::select('select * from m_user where level_id = ?', [1]); //Query tabel m_user dengan level_id = 1

DB::select('select * from m_user where level_id = ? and username = ?', [1, 'admin']);
```

b. DB::insert()

Method ini digunakan untuk memasukkan data pada table database. Method ini **tidak memiliki nilai pengembalian** (*no return*). Contoh

```
DB::insert('insert into m_level(level_kode, level_nama) values(?,?)', ['CUS', 'Pelanggan']);
```

c. DB::update()

Method ini digunakan saat menjalankan *raw query* untuk meng-update data pada database. Method ini **memiliki nilai pengembalian** (*return*) berupa jumlah baris data yang ter-*update*. Contoh

```
DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420

http://www.polinema.ac.id

d. DB::delete()

Method ini digunakan saat menjalankan *raw query* untuk menghapus data dari table. Method ini **memiliki nilai pengembalian** (*return*) berupa jumlah baris data yang telah dihapus. Contoh

```
DB::delete('delete from m_level where level_kode = ?', ['CUS']);
```

Ok, sekarang mari kita coba praktikkan menggunakan DB Façade pada project kita

Praktikum 4 – Implementasi DB Facade

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

```
php artisan make:controller LevelController
```

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

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

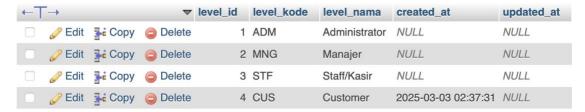
4. Kita coba jalankan di browser dengan url localhost/PWL_POS/public/level dan amati apa yang terjadi pada table m_level di database, *screenshot* perubahan yang ada pada table m_level



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

```
M LevelController.php X M web.php
app > Http > Controllers > ♥ LevelController.php > .
       <?php
       namespace App\Http\Controllers;
       use Illuminate\Http\Request:
       use Illuminate\Support\Facades\DB;
       class LevelController extends Controller
            public function index()
 10
 12
                 // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
 13
                 // return 'Insert data baru berhasil';
 14
                 $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
 16
 17
```

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



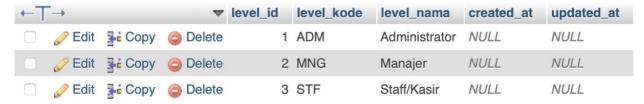
7. Kita coba modifikasi lagi file LevelController untuk melakukan proses hapus data



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

```
M LevelController.php X M web.php
app > Http > Controllers > ♥ LevelController.php > ધ LevelController > ۞ index
        namespace App\Http\Controllers;
        use Illuminate\Http\Request;
        use Illuminate\Support\Facades\DB;
        class LevelController extends Controller
             public function index()
 10
 11
                  // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
 13
                  // return 'Insert data baru berhasil';
 14
                  // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
// return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
 16
 17
                  $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
 18
 20
 21
```





POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

 $Telp.\ (0341)\ 404424-404425,\ Fax\ (0341)\ 404420$

http://www.polinema.ac.id

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

```
namespace App\Http\Controllers;
      use Illuminate\Http\Request;
      use Illuminate\Support\Facades\DB;
      class LevelController extends Controller
           public function index()
10
11
12
                // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
                // return 'Insert data baru berhasil';
14
                // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
// return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
15
16
                // $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
// return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
19
20
                $data = DB::select('select * from m_level');
                return view('level', ['data' => $data]);
23
24
```

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

```
♣ LevelController.php
♠ level.blade.php X
♠ web.php
resources > views > ♥ level.blade.php > ...
 1 <!DOCTYPE html>
           <title>Data Level Pengguna</title>
        </head>
  6
        <body>
           <h1>Data Level Pengguna</h1>
           8
 9
              TD
 10
                 Kode Level
 11
 12
                 Nama Level
              13
 14
              @foreach ($data as $d)
 15
                {{ $d->level_id }}
 16
                {{ $d->level_kode }}
 17
                {{ $d->level_nama }}
 18
 19
              20
              @endforeach
 21
           22
        </body>
```

10. Silahkan dicoba pada browser dan amati apa yang terjadi



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id



Data Level Pengguna

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

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

E. QUERY BUILDER

Query builder adalah fitur yang disediakan Laravel untuk melakukan proses CRUD (create, retrieve/read, update, delete) pada database. Berbeda dengan raw query pada DB Facede yang mengharuskan kita menulis perintah SQL, pada query builder perintah SQL ini diakses menggunakan method. Jadi, kita tidak menulis perintah SQL secara langsung, melainkan cukup memanggil method-method yang ada di query builder.

Query builder membuat kode kita menjadi rapi dan lebih mudah dibaca. Selain itu *query builder* tidak terikat ke satu jenis database, jadi query builder bisa digunakan untuk mengakses berbagai jenis database seperti MySQL, MariaDB, PostgreSQL, SQL Server, dll. Jika suatu saat ingin beralih dari database MySQL ke PostgreSQL, tidak akan banyak kendala. Namun kelemahan dari *query builder* adalah kita harus mengetahui method-method apa saja yang ada di *query builder*.

INFO

Dokumentasi penggunaan Query Builder pada Laravel bisa dicek di laman ini

https://laravel.com/docs/10.x/queries

Ciri khas *query builder* Laravel adalah kita tentukan dahulu target table yang akan kita akses untuk operasi CRUD.

DB::table('<nama-tabel>'); // query builder untuk melakukan operasi CRUD pada tabel yang dituju

Perintah pertama yang dilakukan pada query builder adalah menentukan nama table yang akan dilakukan operasi CRUD. Kemudian baru disusul method yang ingin digunakan sesuai dengan peruntukannya. Contoh

a. Perintah untuk *insert* data dengan method insert()

```
DB::table('m_kategori')->insert(['kategori_kode' => 'SMP', 'kategori_nama' => 'Smartphone']);
```

Query yang dihasilkan dari kode di atas adalah

```
insert into m_kategori(kategori_kode, kategori_nama) values('SMP', 'Smartphone');
```

b. Perintah untuk *update* data dengan method where() dan update()

```
DB::table('m_kategori')->where('kategori_id', 1)->update(['kategori_nama' => 'Makanan Ringan']);
```

Query yang dihasilkan dari kode di atas adalah

```
update m_kategori set kategori_nama = 'Makanan Ringan' where kategori_id = 1;
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

c. Perintah untuk *delete* data dengan method where() dan delete()

```
DB::table('m_kategori')->where('kategori_id', 9) ->delete();
```

Query yang dihasilkan dari kode di atas adalah

```
delete from m_kategori where kategori_id = 9;
```

d. Perintah untuk ambil data

Method Query Builder	Query yang dihasilkan
DB::table('m_kategori')->get();	select * from m_kategori
<pre>DB::table('m_kategori') ->where('kategori_id', 1)->get();</pre>	<pre>select * from m_kategori where kategori_id = 1;</pre>
<pre>DB::table('m_kategori') ->select('kategori_kode') ->where('kategori_id', 1)->get();</pre>	<pre>select kategori_kode from m_kategori where kategori_id = 1;</pre>

Praktikum 5 – Implementasi *Query Builder*

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

```
php artisan make:controller KategoriController
```

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

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

THE SERIES OF TH

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

http://www.polinema.ac.id

```
** KategoriController.php X ** level.blade.php
M LevelController.php
app > Http > Controllers > 🦛 KategoriController.php > ધ KategoriController > 🗞 index
       namespace App\Http\Controllers;
       use Illuminate\Http\Request;
       use Illuminate\Support\Facades\DB;
       class KategoriController extends Controller
            public function index()
 10
 11
 12
                $data = [
                     'kategori_kode' => 'SNK',
'kategori_nama' => 'Snack/Makanan Ringan',
 13
 14
 15
                     'created_at' => now()
 16
 17
                DB::table('m_kategori')->insert($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



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

```
app > Http > Controllers > 🤲 KategoriController.php > ધ KategoriController > 🛇 index
      namespace App\Http\Controllers;
      use Illuminate\Http\Request:
      use Illuminate\Support\Facades\DB;
 8
      class KategoriController extends Controller
           public function index()
 10
11
               /* $data = [
12
                   'kategori kode' => 'SNK',
13
                   'kategori_nama' => 'Snack/Makanan Ringan',
14
15
                   'created_at' => now()
16
              DB::table('m_kategori')->insert($data);
 17
18
              return 'Insert data baru berhasil'; */
19
              $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
20
 21
              return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
 22
```

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



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id



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



POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 - 404425, Fax (0341) 404420 http://www.polinema.ac.id

```
public function index()
12
             /* $data = [
                 'kategori_kode' => 'SNK',
14
                 'kategori_nama' => 'Snack/Makanan Ringan',
15
                 'created at' => now()
16
17
            DB::table('m kategori')->insert($data):
18
            return 'Insert data baru berhasil'; */
19
             // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
20
             // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
22
23
             $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
             return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
25

▼ kategori_id kategori_kode

                                                                  kategori_nama
                                                                                        created_at updated_at

    Ø Edit  
    ♣ Copy  
    Opelete

                                               1 ELEC
                                                                                                      NULL
                                                                   Elektronik
                                                                                        NULL

    Ø Edit  
    ♣ Copy  
    Opelete

                                                                                         NULL
                                                                                                      NULL
                                               2 FASH
                                                                   Fashion

    Ø Edit 
    ♣

    Copy 
    O Delete

                                               3 FOOD
                                                                   Makanan & Minuman NULL
                                                                                                      NULL
                                               4 HOME
                                                                   Peralatan Rumah
                                                                                        NULL
                                                                                                      NULL

    Ø Edit  
    ♣ Copy  
    Oplete

                                               5 AUTO
                                                                   Otomotif
                                                                                        NULL
                                                                                                      NULL
```

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

```
public function index()
11
                * $data = [
12
                   'kategori_kode' => 'SNK',
13
                   'kategori_nama' => 'Snack/Makanan Ringan',
14
15
                  'created_at' => now()
16
17
              DB::table('m_kategori')->insert($data);
18
              return 'Insert data baru berhasil'; */
20
              // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Camilan']);
              // return 'Update data berhasil. Jumlah data yang diupdate: ' . $row.' baris';
21
22
23
              // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
24
              // return 'Delete data berhasil. Jumlah data yang dihapus: ' . $row.' baris';
25
              $data = DB::table('m_kategori')->get();
return view('kategori', ['data' => $data]);
26
27
```

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



POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

```
🙀 kategori.blade.php > 🗭 html > 😭 body > 😭 table > 😭 tr > 😭 td
      <!DOCTYPE html>
              <title>Data Kategori Barang</title>
          </head>
          <body>
              <h1>Data Kategori Barang</h1>

                       ID
10
                       Kode Kategori
11
                       Nama Kategori
                   @foreach ($data as $d)

'\tak{{ $d->kategori_id }}

\{{ $d->kategori_kode }}

\{{ $d->kategori_nama }}

\{{ $d->kategori_nama }}

16
17
18
                   19
                  @endforeach
20
21
               </html>
```

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



Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	ELEC	Elektronik
2	FASH	Fashion
3	FOOD	Makanan & Minuman
4	HOME	Peralatan Rumah
5	AUTO	Otomotif

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

REGERIA PROPERTY PROP

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 http://www.polinema.ac.id

F. ELOQUENT ORM

Eloquent ORM adalah fitur bawaan dari laravel. Eloquent ORM adalah cara pengaksesan database dimana setiap baris tabel dianggap sebagai sebuah object. Kata ORM sendiri merupakan singkatan dari *Object-relational mapping*, yakni suatu teknik programming untuk mengkonversi data ke dalam bentuk object.

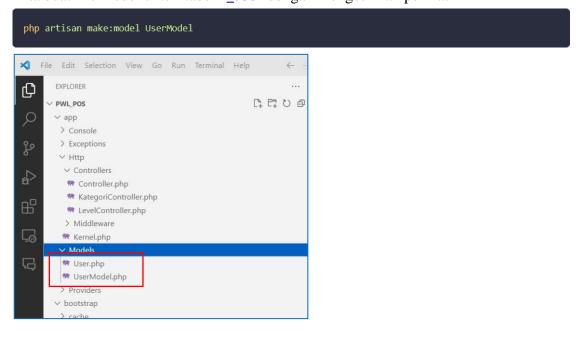
Eloquent ORM memerlukan Model untuk proses konversi data pada tabel menjadi object. Object inilah yang nantinya akan kita akses dari dalam controller. Oleh karena itu membuat Model pada Laravel berarti menggunakan Eloquent ORM. Silahkan cek disini https://laravel.com/docs/10.x/eloquent Perintah untuk membuat model adalah sebagai berikut

Untuk bisa melakukan operasi <u>CRUD</u> (create, read/retrieve, update, delete), kita harus membuat sebuah model sesuai dengan target tabel yang ingin digunakan. Jadi,

dalam 1 model, merepresentasikan 1 tabel database.

Praktikum 6 – Implementasi Eloquent ORM

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



LILL SEGERIA S

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 http://www.polinema.ac.id

- 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

```
app > Models > 🦛 UserModel.php > ધ UserModel
       <?php
  1
  2
  3
       namespace App\Models;
  4
       use Illuminate\Database\Eloquent\Factories\HasFactory;
      use Illuminate\Database\Eloquent\Model;
  6
       class UserModel extends Model
  8
  9
 10
            use HasFactory;
 11
 12
             protected $table = 'm_user';
                                                        // Mendefinisikan nama tabel yang digunakan oleh model ini
            protected $table = m_user; // mendefinisikan nama tabel yang digunakan oleh model i
protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan
 13
 14
 15
```

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

```
routes > 🦬 web.php > ...
  1
      <?php
  2
      use App\Http\Controllers\KategoriController;
      use App\Http\Controllers\LevelController;
      use App\Http\Controllers\UserController;
      use Illuminate\Support\Facades\Route;
      Route::get('/', function () {
 9
 10
          return view('welcome');
 11
 12
      Route::get('/level', [LevelController::class, 'index']);
 13
      Route::get('/kategori', [KategoriController::class, 'index']);
 14
      Route::get('/user', [UserController::class, 'index']);
```

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

```
app > Http > Controllers > 🦬 UserController.php > ...
 1
      <?php
 2
 3
      namespace App\Http\Controllers;
      use App\Models\UserModel;
 5
 6
      use Illuminate\Http\Request;
 8
      class UserController extends Controller
 9
10
          public function index()
11
               // coba_akses_model_UserModel
12
               $user = UserModel::all(); // ambil semua data dari tabel m_user
13
14
               return view('user', ['data' => $user]);
15
```

6. Kemudian kita buat view user.blade.php

THE SERIES OF TH

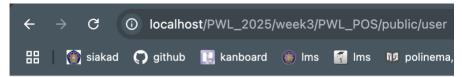
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

```
<!DOCTYPE html>
    <html>
      <head>
         <title>Data User</title>
      </head>
      <body>
         <h1>Data User</h1>
         10
11
               Username
               Nama
12
               ID Level Pengguna
13
            @foreach ($data as $d)
15
16
               {{ $d->user_id }}
17
               {{ $d->username }}
{{ $d->username }}
18
19
               {{ $d->level_id }}
            21
            @endforeach
22
          24
      </body>
```

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

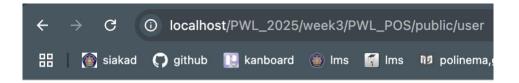
```
app > Http > Controllers > ♥ UserController.php > ...
       <?php
      namespace App\Http\Controllers;
      use App\Models\UserModel;
  6
      use Illuminate\Support\Facades\Hash;
  8
  9
       class UserController extends Controller
 10
           public function index()
11
 12
 13
                // tambah data user dengan Eloquent Model
14
                $data = [
 15
                    'username' => 'customer-1',
                     'nama' => 'Pelanggan',
16
                    'password' => Hash::make('12345'),
 17
 18
                     'level_id' => 4
 19
 20
               UserModel::insert($data); // tambahkan data ke tabel m user
 21
 22
                // coba akses model UserModel
               $user = UserModel::all(); // ambil semua data dari tabel m_user
return view('user', ['data' => $user]);
 23
 24
 25
 26
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

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



Data User

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

10. Kita modifikasi lagi file UserController menjadi seperti berikut

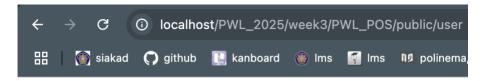
```
class UserController extends Controller
11
          public function index()
12
13
               // tambah data user dengan Eloquent Model
14
               $data = [
                   'nama' => 'Pelanggan Pertama',
15
16
17
               UserModel::where('username', 'customer-1')->update($data); // update data user
18
               // coba akses model UserModel
19
               *Suser = UserModel::all(); // ambil semua data dari tabel m_user return view('user', ['data' => $user]);
20
21
22
23
```



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

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
5	customer-1	Pelanggan Pertama	5

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

G. Penutup

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

- 1. Pada **Praktikum 1 Tahap 5**, apakah fungsi dari APP_KEY pada *file setting* .env Laravel?
 - ➤ APP_KEY di Laravel berfungsi untuk enkripsi data, keamanan session & token, serta validasi CSRF & signed URLs. Kunci ini memastikan bahwa data yang dienkripsi hanya bisa didekripsi oleh aplikasi Laravel yang memiliki APP_KEY yang sama.
- 2. Pada **Praktikum 1**, bagaimana kita men-generate nilai untuk APP KEY?
 - Dengan menggunakan perintah artisan '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?
 - Secara default, Laravel memiliki **tiga file migrasi**, yaitu:
- create_users_table.php → Untuk membuat table users yang menyimpan data pengguna.
- 2. **create_password_resets_table.php** → Untuk menyimpan token reset password.
- 3. **create_failed_jobs_table.php** → Untuk menyimpan informasi tentang pekerjaan (jobs) yang gagal dalam sistem queue.
- 4. Secara *default*, file migrasi terdapat kode \$table->timestamps();, apa tujuan/output dari fungsi tersebut?
 - ➤ Kode ini otomatis menambahkan **dua kolom timestamp** ke dalam tabel, yaitu:
- **created_at** → Menyimpan waktu data dibuat.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI POLITEKNIK NEGERI MALANG



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

- **updated at** → Menyimpan waktu data diperbarui.
- 5. Pada File Migrasi, terdapat fungsi \$table->id(); Tipe data apa yang dihasilkan dari fungsi tersebut?
 - Fungsi \$table->id(); akan menghasilkan **kolom primary key** dengan tipe data **unsigned big integer** (BIGINT UNSIGNED).
- 6. Apa bedanya hasil migrasi pada table m_level, antara menggunakan \$table->id(); dengan menggunakan \$table->id('level id'); ?
 - ➤ Jika \$table->id();, maka akan otomatis membuat kolom id sebagai primary key dengan tipe data BIGINT (UNSIGNED BIG INTEGER) dan auto-increment.

 Sedangkan jika menggunakan \$table->id('level_id'); akan membuat kolom level_id sebagai primary key dengan karakteristik yang sama, hanya namanya berubah sesuai yang ditentukan.
- 7. Pada migration, Fungsi ->unique() digunakan untuk apa?
 - Fungsi ->unique() digunakan untuk **membuat kolom memiliki nilai yang unik**, artinya tidak boleh ada data duplikat dalam kolom tersebut.
- 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') ?
- Di **tabel m_level**, kolom level_id dibuat dengan \$table->id('level_id');, yang berarti bertipe BIGINT UNSIGNED (primary key).
- Di **tabel m_user**, kolom level_id dibuat dengan \$table->unsignedBigInteger('level_id');, agar dapat mereferensikan level_id yang ada di m_level sebagai foreign key.Karena foreign key harus memiliki tipe data yang sama dengan primary key yang dirujuk
- 9. Pada **Praktikum 3 Tahap 6**, apa tujuan dari Class Hash? dan apa maksud dari kode program Hash::make('1234');?

Class Hash digunakan untuk meng-enkripsi password agar tidak tersimpan dalam bentuk teks biasa (plain text).

Hash::make('1234'); berfungsi untuk mengubah string "1234" menjadi hash yang aman sebelum disimpan ke database.

- 10. Pada **Praktikum 4 Tahap 3/5/7**, pada *query builder* terdapat tanda tanya (?), apa kegunaan dari tanda tanya (?) tersebut?
 - Tanda tanya (?) digunakan sebagai placeholder untuk nilai yang akan di-binding secara otomatis ke dalam query SQL.
- 11. Pada **Praktikum 6 Tahap 3**, apa tujuan penulisan kode protected \$table = 'm_user'; dan protected \$primaryKey = 'user_id'; ?
 - > protected \$table = 'm_user'; Menentukan bahwa model ini berhubungan dengan tabel m_user dalam database.



POLITEKNIK NEGERI MALANG JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141 Telp. (0341) 404424 – 404425, Fax (0341) 404420 http://www.polinema.ac.id

- ➤ **protected \$primaryKey = 'user_id'**; Menentukan bahwa primary key dari tabel ini adalah user_id, bukan id (default Laravel).
- 12. Menurut kalian, lebih mudah menggunakan mana dalam melakukan operasi CRUD ke database (*DB Façade / Query Builder / Eloquent ORM*) ? jelaskan
 - ➤ Eloquent ORM lebih mudah digunakan karena mendukung sintaks yang lebih sederhana dan intuitif. Dengan Eloquent, kita bisa melakukan query hanya dengan pemanggilan method yang mirip dengan objek. Selain itu, Eloquent juga mendukung relasi antar tabel secara langsung melalui model, sehingga lebih efisien dibandingkan Query Builder atau DB Facade.

*** Sekian, dan selamat belajar ***