

LAPORAN PRAKTIKUM III
PEMROGRAMAN WEB LANJUT



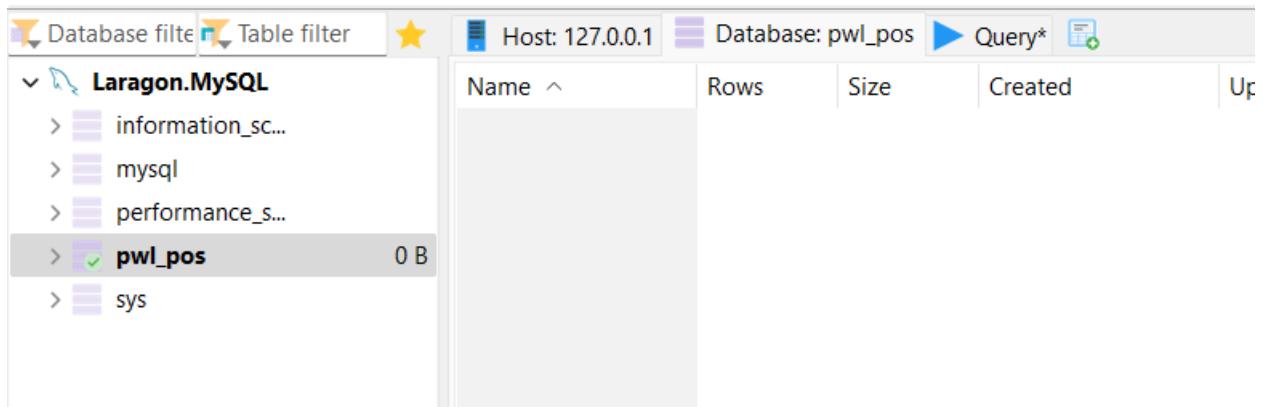
Disusun oleh:

Ricky Putra Pratama Tedjo (2241720204)

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024

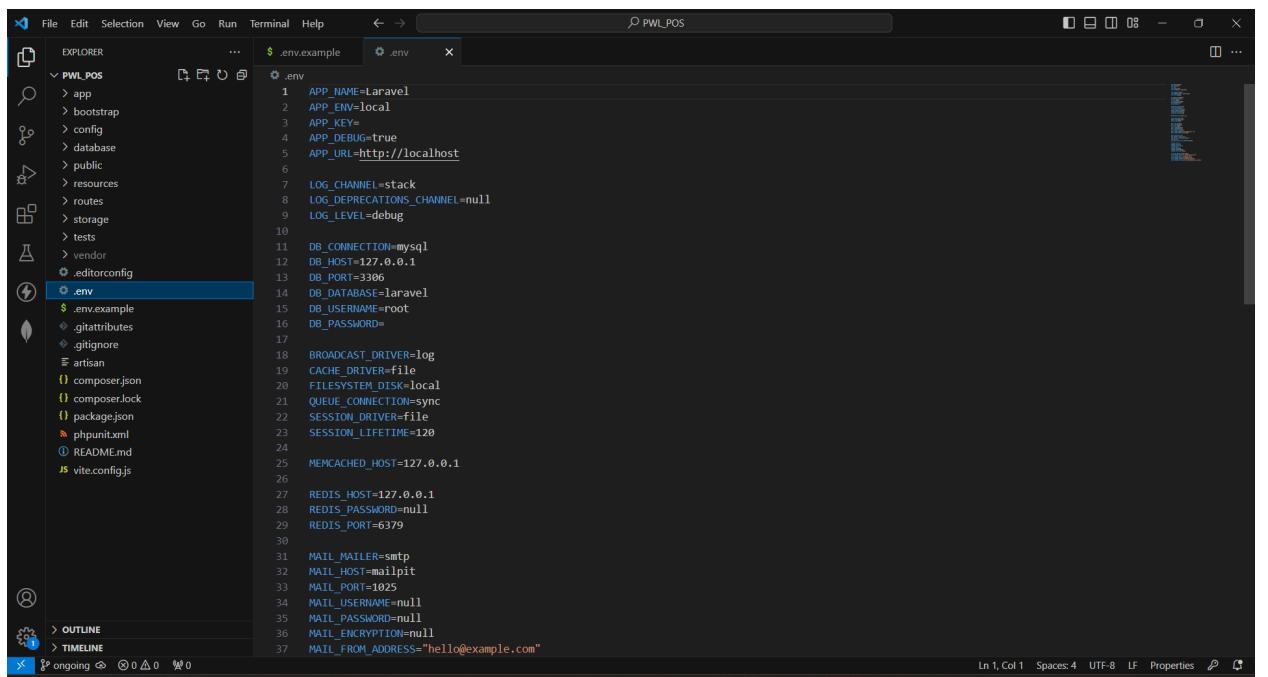
Praktikum 1 - Pengaturan Database

1.



The screenshot shows the Laragon MySQL interface. On the left, there is a tree view of databases under 'Laragon.MySQL': 'information_sc...', 'mysql', 'performance_s...', 'pwl_pos' (which is selected and highlighted in green), and 'sys'. On the right, there is a table with columns 'Name', 'Rows', 'Size', and 'Created'. The table currently has no data, showing '0 B' in the first row.

2.



The screenshot shows the VS Code interface with the '.env' file open in the editor. The file contains environment variables for a Laravel application:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
MEMCACHED_HOST=127.0.0.1
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtp
MAIL_HOST=mailtrap
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_FROM_ADDRESS="hello@example.com"
```

3.

4.

The screenshot shows the VS Code interface with the project folder 'PWL_POS' open in the Explorer sidebar. The terminal tab is active, displaying the command 'php artisan key:generate' and its output: 'INFO Application key set successfully.' The .env file is open in the editor, showing environment variables. The 'APP_URL' variable is highlighted with a red rectangle, indicating it has been modified. The variable value is 'http://localhost'. Other variables visible include APP_NAME=Laravel, APP_ENV=local, APP_KEY, APP_DEBUG=true, LOG_CHANNEL=stack, LOG_DEPRECATIONS_CHANNEL=null, LOG_LEVEL=debug, DB_CONNECTION=mysql, DB_HOST=127.0.0.1, DB_PORT=3306, DB_DATABASE=laravel, DB_USERNAME=root, DB_PASSWORD=, BROADCAST_DRIVER=log, CACHE_DRIVER=file, FILESYSTEM_DISK=local, QUEUE_CONNECTION=sync, SESSION_DRIVER=file, SESSION_LIFETIME=120, MEMCACHED_HOST=127.0.0.1, REDIS_HOST=127.0.0.1, REDIS_PASSWORD=null, and REDIS_PORT=6379.

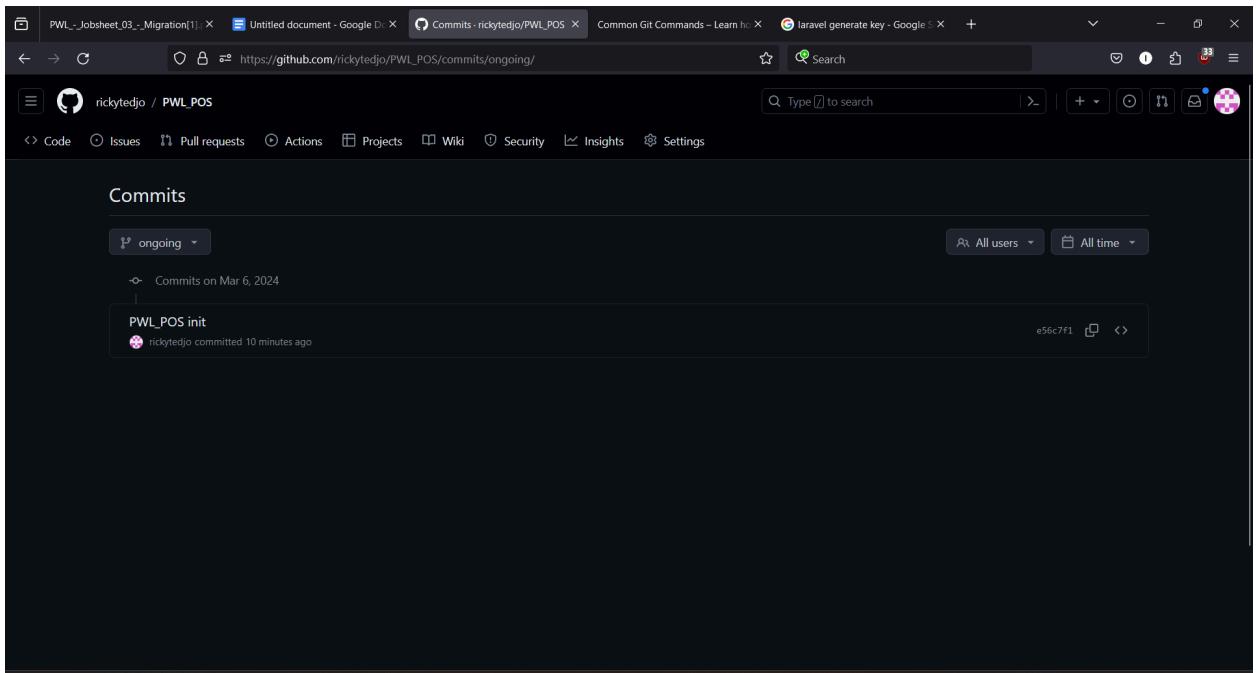
```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:4j/Hvy6MtVlnHJ37vgCSDDjPTBe2mApF3wTrTHYzpfE=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
```

4.

5.

The screenshot shows the VS Code interface with the project folder 'PWL_POS' open in the Explorer sidebar. The terminal tab is active, displaying the command 'php artisan key:generate' and its output: 'INFO Application key set successfully.' The .env file is open in the editor, showing environment variables. The 'APP_URL' variable is highlighted with a red rectangle, indicating it has been modified. The variable value is 'http://localhost'. A new variable 'MAIL_MAILER' is also present with a value of 'smtp'. Other variables visible include APP_DEBUG=true, LOG_CHANNEL=stack, LOG_DEPRECATIONS_CHANNEL=null, LOG_LEVEL=debug, DB_CONNECTION=mysql, DB_HOST=127.0.0.1, DB_PORT=3306, DB_DATABASE=PWL_POS, DB_USERNAME=root, DB_PASSWORD=, BROADCAST_DRIVER=log, CACHE_DRIVER=file, FILESYSTEM_DISK=local, QUEUE_CONNECTION=sync, SESSION_DRIVER=file, SESSION_LIFETIME=120, MEMCACHED_HOST=127.0.0.1, REDIS_HOST=127.0.0.1, REDIS_PASSWORD=null, REDIS_PORT=6379, MAIL_HOST=mailtrap, and MAIL_PORT=1025.

```
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=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailtrap
33 MAIL_PORT=1025
```



Praktikum 2 - Migrasi

2.1 Pembuatan file migrasi tanpa relasi

1.

```
File Edit Selection View Go Run Terminal Help < - > PWL_POS
EXPLORER PWL_POS
  app
  bootstrap
  config
  database
  factories
  migrations
    2014_10_12_000000_create_users_table.php
    2014_10_12_100000_create_password_reset_tokens_table.php
    2019_08_19_000000_create_failed_jobs_table.php
    2019_12_14_000001_create_personal_access_tokens_table.php
  seeder
  .gitignore
  public
  .htaccess
  favicon.ico
  index.php
  robots.txt
  resources
  routes
  storage
  tests
  vendor
  .editorconfig
  .env
  .env.example
  .gitattributes
  .gitignore
  artisan
  composer.json
  composer.lock
  OUTLINE
  TIMELINE
```

\$.env.example

```
env
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=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailpit
33 MAIL_PORT=1025
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\Code\PWL_POS>php artisan key:generate

[INFO] Application key set successfully.

D:\Code\PWL_POS>[]

Ln 14, Col 20 Spaces: 4 UTF-8 LF Properties

A screenshot of a terminal window in a code editor. The terminal shows the command 'php artisan key:generate' being run, which generates a new application key. The output indicates that the key was successfully set. The terminal window is part of a larger interface with an Explorer sidebar showing project files like migrations and .env files.

2.

The screenshot shows the VS Code interface with the following details:

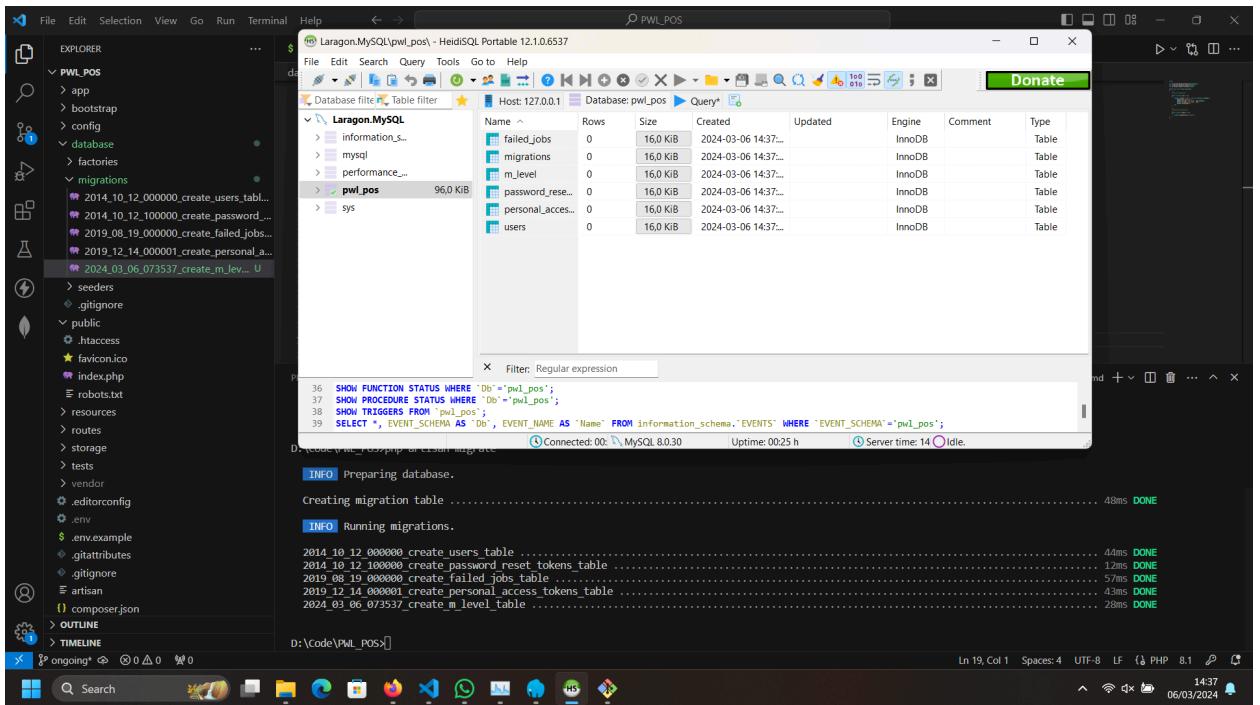
- File Explorer:** Shows the project structure under "PWL_POS".
- Editor:** Displays the code for the migration file `2024_03_06_073537_create_m_level_table.php`. The code defines a new class `2024_03_06_073537_create_m_level` that extends `Migration`. The `up()` method creates a table named `m_level` with columns `level_id`, `level_code` (unique), and `level_name`.
- Terminal:** Shows the command `php artisan key:generate` being run, followed by the message "Application key set successfully".
- Output:** Shows the command `php artisan make:migration create_m_level_table --create=m_level` being run, followed by the message "Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_073537_create_m_level_table.php] created successfully".
- Status Bar:** Shows the current file is `ongoing*`, with 0 errors and 0 warnings, and the PHP version is 8.1.

3.

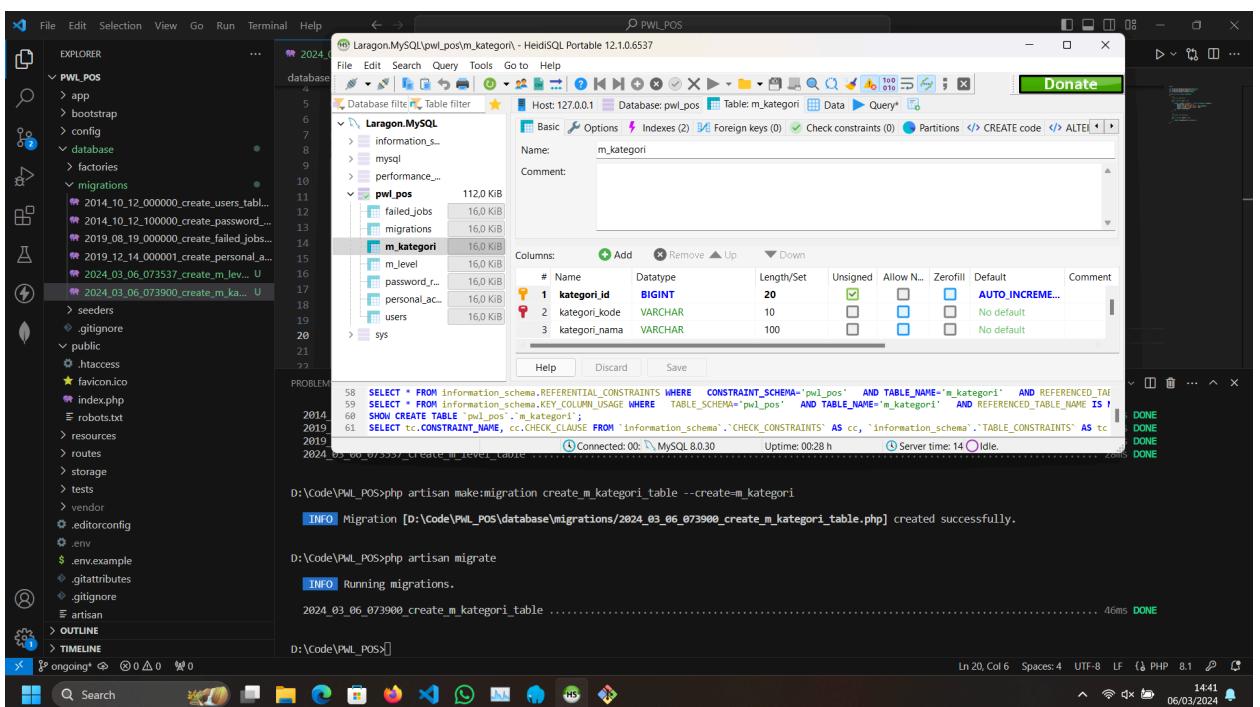
The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "PWL_POS".
- Editor:** Displays the code for the migration file `2024_03_06_073537_create_m_level_table.php`. The `up()` method now includes a closure to handle the unique constraint on `level_code`.
- Terminal:** Shows the command `php artisan key:generate` being run, followed by the message "Application key set successfully".
- Output:** Shows the command `php artisan make:migration create_m_level_table --create=m_level` being run, followed by the message "Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_073537_create_m_level_table.php] created successfully".
- Status Bar:** Shows the current file is `ongoing*`, with 0 errors and 0 warnings, and the PHP version is 8.1.

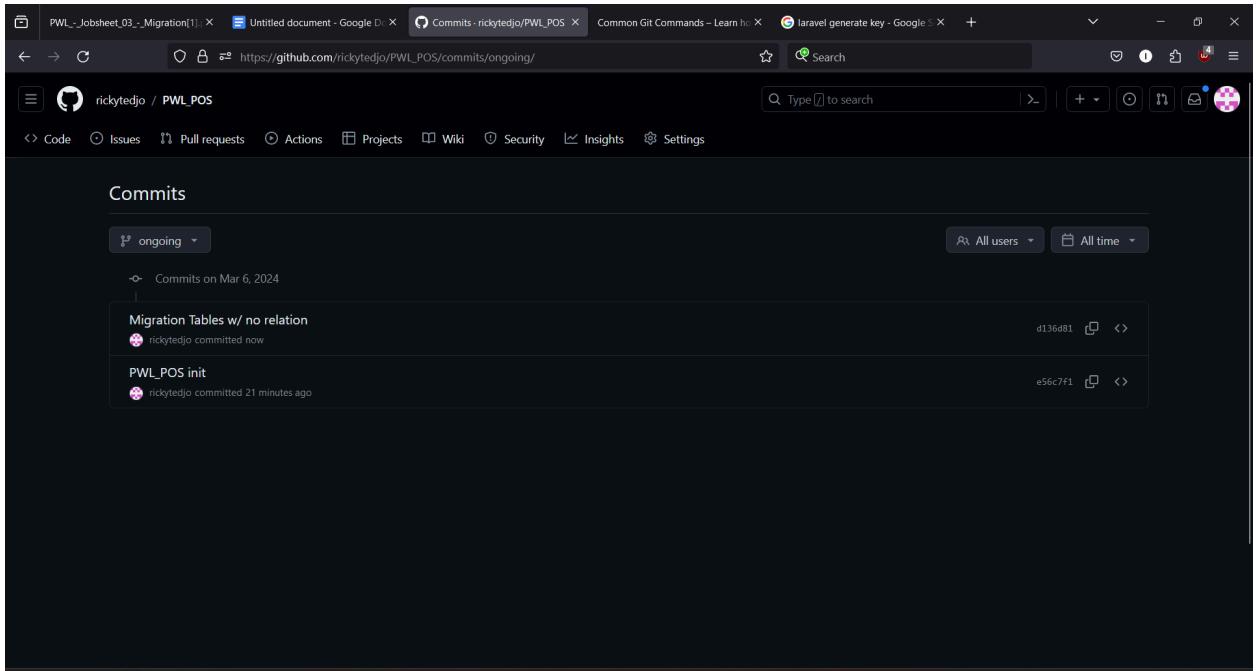
4.



5.



6.



2.2 Pembuatan file migrasi dengan relasi

1.

```

public function up(): void
{
    Schema::table('m_user', function (Blueprint $table) {
        //
        $table->id('user_id');
        $table->unsignedBigInteger('level_id')->index();
        $table->string('username',20)->unique();
        $table->string('name',100);
        $table->string('password');
        $table->timestamps();
    });
    $table->foreign('level_id')->references('level_id')->on('m_level');
}

/**
 * Reverse the migrations.
 */

```

D:\Code\PWL_POS>php artisan make:migration create_m_kategori_table --create=m_kategori
[INFO] Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_073900_create_m_kategori_table.php] created successfully.

D:\Code\PWL_POS>php artisan migrate
[INFO] Running migrations.
2024_03_06_073900_create_m_kategori_table 46ms DONE

D:\Code\PWL_POS>php artisan make:migration create_m_user_table --table=m_user
[INFO] Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_074421_create_m_user_table.php] created successfully.

2.

The screenshot shows a development environment with multiple windows open. On the left is the file explorer for a Laravel project named 'PWL_POS'. In the center, a database management tool (HeidiSQL) is open, showing the 'migrations' table. A specific migration file, '2024_03_06_074421_create_m_user_table.php', is selected. The table structure for 'm_user' is displayed, with columns: user_id (BIGINT, primary key, auto-increment), level_id (BIGINT), and username (VARCHAR(20)). The terminal window at the bottom shows the command 'php artisan migrate' being run, and the output indicates the migration was successful.

3. m_barang

The screenshot shows a development environment with multiple windows open. On the left is the file explorer for a Laravel project named 'PWL_POS'. In the center, a code editor (VS Code) is open, showing the 'migrations' directory. A specific migration file, '2024_03_06_075339_create_m_barang_table.php', is selected. The code defines a migration for creating the 'm_barang' table. The table has columns: barang_id (unsigned big integer, indexed), barang_kode (string, unique), barang_nama (string, length 100), harga_beli (integer), harga_jual (integer), and timestamps. It also includes foreign key relationships with 'm_kategori' and 't_penjualan'. The terminal window at the bottom shows the command 'php artisan make:migration create_t_penjualan_detail_table --create=t_penjualan_detail' being run, and the output indicates the migration was successfully created.

t_penjualan

t_stok

```

File Edit Selection View Go Run Terminal Help < - > PWL_POS
EXPLORER ... 2024_03_06_075339_create_m_barang_table.php U 2024_03_06_075723_create_t_penjualan_table.php U 2024_03_06_080131_create_t_stok_table.php U 2024_03_06_080622_create_t_penjualan_detail_table.php U
database > migrations > 2024_03_06_075723_create_t_penjualan_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->unsignedBigInteger('user_id')->index();
18             $table->string('stok_tanggal', 20)->unique();
19             $table->integer('stok_jumlah');
20             $table->timestamps();
21
22             $table->foreign('user_id')->references('user_id')->on('m_user');
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('t_stok');
32     }
33 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\Code\PWL_POS>php artisan make:migration create_t_stok_table --create=t_stok_table

[INFO] Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_080622_create_t_stok_table.php] created successfully.

D:\Code\PWL_POS>

4 selections Spaces: 4 UTF-8 LF (PHP 8.1 ⚡ 15:09 06/03/2024

t_penjualan_detail

```

File Edit Selection View Go Run Terminal Help < - > PWL_POS
EXPLORER ... 2024_03_06_075339_create_m_barang_table.php U 2024_03_06_075723_create_t_penjualan_table.php U 2024_03_06_080131_create_t_stok_table.php U 2024_03_06_080622_create_t_penjualan_detail_table.php U
database > migrations > 2024_03_06_080131_create_t_stok_table.php > class > up > Closure
10     /**
11      * Run the migrations.
12      */
13     public function up(): void
14     {
15         Schema::create('t_penjualan_detail', function (Blueprint $table) {
16             $table->id('penjualan_id');
17             $table->unsignedBigInteger('barang_id')->index();
18             $table->unsignedBigInteger('user_id')->index();
19             $table->dateTime('penjualan_tanggal');
20             $table->integer('stok_jumlah');
21             $table->timestamps();
22
23             $table->foreign('barang_id')->references('barang_id')->on('m_barang');
24             $table->foreign('user_id')->references('user_id')->on('m_user');
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      */
31     public function down(): void
32     {
33         Schema::dropIfExists('t_penjualan_detail');
34     }
35 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\Code\PWL_POS>php artisan make:migration create_t_penjualan_detail_table --create=t_penjualan_detail_table

[INFO] Migration [D:\Code\PWL_POS\database\migrations\2024_03_06_080622_create_t_penjualan_detail_table.php] created successfully.

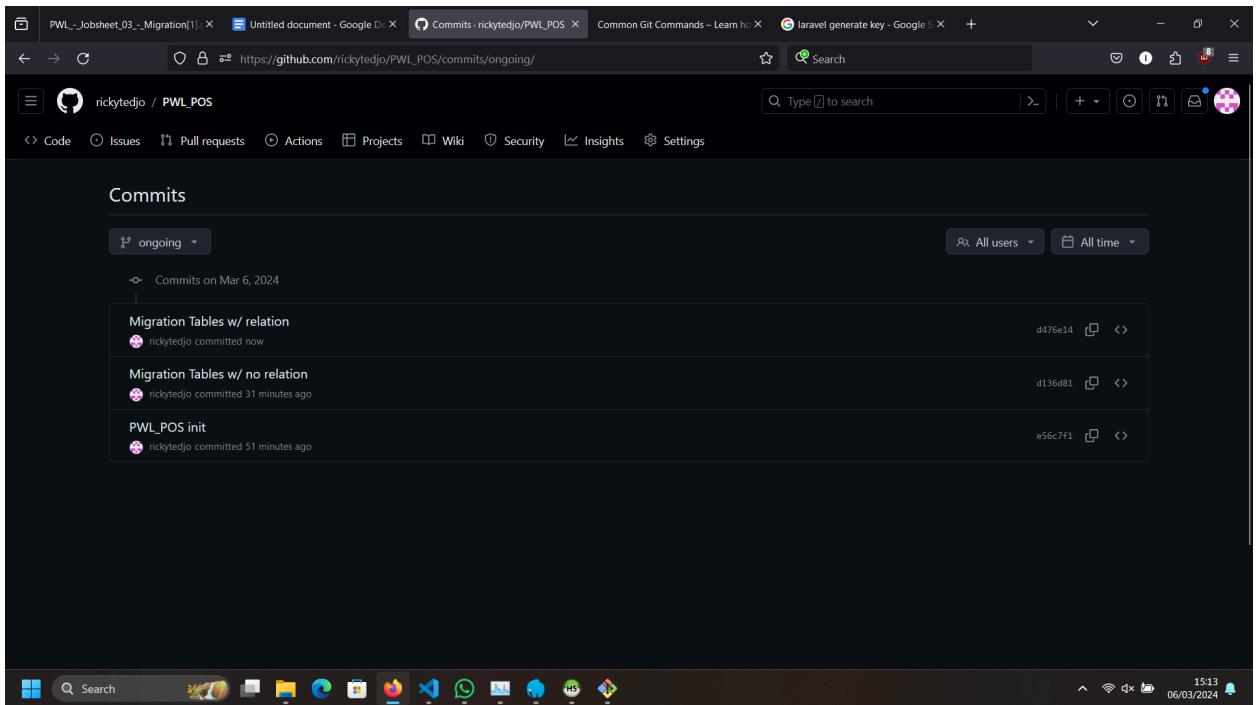
D:\Code\PWL_POS>

Ln 22, Col 80 Spaces: 4 UTF-8 LF (PHP 8.1 ⚡ 15:09 06/03/2024

Hasil Migrasi

Name	Rows	Size	Created	Updated	Engine	Comment	Type
failed_jobs	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table
migrations	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table
m_barang	0	32,0 KiB	2024-03-06 15:10:...		InnoDB		Table
m_kategori	0	16,0 KiB	2024-03-06 14:40:...		InnoDB		Table
m_level	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table
m_user	0	16,0 KiB	2024-03-06 14:52:...		InnoDB		Table
password_reset_tokens	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table
personal_access_tokens	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table
t_penjualan	0	32,0 KiB	2024-03-06 15:10:...		InnoDB		Table
t_penjualan_detail	0	48,0 KiB	2024-03-06 15:10:...		InnoDB		Table
t_stok	0	48,0 KiB	2024-03-06 15:10:...		InnoDB		Table
users	0	16,0 KiB	2024-03-06 14:37:...		InnoDB		Table

4.



Praktikum 3 - Seeder

1.

```

<?php
namespace Database\Seeders;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
class LevelSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        //
    }
}

```

2.

```
D:\Code\PWL_POS>php artisan db:seed --class=LevelSeeder
INFO  Seeding database.
```

Host: 127.0.0.1 Database: pwl_pos Table: m_level Data Query* Query #2* X

pwl_pos.m_level: 3 rows total (approximately)

level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	(NULL)	(NULL)
2	MNG	Manager	(NULL)	(NULL)
3	STF	Staff/Kasir	(NULL)	(NULL)

3.

```

UserSeeder.php U x
12  / * Run the database seeds.
13  */
14  0 references | 0 overrides
15  public function run(): void
16  {
17      $data = [
18          [
19              'user_id'=>1,
20              'level_id'=>1,
21              'username'=>'admin',
22              'nama'=>'Administrator',
23              'password'=>Hash::make('12345')
24          ],
25          [
26              'user_id'=>2,
27              'level_id'=>2,
28              'username'=>'manager',
29              'nama'=>'Manager',
30              'password'=>Hash::make('12345')
31          ],
32          [
33              'user_id'=>3,
34              'level_id'=>3,
35              'username'=>'staff',
36              'nama'=>'Staff/Kasir',
37              'password'=>Hash::make('12345')
38          ]
39      ];
40      DB::table('m_user')->insert($data);
41  }
42 }
```

Host: 127.0.0.1 Database: pwl_pos Table: m_user Data Query* Query #2* X

pwl_pos.m_user: 3 rows total (approximately)

user_id	level_id	username	nama	password	created_at	updated_at
1	1	admin	Administrator	\$2y\$12\$TQVe4ft11HxtwnCE6awkYeDXyE./jyc...	(NULL)	(NULL)
2	2	manager	Manager	\$2y\$12\$RVJzHtpJn1znloBBAAddT.M9HM3.ZDLbD...	(NULL)	(NULL)
3	3	staff	Staff/Kasir	\$2y\$12\$FUKxhvSCquWDNH1Rx1vGxeJTJ5I7YRB...	(NULL)	(NULL)

4. BarangSeeder

BarangSeeder.php

```

14     public function run(): void
15     {
16         $itemNames = [
17             'Levis', 'Emba',
18             'V-Neck Tee', 'Pocket Shirt',
19             'Pullover', 'Sleeveless',
20             'Woolen Hat', 'Wallet',
21             'Silk Boxer', 'Loose Boxer'
22         ];
23         $data = [];
24         for ($i = 0; $i < 10; $i++) {
25             $data[] = [
26                 'kategori_id' => ((($i + 1) <= 5) ? $i + 1 : $i - 4),
27                 'barang_kode' => 'BRG' . $i,
28                 'barang_nama' => $itemNames[$i],
29                 'harga_beli' => random_int(500, 1000),
30                 'harga_jual' => random_int(500, 1000)
31             ];
32         }
33         DB::table('m_barang')->insert($data);
34     }
35 }
36 
```

Laragon MySQL (pwl_pos|m_barang) - HeidiSQL Portable 12.1.0.6537

barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	cri
41	1	BRG1	Levis	990	677	0
42	2	BRG1	Emba	794	595	0
43	3	BRG2	V-Neck Tee	561	794	0
44	4	BRG3	Pocket Shirt	883	676	0
45	5	BRG4	Pullover	904	818	0
46	1	BRG5	Sleeveless	845	624	0
47	2	BRG6	Woolen Hat	560	609	0
48	3	BRG7	Wallet	844	697	0
49	4	BRG8	Silk Boxer	673	517	0
50	5	BRG9	Loose Boxer	667	781	0

Stok Seeder

StokSeeder.php

```

10     /**
11      * Run the database seeds.
12      */
13
14     public function run(): void
15     {
16         $data = [];
17         $userIds = DB::table('m_user')->pluck('user_id');
18         $barangIds = DB::table('m_barang')->pluck('barang_id');
19         for ($i = 1; $i <= 10; $i++) {
20             $data[] = [
21                 'barang_id' => $barangIds[$i-1],
22                 'user_id' => $userIds->random(), // diambilkan oleh
23                 'stok_tanggal' => now(),
24                 'stok_jumlah' => random_int(5, 50),
25             ];
26         }
27     }
28 }
29 
```

Laragon MySQL (pwl_pos|t_stok) - HeidiSQL Portable 12.1.0.6537

stok_id	barang_id	user_id	stok_tanggal	stok_jumlah	created_at	u
11	41	3	2024-03-06 08:45:35	19	(NULL)	0
12	46	2	2024-03-06 08:45:35	10	(NULL)	0
13	42	2	2024-03-06 08:45:35	42	(NULL)	0
14	47	3	2024-03-06 08:45:35	25	(NULL)	0
15	43	2	2024-03-06 08:45:35	40	(NULL)	0
16	48	2	2024-03-06 08:45:35	29	(NULL)	0
17	44	1	2024-03-06 08:45:35	47	(NULL)	0
18	49	2	2024-03-06 08:45:35	27	(NULL)	0
19	45	3	2024-03-06 08:45:35	49	(NULL)	0
20	50	2	2024-03-06 08:45:35	20	(NULL)	0

KategoriSeeder

KategoriSeeder.php

```

class KategoriSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [];
        $kategoriNames = ['jeans', 't-shirt', 'hoodie', 'accessories', 'boxer'];
        for ($i = 0; $i < 5; $i++) {
            $data[] = [
                'kategori_kode' => 'CLI' . $i + 1,
                'kategori_nama' => $kategoriNames[$i],
            ];
        }
        DB::table('m_kategori')->insert($data);
    }
}

```

Database

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	CLI1	jeans	(NULL)	(NULL)
2	CLI2	t-shirt	(NULL)	(NULL)
3	CLI3	hoodie	(NULL)	(NULL)
4	CLI4	accessories	(NULL)	(NULL)
5	CLI5	boxer	(NULL)	(NULL)

PenjualanSeeder

PenjualanSeeder.php

```

class PenjualanSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $data = [];
        $listBuyers = ['Ricky', 'Ahmad', 'Sandi', 'Rifki', 'Novi', 'Fikri', 'Ilham', 'Maul', 'Farid', 'Mail'];
        for ($i = 0; $i < 10; $i++) {
            $data[] = [
                'user_id' => 3, //Transaksi dihandle oleh
                'pembeli' => $listBuyers[$i],
                'penjualan_kode' => 'SELL' . $i + 1,
                'penjualan_tanggal' => now(),
            ];
        }
        DB::table('t_penjualan')->insert($data);
    }
}

```

Database

penjualan_id	user_id	pembeli	penjualan_kode	penjualan_tanggal	created_at
1	3	Ricky	SELL1	2024-03-06 08:45:56	(NULL)
2	3	Ahmad	SELL2	2024-03-06 08:45:56	(NULL)
3	3	Sandi	SELL3	2024-03-06 08:45:56	(NULL)
4	3	Rifki	SELL4	2024-03-06 08:45:56	(NULL)
5	3	Novi	SELL5	2024-03-06 08:45:56	(NULL)
6	3	Ilham	SELL6	2024-03-06 08:45:56	(NULL)
7	3	Fikri	SELL7	2024-03-06 08:45:56	(NULL)
8	3	Maul	SELL8	2024-03-06 08:45:56	(NULL)
9	3	Farid	SELL9	2024-03-06 08:45:56	(NULL)
10	3	Mail	SELL10	2024-03-06 08:45:56	(NULL)

PenjualanDetailSeeder

The screenshot shows a developer's environment with multiple windows open:

- Code Editor:** The main window displays `BarangSeeder.php` with PHP code for generating seed data for a database table.
- Database Management:** A separate window titled "Laragon MySQL\pwl_pos\t_penjualan_detail - HeidiSQL Portable 12.1.0.6537" shows the structure of the `t_penjualan_detail` table and its data. The table has columns: detail_id, penjualan_id, barang_id, jumlah, harga, created_at, and updated_at. The data shows 13 rows of generated seed data.
- Terminal:** The bottom right corner shows a terminal window with the command `INFO Seeding database.` and some SQL queries related to table constraints.

5. Hasil Commit

The screenshot shows the GitHub repository for `PWL_POS`. The "Commits" page displays the following commit history:

- ongping** (selected filter)
- Commits on Mar 6, 2024**
- Seeders** (commit by rickytedjo committed now) - SHA: f93e3d9
- Migration Tables w/ relation** (commit by rickytedjo committed 39 minutes ago) - SHA: d476e14
- Migration Tables w/ no relation** (commit by rickytedjo committed 1 hour ago) - SHA: d136d81
- PWL_POS init** (commit by rickytedjo committed 1 hour ago) - SHA: e56c7f1

Praktikum 4 - Implementasi DB Facade

1.

2.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "PWL_POS".
- Editor:** The "LevelController.php" file is open in the editor tab. The code is as follows:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class LevelController extends Controller
8 {
9 }
10
11
```

Terminal: Shows the command "D:\Code\PWL_POS>[" and the output "INFO Controller [D:\Code\PWL_POS\app\Http\Controllers\LevelController.php] created successfully."

Status Bar: Shows "Ln 9, Col 5" and "Spaces: 4" and "UTF-8".

2.

The screenshot shows the VS Code interface with the following details:

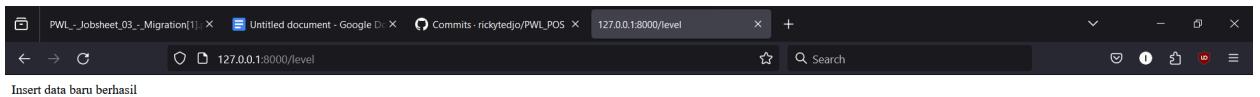
- File Explorer:** Shows the project structure under "PWL_POS".
- Editor:** The "web.php" file is open in the editor tab. The code is as follows:

```
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "web" middleware group. Make something great!
14 |
15 */
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
21 Route::get('level',[LevelController::class, 'index']);
```

Terminal: Shows the command "D:\Code\PWL_POS>[" and the output "INFO Controller [D:\Code\PWL_POS\app\Http\Controllers\LevelController.php] created successfully."

Status Bar: Shows "Ln 20, Col 1" and "Spaces: 4" and "UTF-8".

3.



A screenshot of the HeidiSQL Portable interface. The title bar says "Laragon.MySQL\pwl_pos\m_level - HeidiSQL Portable 12.1.0.6537". The left sidebar shows the database structure under "Laragon.My...". The main pane shows the "m_level" table with 4 rows:

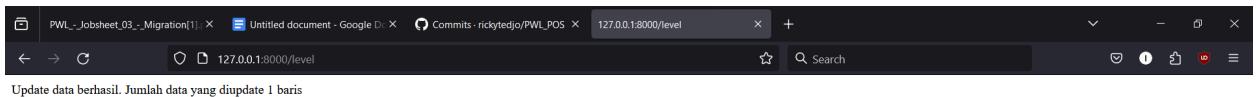
level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	(NULL)	(NULL)
2	MNG	Manager	(NULL)	(NULL)
3	STF	Staff/Kasir	(NULL)	(NULL)
4	CUS	Pelanggan	2024-03-06 08:59:54	(NULL)

The bottom pane shows the MySQL query history:

```
275 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc
276 SELECT * FROM `pwl_pos`.`users` LIMIT 1000;
277 SHOW CREATE TABLE `pwl_pos`.`m_level`;
278 SELECT * FROM `pwl_pos`.`m_level` LIMIT 1000;
```

Status information at the bottom: r1 : c1, Connected: 01, MySQL 8.0.30, Uptime: 01:48 h, Server time: 16, Idle.

4.



A screenshot of the HeidiSQL Portable interface. The database is 'Laragon.My...'. The table 'm_level' is selected, showing 4 rows total. The data is as follows:

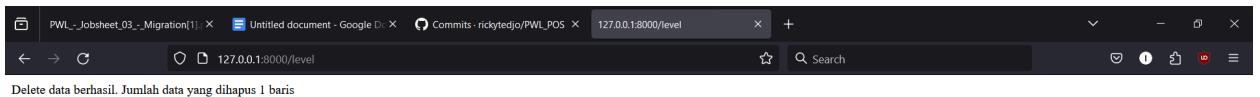
level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	(NULL)	(NULL)
2	MNG	Manager	(NULL)	(NULL)
3	STF	Staff/Kasir	(NULL)	(NULL)
4	CUS	Customer	2024-03-06 08:59:54	(NULL)

The SQL query history at the bottom shows:

```
279 SHOW CREATE TABLE `pwl_pos`.`m_user`;
280 SELECT * FROM `pwl_pos`.`m_user` LIMIT 1000;
281 SHOW CREATE TABLE `pwl_pos`.`m_level`;
282 SELECT * FROM `pwl_pos`.`m_level` LIMIT 1000;
```

System status at the bottom: r1 : c1, Connected: 01, MySQL 8.0.30, Uptime: 01:50 h, Server time: 16, Idle.

5.



A screenshot of HeidiSQL Portable version 12.1.0.6537. The database is "pwl_pos" and the table is "m_level". The table contains 3 rows:

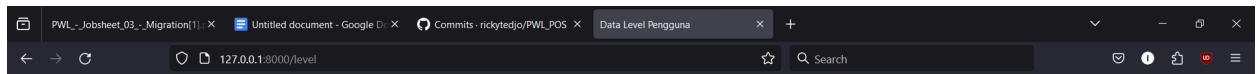
level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	(NULL)	(NULL)
2	MNG	Manager	(NULL)	(NULL)
3	STF	Staff/Kasir	(NULL)	(NULL)

The SQL history pane shows the following queries:

```
292 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='pwl_pos' AND TABLE_NAME='m_level' AND REFERENCED_TABLE_
293 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='pwl_pos' AND TABLE_NAME='m_level' AND REFERENCED_TABLE_NAME IS NOT
294 /* Entering session "Laragon.MySQL" */
295 SELECT * FROM `pwl_pos`.`m_level` LIMIT 1000;
```

6.

```
2 references | 0 implementations
8 class LevelController extends Controller
9 {
10     1 reference | 0 overrides
11         public function index(){
12             $data = DB::select('select * from m_level');
13
14             return view(['level',[ 'data'=>$data]]);
15         }
16     }
1
2     <!DOCTYPE html>
3     <html>
4         <head>
5             <title>Data Level Pengguna</title>
6         </head>
7         <body>
8             <h1>Data Level Pengguna</h1>
9             <table border="1" cellpadding="2" cellspacing="0">
10                <tr>
11                    <th>ID</th>
12                    <th>Kode Level</th>
13                    <th>Nama Level</th>
14                </tr>
15                @foreach ($data as $d)
16                <tr>
17                    <td>{{ $d->level_id}}</td>
18                    <td>{{ $d->level_kode}}</td>
19                    <td>{{ $d->level_nama}}</td>
20                </tr>
21                @endforeach
22            </table>
23        </body>
24    </html>
```



Data Level Pengguna

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

Hasil Commit

The screenshot shows the GitHub interface for the repository 'PWL_POS'. The 'Commits' tab is selected. A dropdown menu shows 'ongoing' is selected. Below it, a filter shows 'Commits on Mar 6, 2024'. The commit list is as follows:

- DB Facade Implementation (bc21e58) - committed now
- Seeders (f93e3d9) - committed 22 minutes ago
- Migration Tables w/ relation (d476e14) - committed 1 hour ago
- Migration Tables w/ no relation (d136d81) - committed 1 hour ago
- PWL_POS init (e56c7f1) - committed 1 hour ago

Praktikum 5 - Implementasi Query Builder

1.

The screenshot shows a code editor with a dark theme. A file named `KategoriController.php` is open in the editor. The code is a simple PHP class definition:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class KategoriController extends Controller
8 {
9     /**
10 }
```

Below the code editor is a terminal window. The terminal tab is selected in the tabs at the top. The terminal output shows the command `php artisan make:controller KategoriController` being run, followed by a success message: `[D:\Code\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.`

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

D:\Code\PWL_POS>php artisan make:controller KategoriController
[INFO] Controller [D:\Code\PWL_POS\app\Http\Controllers\KategoriController.php] created successfully.

D:\Code\PWL_POS>
```

2.

The screenshot shows a code editor interface with the title bar "PWL_POS". The left sidebar is an "EXPLORER" view showing the project structure for "PWL_POS". The "routes" folder contains "web.php". The main editor area displays the "routes/web.php" file:

```
routes > web.php > ...
6
7  /*
8  |-----+
9  | Web Routes
10 |-----+
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider and all of them will
14 | be assigned to the "web" middleware group. Make something great!
15 |
16 */
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
22 Route::get('level',[LevelController::class, 'index']);
23 Route::get('kategori',[KategoriController::class, 'index']);
```

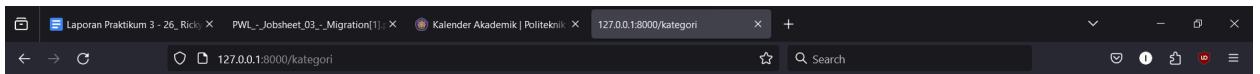
The status bar at the bottom indicates "Ln 23, Col 58" and "PHP 8.1".

The screenshot shows a code editor interface with the title bar "PWL_POS". The left sidebar is an "EXPLORER" view showing the project structure for "PWL_POS". The "Controllers" folder contains "KategoriController.php". The main editor area displays the "Controllers/KategoriController.php" file:

```
app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > index
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\DB;
7
8  class KategoriController extends Controller
9  {
10     /**
11      * @return \Illuminate\Http\Response
12      */
13     public function index()
14     {
15         $data = [
16             'kategori.kode' => 'SNK',
17             'kategori.nama' => 'Snack/Makanan Ringan',
18             'created_at' => now()
19         ];
20     }
21 }
```

The status bar at the bottom indicates "Ln 18, Col 44" and "PHP 8.1".

3.



Laragon.MySQL\pw1_pos\m_kategori - HeidiSQL Portable 12.1.0.6537

File Edit Search Query Tools Go to Help

Database filter Table filter Host: 127.0.0.1 Database: pw1_pos Table: m_kategori Data Query

Donate

pw1_pos.m_kategori: 6 rows total (approximately)

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	CL1	jeans	(NULL)	(NULL)
2	CL2	t-shirt	(NULL)	(NULL)
3	CL3	hoodie	(NULL)	(NULL)
4	CL4	accessories	(NULL)	(NULL)
5	CL5	boxer	(NULL)	(NULL)
6	SNK	Snack/Makanan Ringan	2024-03-10 02:07:16	(NULL)

Laragon.MySQL

- > information_s...
- > mysql
- > performance_...
- ✓ pw1_pos 384,0 KiB
 - failed_jobs 16,0 KiB
 - migrations 16,0 KiB
 - m_barang 48,0 KiB
 - m_kategori 32,0 KiB
 - ✓ m_level 32,0 KiB
 - m_user 48,0 KiB
 - password_f... 16,0 KiB
 - personal_ac... 16,0 KiB
 - t_penjualan 48,0 KiB
 - t_penjualan... 48,0 KiB
 - t_stok 48,0 KiB
 - users 16,0 KiB
- > sys

36 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='pw1_pos' AND TABLE_NAME='m_kategori' AND REFERENCED_TABLE_NAME IS NOT NULL;
37 SHOW CREATE TABLE `pw1_pos`.`m_kategori`;
38 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc WHERE tc.CONSTRAINT_SCHEMA='pw1_pos' AND tc.TABLE_NAME='m_kategori' AND tc.CONSTRAINT_TYPE='CHECK';
39 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;

r1 : c1 Connected: 00:00 h MySQL 8.0.30 Uptime: 00:09 h Server time: 09:07 Idle.

4.

The screenshot shows a developer's environment with multiple windows open:

- Code Editor:** The main window displays the file `KategoriController.php` under the `app/Http/Controllers` directory. The code contains PHP code for updating a database table. A terminal tab below the editor shows the command `php artisan serve` running successfully on port 8000.
- Terminal:** Shows the command `php artisan serve` running successfully on port 8000.
- Browser:** A browser window shows the URL `127.0.0.1:8000/kategori`. The page content is "Update data berhasil. Jumlah data yang diupdate: 1 baris".

```
10 // $data = [
11 //     'kategori_kode'=>'SNK',
12 //     'kategori_nama'=>'Snack/Makanan Ringan',
13 //     'created_at' => now()
14 // ];
15
16
17 // DB::table('m_kategori')->insert($data);
18
19
20 $row = DB::table('m_kategori')->where('kategori.kode','SNK')->update(['kategori.nama'=>'Camilan']);
21
22 return "Update data berhasil. Jumlah data yang diupdate: '$row' baris";
23 }
```

Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.
D:\Code\pwL_POS>php artisan serve
[INFO] Server running on [<http://127.0.0.1:8000>].
Press Ctrl+C to stop the server

2024-03-10 09:07:07 ~ 3s
2024-03-10 09:07:16 ~ 1s
2024-03-10 09:09:57 ~ 0s

Ln 21, Col 80 Spaces: 4 UTF-8 LF (8.1 PHP 8.1 ⚡ 🔍

Laragon.MySQL\pw1_pos\m_kategori - HeidiSQL Portable 12.1.0.6537

File Edit Search Query Tools Go to Help

Database filter Table filter Host: 127.0.0.1 Database: pw1_pos Table: m_kategori Data Query

m_kategori: 6 rows total (approximately)

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	CL1	jeans	(NULL)	(NULL)
2	CL2	t-shirt	(NULL)	(NULL)
3	CL3	hoodie	(NULL)	(NULL)
4	CL4	accessories	(NULL)	(NULL)
5	CL5	boxer	(NULL)	(NULL)
6	SNK	Cemilan	2024-03-10 02:07:16	(NULL)

37 SHOW CREATE TABLE `pw1_pos`.`m_kategori`;
38 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc WHERE tc.CONSTRAINT_SCHEMA='pw1_pos' AND tc.TABLE_NAME='m_kategori' AND tc.CONSTRAINT_TYPE='CHECK';
39 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;
40 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;

r1 : c1 Connected: 00:02 h MySQL 8.0.30 Uptime: 00:12 h Server time: 09:10 Idle.

5.

PWL_POS

KategoriController.php

```

public function index()
{
    // $data = [
    //     'kategori.kode' => 'SNK',
    //     'kategori.nama' => 'Snack/Makanan Ringan',
    //     'created_at' => now()
    // ];

    // DB::table('m_kategori')->insert($data);
    // return 'Insert data baru berhasil';

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

```

Microsoft Windows [Version 10.0.22631.3195]
(c) Microsoft Corporation. All rights reserved.

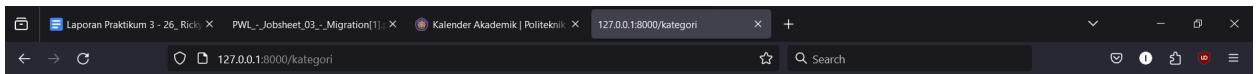
D:\Code\PWL_POS>php artisan serve

[INFO] Server running on [<http://127.0.0.1:8000>].

Press Ctrl+C to stop the server

2024-03-10 09:07:07 ~ 3s
2024-03-10 09:07:16 ~ 1s
2024-03-10 09:09:57 ~ 0s

Ln 21, Col 57 Spaces: 4 UTF-8 LF { PHP 8.1



A screenshot of HeidiSQL Portable version 12.1.0.6537. The database is "Laragon.MySQL\pw1_pos\m_kategori". The table "m_kategori" has 5 rows. The data is as follows:

kategori_id	kategori_kode	kategori_nama	created_at	updated_at
1	Q11	jeans	(NULL)	(NULL)
2	Q12	t-shirt	(NULL)	(NULL)
3	Q13	hoodie	(NULL)	(NULL)
4	Q14	accessories	(NULL)	(NULL)
5	Q15	boxer	(NULL)	(NULL)

The SQL query at the bottom of the interface is:

```
38 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc WHERE tc.CONSTRAINT_SCHEMA='pw1_pos' AND tc.TABLE_NAME='m_kategori' AND tc.CONSTRAINT_TYPE='CHECK'
39 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;
40 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;
41 SELECT * FROM `pw1_pos`.`m_kategori` LIMIT 1000;
```

6.

KategoriController.php U

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

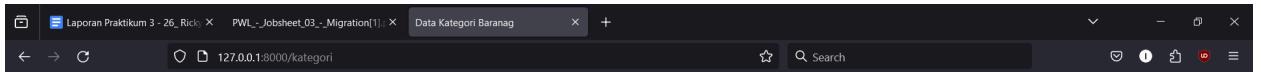
class KategoriController extends Controller
{
    public function index()
    {
        $data = [
            'kategori.kode' => 'SNK',
            'kategori.nama' => 'snack/Makanan Ringan',
            'created_at' => now()
        ];
    }

    // DB::table('m_kategori')->insert($data);
    // return 'Insert data baru berhasil';

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

kategori.blade.php U

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



Data Kategori Barang

ID	Kode Kategori	Nama Kategori
1	CLI1	jeans
2	CLI2	t-shirt
3	CLI3	hoodie
4	CLI4	accessories
5	CLI5	boxer

Hasil Commit

A screenshot of a GitHub commit history for a repository named 'PWL_POS'. The commit was made by 'rickytedjo' 1 minute ago. It has 1 parent commit 'bc21e58' and a commit hash '3f276ce'. The commit message is 'Query Builder Implementation'. The file diff shows changes in 'app/Http/Controllers/KategoriController.php'. The changes are as follows:

Praktikum 6 - Implementasi Eloquent ORM

1.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure under `PWL_POS`, including `app`, `Http`, `Controllers`, `Models`, and `web.php`. `UserModel.php` is selected.
- EDITOR**: The code for `UserModel.php` is displayed:

```
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class UserModel extends Model  
{  
    use Hasfactory;  
}
```
- TERMINAL**: The command `php artisan make:model UserModel` is run, resulting in the message: `[D:\Code\PWL_POS\app\Models\UserModel.php] created successfully.`
- STATUS BAR**: Shows "Ln 1, Col 1" and "Spaces: 4" and "UTF-8".

2.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure under `PWL_POS`, including `app`, `Http`, `Controllers`, `Models`, and `web.php`. `UserModel.php` is selected.
- EDITOR**: The code for `UserModel.php` is displayed:

```
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class UserModel extends Model  
{  
    use Hasfactory;  
  
    protected $table = 'm_user';  
    protected $primaryKey = 'user_id';  
}
```
- TERMINAL**: The command `php artisan make:model UserModel` is run, resulting in the message: `[D:\Code\PWL_POS\app\Models\UserModel.php] created successfully.`
- STATUS BAR**: Shows "Ln 13, Col 39" and "Spaces: 4" and "UTF-8".

3.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "PWL_POS". The "app" folder contains "Console", "Exceptions", "Http", "Controllers" (with "Controller.php", "KategorieController.php", "LevelController.php", "UserController.php"), "Middleware", "Kernel.php", "Models" (with "User.php", "UserModel.php"), "Providers", "bootstrap", "config", "database", "public", "resources" (with "css", "js", "views" containing "kategorie.blade.php", "level.blade.php", "welcome.blade.php"), "routes" (with "api.php", "channels.php", "console.php"), and "OUTLINE" and "TIMELINE" sections.
- Editor:** The "web.php" file is open at line 11. It contains code for registering web routes:

```
11 | -----  
12 |  
13 | Here is where you can register web routes for your application. These  
14 | routes are loaded by the RouteServiceProvider and all of them will  
15 | be assigned to the "web" middleware group. Make something great!  
16 |  
17 |/*  
18 |  
19 |Route::get('/', function () {  
20 |     return view('welcome');  
21 |});  
22 |  
23 |Route::get('/level',[LevelController::class, 'index']);  
24 |Route::get('/kategorie',[KategorieController::class, 'index']);  
25 |Route::get('/user',[UserController::class, 'index']);
```
- Terminal:** The terminal shows the command "php artisan make:model UserModel" being run, followed by the response "[D:\Code\PWL_POS] created successfully."
- Bottom Status Bar:** Shows "Ln 5, Col 41" and "UTF-8 LF".

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, the file structure of the "PWL_POS" project is displayed. It includes the "app" directory which contains "Http", "Controllers", "Models", "Providers", "bootstrap", "config", "database", "public", and "resources" sub-directories. Under "Controllers", there are "Controller.php", "KategoriController.php", "LevelController.php", and "UserController.php".
- Editor:** The main editor area shows the code for "UserController.php". The code defines a class "UserController" that extends "Controller".

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class UserController extends Controller
8 {
9     //
10 }
```
- Terminal:** At the bottom, the terminal window shows the command "php artisan make:model UserModel" being run, followed by the response "INFO Model [D:\Code\PWL_POS\app\Models\UserModel.php] created successfully."
- Bottom Status Bar:** The status bar at the bottom indicates the current file is "ongoing*" and shows other details like spaces, tabs, and encoding.

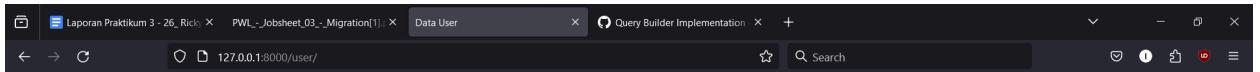
4.

The screenshot shows the VS Code interface with the title bar "PWL_POS". The Explorer sidebar on the left shows the project structure under "app". The main editor area displays the code for "UserController.php". The code defines a class "UserController" extending "Controller". It contains a public function "index()" that retrieves a user from the database and returns a view. The code editor has syntax highlighting and a status bar at the bottom indicating "Ln 14, Col 46".

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
use App\Models\UserModel;  
  
class UserController extends Controller  
{  
    //  
    // 1 reference | 0 overrides  
    public function index()  
    {  
        $user = UserModel::all();  
        return view('user', ['data' => $user]);  
    }  
}  
14
```

The screenshot shows the VS Code interface with the title bar "PWL_POS". The Explorer sidebar on the left shows the project structure under "app". The main editor area displays the code for "user.blade.php". The blade template creates a table to display user data, including columns for ID, Username, Name, and Level Pengguna. It uses a foreach loop to iterate over the data. The code editor has syntax highlighting and a status bar at the bottom indicating "Ln 21, Col 18".

```
<!DOCTYPE html>  
<html>  
    <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>  
</html>
```



Data User

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

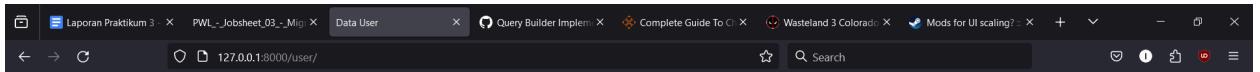
A screenshot of a code editor titled "PWL_POS". The left sidebar shows a file tree for the "app" directory, including "UserController.php" which is currently selected. The right pane displays the PHP code for the "UserController" class:

```
use Illuminate\Support\Facades\Hash;
```

```
class UserController extends Controller
{
    ...
    public function index()
    {
        $data = [
            'username' => 'customer-1',
            'nama' => 'Pelanggan',
            'password' => Hash::make('12345'),
            'level_id' => 3
        ];
        UserModel::insert($data);

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

5.

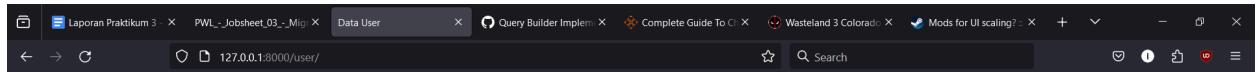


ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
5	customer-1	Pelanggan	3

A screenshot of a code editor (VS Code) showing the 'UserController.php' file under the 'app/Http\Controllers' directory. The code is as follows:

```
use Illuminate\Support\Facades\Hash;
class UserController extends Controller
{
    public function index()
    {
        $data = [
            'nama'=>'Pelanggan Pertama'
        ];
        UserModel::where('username','customer-1')->update($data);
        $user = UserModel::all();
        return view('user', ['data'=>$user]);
    }
}
```

6.



Data User

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

Hasil Commit

A screenshot of a GitHub commit page. The URL is https://github.com/rickytedjo/PWL_POS/commit/3383dba3fa74772728749b8f43817bef2e212733. The commit is titled 'Eloquent ORM Implementation' and was made by 'rickytedjo' on 'committed now'. It has 1 parent commit '3f276ce' and a commit hash '3383dba'. The commit message is 'Eloquent ORM Implementation'. The code changes section shows 4 changed files with 66 additions and 3 deletions. The file 'app/Http/Controllers/UserController.php' is shown with the following diff:

Soal

1. Untuk dijadikan key dalam enkripsi dan dekripsi
2. php artisan key:generate
3. 4 file yang digunakan untuk akses developer
4. Untuk menghasilkan data waktu dari perangkat yang digunakan
5. Untuk menghasilkan primary key

6. Jika diisi, output primary key memiliki nama kolom yang ditentukan, sedangkan jika tidak diisi akan dinamakan id secara *default*.
7. Agar isi field yang berkaitan dengan kolom tersebut selalu unik
8. Fungsi id menghasilkan kolom primary key yang bertipe data bigint, sedangkan tujuan fungsi unsignedbiginteger digunakan untuk menghasilkan kolom bertipe data bigint yang nantinya dapat dijadikan foreign key
9. Hash digunakan untuk mengenkripsi data
10. Tanda tersebut digunakan untuk prepared statement
11. Untuk menetapkan tabel yang akan diambil data nya
12. Eloquent ORM karena fungsi-fungsi yang digunakan lebih sederhana dibandingkan metode lain