

LAPORAN MODUL 3 PRAKTIKUM
Mata Kuliah Pemrograman Web



Nama : Putri Maulida
Nim : C030321051
Kelas : TI-4B
Praktek ke : 3

PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI BANJARMASIN
2023

MODUL 3

Tujuan

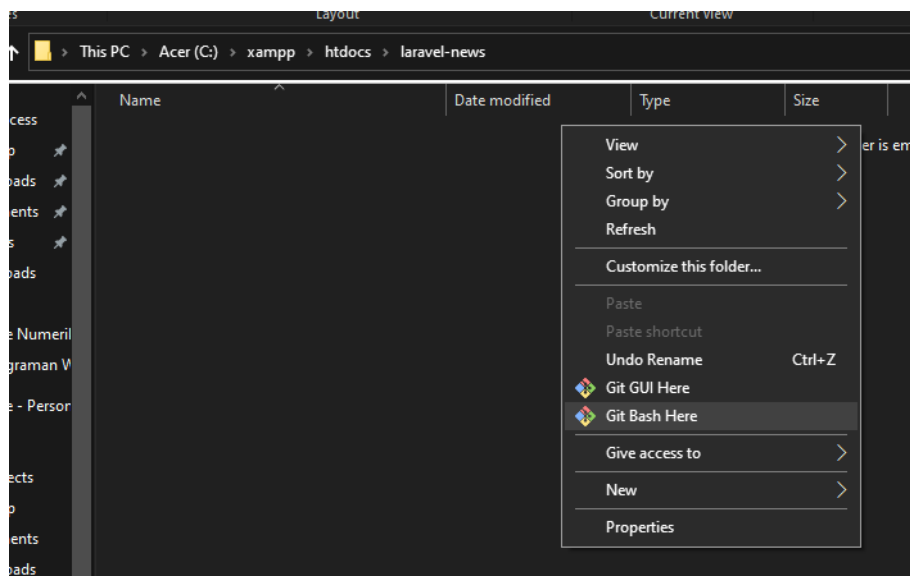
Setelah mempelajari modul ini anda diharapkan mampu :

- Memahami teknik *Model View Controller* (MVC) menggunakan bahasa pemrograman PHP.
- Merancang program menggunakan teknik *Model View Controller* (MVC).

Praktikum

Install Laravel

Masuk ke C:\xampp\htdocs buat folder laravel-news, kemudian masuk. Klik kanan pada mouse, pilih Git Bash Here



akan muncul sebuah CMD, masukkan perintah *composer create-project laravel/laravel:^9.0 c030321051-project3*

```
C:\xampp\htdocs\laravel-news

putri@LAPTOP-1KHBPEIC MINGW64 /c/xampp/htdocs/laravel-news
$ composer create-project laravel/laravel:^9.0 c030321051-project3
Creating a "laravel/laravel:^9.0" project at "./c030321051-project3"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.5.2)
- Installing laravel/laravel (v9.5.2): Extracting archive
Created project in C:\xampp\htdocs\laravel-news\c030321051-project3
> @php -r "file_exists('.env') || copy('.env.example', '.env');";"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 108 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (2.0.0)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (4.0.1)
- Locking fakerphp/faker (v1.21.0)
```

Ketikkan perintah `composer create-project laravel/laravel c030321051-project3`

```
putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs
$ composer create-project laravel/laravel c030321051-project3
Creating a "laravel/laravel" project at "./c030321051-project3"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v10.1.1)
- Downloading laravel/laravel (v10.1.1)
- Installing laravel/laravel (v10.1.1): Extracting archive
Created project in C:\xampp\htdocs\c030321051-project3
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (4.0.1)
- Locking fakerphp/faker (v1.21.0)
```

Symbolic Link

Ketikkan perintah `cd c030321051-project3` agar Laravel bisa membaca semua file yang ada di folder c030321051-project3

```
putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs
$ cd c030321051-project3

putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
```

Lalu, ketikkan perintah `php artisan storage:link`

```
putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$ php artisan storage:link

INFO The [C:\xampp\htdocs\c030321051-project3\public\storage] link has been connected to [C:\xampp\htdocs\c030321051-project3\storage\app\public].
```

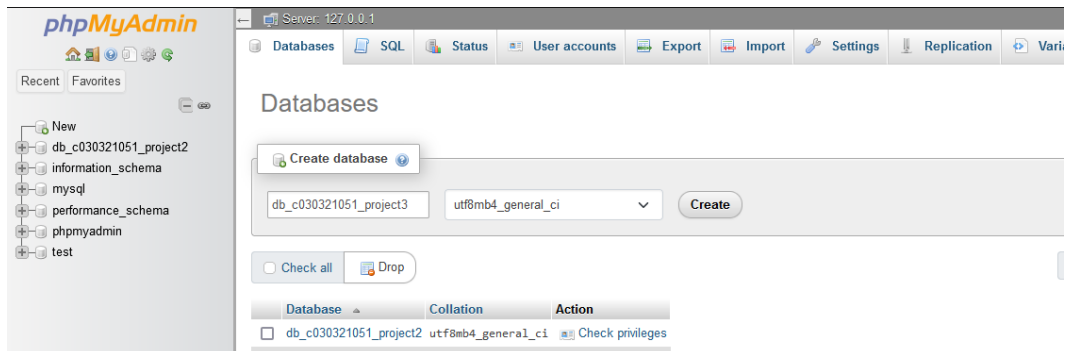
Di dalam folder c030321051-project3, akan muncul folder baru, yaitu folder storage

This PC > Acer (C:) > xampp > htdocs > c030321051-project3 >				
Name	Date modified	Type	Size	
app	19/04/2023 0:21	File folder		
bootstrap	19/04/2023 0:21	File folder		
config	19/04/2023 0:21	File folder		
database	19/04/2023 0:21	File folder		
public	19/04/2023 0:21	File folder		
resources	19/04/2023 0:21	File folder		
routes	19/04/2023 0:21	File folder		
storage	19/04/2023 0:21	File folder		
tests	19/04/2023 0:21	File folder		
vendor	25/04/2023 23:57	File folder		
.editorconfig	19/04/2023 0:21	EDITORCONFIG File	1 KB	
.env	25/04/2023 23:57	ENV File	2 KB	
.env.example	19/04/2023 0:21	EXAMPLE File	2 KB	
.gitattributes	19/04/2023 0:21	Text Document	1 KB	
.gitignore	19/04/2023 0:21	Text Document	1 KB	
artisan	19/04/2023 0:21	File	2 KB	
composer.json	19/04/2023 0:21	JSON File	2 KB	
composer.lock	25/04/2023 23:53	LOCK File	280 KB	

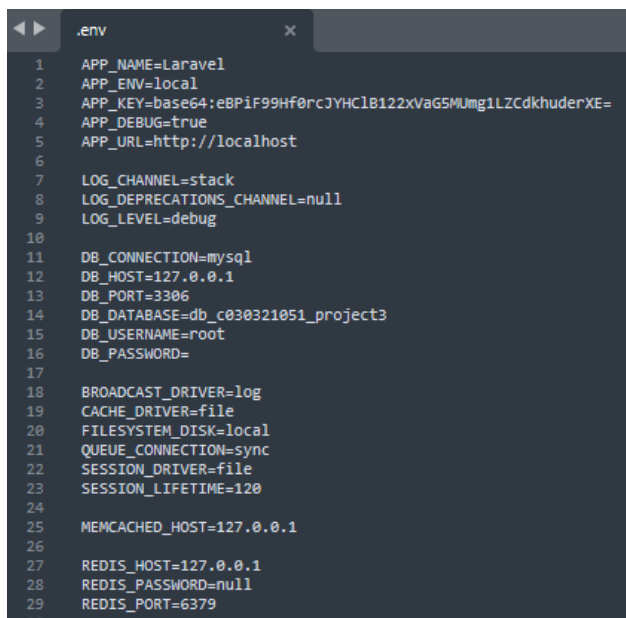
Database

Koneksi Laravel ke MySql

Buat database dengan nama db_c030321051_project3

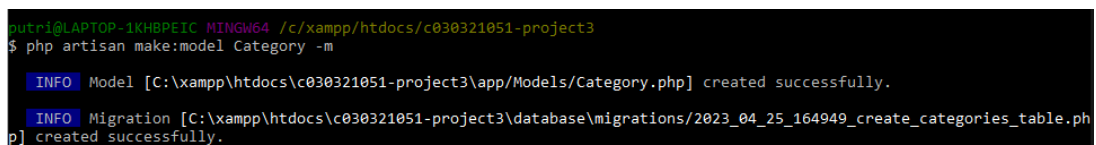


ubah code DB_DATABASE di file .env



Membuat Model dan Migration Category

Masukkan perintah *php artisan make:model Category -m*



Buka file **app/Models/Category.php**, masukkan perintah seperti gambar dibawah

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Category extends Model
9  {
10     use HasFactory;
11     protected $fillable = [
12         'name', 'slug', 'image'
13     ];
14 }
15

```

buka file **database/migrations/2023_**_**_164949_create_categories_table.php**,
sesuaikan code seperti gambar dibawah

```

9
10  /**
11   * Run the migrations.
12   */
13  public function up(): void
14  {
15      Schema::create('categories', function (Blueprint $table) {
16          $table->id();
17          $table->string('image');
18          $table->string('name');
19          $table->string('slug')->unique();
20          $table->timestamps();
21      });
22  }
23
24  /**
25   * Reverse the migrations.
26   */
27  public function down(): void
28  {
29      Schema::dropIfExists('categories');
30  }
31

```

Membuat Model dan Migration Post

Ketikkan perintah *php artisan make:model Post -m* di CMD untuk membuat

```

Putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$ php artisan make:model Post -m

INFO Model [C:\xampp\htdocs\c030321051-project3\app\Models\Post.php] created successfully.
INFO Migration [C:\xampp\htdocs\c030321051-project3\database\migrations\2023_04_26_160941_create_posts_table.php] created successfully.

Putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$

```

Buka file **app/Models/Post.php**, lakukan konfigurasi seperti dibawah

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Post extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = [
13         'title', 'slug', 'category_id', 'user_id', 'content', 'image', 'description'
14     ];
15 }
16

```

Buka `database/migrations/2023_**_**_160941_create_posts_table.php`, sesuaikan code seperti gambar dibawah

```
2023_04_26_160941_create_posts_table.php x
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create('posts', function (Blueprint $table) {
15              $table->id();
16              $table->string('title');
17              $table->string('slug')->unique();
18              $table->unsignedInteger('category_id');
19              $table->unsignedInteger('user_id');
20              $table->text('content');
21              $table->string('image');
22              $table->text('description');
23              $table->timestamps();
24          });
25      }
26
27      /**
28       * Reverse the migrations.
29       */
30      public function down(): void
31      {
32          Schema::dropIfExists('posts');
33      }
34  };
35
```

Membuat Model dan Migration Comment

Ketikkan perintah `php artisan make:model Comment -m`

```
putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$ php artisan make:model Comment -m

INFO Model [C:\xampp\htdocs\c030321051-project3\app\Models\Comment.php] created successfully.
INFO Migration [C:\xampp\htdocs\c030321051-project3\database\migrations\2023_04_30_141525_create_comments_table.php]
created successfully.
```

Buka file `app/Models/Comment.php`, lakukan konfigurasi seperti dibawah

```
Comment.php x
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Comment extends Model
9  {
10      use HasFactory;
11      protected $fillable = [
12          'post_id', 'name', 'email', 'comment'
13      ];
14  }
15
```

Buka `database/migrations/2023_**_**_141525_create_comments_table.php`,
sesuaikan code seperti gambar dibawah

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create('comments', function (Blueprint $table) {
15              $table->id();
16              $table->unsignedInteger('post_id');
17              $table->string('name');
18              $table->string('email');
19              $table->text('comment');
20              $table->timestamps();
21          });
22      }
23
24      /**
25       * Reverse the migrations.
26       */
27      public function down(): void
28      {
29          Schema::dropIfExists('comments');
30      }
31  };
32

```

Ketikkan `php artisan migrate` untuk membuat beberapa tabel dan column di database

```

putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$ php artisan migrate

INFO: Preparing database.

Creating migration table ..... 196ms DONE

INFO: Running migrations.

2014_10_12_000000_create_users_table ..... 463ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 860ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 501ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 768ms DONE
2023_04_25_164949_create_categories_table ..... 402ms DONE
2023_04_26_160941_create_posts_table ..... 562ms DONE
2023_04_30_141525_create_comments_table ..... 232ms DONE

putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$

```

Table	Action	Rows	Type	Collation	Size	Overhead
categories	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K1B	-
comments	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K1B	-
failed_jobs	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K1B	-
migrations	✱ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_unicode_ci	16.0 K1B	-
password_reset_tokens	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K1B	-
personal_access_tokens	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 K1B	-
posts	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K1B	-
users	✱ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K1B	-
8 tables	Sum	7	InnoDB	utf8mb4_general_ci	224.0 K1B	0 B

Ubah code **app/Models/User.php** seperti gambar dibawah

```
User.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Casts\Attribute;
6 use Illuminate\Database\Eloquent\Model;
7
8 class User extends Model
9 {
10
11     protected function name(): Attribute
12     {
13         return Attribute::make(
14             get: fn ($value) => strtoupper($value),
15         );
16     }
17 }
```

Setting relasi & Accessor Model Category

Ubah code **app/Models/Category.php** seperti gambar dibawah

```
Category.php
1 <?php
2 namespace App\Models;
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Casts\Attribute;
5 use Illuminate\Database\Eloquent\Model;
6 class Category extends Model
7 {
8     use HasFactory;
9
10     protected $fillable = [
11         'name', 'slug', 'image'
12     ];
13
14     public function posts()
15     {
16         return $this->hasMany(Post::class);
17     }
18
19     protected function image(): Attribute
20     {
21         return Attribute::make(
22             get: fn ($value) => asset('/storage/categories/' . $value),
23         );
24     }
25 }
```

Setting relasi & Accessor Model Comment

Ubah code **app/Models/Comment.php** seperti gambar dibawah

```
Comment.php
1 <?php
2 namespace App\Models;
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Casts\Attribute;
5 use Illuminate\Database\Eloquent\Model;
6 class Comment extends Model
7 {
8     use HasFactory;
9
10     protected $fillable = [
11         'post_id', 'name', 'email', 'comment'
12     ];
13     protected function createdAt(): Attribute
14     {
15         return Attribute::make(
16             get: fn ($value) => \Carbon\Carbon::locale('id')->parse($value)->translatedFormat('l, d F Y'),
17         );
18     }
19     protected function updatedAt(): Attribute
20     {
21         return Attribute::make(
22             get: fn ($value) => \Carbon\Carbon::locale('id')->parse($value)->translatedFormat('l, d F Y'),
23         );
24     }
25 }
```


Setting relasi & Accessor Model Post

Ubah code **app/Models/Post.php** seperti gambar dibawah

```
Post.php
1 <?php
2 namespace App\Models;
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Castable;
5 use Illuminate\Database\Eloquent\Model;
6 class Post extends Model
7 {
8     use HasFactory;
9
10     protected $fillable = [
11         'title', 'slug', 'category_id', 'user_id', 'content', 'image', 'description'
12     ];
13
14     public function category()
15     {
16         return $this->belongsTo(Category::class);
17     }
18
19     public function user()
20     {
21         return $this->belongsTo(User::class);
22     }
23
24     public function comments()
25     {
26         return $this->hasMany(Comment::class);
27     }
28
29     protected function image(): Attribute
30     {
31         return Attribute::make(
32             get: fn ($value) => asset('/storage/posts/' . $value),
33         );
34     }
35
36     protected function createdAt(): Attribute
37     {
38         return Attribute::make(
39             get: fn ($value) => \Carbon\Carbon::locale('id')->parse($value)->translatedFormat('l, d F Y'),
40         );
41     }
42
43     protected function updatedAt(): Attribute
44     {
45         return Attribute::make(
46             get: fn ($value) => \Carbon\Carbon::locale('id')->parse($value)->translatedFormat('l, d F Y'),
47         );
48     }
49 }
```

Make Seeder User

Buat file didalam folder **database\seeders\UserSeeder.php** dengan mengetikkan perintah *php artisan make:seeder UserSeeder* di CMD

```
Putri@LAPTOP-1KHBPEIC MINGW64 /c:/xampp/htdocs/c030321051-project3
$ php artisan make:seeder UserSeeder

INFO Seeder [C:\xampp\htdocs\c030321051-project3\database\seeders\UserSeeder.php] created successfully.
```

```
UserSeeder.php
1 <?php
2 namespace Database\Seeders;
3 use Illuminate\Database\Seeder;
4 use Illuminate\Support\Facades\DB;
5 use Illuminate\Support\Facades\Hash;
6 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7 class UserSeeder extends Seeder
8 {
9     public function run()
10     {
11         DB::table('users')->insert([
12             'name' => 'Putri Maulida',
13             'email' => 'c030321051@mahasiswa.poliban.ac.id',
14             'password' => Hash::make('12345678'),
15         ]);
16     }
17 }
```

Dump UserSeeder

ketikkan perintah *php artisan make:seeder UserSeeder* untuk men generate file autoload.php

```
putri@LAPTOP-1KHBPEIC MINGW64 /c/xampp/htdocs/c030321051-project3
$ composer dump-autoload
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

Generated optimized autoload files containing 5780 classes
```

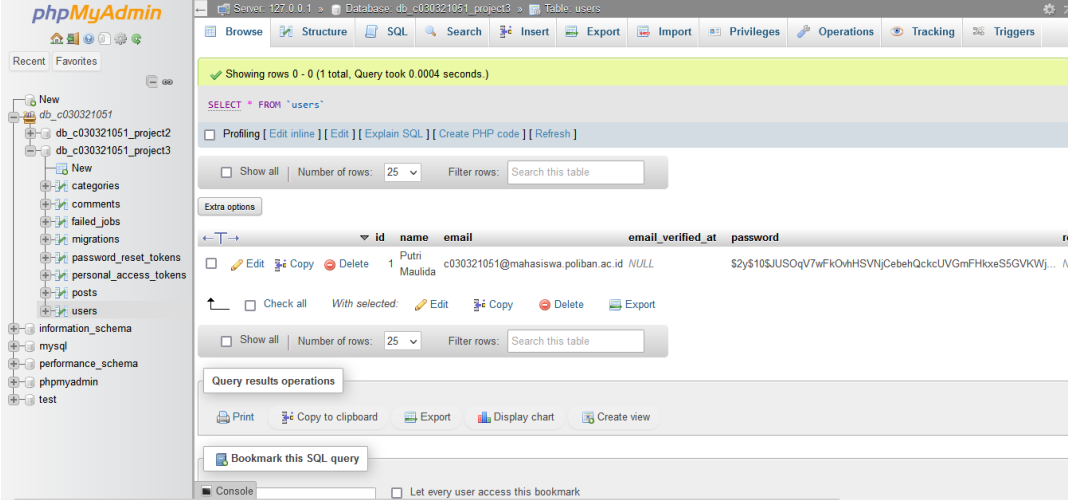
dan juga perintah *php artisan db:seed --class=UserSeeder*

```
putri@LAPTOP-1KHBPEIC MINGW64 /c/xampp/htdocs/c030321051-project3
$ php artisan db:seed --class=UserSeeder

INFO Seeding database.

putri@LAPTOP-1KHBPEIC MINGW64 /c/xampp/htdocs/c030321051-project3
$
```

Di dalam table users, otomatis terdapat sebuah user yang sudah dibuat sebelumnya



The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar displays a tree view of the database structure, including tables like 'categories', 'comments', 'failed_jobs', 'migrations', 'password_reset_tokens', 'personal_access_tokens', 'posts', 'users', 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The main area shows the 'users' table with the following data:

id	name	email	email_verified_at	password
1	Putri Maulida	c030321051@mahasiswa.poliban.ac.id	NULL	\$2y\$10\$JUS0qV7wFkO+hHSVNjCebhQckcUVGmFHxeS5GVKwj...

The interface also includes a SQL query editor with the query `SELECT * FROM `users`` and various options for displaying and managing the data.