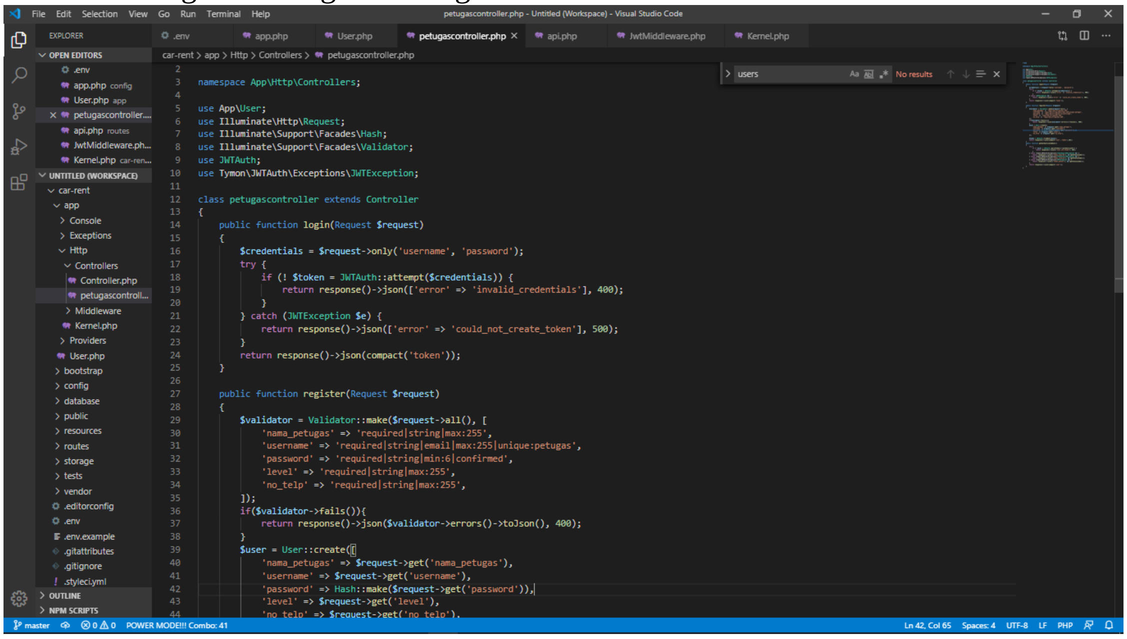
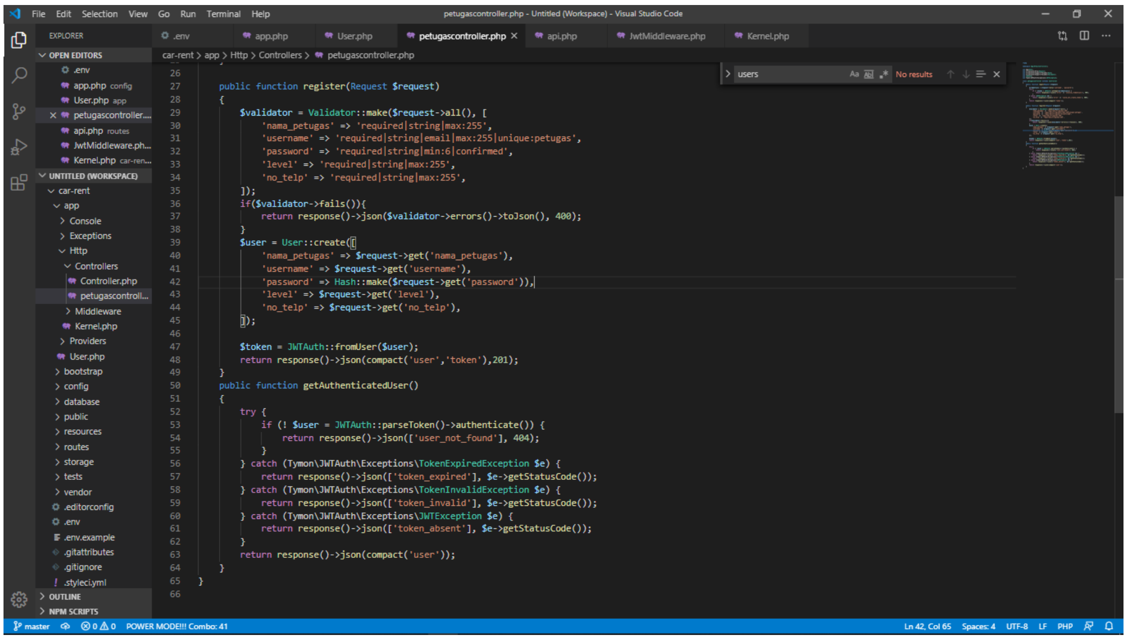
1. Controller Login dan Register Petugas

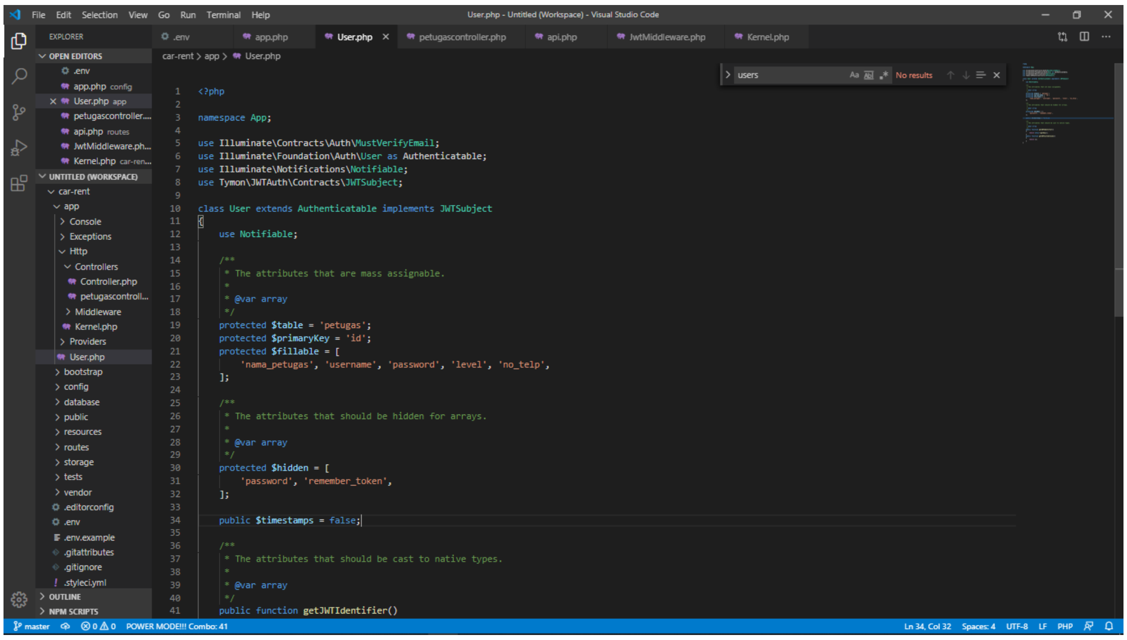


```php

namespace App\Http\Controllers;

use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;

class petugascontroller extends Controller
{
    public function login(Request $request)
    {
        $credentials = $request->only('username', 'password');
        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 400);
            }
        } catch (JWTException $e) {
            return response()->json(['error' => 'could_not_create_token'], 500);
        }
        return response()->json(compact('token'));
    }

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'username' => 'required|string|email|max:255|unique:petugas',
            'password' => 'required|string|min:6|confirmed',
            'level' => 'required|string|max:255',
            'no_telp' => 'required|string|max:255',
        ]);
        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }
        $user = User::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'username' => $request->get('username'),
            'password' => Hash::make($request->get('password')),
            'level' => $request->get('level'),
            'no_telp' => $request->get('no_telp'),
```

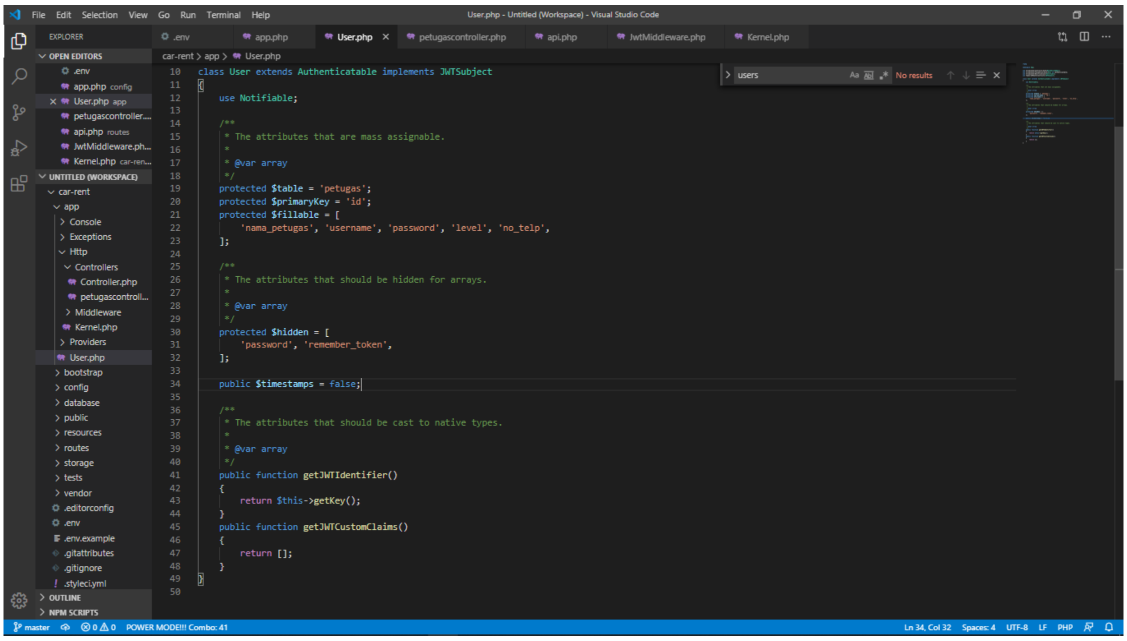


```php

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'username' => 'required|string|email|max:255|unique:petugas',
            'password' => 'required|string|min:6|confirmed',
            'level' => 'required|string|max:255',
            'no_telp' => 'required|string|max:255',
        ]);
        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }
        $user = User::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'username' => $request->get('username'),
            'password' => Hash::make($request->get('password')),
            'level' => $request->get('level'),
            'no_telp' => $request->get('no_telp'),
        ]);

        $token = JWTAuth::fromUser($user);
        return response()->json(compact('user','token'),201);
    }
    public function getAuthenticatedUser()
    {
        try {
            if (! $user = JWTAuth::parseToken()->authenticate()) {
                return response()->json(['user_not_found'], 404);
            }
        } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
            return response()->json(['token_expired'], $e->getStatusCode());
        } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
            return response()->json(['token_invalid'], $e->getStatusCode());
        } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
            return response()->json(['token_absent'], $e->getStatusCode());
        }
        return response()->json(compact('user'));
    }
}

```

2. Model Petugas



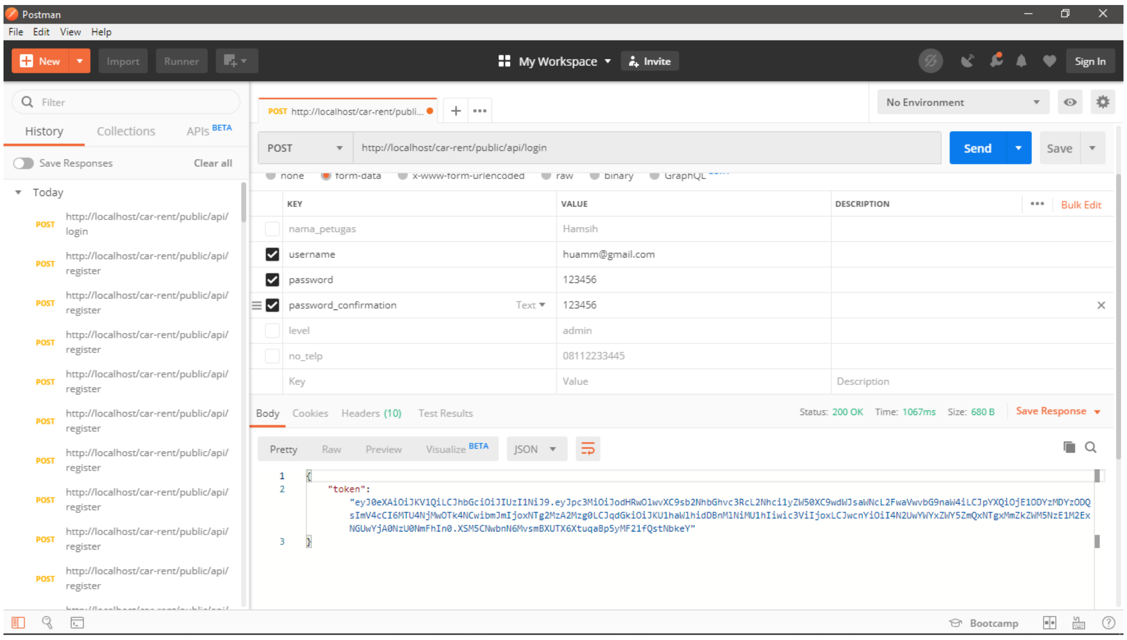
```php
<?php

namespace App;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Tymon\JWTAuth\Contracts\JWTSubject;

class User extends Authenticatable implements JWTSubject
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $table = 'petugas';
    protected $primaryKey = 'id';
    protected $fillable = [
        'nama_petugas', 'username', 'password', 'level', 'no_telp',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public $timestamps = false;

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    public function getJWTIdentifier()
```



```php
class User extends Authenticatable implements JWTSubject
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $table = 'petugas';
    protected $primaryKey = 'id';
    protected $fillable = [
        'nama_petugas', 'username', 'password', 'level', 'no_telp',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public $timestamps = false;

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    public function getJWTIdentifier()
```

```php
    {
        return $this->getKey();
    }
    public function getJWTCustomClaims()
    {
        return [];
    }
}
```

3. Hasil Login

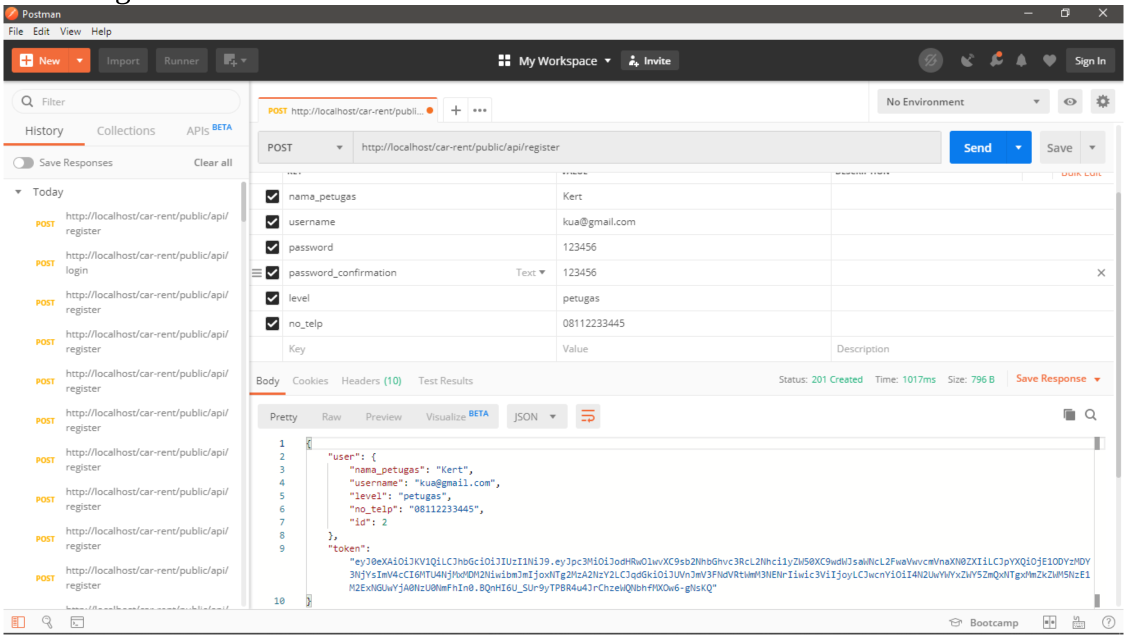

POST http://localhost/car-rent/public/api/login

	KEY	VALUE	DESCRIPTION
☐	nama_petugas	Hamsih	
✓	username	huamm@gmail.com	
✓	password	123456	
✓	password_confirmation	123456	
☐	level	admin	
☐	no_telp	08112233445	

Status: 200 OK Time: 1067ms Size: 680 B

```
{
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL2Nhci1yZW50XC9wdWJsaWNcL2FwaVwvbG9naW4iLCJpYXQiOjE1ODYzMDYzODQsImV4cCI6MTU4NjMwOTk4NCwibmJmIjoxNTg2MzA2Mzg0LCJqdGkiOiJKU1haWlhidDBnM1NiMU1hIiwic3ViIjoxLCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxMmZkZWM5NzE1M2ExNGUwYjA0NzU0NmFhIn0.XSM5CNwbnN6MvsmBXUTX6XtuqaBp5yMF21fQstNbkeY"
}
```

4. Hasil Register

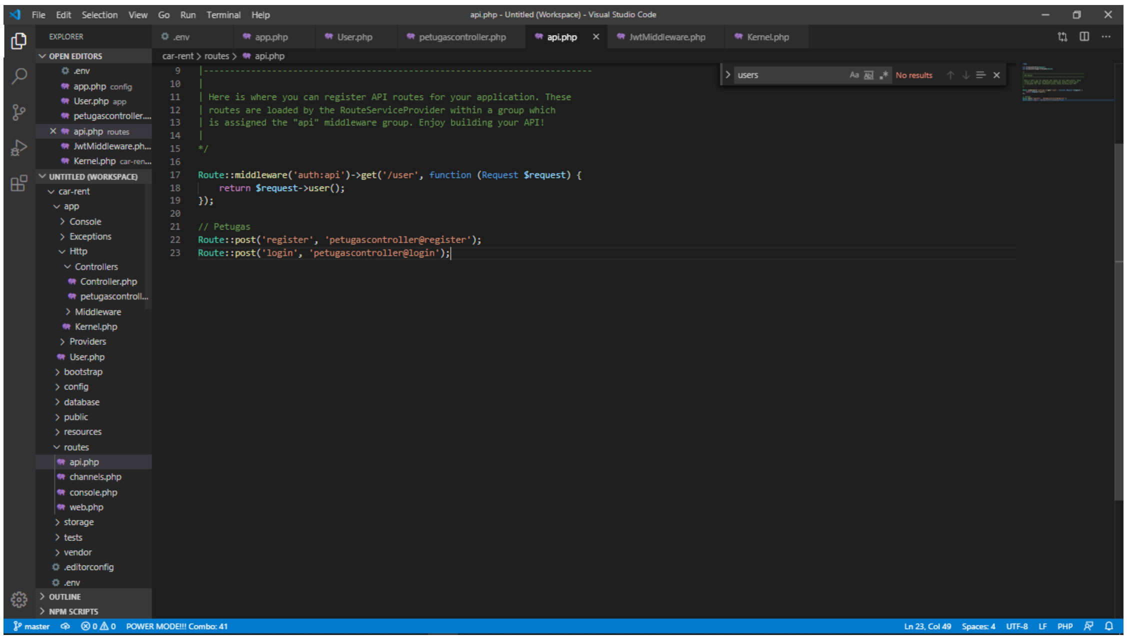

POST http://localhost/car-rent/public/api/register

	KEY	VALUE	DESCRIPTION
✓	nama_petugas	Kert	
✓	username	kua@gmail.com	
✓	password	123456	
✓	password_confirmation	123456	
✓	level	petugas	
✓	no_telp	08112233445	

Status: 201 Created Time: 1017ms Size: 796 B

```
{
    "user": {
        "nama_petugas": "Kert",
        "username": "kua@gmail.com",
        "level": "petugas",
        "no_telp": "08112233445",
        "id": 2
    },
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL2Nhci1yZW50XC9wdWJsaWNcL2FwaVwvcmVnaXN0ZXIiLCJpYXQiOjE1ODYzMDY3NjYsImV4cCI6MTU4NjMxMDM2NiwibmJmIjoxNTg2MzA2NzY2LCJqdGkiOiJUVnJmV3FNdVRtWmM3NENrIiwic3ViIjoyLCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxMmZkZWM5NzE1M2ExNGUwYjA0NzU0NmFhIn0.BQnHI6U_SUr9yTPBR4u4JrChzeWQNbhfMXOw6-gNsKQ"
}
```

5. Route


api.php - Untitled (Workspace) - Visual Studio Code
car-rent > routes > api.php

```php
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

// Petugas
Route::post('register', 'petugascontroller@register');
Route::post('login', 'petugascontroller@login');
```