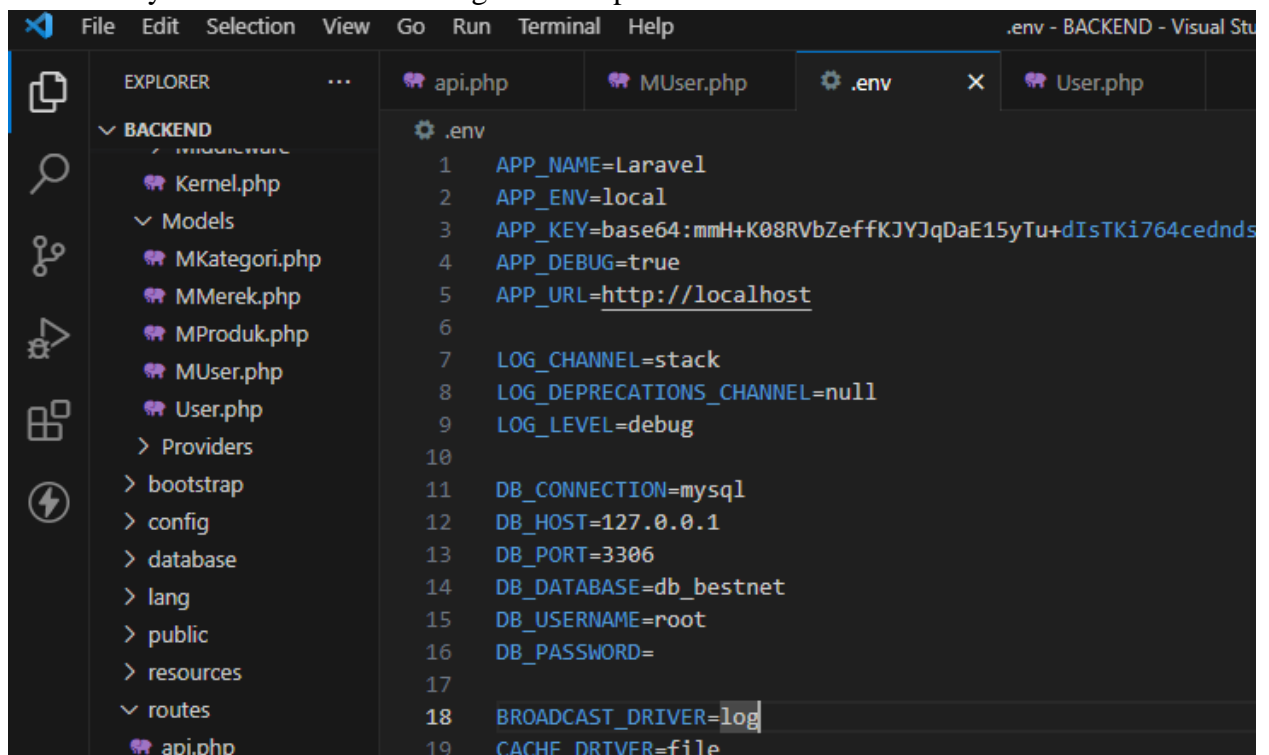


# Laporan Manual Book BackEnd PWBS-TA-Kopikap

Dalam hal ini saya sebagai backend dalam tim kopikap akan menjelaskan bagian yang akan saya buat. Pertama saya akan melakukan konfigurasi database ke dalam project. Selanjutnya saya akan membuat api yang akan nantinya di pakai oleh frontend. Api yang saya bikin disini ada empat yaitu api produk, merek, kategori, tampil produk perkategori, tampil merek perkategori, api pencarian, dan api upload gambar yang dimana keempat api ini akan digunakan oleh frontend.

## 1. Pembuatan api produk

- 1) Pertama saya akan melakukan setting database pada file .env



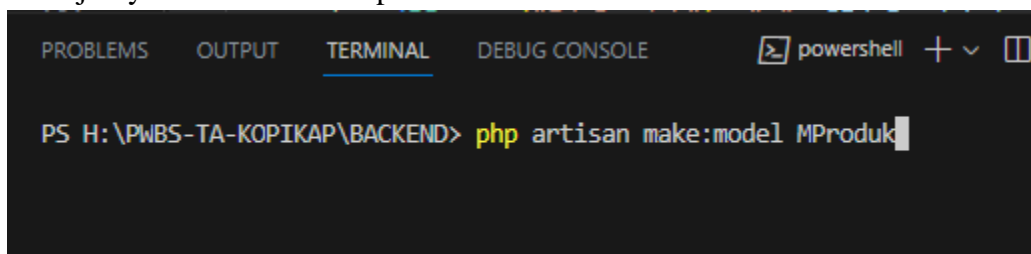
The screenshot shows the Visual Studio Code editor with the .env file open. The Explorer sidebar on the left shows the project structure with the Backend folder expanded, listing files like Kernel.php, Models, MKategori.php, MMerek.php, MProduk.php, MUser.php, User.php, Providers, bootstrap, config, database, lang, public, resources, routes, and api.php. The .env file content is as follows:

```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:mmH+K08RVbZeffKJYJqDaE15yTu+dIsTKi764cednds
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=db_bestnet
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
```

- 2) Selanjutnya melakukan Setting route api untuk membuat route api produk pada halaman api.php

```
Route::middleware('auth:sanctum')->get('/user', function (Request $request) {  
    return $request->user();  
});  
  
// Route untuk tampil data produk  
Route::get('/tampil', [Produk::class, 'tampil']);  
// Route Untuk detail produk  
Route::get('/detail/{parameter}', [Produk::class, 'detail']);  
// Route Untuk hapus data produk  
Route::delete('/delete/{parameter}', [Produk::class, 'delete']);  
// Route Untuk tambah data produk  
Route::post('/insert', [Produk::class, 'insert']);  
// Route untuk update data produk  
Route::put('/updateProduk/{parameter}', [Produk::class, 'updateProduk']);
```

- 3) Selanjutnya kita buat Model produk



The screenshot shows a PowerShell terminal window with the following tabs: PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The terminal title bar indicates it is a powershell window. The command entered in the terminal is: `PS H:\PWBS-TA-KOPIKAP\BACKEND> php artisan make:model MProduk`. The cursor is at the end of the command.

- 4) Selanjutnya buat model produk dan buat function untuk mengambil data dari database disini saya kasih nama function nya tampilData()

```

class MProduk extends Model
{
    protected $table = 'tbl_produk';

    function tampilData()
    {
        $query = DB::table('tbl_produk')
            ->select(
                "Id_Produk",
                "Nama_Produk",
                "Harga",
                "Stok_Produk",
                "Spesifikasi",
                "Foto_Produk",
                "Kategori",
                "Model"
            )
            ->orderBy("Id_Produk")
            ->get();

        return $query;
    }
}

```

- 5) Selanjutnya buat function detailData() di model produk untuk melihat detail dari data

```

function detailData($parameter){
    $query = DB::table('tbl_produk')
        ->select(
            "Id_Produk",
            "Nama_Produk",
            "Harga",
            "Stok_Produk",
            "Spesifikasi",
            "Foto_Produk",
            "Kategori",
            "Model",
        )
        ->where(DB::raw("Id_produk"), "=", $parameter)
        ->orderBy("Id_Produk")
        ->get();

    return $query;
}

```

- 6) Selanjutnya buat function untuk menghapus data di sini saya kasih nama deleteData()

```
// buat fungsi delete data
function deleteData($parameter)
{
    DB::table("tbl_produk")
        ->where(DB::raw("Id_Produk"), '=', $parameter)
        ->delete();
}
```

- 7) Selanjutnya buat function tambah data di modelproduk untuk menambahkan data, disini saya kasih nama saveData()

```
8) // buat fungsi tambah data
9) function saveData($Id_produk, $Nama_Produk, $Harga, $Stok_Produk,
    $Spesifikasi, $Foto_Produk,$Kategori, $Model)
10) {
11)     DB::table("tbl_produk")
12)     ->insert([
13)         "Id_Produk" => $Id_produk,
14)         "Nama_Produk" => $Nama_Produk,
15)         "Harga" => $Harga,
16)         "Stok_Produk" => $Stok_Produk,
17)         "Spesifikasi" => $Spesifikasi,
18)         "Foto_Produk" => $Foto_Produk,
19)         "Kategori" => $Kategori,
20)         "Model" => $Model
21)     ]);
22) }
```

- 8) Selanjutnya kita buat function untuk cek update data apakah data tersebut ada atau tidak

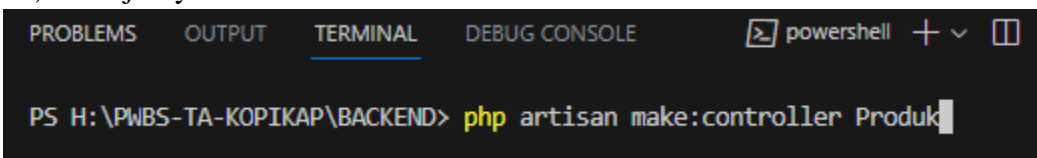
```
// Fungsi Untuk Cek Ubah Data
function checkUpdate($Id_Produk_Lama, $Id_Produk_baru)
{
    // tampilkan data
    $query = DB::table("tbl_produk")
        ->select("Id_Produk")
        ->where("Id_Produk", "=", $Id_Produk_baru)
        ->where(DB::raw("(Id_Produk)"), "!=" , $Id_Produk_Lama)
        ->get();

    return $query;
}
```

9) Setelah kita membuat function untuk cekupdate selanjutnya kita membuat function untuk update data.

```
// Update Data Kamar
function updateData(
    $Id_Produk,
    $Nama_Produk,
    $Harga,
    $Stok_Produk,
    $Spesifikasi,
    $Foto_Produk,
    $Kategori,
    $Model,
    $Id_Produk_Lama
) {
    DB::table("tbl_produk")
        ->where(DB::raw("Id_Produk"), "=", $Id_Produk_Lama)
        ->update([
            "Id_Produk" => $Id_Produk,
            "Nama_Produk" => $Nama_Produk,
            "Harga" => $Harga,
            "Stok_Produk" => $Stok_Produk,
            "Spesifikasi" => $Spesifikasi,
            "Foto_Produk" => $Foto_Produk,
            "Kategori" => $Kategori,
            "Model" => $Model
        ]);
}
```

10) Selanjutnya buat controller Produk



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal title bar indicates it is a PowerShell session. The command prompt shows the user is in the directory 'H:\PWBS-TA-KOPIKAP\BACKEND' and has executed the command 'php artisan make:controller Produk'.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  powershell + -
PS H:\PWBS-TA-KOPIKAP\BACKEND> php artisan make:controller Produk
```

11) Selanjutnya kita buat function untuk memanggil model Mproduk di controller kita

```
function __construct()  
{  
    $this->model = new MProduk();  
}
```

```
use Illuminate\Http\Request;  
use App\Models\MProduk;
```

12) Selanjutnya kita buat fungsi tampil untuk menampilkan data di controller kita .

```
// Function Untuk Tampil Data  
function tampil()  
{  
    // ambil fungsi dari viewData(dari Mkaryawan)  
    $data = $this->model->tampilData();  
  
    // tampilkan hasil dari "tbkaryawan"  
    return response([  
        "Produk" => $data  
    ], http_response_code());  
}
```

13) Selanjutnya kita buat fungsi detail untuk melihat detailData() di controller kita

```
function detail($parameter)  
{  
    // ambil fungsi dari viewData(dari Mkaryawan)  
    $data = $this->model->detailData($parameter);  
  
    // tampilkan hasil dari "tbkaryawan"  
    return response([  
        "Produk" => $data  
    ], http_response_code());  
}
```

14) Selanjutnya kita buat fungsi delete di controller

```

// buat fungsi untuk delete data
function delete($parameter)
{
    // cek data dari tbl_karyawan
    //(berdasarkan nik)
    $data = $this->model->detailData($parameter);

    // jika data ditemukan
    if (count($data) != 0) {
        // lakukan penghapusan data
        $data = $this->model->deleteData($parameter);
        // buat pesan dan status hasil penghapusan data
        $status = 1;
        $pesan = "Data Berhasil di Hapus";
    }
    // jika data tidak ditemukan
    else {
        // tampilkan pesan data gagal dihapus
        $status = 1;
        $pesan = "Data Gagal di Hapus ! (NIK tidak ditemuka";
    }

    // tampilkan hasil respon

    return response([
        "status" => $status,
        "pesan" => $pesan
    ], http_response_code());
}

```

Activate Windows

15) Selanjutnya buat fungsi tambah data di controller kita

```

function insert(Request $req)
{
    // ambil data hasil input
    $data = array(
        "Id_Produk" => $req->Id_Produk,
        "Nama_Produk" => $req->Nama_Produk,
        "Harga" => $req->Harga,
        "Stok_Produk" => $req->Stok_Produk,
        "Spesifikasi" => $req->Spesifikasi,
        "Foto_Produk" => $req->Foto_produk,
        "Kategori" => $req->Kategori,
        "Model" => $req->Model,
    );
    // baruu
    $parameter = ($data["Id_Produk"]);
    // cek apakah data karyawan (nik) sudah pernah tersimp
    $check = $this->model->detailData($parameter);

```

```

// jika data tidak ditemukan
if (count($check) == 0) {
    // lakukan proses penyimpanan
    $this->model->saveData($data["Id_Produk"], $data["Nama_Produk"]);
    // buat pesan dan status hasil penyimpanan data
    $status = 1;
    $pesan = "Data Berhasil disimpan";
}
// jika data tidak ditemukan
else {
    // tampilkan pesan data gagal disimpan
    $status = 0;
    $pesan = "Data Gagal disimpan";
}
// tampilkan hasil respon

return response([
    "status" => $status,
    "pesan" => $pesan
], http_response_code());

```



16) Selanjutnya buat fungsi update pada controller produk.

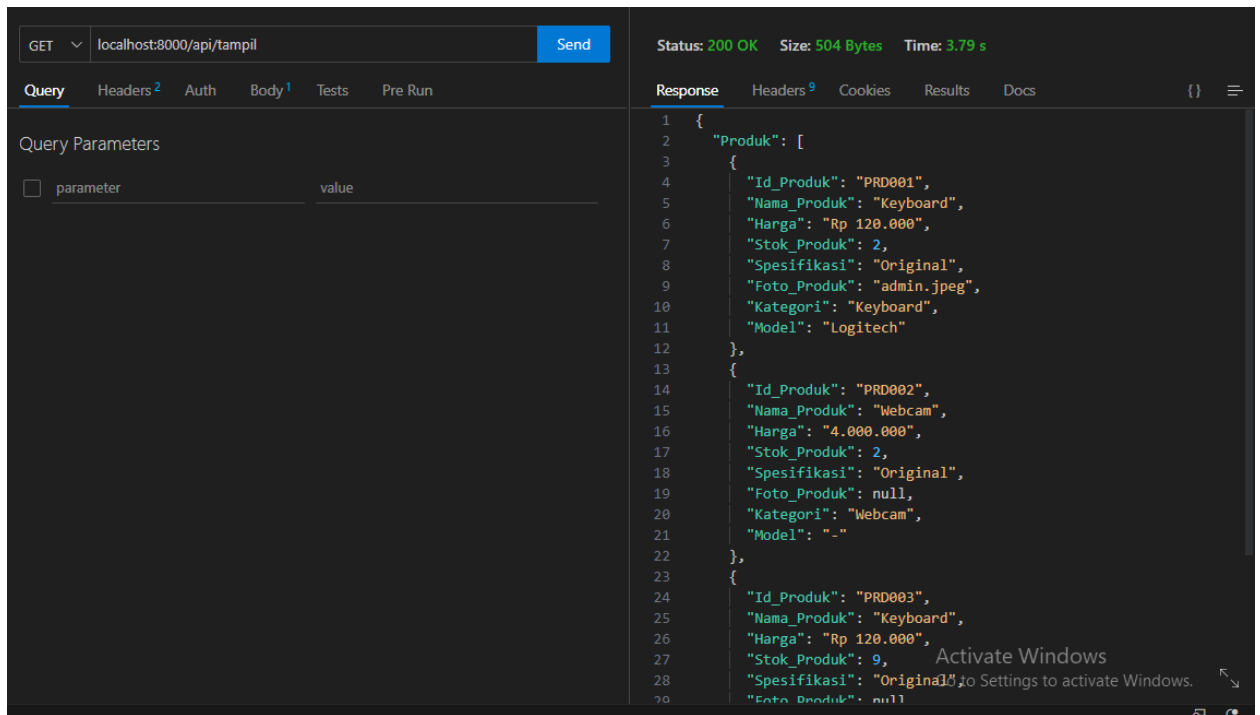
```
// Function untuk Update Data Kamar
function updateProduk(
    $parameter,
    Request $req
) {
    // Ambil data hasil input
    $data = array(
        "Id_Produk" => $req->Id_Produk,
        "Nama_Produk" => $req->Nama_Produk,
        "Harga" => $req->Harga,
        "Stok_Produk" => $req->Stok_Produk,
        "Spesifikasi" => $req->Spesifikasi,
        "Foto_Produk" => $req->Foto_Produk,
        "Kategori" => $req->Kategori,
        "Model" => $req->Model,
    );
}
```

```

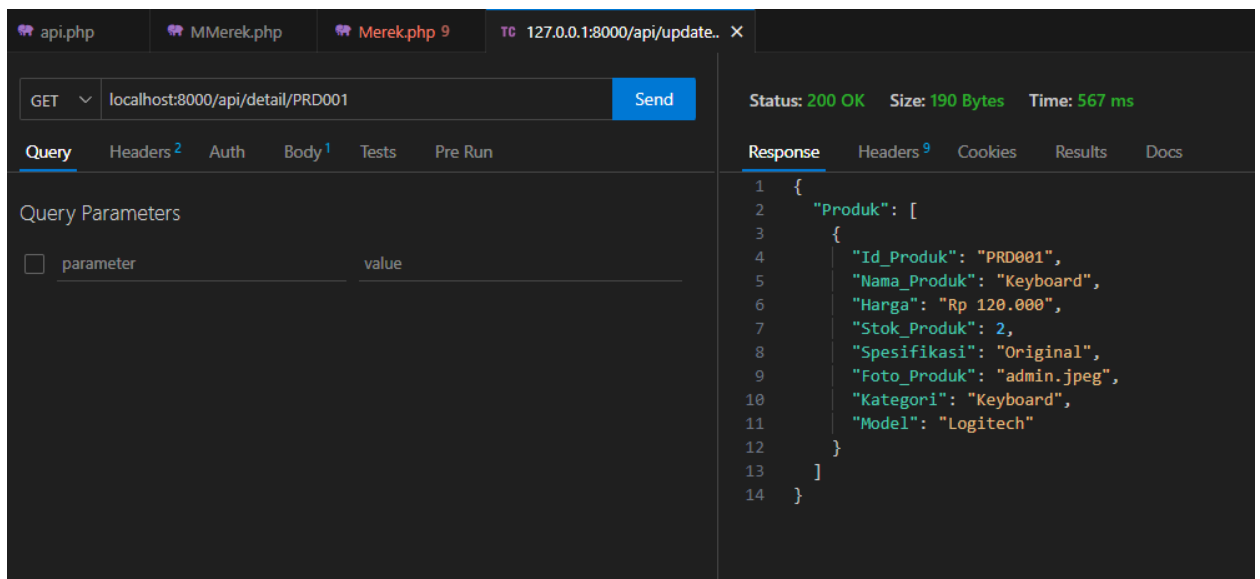
// Cek apakah data kamar tersedia/tidak
$cek = $this->model->checkUpdate($parameter, $data["Id_Produk"]);
// Jika data tidak ditemukan
if (count($cek) == 0) {
    // Ubah data kamar
    $this->model->updateData(
        $data["Id_Produk"],
        $data["Nama_Produk"],
        $data["Harga"],
        $data["Stok_Produk"],
        $data["Spesifikasi"],
        $data["Foto_Produk"],
        $data["Kategori"],
        $data["Model"],
        $parameter
    );
    // tampilkan pesan
    $status = "1";
    $pesan = "Data Berhasil di Ubah";
}
// Jika data tidak ditemukan
else {
    $status = 0;
    $pesan = "Data Gagal Diubah ! (Kode_Kamar Sudah Pernah Tersimpan)";
}
// Tampilkan pesan
return response([
    "status" => $status,
    "pesan" => $pesan
], http_response_code());
}

```

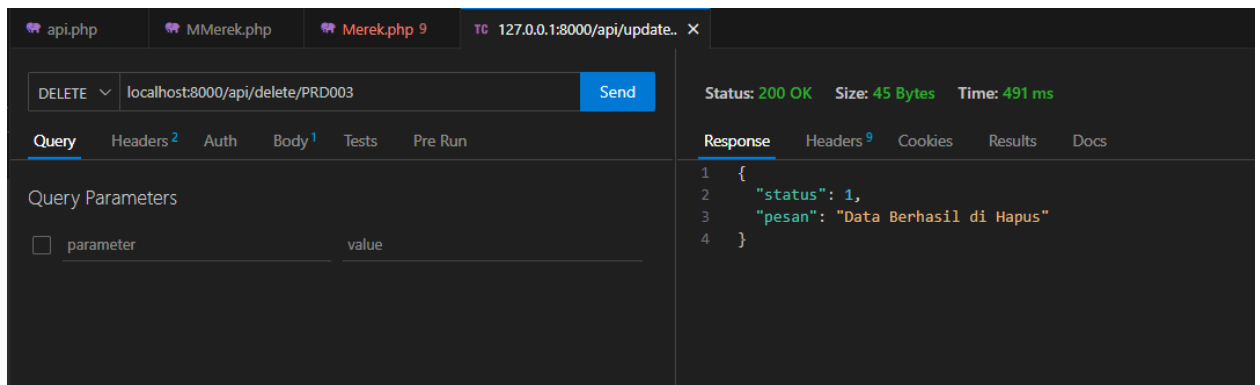
17) Selanjutnya setelah kita selesai buat model Mproduk dan Controller Produk kita testing apakah api yang kita buat sudah bisa tampil atau belum. Disini saya akan testing untuk tampil produk dengan menggunakan thunder client.



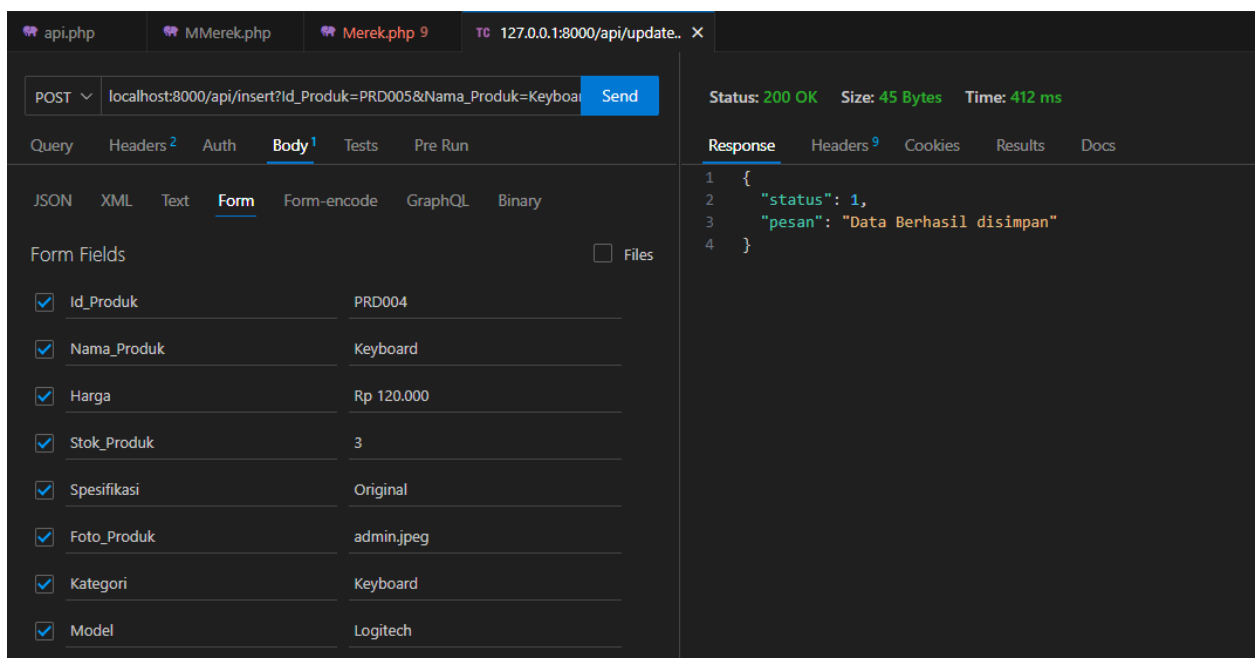
18) Testing api untuk detail produk.



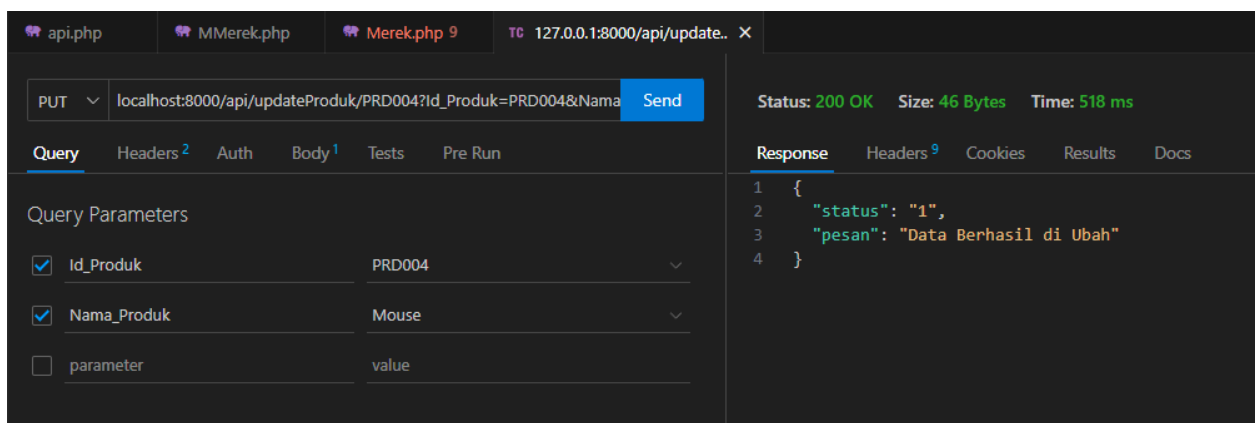
19) Testing api untuk delete produk.



20) Testing api untuk menyimpan data produk.



21) Testing api untuk update data produk

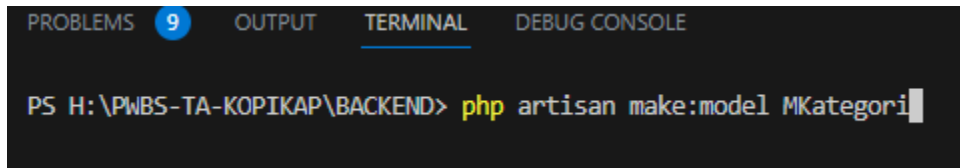


## 2. Membuat Api Kategori

- 1) Buka route/api lalu buat route CRUD untuk api Kategori

```
// Route untuk tampil data kategori
Route::get('/tampilkategori', [Kategori::class, 'tampilkategori']);
// Route untuk tampil data kategori
Route::get('/detailkategori/{parameter}', [Kategori::class, 'detailkategori']);
// Route Untuk hapus data kategori
Route::delete('/deletekategori/{parameter}', [Kategori::class, 'deletekategori']);
// Route Untuk tambah data kategori
Route::post('/insertkategori', [Kategori::class, 'insertkategori']);
// Route untuk update data kategori
Route::put('/updateKategori/{parameter}', [Kategori::class, 'updateKategori']);
```

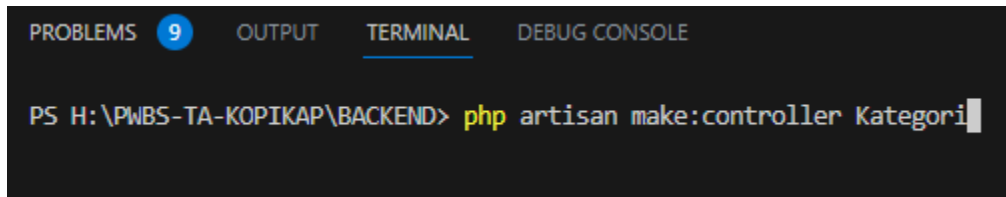
- 2) Buat model untuk api kategori dari terminal



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The command entered is `php artisan make:model MKategori`.

```
PS H:\PWBS-TA-KOPIKAP\BACKEND> php artisan make:model MKategori
```

- 3) Buat controller untuk api kategori dari terminal



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The command entered is `php artisan make:controller Kategori`.

```
PS H:\PWBS-TA-KOPIKAP\BACKEND> php artisan make:controller Kategori
```

- 4) Selanjutnya kita buat function `tampilDataKategori()` pada class model `Mkategori`

```

protected $table = 'tbl_kategori';

function tampilDataKategori()
{
    $query = DB::table('tbl_kategori')
        ->select(
            "Id_Kategori",
            "Nama_Kategori",
        )
        ->orderBy("Id_Kategori")
        ->get();

    return $query;
}

```

- 5) Selanjutnya buat function detailDataKategori() untuk melihat detail pada data.

```

function detailDataKategori($parameter){
    $query = DB::table('tbl_kategori')
        ->select(
            "Id_Kategori",
            "Nama_Kategori",
        )
        ->where(DB::raw("Id_Kategori"), "=", $parameter)
        ->orderBy("Id_Kategori")
        ->get();

    return $query;
}

```

- 6) Selanjutnya kita buat function untuk deleteDataKategori() untuk menghapus data kategori.

```

// buat fungsi delete data
function deleteDataKategori($parameter)
{
    DB::table("tbl_kategori")
        ->where(DB::raw("Id_Kategori"), '=', $parameter)
        ->delete();
}

```

7) Selanjutnya kita buat fungsi simpan data pada model MKategori

```
// buat fungsi tambah data
function saveDataKategori($Id_Kategori, $Nama_Kategori)
{
    DB::table("tbl_kategori")
    ->insert([
        "Id_Kategori" => $Id_Kategori,
        "Nama_Kategori" => $Nama_Kategori,
    ]);
}
```

8) Selanjutnya buat fungsi untuk cek update pada model Kategori

```
// Fungsi Untuk Cek Ubah Data
function CekUpdateKategori($Id_Kategori_Lama, $Id_Kategori_baru)
{
    // tampilkan data
    $query = DB::table("tbl_kategori")
        ->select("Id_Kategori")
        ->where("Id_Kategori", "=", $Id_Kategori_baru)
        ->where(DB::raw("(Id_Kategori)"), "!=" , $Id_Kategori_Lama)
        ->get();

    return $query;
}
```

9) Selanjutnya kita buat function untuk update data pada model MKategori

```
// Update Data Kamar
function updateDataKategori(
    $Id_Kategori,
    $Nama_Kategori,
    $Id_Kategori_Lama
) {
    DB::table("tbl_kategori")
        ->where(DB::raw("Id_Kategori"), "=", $Id_Kategori_Lama)
        ->update([
            "Id_Kategori" => $Id_Kategori,
            "Nama_Kategori" => $Nama_Kategori,
        ]);
}
```

- 10) Setelah selesai kita buat fungsi CRUD di model MKategori maka kita buat function CRUD pada controller Kategori

```
function __construct()
{
    $this->model = new MKategori();
}

// Function Untuk Tampil Data
function tampilkategori()
{
    // ambil fungsi dari viewData(dari Mkaryawan)
    $data = $this->model->tampilDataKategori();

    // tampilkan hasil dari "tbkaryawan"
    return response([
        "Kategori" => $data
    ], http_response_code());
}
```

- 11) Buat fungsi untuk detailkategori pada Controller Kategori.



```

function detailKategori($parameter)
{
    // ambil fungsi dari viewData(dari Mkaryawan)
    $data = $this->model->detailDataKategori($parameter);

    // tampilkan hasil dari "tbkaryawan"
    return response([
        "Kategori" => $data
    ], http_response_code());
}

```

12) Buat Fungsi untuk menghapus data pada Controller Kategori.

```

// buat fungsi untuk delete data
function deleteKategori($parameter)
{
    // cek data dari tbl_karyawan
    //(berdasarkan nik)
    $data = $this->model->detailDataKategori($parameter);

    // jika data ditemukan
    if (count($data) != 0) {
        // lakukan penghapusan data
        $data = $this->model->deleteDataKategori($parameter);
        // buat pesan dan status hasil penghapusan data
        $status = 1;
        $pesan = "Data Berhasil di Hapus";
    }
    // jika data tidak ditemukan
    else {
        // tampilkan pesan data gagal dihapus
        $status = 1;
        $pesan = "Data Gagal di Hapus ! (NIK tidak ditemukan !)";
    }

    // tampilkan hasil respon

    return response([
        "status" => $status,
        "pesan" => $pesan
    ], http_response_code());
}

```

13) Buat Fungsi tambah data pada Controller Kategori

```
// buat function untuk insert data
function insertkategori(Request $req)
{
    // ambil data hasil input
    $data = array(
        "Id_Kategori" => $req->Id_Kategori,
        "Nama_Kategori" => $req->Nama_Kategori,
    );
    // baruu
    $parameter =($data["Id_Kategori"]);
    // cek apakah data karyawan (nik) sudah pernah tersimpan/belum
    $check = $this->model->detailDataKategori($parameter);

    // jika data tidak ditemukan
    if (count($check) == 0) {
        // lakukan proses penyimpanan
        $this->model->saveDataKategori($data["Id_Kategori"], $data["Nama_Kategori"]);
        // buat pesan dan status hasil penyimpanan data
        $status = 1;
        $pesan = "Data Berhasil disimpan";
    }
    // jika data tidak ditemukan
    else {

        // tampilkan pesan data gagal disimpan
        $status = 0;
        $pesan = "Data Gagal disimpan";
    }
    // tampilkan hasil respon

    return response([
        "status" => $status
```

14) Buat Fungsi untuk update data pada Controller Kategori

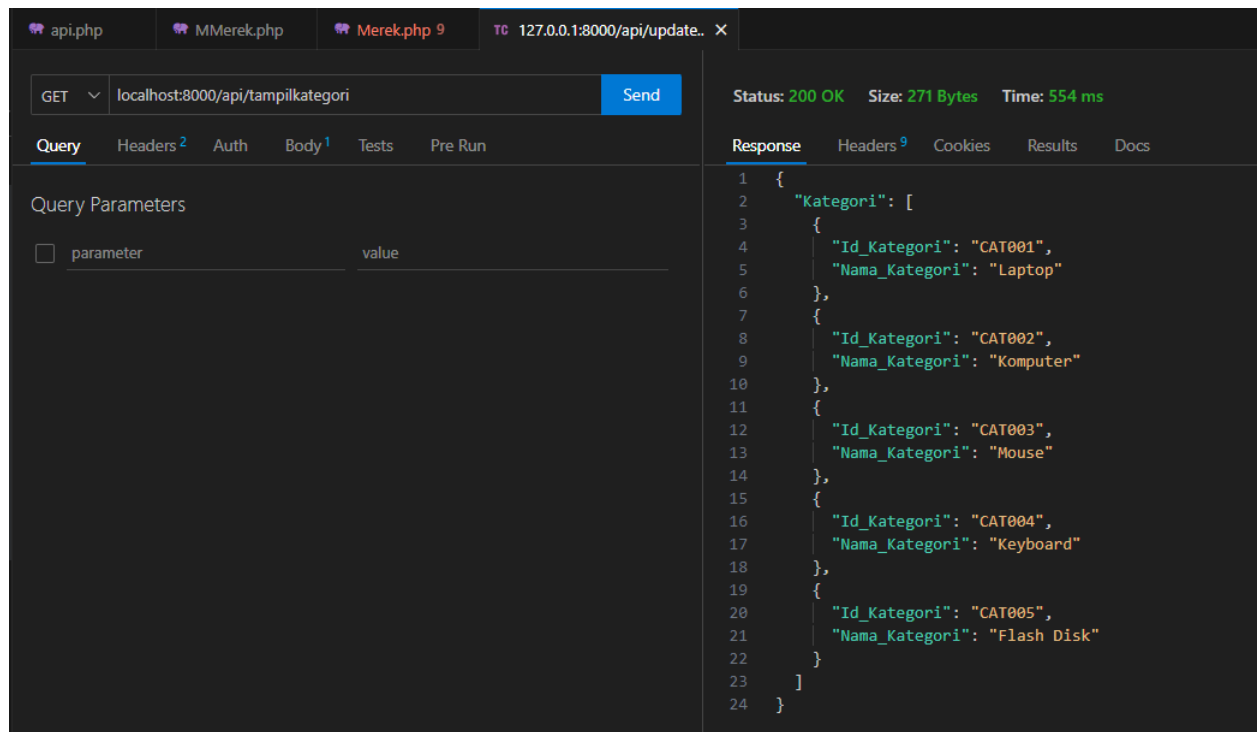
```

// Function untuk Update Data Kamar
function updateKategori(
    $parameter,
    Request $req
) {
    // Ambil data hasil input
    $data = array(
        "Id_Kategori" => $req->Id_Kategori,
        "Nama_Kategori" => $req->Nama_Kategori,
    );

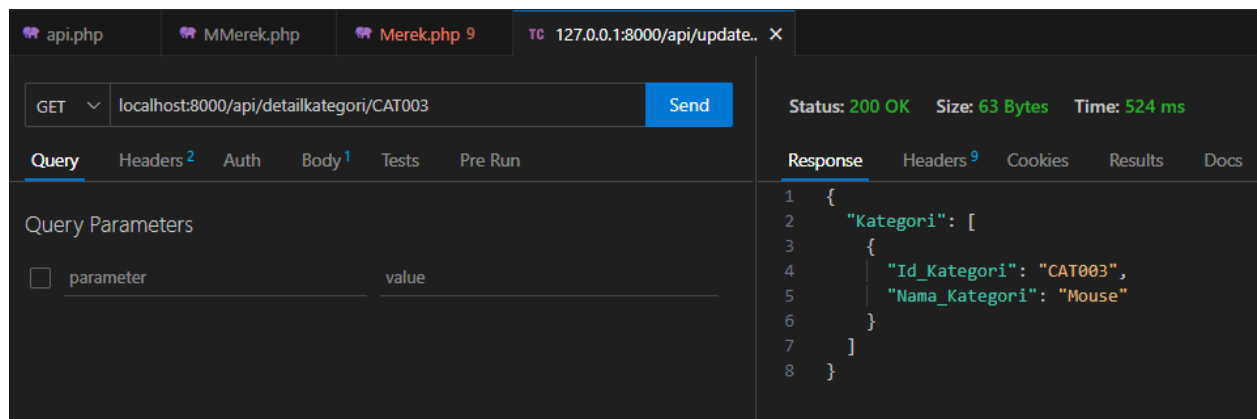
    // Cek apakah data kamar tersedia/tidak
    $cek = $this->model->CekUpdateKategori($parameter, $data["Id_Kategori"]);
    // Jika data tidak ditemukan
    if (count($cek) == 0) {
        // Ubah data kamar
        $this->model->updateDataKategori(
            $data["Id_Kategori"],
            $data["Nama_Kategori"],
            $parameter
        );
        // tampilkan pesan
        $status = "1";
        $pesan = "Data Berhasil di Ubah";
    }
    // Jika data tidak ditemukan
    else {
        $status = 0;
        $pesan = "Data Gagal Diubah ! (Kode_Kamar Sudah Pernah Tersimpan)";
    }
    // Tampilkan pesan
    return response([

```

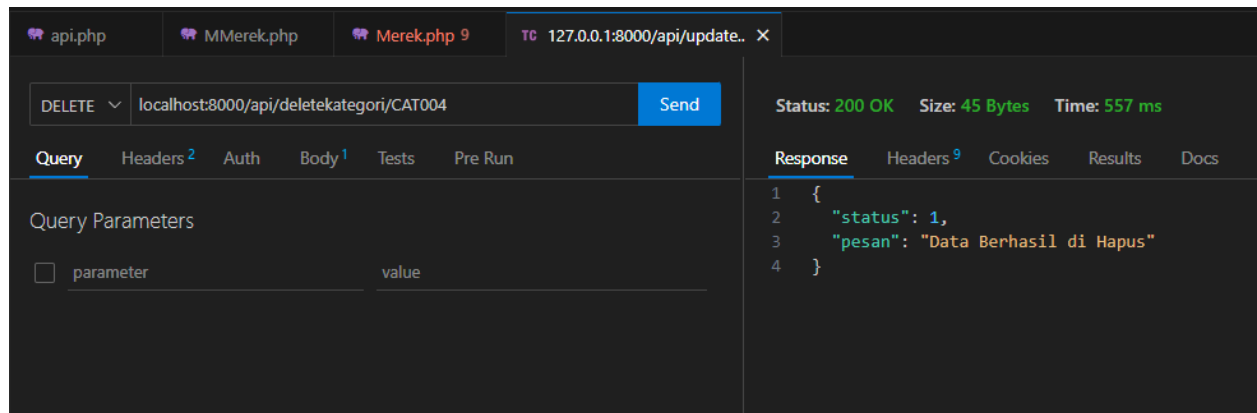
- 15) Setelah selesai kita membuat fungsi CRUD di model MKategori dan di Controller Kategori selanjutnya kita akan melakukan testing apakah api kategori yang sudah kita buat bisa tampil atau belum. Disini saya akan melakukan testing untuk tampil data kategori.



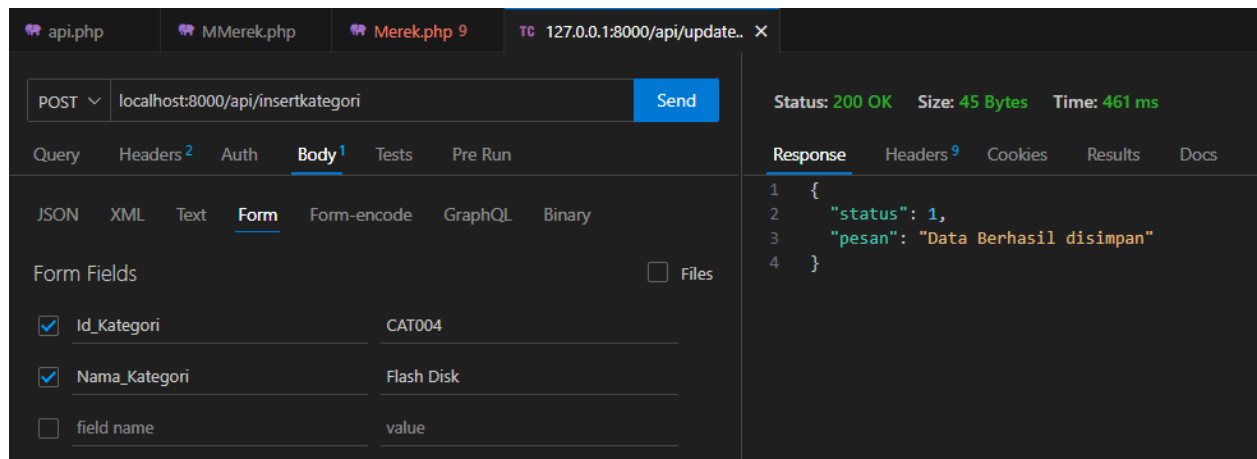
16) Testing api untuk detail pada api kategori.



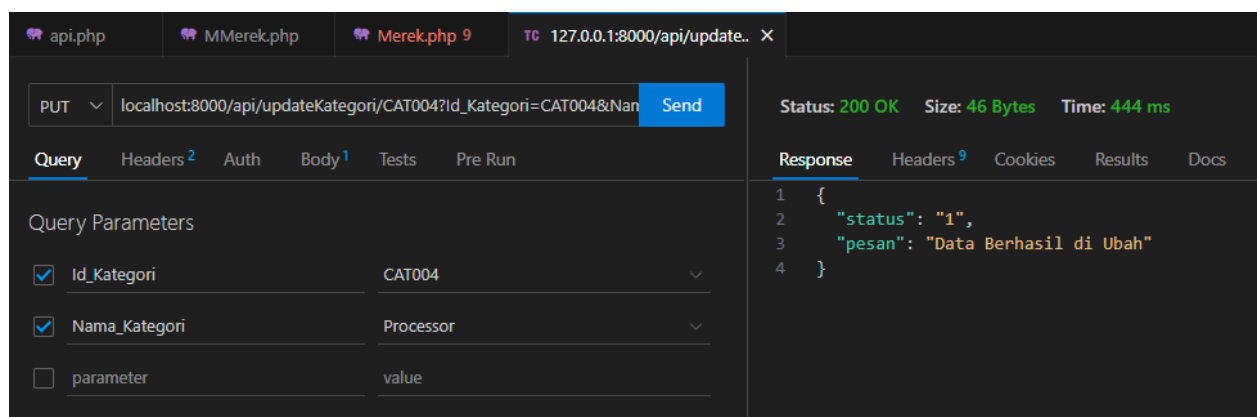
17) Testing api untuk delete pada api kategori.



18) Testing api insert kategori untuk menyimpan data pada api kategori.



19) Testing api untuk update data pada api kategori.

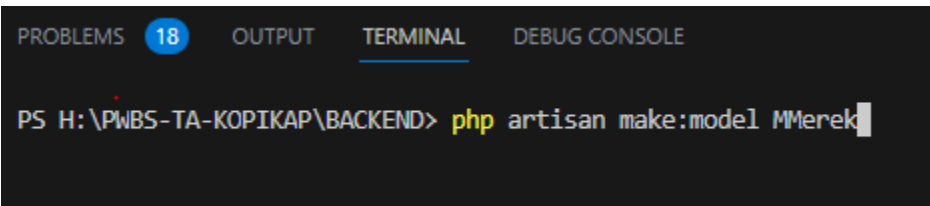


### 3. Buat Api Merek

- 1) Buka Route/api.php selanjutnya kita buat route untuk CRUD Merek pada setting route/api.php

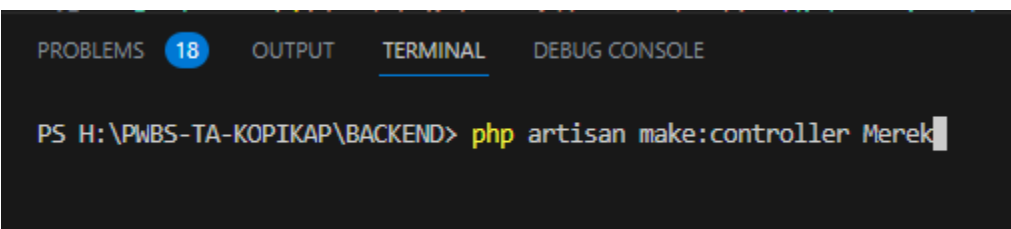
```
// Route untuk tampil data merek
Route::get('/tampilmerk', [Merek::class, 'tampilmerk']);
// Route untuk tampil data merek
Route::get('/detailmerk/{parameter}', [Merek::class, 'detailmerk']);
// Route Untuk hapus data merek
Route::delete('/deletemerek/{parameter}', [Merek::class, 'deletemerek']);
// Route Untuk tambah data merek
Route::post('/insertmerk', [Merek::class, 'insertmerk']);
// Route untuk update data kategori
Route::put('/updateMerek/{parameter}', [Merek::class, 'updateMerek']);
```

- 2) Selanjutnya kita buat model untuk api Merek.



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS' (with a blue circle containing the number 18), 'OUTPUT', 'TERMINAL' (which is selected and underlined), and 'DEBUG CONSOLE'. Below the tabs, the command prompt shows the path 'PS H:\PWBS-TA-KOPIKAP\BACKEND>' followed by the command 'php artisan make:model MMerek' with a cursor at the end.

- 3) Selanjutnya kita buat Controller untuk api Merek



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS' (with a blue circle containing the number 18), 'OUTPUT', 'TERMINAL' (which is selected and underlined), and 'DEBUG CONSOLE'. Below the tabs, the command prompt shows the path 'PS H:\PWBS-TA-KOPIKAP\BACKEND>' followed by the command 'php artisan make:controller Merek' with a cursor at the end.

- 4) Selanjutnya kita buat fungsi tampil datamerek pada model MMerek

```

protected $table = 'tbl_merek';

function tampilDataMerek()
{
    $query = DB::table('tbl_merek')
        ->select(
            "Id_Merek",
            "Nama_Merek",
        )
        ->orderBy("Id_Merek")
        ->get();

    return $query;
}

```

- 5) Buat Fungsi detailDataMerek() pada MMerek.

```

function detailDataMerek($parameter){
    $query = DB::table('tbl_merek')
        ->select(
            "Id_Merek",
            "Nama_Merek",
        )
        ->where(DB::raw("Id_Merek"), "=", $parameter)
        ->orderBy("Id_Merek")
        ->get();

    return $query;
}

```

- 6) Buat fungsi deleteDataMerek untuk menghapus data merek.

```

// buat fungsi delete data
function deleteDataMerek($parameter)
{
    DB::table("tbl_merek")
        ->where(DB::raw("Id_Merek"), '=', $parameter)
        ->delete();
}

```

7) Buat fungsi InsertDataMerek() untuk menyimpan data.

```
// buat fungsi tambah data
function saveDataMerek($Id_Merek, $Nama_Merek)
{
    DB::table("tbl_merek")
    ->insert([
        "Id_Merek" => $Id_Merek,
        "Nama_Merek" => $Nama_Merek,
    ]);
}
```

8) Buat fungsi cek update pada model MMerek.

```
// Fungsi Untuk Cek Ubah Data
function CekUpdateMerek($Id_Merek_Lama, $Id_Merek_baru)
{
    // tampilkan data
    $query = DB::table("tbl_merek")
    ->select("Id_Merek")
    ->where("Id_Merek", "=", $Id_Merek_baru)
    ->where(DB::raw("(Id_Merek)"), "!=", $Id_Merek_Lama)
    ->get();

    return $query;
}
```

9) Buat fungsi updatedata pada model MMerek



```
// Update Data Kamar
function updateDataMerek(
    $Id_Merek,
    $Nama_Merek,
    $Id_Merek_Lama
) {
    DB::table("tbl_merek")
        ->where(DB::raw("Id_Merek"), "=", $Id_Merek_Lama)
        ->update([
            "Id_Merek" => $Id_Merek,
            "Nama_Merek" => $Nama_Merek,
        ]);
}
```

- 10) Setelah selesai buat function CRUD di model MMerek selanjutnya kita akan membuat function untuk CRUD di controller Merek. Yang pertama kita buat fungsi untuk tampilmerek() pada controller Merek.

```
function __construct()
{
    $this->model = new MMerek();
}

// Function Untuk Tampil Data
function tampilmerek()
{
    // ambil fungsi dari viewData(dari Mkaryawan)
    $data = $this->model->tampilDataMerek();

    // tampilkan hasil dari "tbkaryawan"
    return response([
        "Merek" => $data
    ], http_response_code());
}
```

- 11) Selanjutnya buat function detailmerek() pada controller MMerek.

```

function detailmerek($parameter)
{
    // ambil fungsi dari viewData(dari Mkaryawan)
    $data = $this->model->detailDataMerek($parameter);

    // tampilkan hasil dari "tbkaryawan"
    return response([
        "Merek" => $data
    ], http_response_code());
}

```

12) Selanjutnya buat fungsi deletemerek() pada controller Merek.

```

// buat fungsi untuk delete data
function deletemerek($parameter)
{
    // cek data dari tbl_karyawan
    //(berdasarkan nik)
    $data = $this->model->detailDataMerek($parameter);

    // jika data ditemukan
    if (count($data) != 0) {
        // lakukan penghapusan data
        $data = $this->model->deleteDataMerek($parameter);
        // buat pesan dan status hasil penghapusan data
        $status = 1;
        $pesan = "Data Berhasil di Hapus";
    }
    // jika data tidak ditemukan
    else {
        // tampilkan pesan data gagal dihapus
        $status = 1;
        $pesan = "Data Gagal di Hapus ! (NIK tidak ditemukan !)";
    }

    // tampilkan hasil respon

    return response([
        "status" => $status,
        "pesan" => $pesan
    ], http_response_code());
}

```

13) Selanjutnya kita buat fungsi updateMerek() pada Controller Merek.

```
// buat function untuk insert data
function insertmerek(Request $req)
{
    // ambil data hasil input
    $data = array(
        "Id_Merek" => $req->Id_Merek,
        "Nama_Merek" => $req->Nama_Merek,
    );
    // baruu
    $parameter =($data["Id_Merek"]);
    // cek apakah data karyawan (nik) sudah pernah tersimpan/belum
    $check = $this->model->detailDataMerek($parameter);

    // jika data tidak ditemukan
    if (count($check) == 0) {
        // lakukan proses penyimpanan
        $this->model->saveDataMerek($data["Id_Merek"], $data["Nama_Merek"]);
        // buat pesan dan status hasil penyimpanan data
        $status = 1;
        $pesan = "Data Berhasil disimpan";
    }
    // jika data tidak ditemukan
    else {

        // tampilkan pesan data gagal disimpan
        $status = 0;
        $pesan = "Data Gagal disimpan";
    }
    // tampilkan hasil respon
}
```

14) Selanjutnya kita buat fungsi updateMerek() pada Controller Merek.

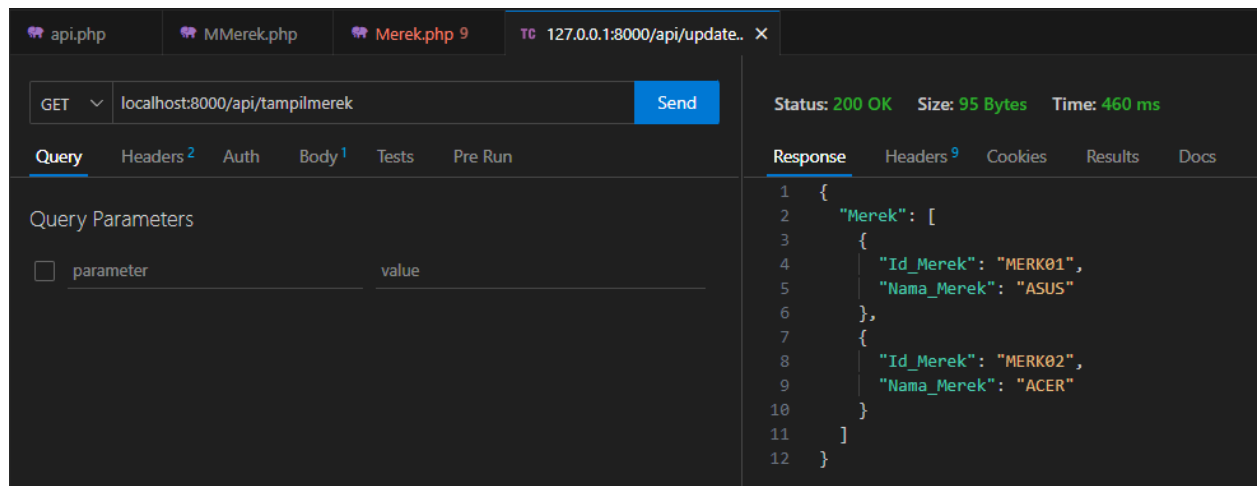
```

function updateMerek($parameter, Request $req)
{
    // Ambil data hasil input
    $data = array(
        "Id_Merek" => $req->Id_Merek,
        "Nama_Merek" => $req->Nama_Merek,
    );

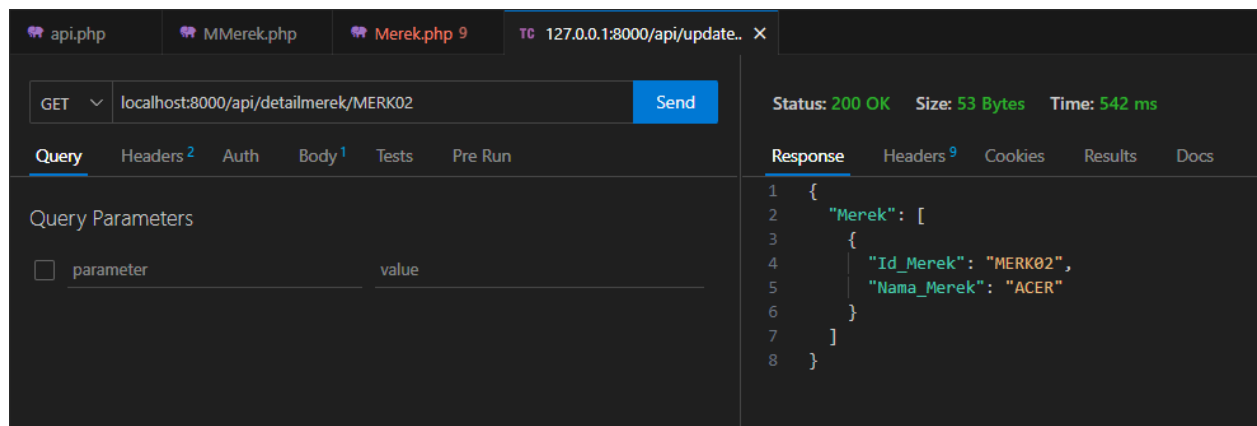
    // Cek apakah data kamar tersedia/tidak
    $cek = $this->model->CekUpdateMerek($parameter, $data["Id_Merek"]);
    // Jika data tidak ditemukan
    if (count($cek) == 0) {
        // Ubah data kamar
        $this->model->updateDataMerek(
            $data["Id_Merek"],
            $data["Nama_Merek"],
            $parameter
        );
        // tampilkan pesan
        $status = "1";
        $pesan = "Data Berhasil di Ubah";
    }
    // Jika data tidak ditemukan
    else {
        $status = 0;
        $pesan = "Data Gagal Diubah ! (Kode_Kamar Sudah Pernah Tersimpan)";
    }
    // Tampilkan pesan
    return response([
        "status" => $status,
        "pesan" => $pesan
    ], http_response_code());
}

```

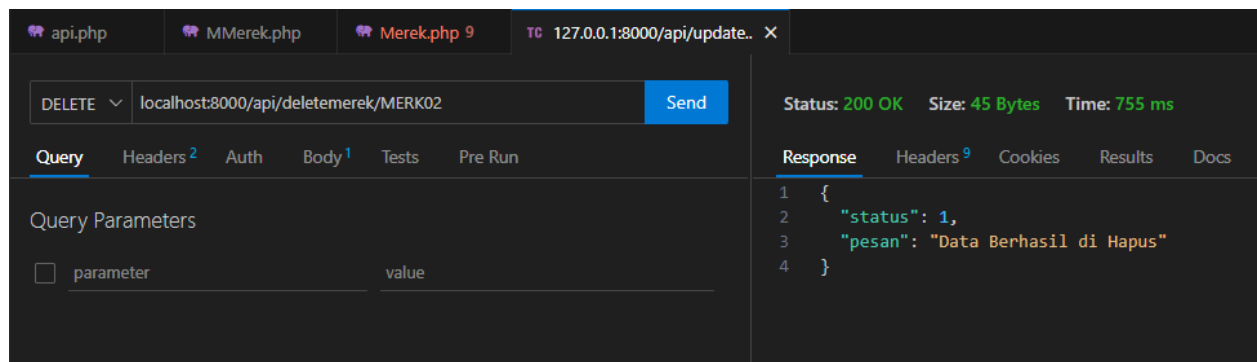
- 15) Setelah selesai kita membuat function CRUD di model MMerek dan di Controller Merek selanjutnya kita akan melakukan testing apakah api merek yang kita sudah buat sudah bisa atau belum. Disini saya akan coba testing untuk menampilkan data pada api merek.



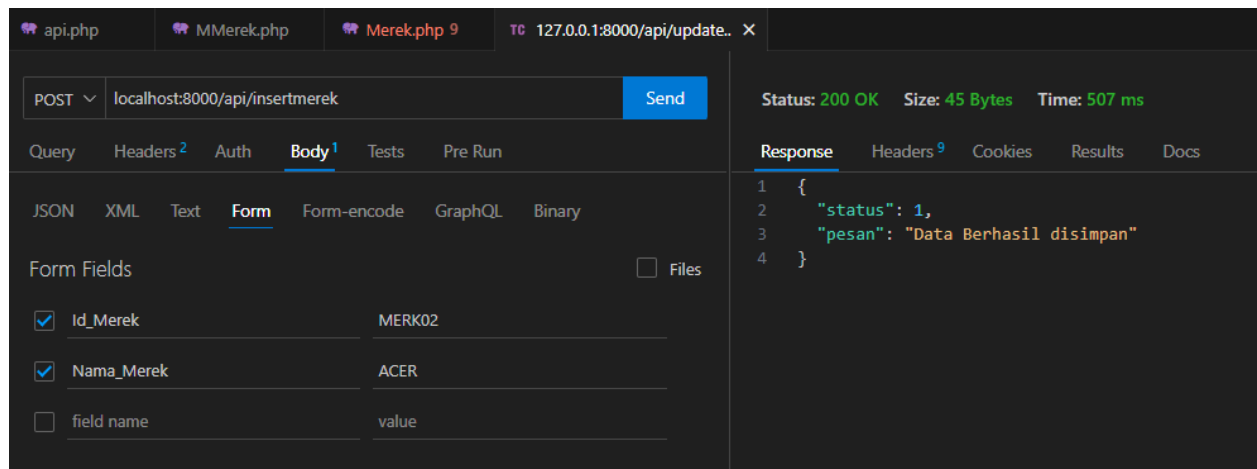
16) Testing api untuk melihat detail data pada api merek.



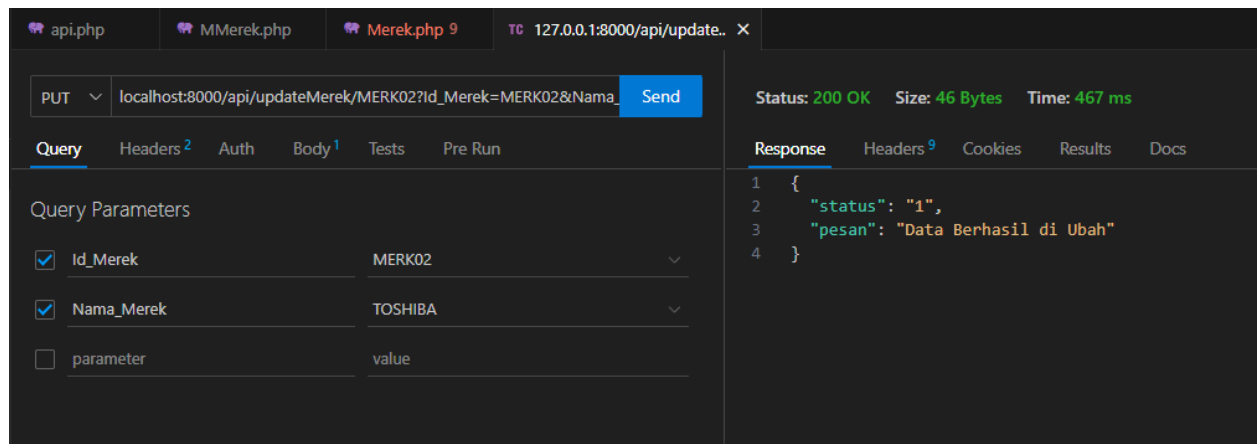
17) Testing api untuk menghapus data pada api merek.



18) Testing api untuk menyimpan data pada api merek.



19) Testing api untuk mengubah data pada api merek,



## 4. Buat Api Tampil per Kategori

### 1) Tampil Laptop

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan laptop

```
// Route untuk tampil produk per laptop
Route::get('/tampilLaptop', [Produk::class, 'tampilLaptop']);
```

- Selanjutnya buat model untuk tampil laptop di MProduk()

```
// tampil produk laptop
function tampilLaptop()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select(['tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk',
    'tbl_produk.Harga', 'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek'])
    ->where('Nama_Kategori', 'Laptop')
    ->get();

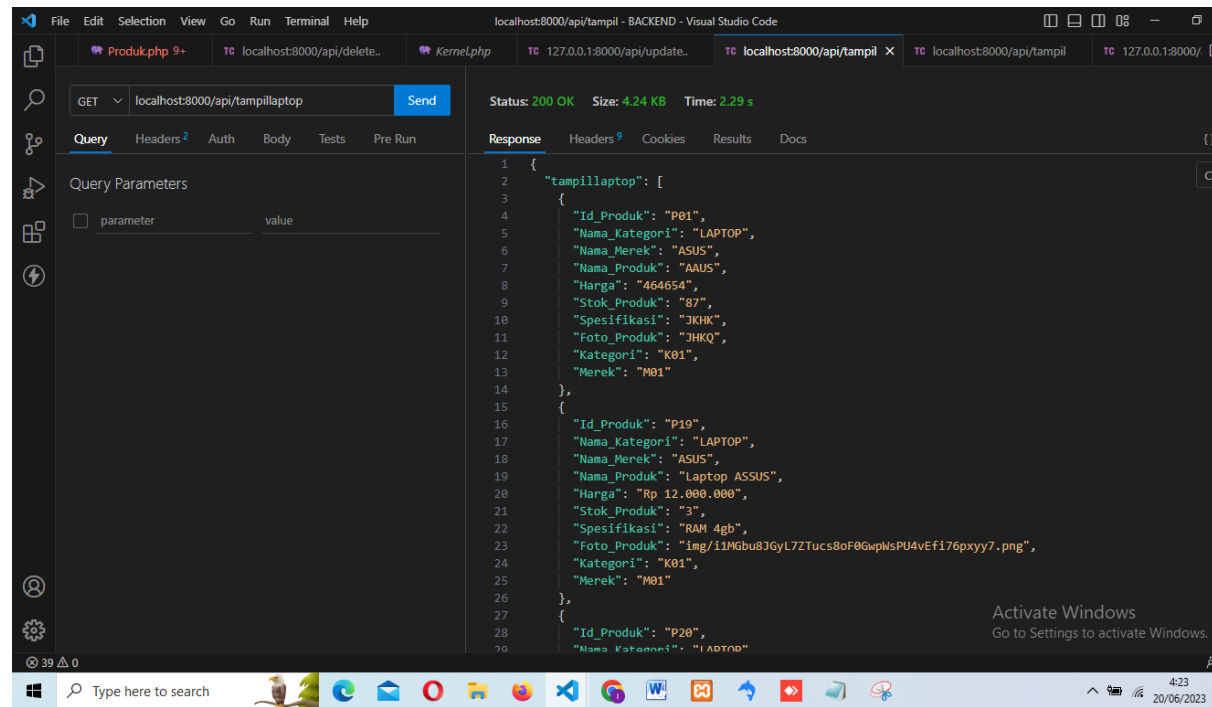
    return $query;
}
```

- Setelah selesai buat model untuk tampil laptop selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori laptop
function tampilLaptop()
{
    // ambil function tampilLaptop dari MProduk
    $data = $this->model->tampilLaptop();

    // Tampilkan hasil produk per kategori Laptop
    return response([
        "tampilLaptop" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori laptop



## 2) Tampil Monitor

- Pertama buat route untuk api tampil per kategori disini saya akan menampilkan monitor

```
// Route untuk tampil produk per monitor
Route::get('/tampilmonitor', [Produk::class, 'tampilmonitor']);
```



- Selanjutnya buat model untuk tampil monitor di MProduk()

```
// tampil produk monitor
function tampilMonitor()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Monitor')
    ->get();

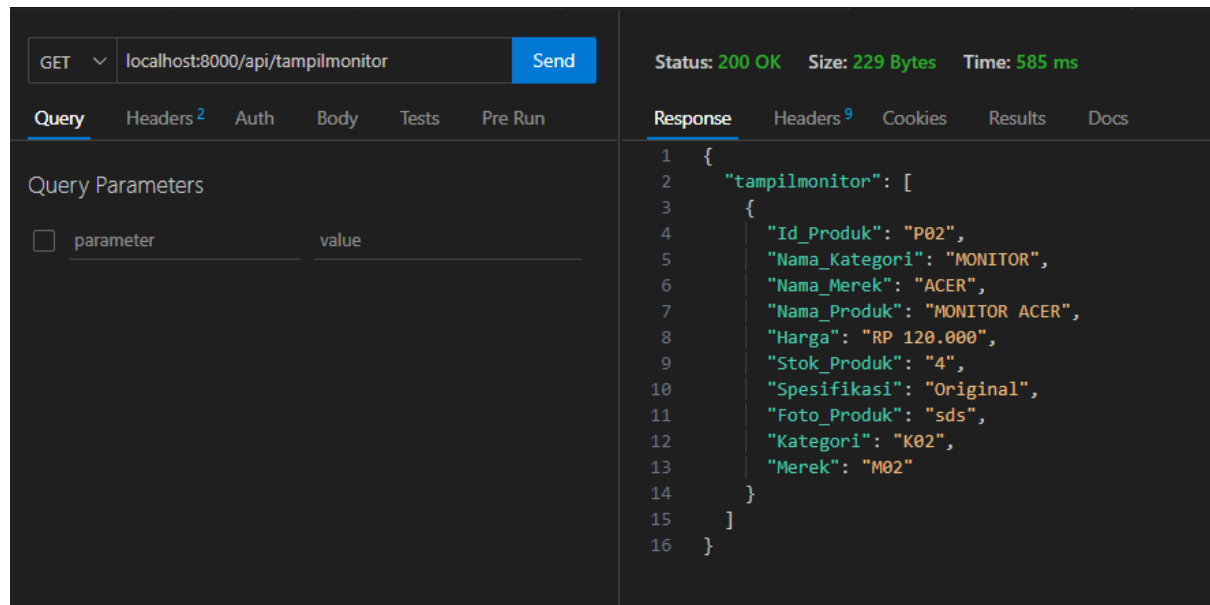
    return $query;
}
```

- Setelah selesai buat model untuk tampil laptop selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori monitor
function tampilmonitor()
{
    // ambil function tampilMonitor dari MProduk
    $data = $this->model->tampilMonitor();

    // Tampilkan hasil produk per kategori Monitor
    return response([
        "tampilmonitor" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori monitor



### 3) Tampil Keyboard

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Keyboard

```

// Route untuk tampil produk per keyboard
Route::get('/tampilkeyboard', [Produk::class, 'tampilkeyboard']);

```

- Selanjutnya buat model untuk tampil monitor di MProduk()

```

// tampil produk keyboard
function tampilKeyboard()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori.Nama_Kategori', 'tbl_merek.Nama_Merek', 'tbl_produk.Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori.Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Keyboard')
    ->get();

    return $query;
}

```

- Setelah selesai buat model untuk tampil keyboard selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori keyboard
function tampilkeyboard()
{
    // ambil function tampilKeyboard dari MProduk
    $data = $this->model->tampilKeyboard();

    // Tampilkan hasil produk per kategori Keyboard
    return response([
        "tampilkeyboard" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori keyboard

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilkeyboard
- Status:** 200 OK
- Size:** 224 Bytes
- Time:** 643 ms
- Response:**

```
1 {
2   "tampilkeyboard": [
3     {
4       "Id_Produk": "P03",
5       "Nama_Kategori": "KEYBOARD",
6       "Nama_Merek": "MSI",
7       "Nama_Produk": "KEYBOARD",
8       "Harga": "RP 80.000",
9       "Stok_Produk": "9",
10      "Spesifikasi": "ORIGINAL",
11      "Foto_Produk": "SS",
12      "Kategori": "K05",
13      "Merek": "M04"
14    }
15  ]
16 }
```

#### 4) Tampil Mouse

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Mouse

```
// Route untuk tampil produk per mouse
Route::get('/tampilmouse', [Produk::class, 'tampilmouse']);
```

- Selanjutnya buat model untuk tampil Mouse di MProduk()

```
// tampil produk mouse
function tampilMouse()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'MOUSE')
    ->get();

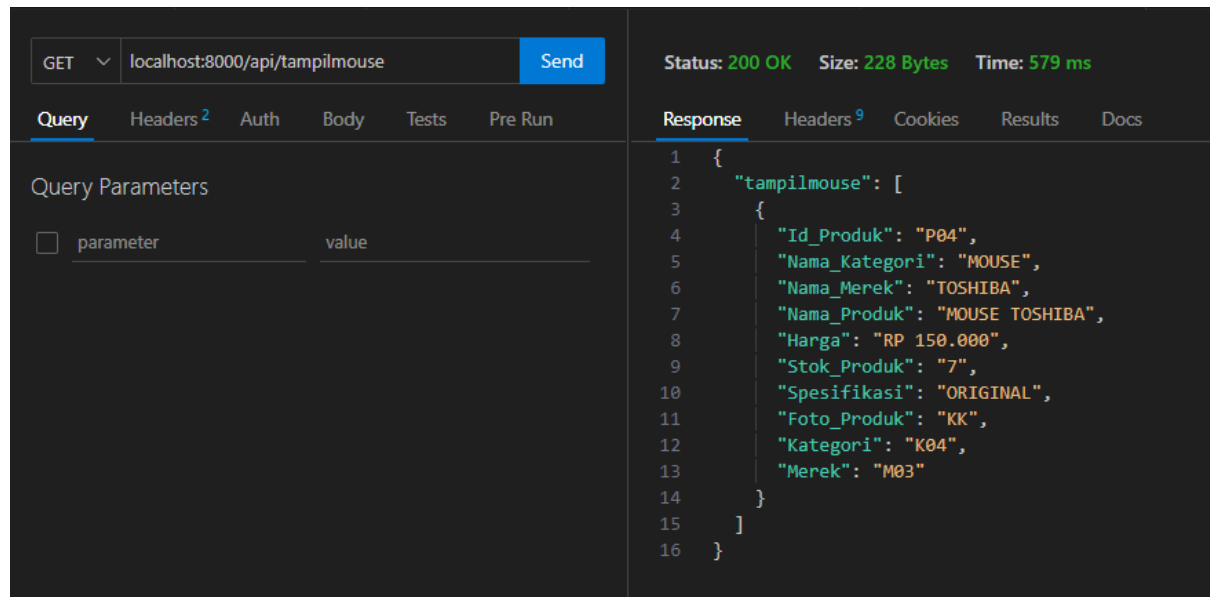
    return $query;
}
```

- Setelah selesai buat model untuk tampil Mouse selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori mouse
function tampilmouse()
{
    // ambil function tampilMouse dari MProduk
    $data = $this->model->tampilMouse();

    // Tampilkan hasil produk per kategori Mouse
    return response([
        "tampilmouse" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori mouse



## 5) Tampil Printer

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Printer

```

// Route untuk tampil produk per printer
Route::get('/tampilprinter', [Produk::class, 'tampilprinter']);

```

- Selanjutnya buat model untuk tampil Printer di MProduk()

```

// tampil produk mouse
function tampilMouse()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'MOUSE')
    ->get();

    return $query;
}

```

- Setelah selesai buat model untuk tampil Printer selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori printer
function tampilprinter()
{
    // ambil function tampilPrinter dari MProduk
    $data = $this->model->tampilPrinter();

    // Tampilkan hasil produk per kategori Printer
    return response([
        "tampilprinter" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori printer

Query	Headers <sup>2</sup>	Auth	Body	Tests	Pre Run	Response	Headers <sup>9</sup>	Cookies	Results	Docs
Query Parameters										
<input type="checkbox"/> parameter										
value										
						1 {				
						2 "tampilprinter": [				
						3 {				
						4 "Id_Produk": "P05",				
						5 "Nama_Kategori": "PRINTER",				
						6 "Nama_Merek": "SAMSUNG",				
						7 "Nama_Produk": "PRINTER SAMSUNG",				
						8 "Harga": "RP 3.000.000",				
						9 "Stok_Produk": "3",				
						10 "Spesifikasi": "ORIGINAL",				
						11 "Foto_Produk": "--",				
						12 "Kategori": "K03",				
						13 "Merek": "M06"				
						14 }				
						15 ]				
						16 }				

## 6) Tampil Speaker

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Speaker

```
// Route untuk tampil produk per speaker  
Route::get('/tampilspeaker', [Produk::class, 'tampilspeaker']);
```

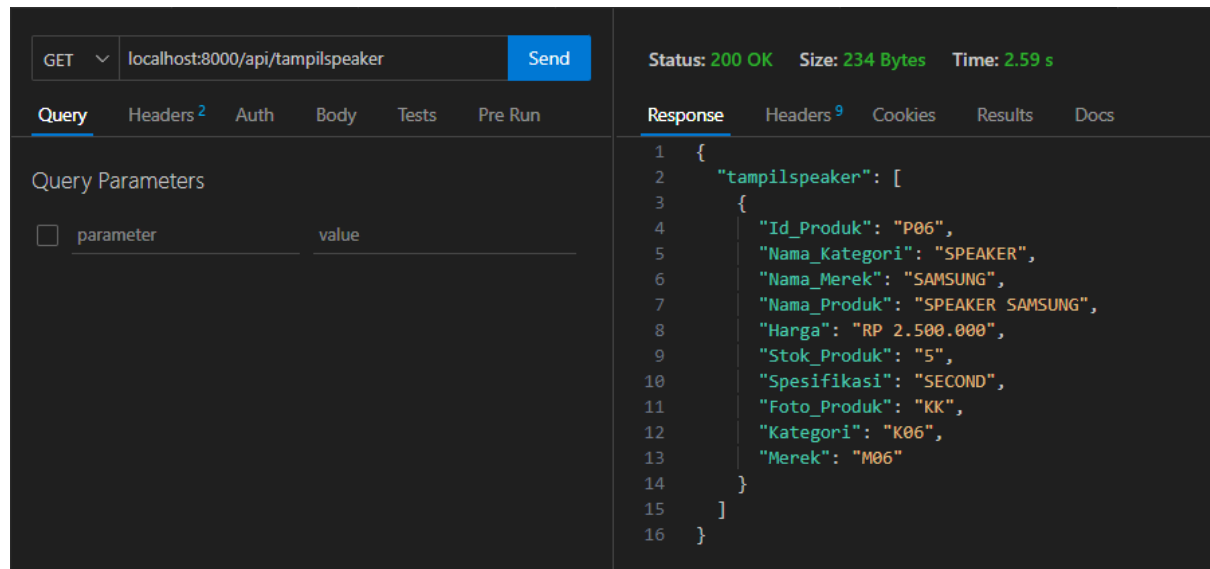
- Selanjutnya buat model untuk tampil Speaker di MProduk()

```
// tampil produk speaker  
function tampilSpeaker()  
{  
    $query = DB::table('tbl_produk')  
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')  
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')  
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',  
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',  
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')  
    ->where('Nama_Kategori', 'Speaker')  
    ->get();  
  
    return $query;  
}
```

- Setelah selesai buat model untuk tampil Speaker selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori speaker  
function tampilspeaker()  
{  
    // ambil function tampilSpeaker dari MProduk  
    $data = $this->model->tampilSpeaker();  
  
    // Tampilkan hasil produk per kategori Speaker  
    return response([  
        "tampilspeaker" => $data  
    ], http_response_code());  
}
```

- Testing api untuk menampilkan produk per kategori Speaker



## 7) Tampil Headset

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Headset

```

// Route untuk tampil produk per headset
Route::get('/tampilheadset', [Produk::class, 'tampilheadset']);

```

- Selanjutnya buat model untuk tampil Headset di MProduk()

```

// tampil produk headset
function tampilHeadset()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Headset')
    ->get();

    return $query;
}

```



- Setelah selesai buat model untuk tampil Headset selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori headset
function tampilheadset()
{
    // ambil function tampilHeadset dari MProduk
    $data = $this->model->tampilHeadset();

    // Tampilkan hasil produk per kategori Headset
    return response([
        "tampilheadset" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Headset

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilheadset
- Status:** 200 OK
- Size:** 224 Bytes
- Time:** 816 ms
- Response:** A JSON object with the following structure:

```
{
  "tampilheadset": [
    {
      "Id_Produk": "P07",
      "Nama_Kategori": "HEADSET",
      "Nama_Merek": "HP",
      "Nama_Produk": "HEADSET HP",
      "Harga": "RP 500.000",
      "Stok_Produk": "8",
      "Spesifikasi": "ORIGINAL",
      "Foto_Produk": "OP",
      "Kategori": "K07",
      "Merek": "M07"
    }
  ]
}
```

## 8) Tampil Catridge

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Catridge

```
// Route untuk tampil produk per catridge
Route::get('/tampilcatridge', [Produk::class, 'tampilcatridge']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk catridge
function tampilCatridge()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Catridge')
    ->get();

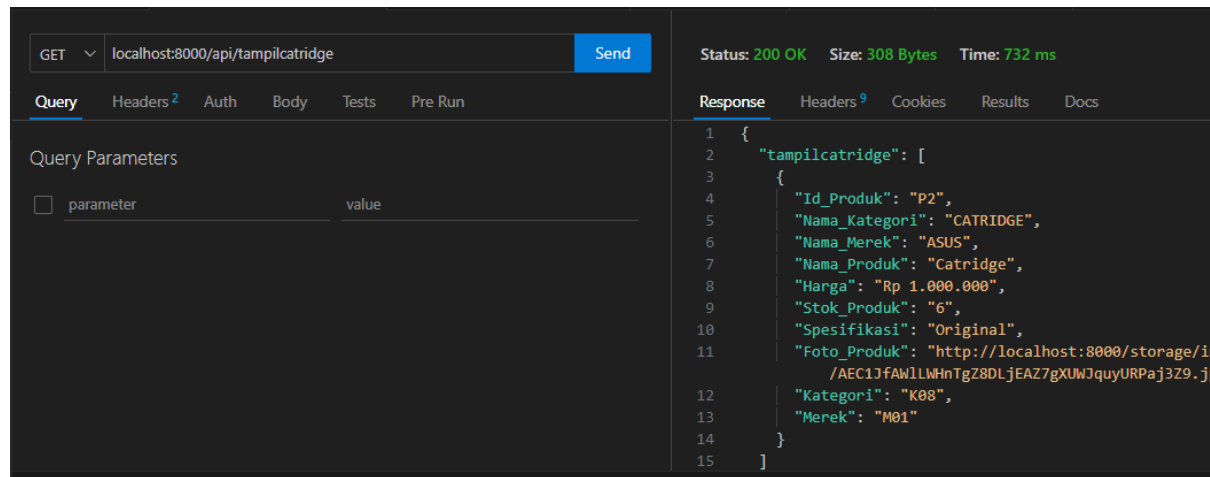
    return $query;
}
```

- Setelah selesai buat model untuk tampil Catridge selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori catridge
function tampilcatridge()
{
    // ambil function tampilCatridge dari MProduk
    $data = $this->model->tampilCatridge();

    // Tampilkan hasil produk per kategori Catridge
    return response([
        "tampilcatridge" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Catridge



## 9) Tampil Hardisk

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Hardisk

```
// Route untuk tampil produk per hardisk
Route::get('/tampilhardisk', [Produk::class, 'tampilhardisk']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk hardisk
function tampilHardisk()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Hardisk')
    ->get();

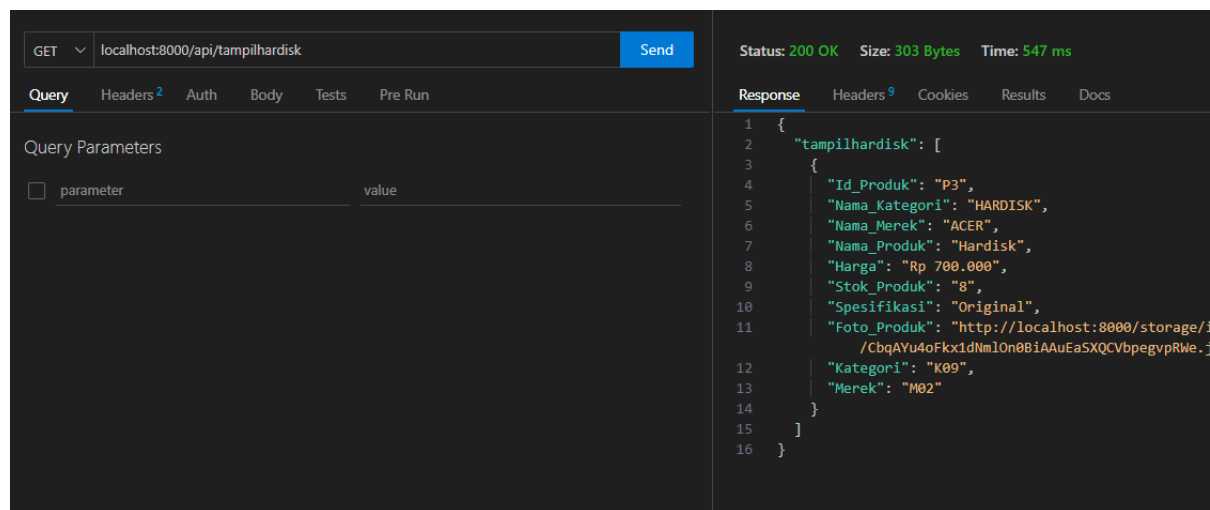
    return $query;
}
```

- Setelah selesai buat model untuk tampil Hardisk selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori hardisk
function tampilhardisk()
{
    // ambil function tampilHardisk dari MProduk
    $data = $this->model->tampilHardisk();

    // Tampilkan hasil produk per kategori Hardisk
    return response([
        "tampilhardisk" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Hardisk



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilhardisk
- Status:** 200 OK
- Size:** 303 Bytes
- Time:** 547 ms
- Response Body:**

```
{
  "tampilhardisk": [
    {
      "Id_Produk": "P3",
      "Nama_Kategori": "HARDISK",
      "Nama_Merek": "ACER",
      "Nama_Produk": "Hardisk",
      "Harga": "Rp 700.000",
      "Stok_Produk": "8",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/
/CbqAYu4oFkx1dNm1on0BiAAuEaSXQCVbpegvPRWe.
",
      "Kategori": "K09",
      "Merek": "M02"
    }
  ]
}
```

## 10) Tampil Flashdisk

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Flashdisk

```
// Route untuk tampil produk per flashdisk
Route::get('/tampilflashdisk', [Produk::class, 'tampilflashdisk']);
// Route untuk tampil produk per kategori flashdisk
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk flashdisk
function tampilFlashdisk()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Flashdisk')
    ->get();

    return $query;
}
```

- Setelah selesai buat model untuk tampil Flashdisk selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori flashdisk
function tampilflashdisk()
{
    // ambil function tampilFlashdisk dari MProduk
    $data = $this->model->tampilFlashdisk();

    // Tampilkan hasil produk per kategori Flashdisk
    return response([
        "tampilflashdisk" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Hardisk

Query	Headers	Auth	Body	Tests	Pre Run	Status	Size	Time
GET localhost:8000/api/tampilflashdisk						200 OK	317 Bytes	511 ms
<b>Query Parameters</b> <input type="checkbox"/> parameter value						<b>Response</b> <pre> 1 { 2   "tampilflashdisk": [ 3     { 4       "Id_Produk": "P4", 5       "Nama_Kategori": "FLASHDISK", 6       "Nama_Merek": "ACER", 7       "Nama_Produk": "Flashdisk Sandisk", 8       "Harga": "Rp 80.000", 9       "Stok_Produk": "10", 10      "Spesifikasi": "Original", 11      "Foto_Produk": "http://localhost:8000/storage/images/zALmRPDy7L08QewFD99JjnowuVag4McK4oxREepn.jpg", 12      "Kategori": "K10", 13      "Merek": "M02" 14    } 15  ] 16 }</pre>		

## 11) Tampil RAM

- Pertama buat route untuk api tampil per kategori disini saya akan menampilkan RAM

```
// Route untuk tampil produk per ram
Route::get('/tampilram', [Produk::class, 'tampilram']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk ram
function tampilRAM()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'RAM')
    ->get();

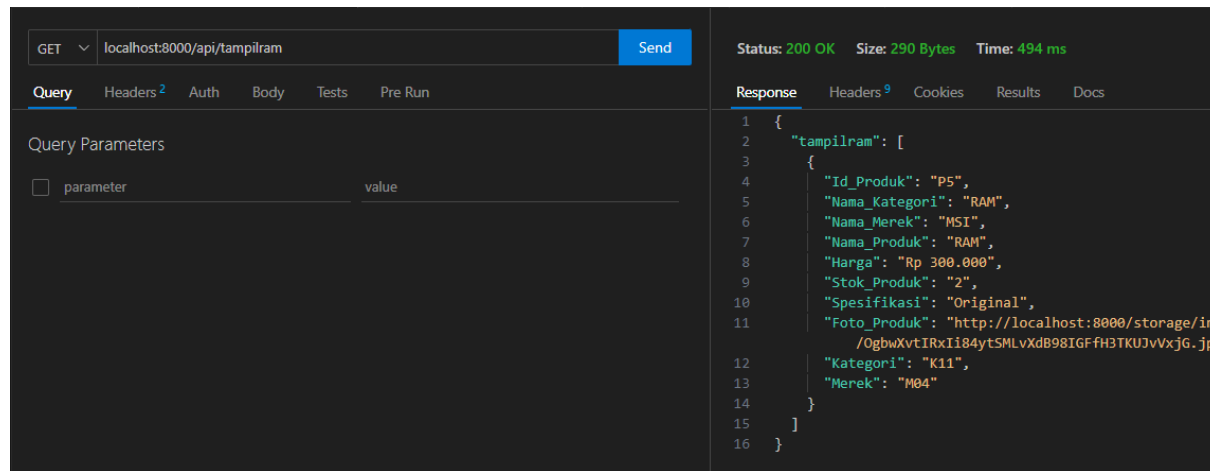
    return $query;
}
```

- Setelah selesai buat model untuk tampil RAM selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori RAM
function tampilram()
{
    // ambil function tampilRAM dari MProduk
    $data = $this->model->tampilRAM();

    // Tampilkan hasil produk per kategori RAM
    return response([
        "tampilram" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori RAM



## 12) Tampil PowerSupply

- Pertama buat route untuk api tampil per kategori disini saya akan menampilkan PowerSupply

```

// Route untuk tampil produk per powersupply
Route::get('/tampilpowersupply', [Produk::class, 'tampilpowersupply']);

```

- Selanjutnya buat model untuk tampil di MProduk()

```

// tampil produk powersupply
function tampilPowersupply()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Powersupply')
    ->get();

    return $query;
}

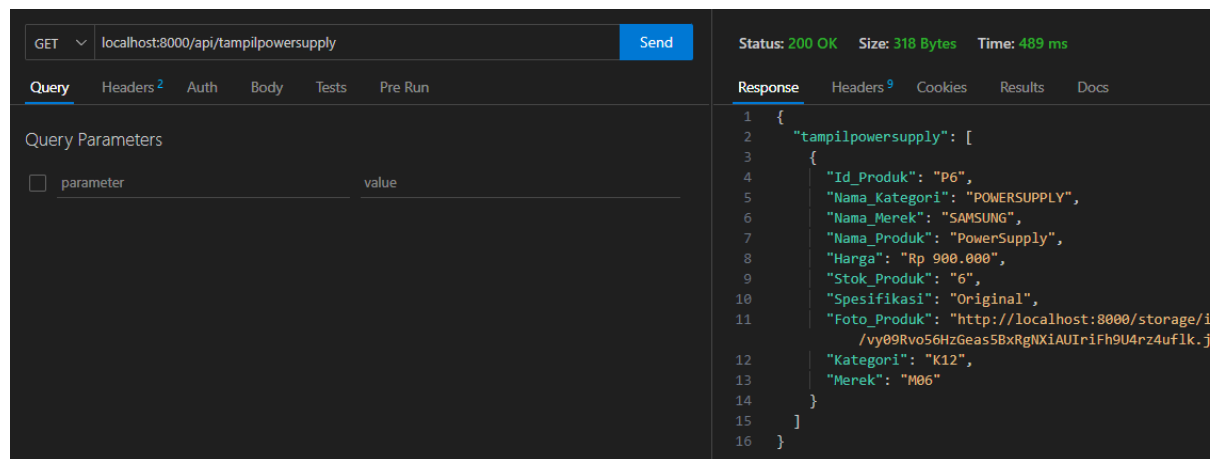
```

- Setelah selesai buat model untuk tampil PowerSupply selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori powersupply
function tampilpowersupply()
{
    // ambil function tampilPowersupply dari MProduk
    $data = $this->model->tampilPowersupply();

    // Tampilkan hasil produk per kategori Powersupply
    return response([
        "tampilpowersupply" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori PowerSupply



The screenshot shows a REST client interface with the following details:

- Query:** GET localhost:8000/api/tampilpowersupply
- Status:** 200 OK
- Size:** 318 Bytes
- Time:** 489 ms
- Response:**

```
1 {
2   "tampilpowersupply": [
3     {
4       "Id_Produk": "P6",
5       "Nama_Kategori": "POWERSUPPLY",
6       "Nama_Merek": "SAMSUNG",
7       "Nama_Produk": "PowerSupply",
8       "Harga": "Rp 900.000",
9       "Stok_Produk": "6",
10      "Spesifikasi": "Original",
11      "Foto_Produk": "http://localhost:8000/storage/1/vy09Rvo56HzGeas5BxRgNXiAUirFh9U4rz4uf1k.j
12      "Kategori": "K12",
13      "Merek": "M06"
14    }
15  ]
16 }
```

### 13) Tampil Motherboard

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Motherboard

```
// Route untuk tampil produk per motherboard
Route::get('/tampilmotherboard', [Produk::class, 'tampilmotherboard']);
```



- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk motherboard
function tampilMotherboard()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Motherboard')
    ->get();

    return $query;
}
```

- Setelah selesai buat model untuk tampil Motherboard selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori motherboard
function tampilmotherboard()
{
    // ambil function tampilMotherboard dari MProduk
    $data = $this->model->tampilMotherboard();

    // Tampilkan hasil produk per kategori Motherboard
    return response([
        "tampilmotherboard" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Motherboard

GET localhost:8000/api/tampilmotherboard Send

Status: 200 OK Size: 317 Bytes Time: 524 ms

Query Headers<sup>2</sup> Auth Body Tests Pre Run

Query Parameters

parameter	value
-----------	-------

Response Headers<sup>9</sup> Cookies Results Docs

```
1 {
2   "tampilmotherboard": [
3     {
4       "Id_Produk": "P7",
5       "Nama_Kategori": "MOTHERBOARD",
6       "Nama_Merek": "ASUS",
7       "Nama_Produk": "Motherboard",
8       "Harga": "Rp 1.400.000",
9       "Stok_Produk": "2",
10      "Spesifikasi": "Original",
11      "Foto_Produk": "http://localhost:8000/storage/images/7trJUVxxZMnds1M3gTsfpHfHaww61q70SPRUWfVS.jpg",
12      "Kategori": "K13",
13      "Merek": "M01"
14    }
15  ]
16 }
```

#### 14) Tampil Processor

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Processor

```
// Route untuk tampil produk per processor
Route::get('/tampilprocessor', [Produk::class, 'tampilprocessor']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk processor
function tampilProcessor()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Processor')
    ->get();

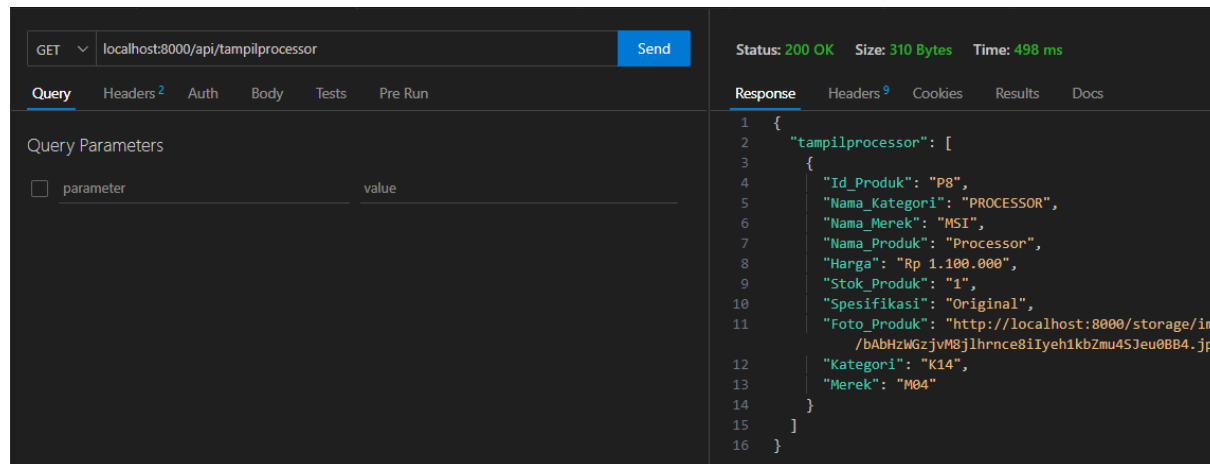
    return $query;
}
```

- Setelah selesai buat model untuk tampil Processor selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori processor
function tampilprocessor()
{
    // ambil function tampilProcessor dari MProduk
    $data = $this->model->tampilProcessor();

    // Tampilkan hasil produk per kategori Processor
    return response([
        "tampilprocessor" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Processor



## 15) Tampil Cooler

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Cooler

```
// Route untuk tampil produk per cooler
Route::get('/tampilcooler', [Produk::class, 'tampilcooler']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk cooler
function tampilCooler()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Cooler')
    ->get();

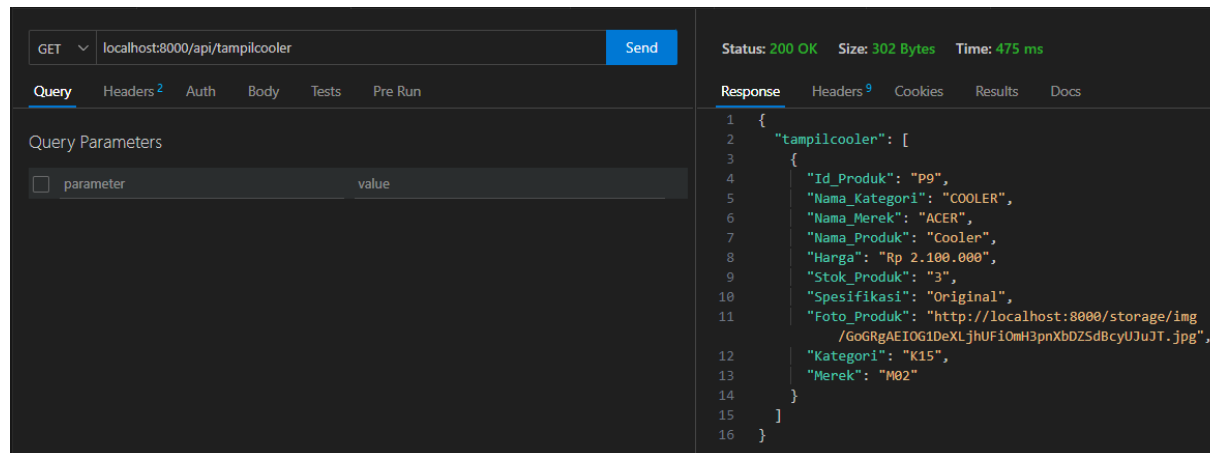
    return $query;
}
```

- Setelah selesai buat model untuk tampil Cooler selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori cooler
function tampilcooler()
{
    // ambil function tampilCooler dari MProduk
    $data = $this->model->tampilCooler();

    // Tampilkan hasil produk per kategori Cooler
    return response([
        "tampilcooler" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Cooler



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilcooler
- Status:** 200 OK
- Size:** 302 Bytes
- Time:** 475 ms
- Response Body:**

```
{
  "tampilcooler": [
    {
      "Id_Produk": "P9",
      "Nama_Kategori": "COOLER",
      "Nama_Merek": "ACER",
      "Nama_Produk": "Cooler",
      "Harga": "Rp 2.100.000",
      "Stok_Produk": "3",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/img/GoGRgAEIOG1DeXLjhUF1OmH3pnXbDZSd8cyU3u3T.jpg",
      "Kategori": "K15",
      "Merek": "M02"
    }
  ]
}
```

## 16) Tampil Casing PC

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Casing PC

```
// Route untuk tampil produk per casingpc
Route::get('/tampilcasingpc', [Produk::class, 'tampilcasingpc']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk casingpc
function tampilCasingPC()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'CasingPC')
    ->get();

    return $query;
}
```

- Setelah selesai buat model untuk tampil Casing PC selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori casingpc
function tampilcasingpc()
{
    // ambil function tampilCasingPC dari MProduk
    $data = $this->model->tampilCasingPC();

    // Tampilkan hasil produk per kategori CasingPC
    return response([
        "tampilcasingpc" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori CasingPC

GET

localhost:8000/api/tampilcasingpc

Send

Query

Headers<sup>2</sup>

Auth

Body

Tests

Pre Run

Query Parameters

☐ parameter

value

Status: 200 OK Size: 310 Bytes Time: 503 ms

Response

Headers<sup>9</sup>

Cookies

Results

Docs

```

1  {
2      "tampilcasingpc": [
3          {
4              "Id_Produk": "P10",
5              "Nama_Kategori": "CASINGPC",
6              "Nama_Merek": "LENOVO",
7              "Nama_Produk": "Casing PC",
8              "Harga": "Rp 600.000",
9              "Stok_Produk": "7",
10             "Spesifikasi": "Original",
11             "Foto_Produk": "http://localhost:8000/storage/07tGrBmUXF7r1Ew0UJeDowpkGsaXmQHudIPjv1d.j
12             "Kategori": "K16",
13             "Merek": "M08"
14         }
15     ]
16 }
```

## 17) Tampil NetworkAdapter

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan NetworkAdapter

```
// Route untuk tampil produk per networkadapter
Route::get('/tampilnetworkadapter', [Produk::class, 'tampilnetworkadapter']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk networkadapter
function tampilNetworkAdapter()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'NetworkAdapter')
    ->get();

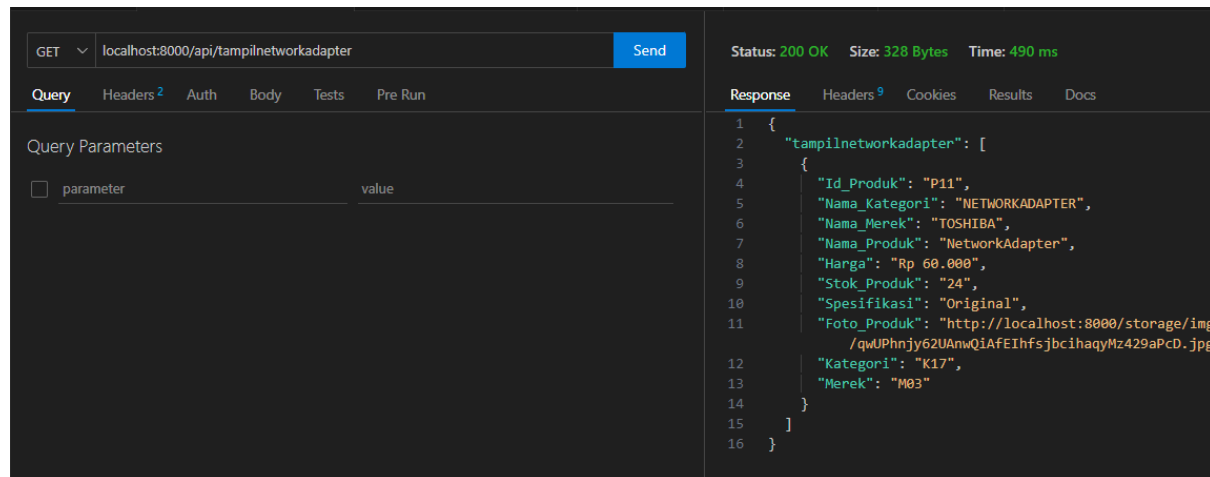
    return $query;
}
```

- Setelah selesai buat model untuk tampil NetworkAdapter selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori networkadapter
function tampilnetworkadapter()
{
    // ambil function tampilNetworkAdapter dari MProduk
    $data = $this->model->tampilNetworkAdapter();

    // Tampilkan hasil produk per kategori NetworkAdapter
    return response([
        "tampilnetworkadapter" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori NetworkAdapter



## 18) Tampil Webcam

- Pertama buat route untuk api tampil perkategori disini saya akan menampilkan Webcam

```
// Route untuk tampil produk per webcam
Route::get('/tampilwebcam', [Produk::class, 'tampilwebcam']);
```

- Selanjutnya buat model untuk tampil di MProduk()

```
// tampil produk webcam
function tampilWebcam()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_kategori', 'tbl_kategori.Id_Kategori', '=', 'tbl_produk.Kategori')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_kategori>Nama_Kategori', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_kategori>Nama_Kategori', 'tbl_produk.Merek')
    ->where('Nama_Kategori', 'Webcam')
    ->get();

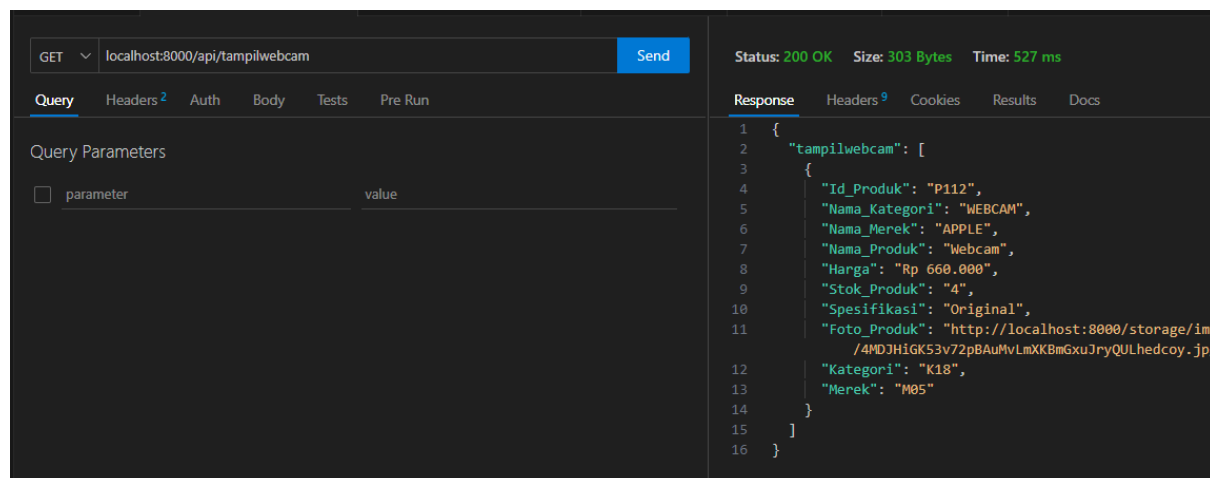
    return $query;
}
```

- Setelah selesai buat model untuk tampil Webcam selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per kategori webcam
function tampilwebcam()
{
    // ambil function tampilWebcam dari MProduk
    $data = $this->model->tampilWebcam();

    // Tampilkan hasil produk per kategori Webcam
    return response([
        "tampilwebcam" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per kategori Webcam



The screenshot shows a REST client interface with the following details:

- Query:** GET localhost:8000/api/tampilwebcam
- Status:** 200 OK
- Size:** 303 Bytes
- Time:** 527 ms
- Response:**

```
1 {
2   "tampilwebcam": [
3     {
4       "Id_Produk": "P112",
5       "Nama_Kategori": "WEBCAM",
6       "Nama_Merek": "APPLE",
7       "Nama_Produk": "Webcam",
8       "Harga": "Rp 660.000",
9       "Stok_Produk": "4",
10      "Spesifikasi": "Original",
11      "Foto_Produk": "http://localhost:8000/storage/images/4MDJH1GK53v72pBAuMvLmXKBmGxuJryQULhedcoy.jpg",
12      "Kategori": "K18",
13      "Merek": "M05"
14    }
15  ]
16 }
```



## 5. Buat Api Tampil per Merek

### 1) Tampil Asus

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek Asus

```
// Route untuk tampil produk per asus
Route::get('/tampilasus', [Produk::class, 'tampilasus']);
```

- Selanjutnya buat model untuk tampil merek Asus di MProduk()

```
// tampil produk merek ASUS
function tampilAsus()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_produk.Merek')
    ->where('Nama_Merek', 'Asus')
    ->get();

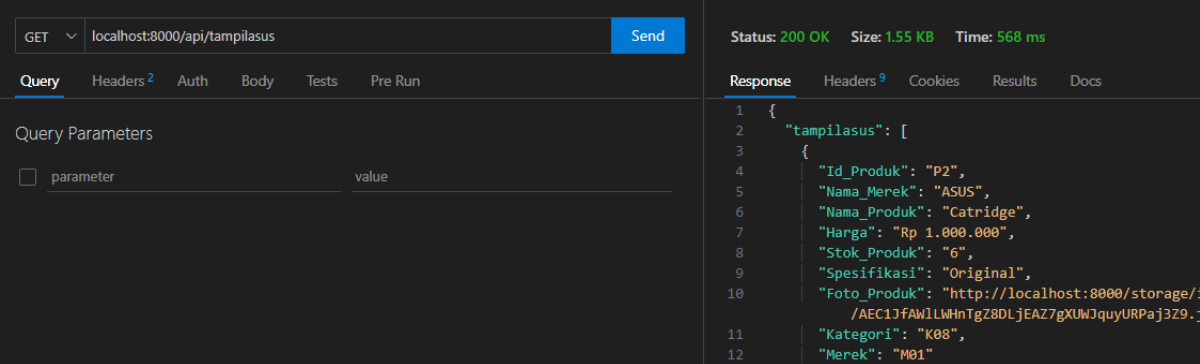
    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek Asus selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek asus
function tampilasus()
{
    // ambil function tampilAsus dari MProduk
    $data = $this->model->tampilAsus();

    // Tampilkan hasil produk per merek ASUS
    return response([
        "tampilasus" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek Asus



GET localhost:8000/api/tampilasus Send

Status: 200 OK Size: 1.55 KB Time: 568 ms

Query Parameters

parameter	value

Response

```

1 {
2   "tampilasus": [
3     {
4       "Id_Produk": "P2",
5       "Nama_Merek": "ASUS",
6       "Nama_Produk": "Catridge",
7       "Harga": "Rp 1.000.000",
8       "Stok_Produk": "6",
9       "Spesifikasi": "Original",
10      "Foto_Produk": "http://localhost:8000/storage/
11      /AEC1JfAWllWmTgZ8DLjEAZ7gXUNJquyURPaj3Z9.
12      "Kategori": "K08",
13      "Merek": "M01"
14    }
15  ]
16 }

```

## 2) Tampil Acer

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek Acer

```
// Route untuk tampil produk per acer
Route::get('/tampilacer', [Produk::class, 'tampilacer']);
```

- Selanjutnya buat model untuk tampil merek Acer di MProduk()

```
// tampil produk merek ACER
function tampilAcer()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_merek.Nama_Merek', 'tbl_produk.Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori'
    | 'tbl_produk.Merek')
    ->where('Nama_Merek', 'Acer')
    ->get();

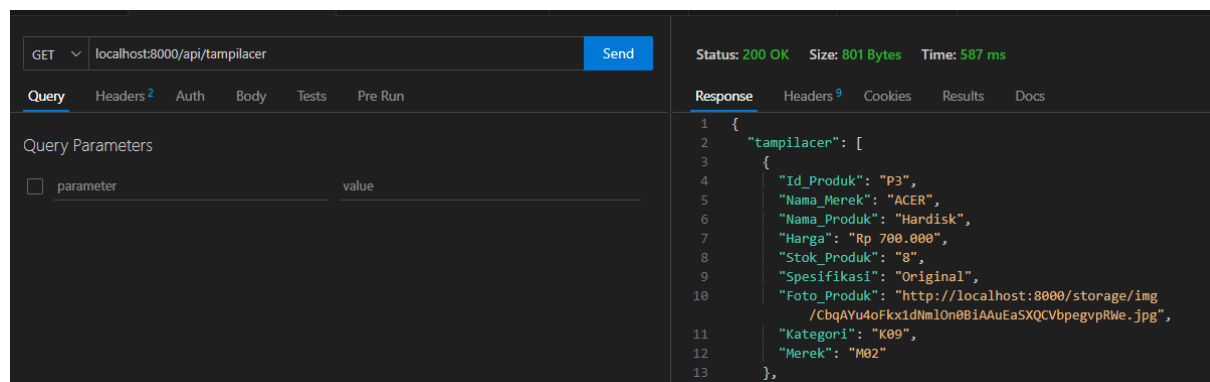
    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek Acer selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek acer
function tampilacer()
{
    // ambil function tampilAcer dari MProduk
    $data = $this->model->tampilAcer();

    // Tampilkan hasil produk per merek ACER
    return response([
        "tampilacer" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek Acer



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilacer
- Status:** 200 OK
- Size:** 801 Bytes
- Time:** 587 ms
- Response Body:**

```
{
  "tampilacer": [
    {
      "Id_Produk": "P3",
      "Nama_Merek": "ACER",
      "Nama_Produk": "Hardisk",
      "Harga": "Rp 700.000",
      "Stok_Produk": "8",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/img/CbqAYu4oFkxIdNm1On0BiAAuEaSXQCVbpegvpRWe.jpg",
      "Kategori": "K09",
      "Merek": "M02"
    }
  ]
}
```

### 3) Tampil Toshiba

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek Toshiba

```
// Route untuk tampil produk per toshiba
Route::get('/tampiltohiba', [Produk::class, 'tampiltoshiba']);
```

- Selanjutnya buat model untuk tampil merek Toshiba di MProduk()

```
// tampil produk merek TOSHIBA
function tampilToshiba()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_produk.Merek')
    ->where('Nama_Merek', 'Toshiba')
    ->get();

    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek Toshiba selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek toshiba
function tampilToshiba()
{
    // ambil function tampilToshiba dari MProduk
    $data = $this->model->tampilToshiba();

    // Tampilkan hasil produk per merek Toshiba
    return response([
        "tampilToshiba" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek Toshiba

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilToshiba
- Status:** 200 OK
- Size:** 288 Bytes
- Time:** 699 ms
- Response:** A JSON array containing one object representing a Toshiba product.

```
{
  "tampilToshiba": [
    {
      "Id_Produk": "P11",
      "Nama_Merek": "TOSHIBA",
      "Nama_Produk": "NetworkAdapter",
      "Harga": "Rp 60.000",
      "Stok_Produk": "24",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/1/qwUPhnjy62UAnwQiAfEIhfsjbcihagyMz429aPcD.j",
      "Kategori": "K17",
      "Merek": "M03"
    }
  ]
}
```

#### 4) Tampil MSI

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek MSI

```
// Route untuk tampil produk per msi
Route::get('/tampilmsi', [Produk::class, 'tampilmsi']);
// Route untuk tampil produk per merk
```

- Selanjutnya buat model untuk tampil merek MSI di MProduk()

```
// tampil produk merek MSI
function tampilMSI()
{
    $query = DB::table('tbl_produk')
        ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
        ->select('tbl_produk.Id_Produk', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
            'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
            'tbl_produk.Merek')
        ->where('Nama_Merek', 'MSI')
        ->get();

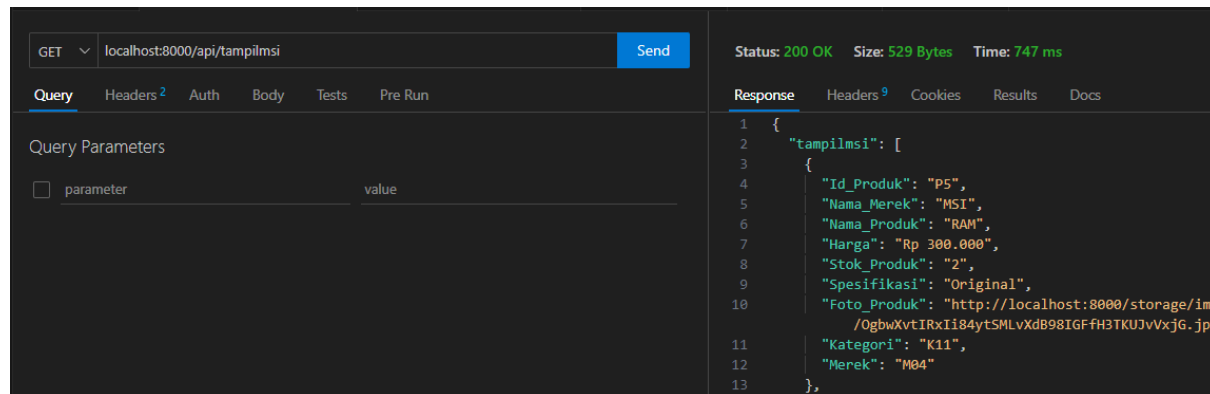
    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek MSI selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek MSI
function tampilmsi()
{
    // ambil function tampilMSI dari MProduk
    $data = $this->model->tampilMSI();

    // Tampilkan hasil produk per merek MSI
    return response([
        "tampilmsi" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek MSI



## 5) Tampil Apple

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek Apple

```

// Route untuk tampil produk per apple
Route::get('/tampilapple', [Produk::class, 'tampilapple']);

```

- Selanjutnya buat model untuk tampil merek Apple di MProduk()

```

// tampil produk merek apple
function tampilApple()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_merek.Nama_Merek', 'tbl_produk.Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori'
    | 'tbl_produk.Merek')
    ->where('Nama_Merek', 'Apple')
    ->get();

    return $query;
}

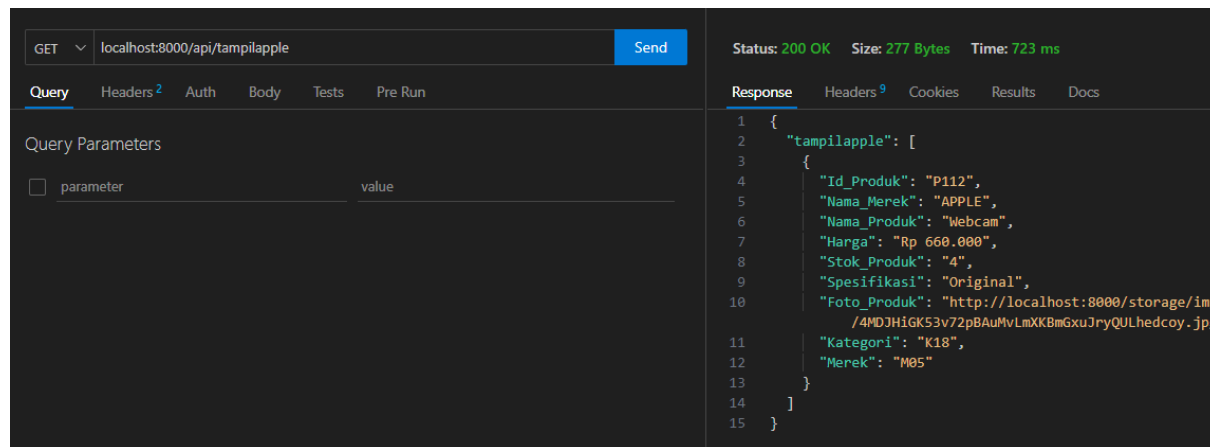
```

- Setelah selesai buat model untuk tampil Merek Apple selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek apple
function tampilapple()
{
    // ambil function tampilApple dari MProduk
    $data = $this->model->tampilApple();

    // Tampilkan hasil produk per merek Apple
    return response([
        "tampilapple" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek Apple



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilapple
- Status:** 200 OK
- Size:** 277 Bytes
- Time:** 723 ms
- Response Body:**

```
{
  "tampilapple": [
    {
      "Id_Produk": "P112",
      "Nama_Merek": "APPLE",
      "Nama_Produk": "Webcam",
      "Harga": "Rp 660.000",
      "Stok_Produk": "4",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/images/4MDJHIGK53v72pBAuMvLmXKBmGxuJryQULhedcoy.jpg",
      "Kategori": "K18",
      "Merek": "M05"
    }
  ]
}
```

## 6) Tampil Samsung

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek Samsung

```
// Route untuk tampil produk per samsung
Route::get('/tampilsamsung', [Produk::class, 'tampilsamsung']);
```

- Selanjutnya buat model untuk tampil merek Samsung di MProduk()

```
// tampil produk merek samsung
function tampilSamsung()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk','tbl_merek>Nama_Merek','tbl_produk>Nama_Produk','tbl_produk.Harga',
    'tbl_produk.Stok_Produk','tbl_produk.Spesifikasi','tbl_produk.Foto_Produk','tbl_produk.Kategori',
    'tbl_produk.Merek')
    ->where('Nama_Merek', 'Samsung')
    ->get();

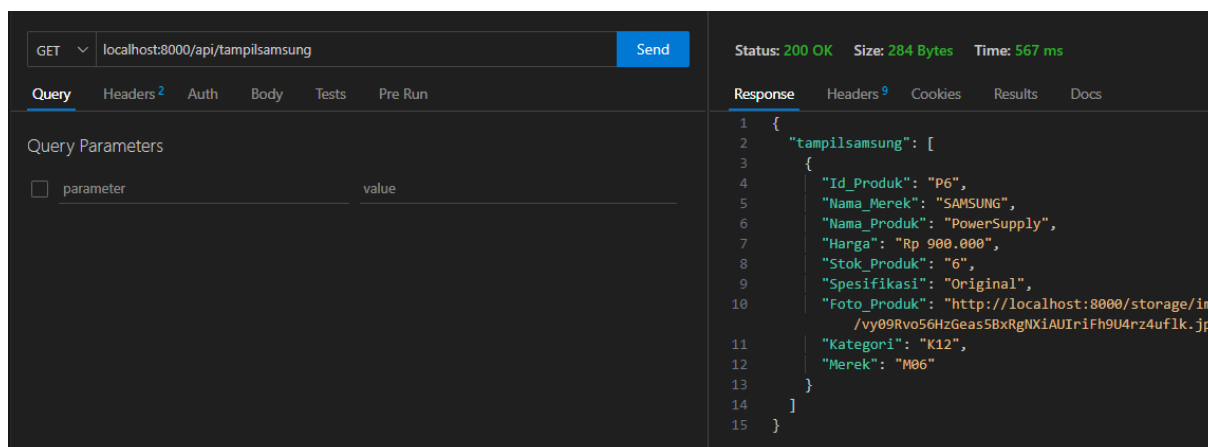
    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek Samsung selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek samsung
function tampilsamsung()
{
    // ambil function tampilSamsung dari MProduk
    $data = $this->model->tampilSamsung();

    // Tampilkan hasil produk per merek Samsung
    return response([
        "tampilsamsung" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek samsung



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilsamsung
- Status:** 200 OK
- Size:** 284 Bytes
- Time:** 567 ms
- Response Body:**

```
{
  "tampilsamsung": [
    {
      "Id_Produk": "P6",
      "Nama_Merek": "SAMSUNG",
      "Nama_Produk": "PowerSupply",
      "Harga": "Rp 900.000",
      "Stok_Produk": "6",
      "Spesifikasi": "Original",
      "Foto_Produk": "http://localhost:8000/storage/in
/vy09Rvo56HzGeas5BxRgNXiAUiriFh9U4rz4uf1k.jp
",
      "Kategori": "K12",
      "Merek": "M06"
    }
  ]
}
```



## 7) Tampil HP

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek HP

```
// Route untuk tampil produk per hp
Route::get('/tampilhp', [Produk::class, 'tampilhp']);
```

- Selanjutnya buat model untuk tampil merek HP di MProduk()

```
// tampil produk merek hp
function tampilHP()
{
    $query = DB::table('tbl_produk')
->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
->select('tbl_produk.Id_Produk', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
        'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
        'tbl_produk.Merek')
->where('Nama_Merek', 'HP')
->get();

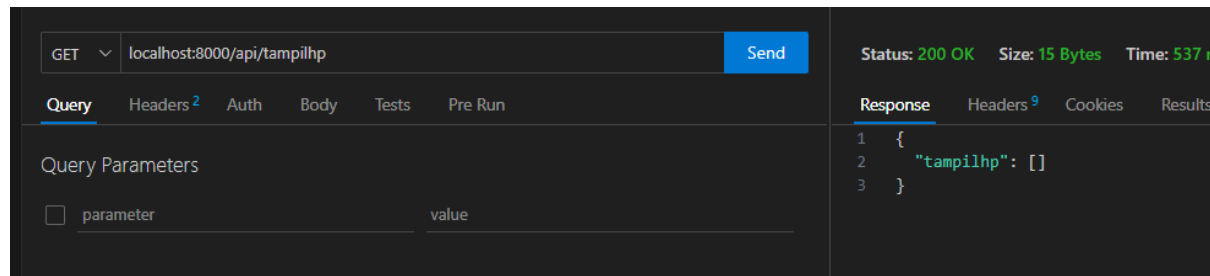
    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek HP selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek HP
function tampilhp()
{
    // ambil function tampilHP dari MProduk
    $data = $this->model->tampilHP();

    // Tampilkan hasil produk per merek HP
    return response([
        "tampilhp" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek HP



## 8) Tampil Lenovo

- Pertama buat route untuk api tampil permerek disini saya akan menampilkan Merek HP

```
// Route untuk tampil produk per lenovo
Route::get('/tampillenovo', [Produk::class, 'tampillenovo']);
```

- Selanjutnya buat model untuk tampil merek Lenovo di MProduk()

```
// tampil produk merek lenovo
function tampillenovo()
{
    $query = DB::table('tbl_produk')
    ->join('tbl_merek', 'tbl_merek.Id_Merek', '=', 'tbl_produk.Merek')
    ->select('tbl_produk.Id_Produk', 'tbl_merek>Nama_Merek', 'tbl_produk>Nama_Produk', 'tbl_produk.Harga',
    'tbl_produk.Stok_Produk', 'tbl_produk.Spesifikasi', 'tbl_produk.Foto_Produk', 'tbl_produk.Kategori',
    'tbl_produk.Merek')
    ->where('Nama_Merek', 'Lenovo')
    ->get();

    return $query;
}
```

- Setelah selesai buat model untuk tampil Merek Lenovo selanjutnya kita panggil di controller.

```
// Function untuk tampil produk per merek Lenovo
function tampilLenovo()
{
    // ambil function tampilLenovo dari MProduk
    $data = $this->model->tampilLenovo();

    // Tampilkan hasil produk per merek Lenovo
    return response([
        "tampilLenovo" => $data
    ], http_response_code());
}
```

- Testing api untuk menampilkan produk per merek Lenovo

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/tampilLenovo
- Status:** 200 OK
- Size:** 281 Bytes
- Time:** 585 ms
- Response:**

```
1 {
2   "tampilLenovo": [
3     {
4       "Id_Produk": "P10",
5       "Nama_Merek": "LENOVO",
6       "Nama_Produk": "Casing PC",
7       "Harga": "Rp 600.000",
8       "Stok_Produk": "7",
9       "Spesifikasi": "Original",
10      "Foto_Produk": "http://localhost:8000/storage/img
11                        /07tGr8mUXF7r1Ew0UJeDowpkGsaXmQHiudIPjvld.jpg",
12      "Kategori": "K16",
13      "Merek": "M08"
14    }
15  ]
16 }
```

## 6. Buat Api Pencarian

- Pertama buat route untuk api Pencarian

```
// Route untuk PENCARIAN
Route::get('/search/{Nama_Produk}', [Produk::class, 'search']);

Route::get('/search1/{Nama_Produk}', [Produk::class, 'search1']);
```

- Selanjutnya buat fungsi Pencarian di controller Produk()

```
// buat fungsi search
public function search($Nama_Produk)
{
    return MProduk::where("Nama_Produk", "like", "%".$Nama_Produk."%")->get();
}

public function search1($Nama_Produk)
{
    $results = MProduk::where('Nama_Produk', 'LIKE', '%'.$Nama_Produk.'%')->get();

    return response([
        "cari" => $results
    ], 200);
}
```

- Testing api untuk Pencarian

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:8000/api/search/Processor
- Status:** 200 OK
- Size:** 243 Bytes
- Time:** 478 ms
- Response:** A JSON array containing one object with the following fields:
  - `"Id_Produk": "P8",`
  - `"Nama_Produk": "Processor",`
  - `"Harga": "Rp 1.100.000",`
  - `"Stok_Produk": "1",`
  - `"Spesifikasi": "Original",`
  - `"Foto_Produk": "http://localhost:8000/storage/images/bAbHzWGzjvM8jIhrnce8iIyeh1kbZmu45Jeu0BB4.jpg",`
  - `"Kategori": "K14",`
  - `"Merek": "M04"`

## 7. Buat Api Upload Foto

- Pertama buat route api untuk Upload Foto

```
// Route Untuk upload foto
Route::post('/uploadgambar', [Produk::class, 'uploadgambar']);
```

- Selanjutnya buat fungsi upload foto di controller Produk()

```
// buat function untuk insert data
function uploadgambar(Request $req)
{
    $req->validate([
        'Foto_Produk' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ]);

    // ambil data hasil input
    $data = array(
        "Id_Produk" => $req->Id_Produk,
        "Nama_Produk" => $req->Nama_Produk,
        "Harga" => $req->Harga,
        "Stok_Produk" => $req->Stok_Produk,
        "Spesifikasi" => $req->Spesifikasi,
        "Foto_Produk" => $req->Foto_Produk,
        "Kategori" => $req->Kategori,
        "Merek" => $req->Merek,
    );

    // $imagepath = Storage::put('public/storage/img', $req->file("Foto_Produk"));
    // $imagepath=Storage::put('local/images', $req->file('Foto_Produk'));
    $imagepath= $req->file('Foto_Produk')->store('public/img');
    $url = asset(Storage::url($imagepath));

    // baruu
    $parameter =($data["Id_Produk"]);
    // cek apakah data produk (Id_Produk) sudah pernah tersimpan/belum
    $check = $this->model->detailData($parameter);
```

```

// baruu
$parameter = ($data["Id_Produk"]);
// cek apakah data produk (Id_Produk) sudah pernah tersimpan/belum
$check = $this->model->detailData($parameter);

// jika data tidak ditemukan
if (count($check) == 0) {
    // lakukan proses penyimpanan
    $this->model->saveData([$data["Id_Produk"], $data["Nama_Produk"], $data["Harga"], $data["Stok_Produk"],
    | $data["Spesifikasi"], $url,$data
    | ["Kategori"], $data["Merek"]]);
    // buat pesan dan status hasil penyimpanan data
    $status = 1;
    $pesan =
    | "Data Berhasil disimpan";
}
// jika data tidak ditemukan
else {

    // tampilkan pesan data gagal disimpan
    $status = 0;
    $pesan = "Data Gagal disimpan";
}
// tampilkan hasil respon

return response([
    "status" => $status,
    "pesan" => $pesan
], http_response_code());

```

- Testing api untuk upload foto

POST

localhost:8000/api/uploadgambar

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

☒

Id\_Produk

P13

☒

Nama\_Produk

Laptop ASUS

☒

Harga

Rp 6.000.000

☒

Stok\_Produk

2

☒

Spesifikasi

Original

☒

Kategori

K01

☒

Merek

M01

☐

field name

value

Files

☒

Foto\_Produk

Choose File

kk.jpg

☐

field name

Choose File

Select file

Status: 200 OK

Size: 45 Bytes

Time: 736 ms

Response

Headers<sup>9</sup>

Cookies

Results

```

1 {
2   "status": 1,
3   "pesan": "Data Berhasil disimpan"
4 }

```

- Hasil Upload akan tampil di Database.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Id_Produk	Nama_Produk	Harga	Stok_Produk	Spesifikasi	Foto_Produk	Kategori
<input type="checkbox"/> P10	Casing PC	Rp 600.000	7	Original	http://localhost:8000/storage/img/07tGrBmUXF7r1Ew0UJeDowpkGsaXmQHiu	K16
<input type="checkbox"/> P11	NetworkAdapter	Rp 60.000	24	Original	http://localhost:8000/storage/img/qwUPhnjy62UAnwQ1AfeIhfsjbcihaqyMz	K17
<input type="checkbox"/> P112	Webcam	Rp 660.000	4	Original	http://localhost:8000/storage/img/4MDJHiGK53v72pBAuMvLnXKBmGxuJryQU	K18
<input type="checkbox"/> P13	Laptop ASUS	Rp 6.000.000	2	Original	http://localhost:8000/storage/img/baliaUHSRM21phabPBxYSRicZRi2AeMQ	K01
<input type="checkbox"/> P2	Catridge	Rp 1.000.000	6	Original	http://localhost:8000/storage/img/AEC1JfAW1LWHnTgZ8DLjEAZ7gXUWJquyU	K08
<input type="checkbox"/> P3	Hardisk	Rp 700.000	8	Original	http://localhost:8000/storage/img/CbqAYu4oFkxldNm1On0BiAAuEaSXQCvop	K09
<input type="checkbox"/> P34	Laptop	Rp 1.000.000	3	RAM 4gb	http://localhost:8000/public/storage/img/uId6KbiReeoTKoGeWBbE6YoGH	K01
<input type="checkbox"/> P35	Laptop	Rp 1.000.000	3	RAM 4gb	http://localhost:8000/storage/img/JmIZ7Yrm6uJKIageW0GdpUZAfJ7VdMKS3	K01
<input type="checkbox"/> P36	Laptop	Rp 1.000.000	3	RAM 4gb	http://localhost:8000/storage/img/WsqwIkjd7vCkzkWR4RGcFhCrEXAcB6AS	K02
<input type="checkbox"/> P39	Laptop	Rp 1.000.000	3	RAM 4gb	http://localhost:8000/storage/img/WmJF4P10xs03eeWz4ucBnATTVftrVbTk	K02