

## Dokumentasi :

Framework: Laravel

Libray: bootstrap5, datatable, sweetalert2

Get Api disini saya menggunakan curl yang saya taruh di folder service dengan nama api.php

Berikut Isi dari api.php

```
// Untuk cek password dan username saya menggunakan POSTMAN
$username = 'tesprogrammer090523C22';
$password = md5('bisacoding-09-05-23');
$post = [
    'username' => $username,
    'password' => $password,
];
// API tes programmer
$url = 'https://recruitment.fastprint.co.id/tes/api_tes_programmer';
$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_ANY);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);

$result = curl_exec($ch);
curl_close($ch);

return $result;
```

Berikut data berupa json

```
← → ↻ ⓘ 127.0.0.1:8000/produk-api

array:3 [▼
  "error" => 0
  "version" => "200722.2.1"
  "data" => array:30 [▼
    0 => array:6 [▼
      "no" => "7"
      "id_produk" => "6"
      "nama_produk" => "ALCOHOL GEL POLISH CLEANSER GP-CLN01"
      "kategori" => "L QUEENLY"
      "harga" => "12500"
      "status" => "bisa dijual"
    ]
    1 => array:6 [▼
      "no" => "10"
      "id_produk" => "9"
      "nama_produk" => "ALUMUNIMUM FOIL ALL IN ONE BULAT 23mm IM"
      "kategori" => "L MTH AKSESORIS (IM)"
      "harga" => "1000"
      "status" => "bisa dijual"
    ]
    2 => array:6 [▼
      "no" => "12"
      "id_produk" => "11"
      "nama_produk" => "ALUMUNIMUM FOIL ALL IN ONE BULAT 30mm IM"
      "kategori" => "L MTH AKSESORIS (IM)"
      "harga" => "1000"
      "status" => "bisa dijual"
    ]
  ]
]
```

setelah mendapatkan data dari api selanjutnya membuat migrasi database dengan nama tabel products yang berisi beberapa kolom berikut:

```
$table->increments('id_produk');
$table->string('nama_produk');
$table->double('harga');
$table->string('kategori');
$table->string('status');
$table->timestamps();
```

setelah itu jangan lupa untuk menjalankan dengan mengetikkan php artisan migration.

*Note : sebelum migration jangan lupa untuk mensetting database di file .env*

Jika sudah migration maka upload data api ke database mysql, saya menggunakan Xampp berikut function yang saya gunakan untuk upload api ke mysql :

```
public function uploadFromApi()
{
    $apiProduct = new Api;
    $getApiProduct = $apiProduct->product();

    $data = json_decode($getApiProduct, true);
    $banyakData = $data['data'];

    for ($i=0; $i < count($banyakData) ; $i++) {
```

```

        $idProduct = $data['data'][$i]['id_produk'];
        $namaProduct = $data['data'][$i]['nama_produk'];
        $hargaProduct = $data['data'][$i]['harga'];
        $kategoriProduct = $data['data'][$i]['kategori'];
        $statusProduct = $data['data'][$i]['status'];

        // upload data api ke tabel product
        Product::create([
            'id_produk' => $idProduct,
            'nama_produk' => $namaProduct,
            'harga' => $hargaProduct,
            'kategori' => $kategoriProduct,
            'status' => $statusProduct
        ]);
    }
}

```

Selanjutnya tampilkan data pada index yang mana hanya menampilkan status ‘bisa dijual’.

Berikut query yang saya gunakan untuk menampilkan status ‘bisa dijual’, saya juga menambahkan orderBy yang mana untuk fungsi descending agar saat menambahkan data baru, data baru akan tampil diatas.

```

$products = Product::where('status', 'bisa dijual')->orderBy('id_produk',
'desc')->get();

```

Ini adalah kode validasi yang digunakan agar nama produk harus di isi dan harga harus berupa angka pada controller:

```

$dataValidated = $request->validate([
    'nama_produk' => 'required',
    'harga' => 'numeric',
    'kategori' => 'nullable',
    'status' => 'nullable'
]);

```

Selanjutnya untuk bagian hapus saya menggunakan jquery ajax, kode function yang saya gunakan seperti berikut

```

function deleteData(event) {
    event.preventDefault();
    var id_produk =
event.target.querySelector('input[name="id_produk"]').value;

```

```

        var nama_produk =
event.target.querySelector('input[name="nama_produk"]').value;

        var confirmDelete = confirm('Apakah kamu yakin ingin menghapus produk
' + nama_produk + '?');

        if (confirmDelete == true) {
            var url = "{ route('produk.delete') }";
            var fd = new FormData($(event.target)[0]);
            $.ajax({
                type: "post",
                url: url,
                data: fd,
                contentType: false,
                processData: false,
                success: function(response) {
                    Swal.fire(
                        'Good job!',
                        response.data,
                        'success'
                    )
                    setTimeout(function() {
                        window.location.reload();
                    }, 1000);
                }
            });
        }
    }
}

```

## Tampilan Route

```

// route untuk upload data dari api ke database
Route::get('/produk-api', [ProductController::class, 'uploadFromApi']);

// route tampilan daftar product
Route::get('/', [ProductController::class, 'index'])->name('produk.daftar');
// route menampilkan tambah data produk
Route::get('/create', [ProductController::class, 'create'])->
>name('produk.create');
// route memasukkan data produk ke database
Route::post('/store', [ProductController::class, 'store'])->name('produk.store');
// route mengedit data produk
Route::post('/edit', [ProductController::class, 'edit'])->name('produk.edit');
// route menghapus data produk

```

```
Route::delete('/delete', [ProductController::class, 'destroy'])->name('produk.delete');
```

## Tampilan aplikasi yang saya buat

The screenshot displays two views of a web application. The top view is the 'Daftar Produk' (Product List) page, and the bottom view is the 'Tambah Produk' (Add Product) page. Both views are part of a system named 'MSubeqi'.

**Daftar Produk View:**

- Header:** 'Daftar Produk' title, 'Show 10 entries' dropdown, and a search bar.
- Table:** A table with 10 columns: No, Nama Produk, Harga, Kategori, Status, and Aksi. It lists 10 products with details like 'BOTOL KOTAK 100ML LK', 'BOTOL 1000ML ORANGE KHUSUS UNTUK EPSON R1900/R2000 IM - 4190 (T087920)', etc.
- Footer:** 'Showing 1 to 10 of 16 entries' and pagination controls (Previous, 1, 2, Next).

**Tambah Produk View:**

- Header:** 'Tambah Produk' title.
- Form:** A form with four input fields: 'Nama Produk' (with a note '\*nama produk harus di isi'), 'Harga Produk' (with a note '\*input harga hanya bisa angka'), 'Kategori Produk' (dropdown), and 'Status' (dropdown).
- Button:** A blue 'Simpan' (Save) button.