

Nama: Andhika Pratama  
NIM: 09021182328025  
Kelas: TI L3 23

## Pemrograman Web II - Progress Report 2

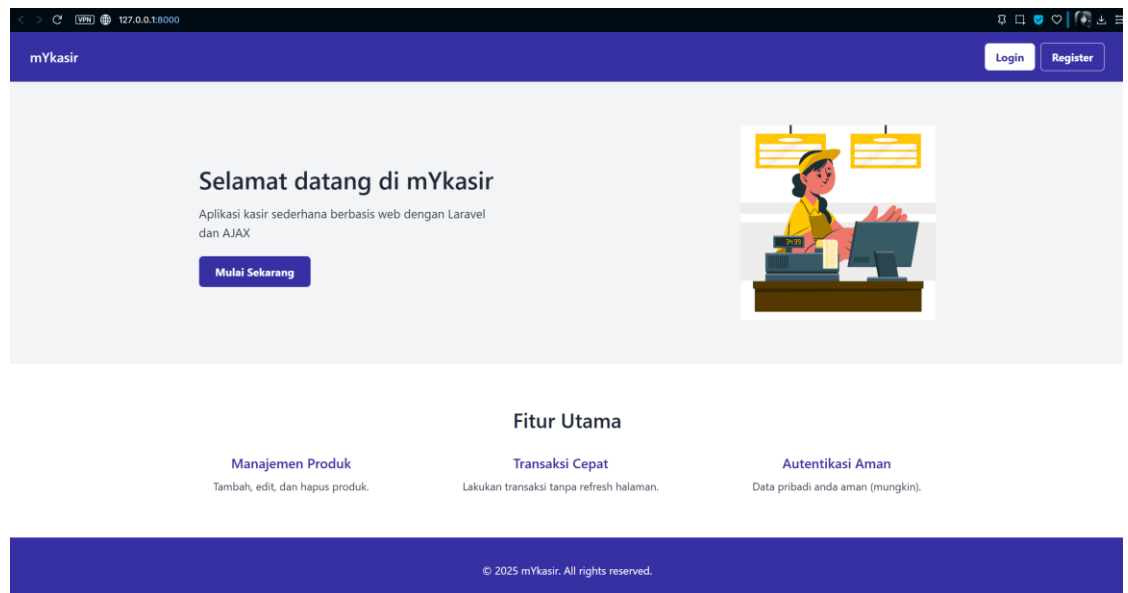
1. Implementasi project yang telah di lakukan (View/Blade, Javascript, CSS, Controller, atau Desain Model / DB)

note: untuk kodingannya, hanya saya ss sebagian saja pak/bu, selebihnya ada di:

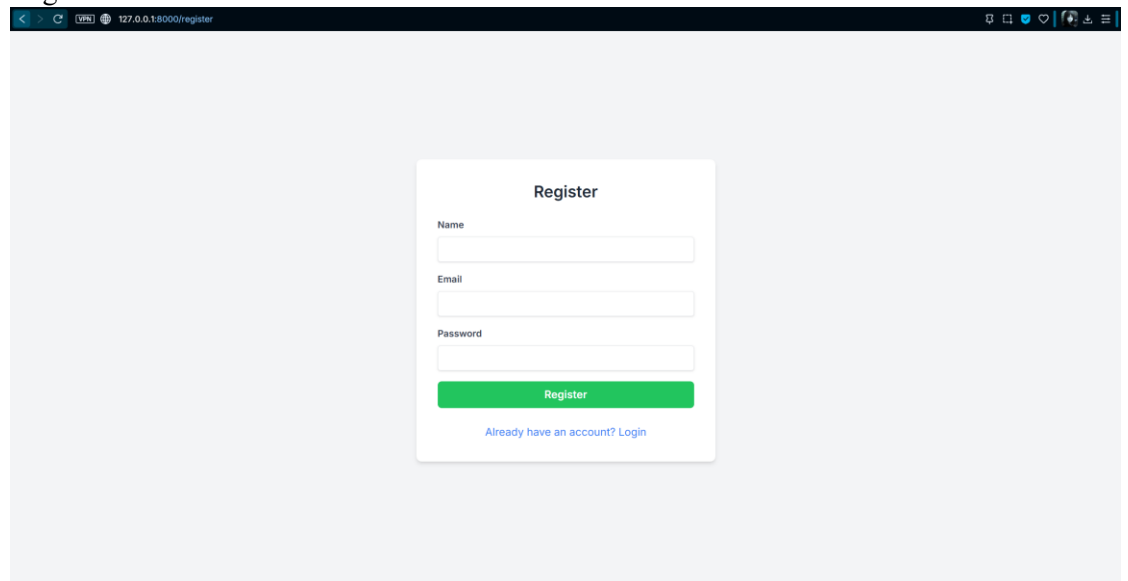
<https://github.com/aristelcrowley/mYkasir>

### a. Implementasi Project

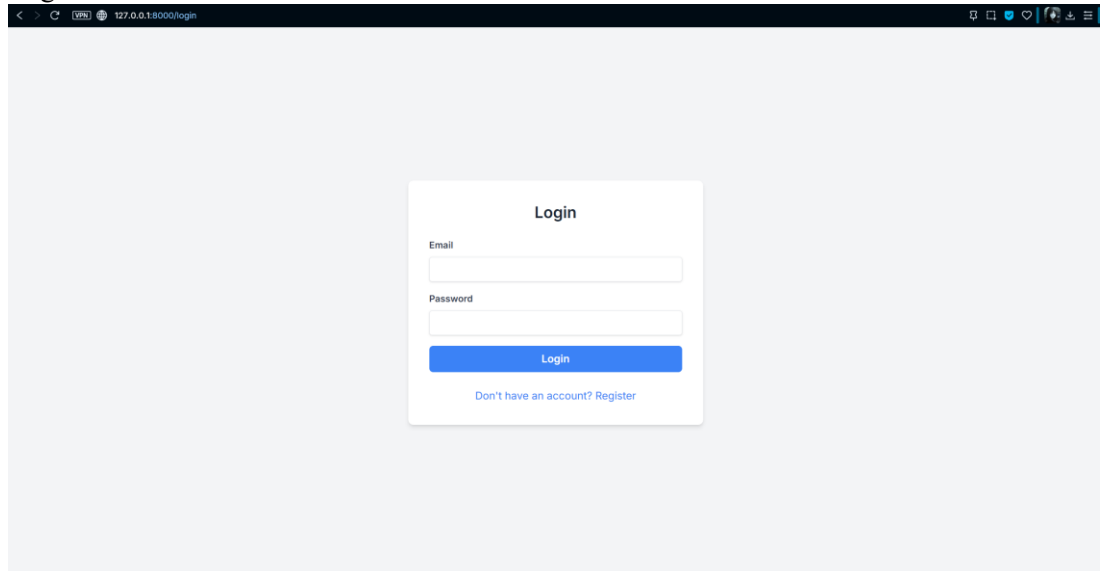
#### 1) Home



#### 2) Register



### 3) Login



127.0.0.1:8000/login

### Login

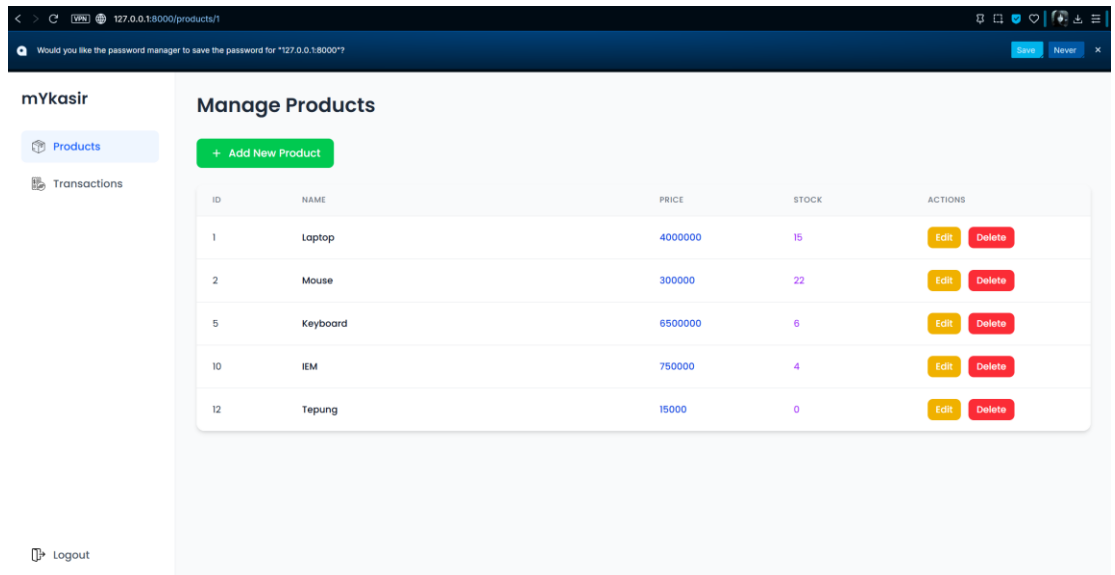
Email

Password

Login

[Don't have an account? Register](#)

### 4) Products



127.0.0.1:8000/products/1

Would you like the password manager to save the password for "127.0.0.1:8000"? [Save](#) [Never](#)

mYkasir

- Products
- Transactions

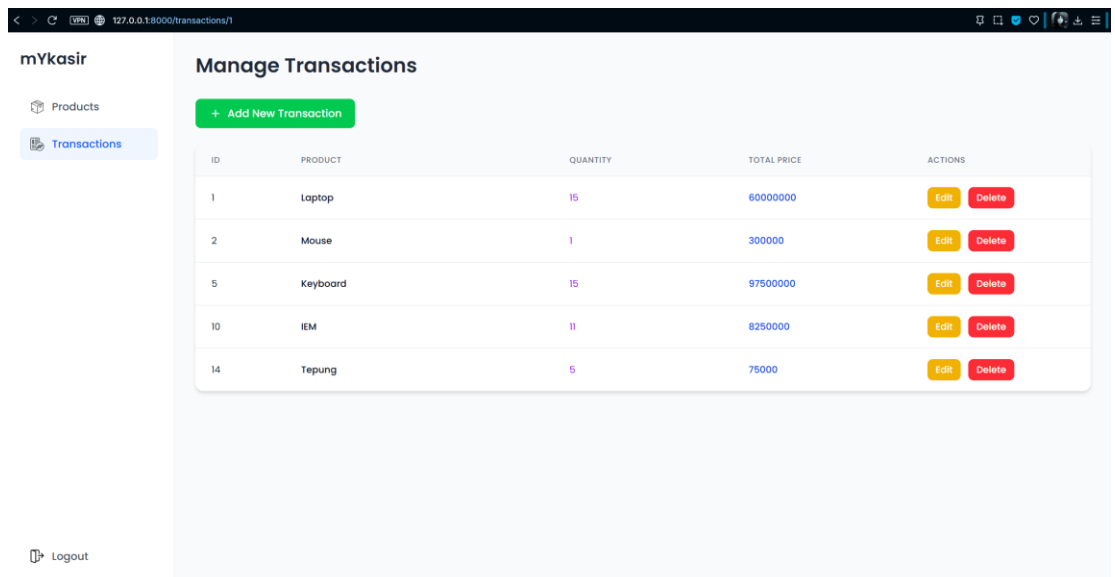
Logout

### Manage Products

+ Add New Product

ID	NAME	PRICE	STOCK	ACTIONS
1	Laptop	4000000	15	<a href="#">Edit</a> <a href="#">Delete</a>
2	Mouse	300000	22	<a href="#">Edit</a> <a href="#">Delete</a>
5	Keyboard	6500000	6	<a href="#">Edit</a> <a href="#">Delete</a>
10	IEM	750000	4	<a href="#">Edit</a> <a href="#">Delete</a>
12	Tepung	15000	0	<a href="#">Edit</a> <a href="#">Delete</a>

### 5) Transactions



127.0.0.1:8000/transactions/1

mYkasir

- Products
- Transactions

Logout

### Manage Transactions

+ Add New Transaction

ID	PRODUCT	QUANTITY	TOTAL PRICE	ACTIONS
1	Laptop	15	60000000	<a href="#">Edit</a> <a href="#">Delete</a>
2	Mouse	1	300000	<a href="#">Edit</a> <a href="#">Delete</a>
5	Keyboard	15	97500000	<a href="#">Edit</a> <a href="#">Delete</a>
10	IEM	11	8250000	<a href="#">Edit</a> <a href="#">Delete</a>
14	Tepung	5	75000	<a href="#">Edit</a> <a href="#">Delete</a>

b. Views/Blade.php (home) (tidak pakai css terpisah, tailwind)

```
resources > views > home.blade.php > html > body.bg-gray-100 > header > nav.bg-indigo-800.text-white.py-4.px-8.flex.justify-between.items-center > div.navbar-right.flex.items-center
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>mKasir - Home</title>
7   <script src="https://cdn.tailwindcss.com"></script>
8   <link href="https://fonts.googleapis.com/css2?family=Segoe+UI:wght@400;600;700&display=swap" rel="stylesheet">
9   <style>
10     body {
11       font-family: 'Segoe UI', sans-serif; /* Use Segoe UI or a close alternative */
12     }
13   </style>
14 </head>
15 <body class="bg-gray-100">
16   <header>
17     <nav class="bg-indigo-800 text-white py-4 px-8 flex justify-between items-center">
18       <div class="navbar-left">
19         <h1 class="text-xl font-semibold">mKasir</h1>
20       </div>
21       <div class="navbar-right flex items-center">
22         <a href="/login" class="bg-white text-indigo-800 py-2 px-4 rounded-md font-bold mr-2 hover:bg-gray-100 transition duration-300">Log
23         <a href="/register" class="bg-transparent border border-white text-white py-2 px-4 rounded-md font-bold hover:bg-white hover:text-
24       </div>
25     </nav>
26   </header>
27
28   <main>
29     <section class="bg-gray-100 py-16 px-8 flex justify-between items-center max-w-6xl mx-auto">
30       <div class="hero-content max-w-md">
31         <h2 class="text-4xl font-semibold text-gray-800 mb-4">Selamat datang di mKasir</h2>
32         <p class="text-lg text-gray-700 mb-8">Aplikasi kasir sederhana berbasis web dengan Laravel dan AJAX</p>
33         <a href="/register" class="bg-indigo-800 text-white py-3 px-6 rounded-md font-bold hover:bg-indigo-700 transition duration-300">Mu
```

c. Javascript (transactions)

```
resources > views > transactions.blade.php > html > body.bg-gray-50.antialiased.h-screen.flex.overflow-hidden > script > ready() callback > fetchTransactions > xhrFields
```

```
22 <body class="bg-gray-50 antialiased h-screen flex overflow-hidden">
188
189   <script>
190     $(document).ready(function() {
191       function fetchTransactions() {
192         url: '/api/transactions/${userId}',
193         type: 'GET',
194         dataType: 'json',
195         xhrFields: {
196           withCredentials: true
197         },
198       success: function(response) {
199         if (response.status === 'success') {
200           let transactions = response.data;
201           let tableBody = $('#transactionsTable').find('tbody');
202           tableBody.empty();
203
204           transactions.forEach(transaction => {
205             tableBody.append(
206               <tr class="hover:bg-gray-50">
207                 <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-800">${transaction.id}</td>
208                 <td class="px-6 py-4 text-sm font-medium text-gray-900">${transaction.product.name}</td>
209                 <td class="px-6 py-4 whitespace-nowrap text-sm text-purple-600">${transaction.quantity}</td>
210                 <td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-blue-700">${transaction.total_price}</td>
211                 <td class="px-6 py-4 whitespace-nowrap text-sm">
212                   <button data-id="${transaction.id}" class="edit-btn bg-yellow-500 hover:bg-yellow-600 text-white font-m
213                   <button data-id="${transaction.id}" class="delete-btn bg-red-500 hover:bg-red-600 text-white font-medium
214                 </td>
215             </tr>
216           );
217         });
218       } else {
219         showErrorModal('Failed to fetch transactions: ' + response.message);
220       }
221     });
222   </script>
```

d. Controller (products)

```
app > Http > Controllers > ProductController.php > ProductController > show
```

```
8 use Illuminate\Support\Facades\Validator;
9
10 class ProductController extends Controller
11 {
12   public function index($userId)
13   {
14     $products = Product::where('user_id', $userId)->get();
15     return response()->json($products, 200);
16   }
17
18   public function store(Request $request)
19   {
20     $validator = Validator::make($request->all(), [
21       'user_id' => 'required|integer|exists:users,id',
22       'name' => 'required|string|max:255',
23       'price' => 'required|numeric|min:0',
24       'stock' => 'nullable|integer|min:0',
25     ]);
26
27     if ($validator->fails()) {
28       return response()->json(['message' => 'Validation error', 'errors' => $validator->errors()], 422);
29     }
30
31     $product = new Product();
32     $product->user_id = $request->input('user_id');
33     $product->name = $request->input('name');
34     $product->price = $request->input('price');
35     $product->stock = $request->input('stock', 0);
36     $product->save();
37
38     return response()->json($product, 201);
39   }
40 }
```

## e. Model (products)

```

EXPLORER
  MYKASIR
    app
      Http
      middleware
      Models
        Product.php
        Transaction.php
        User.php
      Providers
      bootstrap
      config
      database
      docs
      public
      assets
      .htaccess
      favicon.ico
      index.php
      robots.txt
      resources
      css
      js
      views
      home.blade.php
      login.blade.php
      products.blade.php
      register.blade.php
      transactions.blade.php
    OUTLINE
    TIMELINE

Transaction.php
  1 <?php
  2
  3 namespace App\Models;
  4
  5 use Illuminate\Database\Eloquent\Factories\HasFactory;
  6 use Illuminate\Database\Eloquent\Model;
  7
  8 class Transaction extends Model
  9 {
 10     use HasFactory;
 11
 12     protected $fillable = [
 13         'product_id',
 14         'quantity',
 15         'total_price',
 16         'user_id'
 17     ];
 18
 19     public function product()
 20     {
 21         return $this->belongsTo(Product::class);
 22     }
 23
 24     public function user()
 25     {
 26         return $this->belongsTo(User::class);
 27     }
 28 }
 29

```

## f. DB



2. Permasalahan yang dihadapi selama fase implementasi (Tuliskan solusi jika sudah ditemukan)

Permasalahan pertama yang saya hadapi selama mengimplementasikan ide saya dalam kodingan ini adalah menerapkan autentikasi. Saya kesulitan untuk menggunakan auth:sanctum, yang menggunakan authorization header sebagai autentikasi, ataupun tools autentikasi bawaan lain dari laravel yang membuat user bisa mengakses seluruh halaman terproteksi tanpa login. Saya akhirnya memilih untuk membuat middleware sendiri berupa autentikasi berdasarkan auth\_token, yang akan dicek validasinya dengan table dari auth:sanctum, tapi dicocokkan dengan auth\_token yang di set (plain-textnya) di cookies.

Permasalahan saya selanjutnya adalah berusaha melindungi products/transactions users berdasarkan web routes, bukan api routes. Misalnya web url /products/13 akan menampilkan seluruh produk yang dimiliki oleh user dengan id 13. Saya tidak berhasil melindungi products yang dimiliki masing masing user (user bisa pergi ke url dengan user id lain dan melihat items mereka). Saya menemukan cara mengatasinya dengan melindungi api routes nya saja (di masing masing controller). Jadi sistem melakukan check user\_id dari PersonalAccessToken user yang sedang ter-login, dan melakukan validasi tersebut di fetchproducts/fetchtransactions (controller index). Jadi, jika user berusaha mengakses item yang bukan miliknya, akan dilempar error 403:forbidden sebelum user tersebut dapat mengecek item apapun yang ada di halaman tersebut. Cara ini masih bisa di improve dengan cara “membungkus” api routesnya itu sendiri dengan middleware baru saja (sekalian agar bisa redirect ke item miliknya sendiri) ditambah dengan middleware autentikasi dipermasalahan pertama, tidak perlu cek di controller. Namun saya baru kepikiran setelah project ini telah selesai saya kerjakan. Mungkin untuk selanjutnya akan saya implementasikan cara tersebut.