

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

Pemrograman Web II - Progress Report I

1. Perubahan topik project, disertai alasan mengapa berubah

Saya melakukan perubahan topik project, yang awalnya “FictionScope”, sebuah website yang digunakan untuk mereview berbagai jenis fiksi, seperti film, anime, novel, dsb. Sekarang project saya bernama “mYkasir” yang merupakan sebuah website kasir sederhana dimana user bisa melakukan operasi crud untuk produk dan transaksi

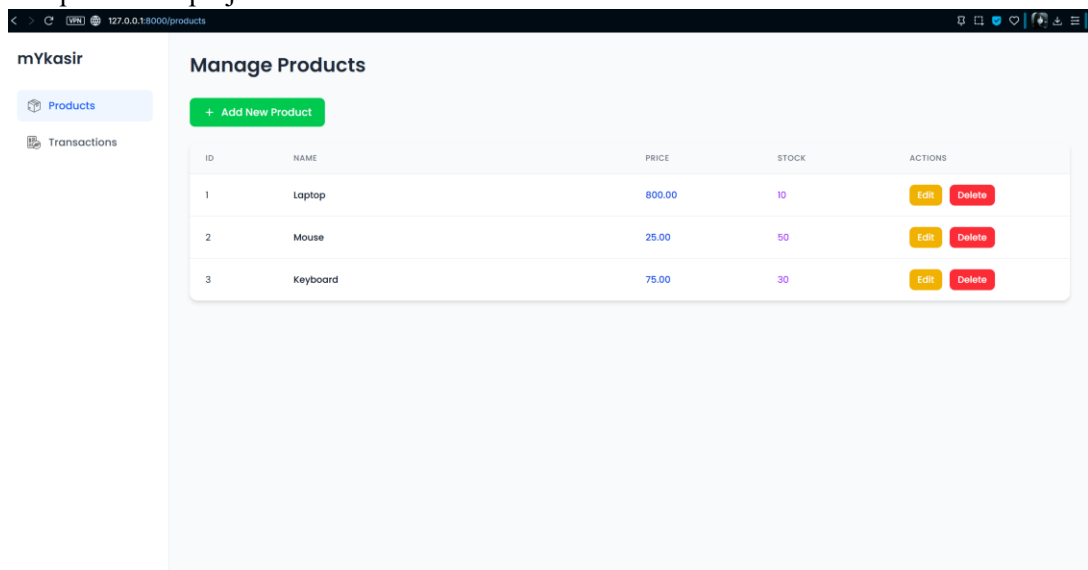
Alasan saya memilih untuk mengubah topik project saya adalah karena menurut saya cakupan dari website sebelumnya terlalu luas, dan terlalu kompleks. Saya merasa tidak sanggup untuk mengerjakannya dengan skill yang saya miliki saat ini. Ditambah banyaknya project project lain, saya merasa tidak akan punya waktu untuk belajar sekaligus mengerjakan website sesuai dengan kriteria awal yang saya berikan.

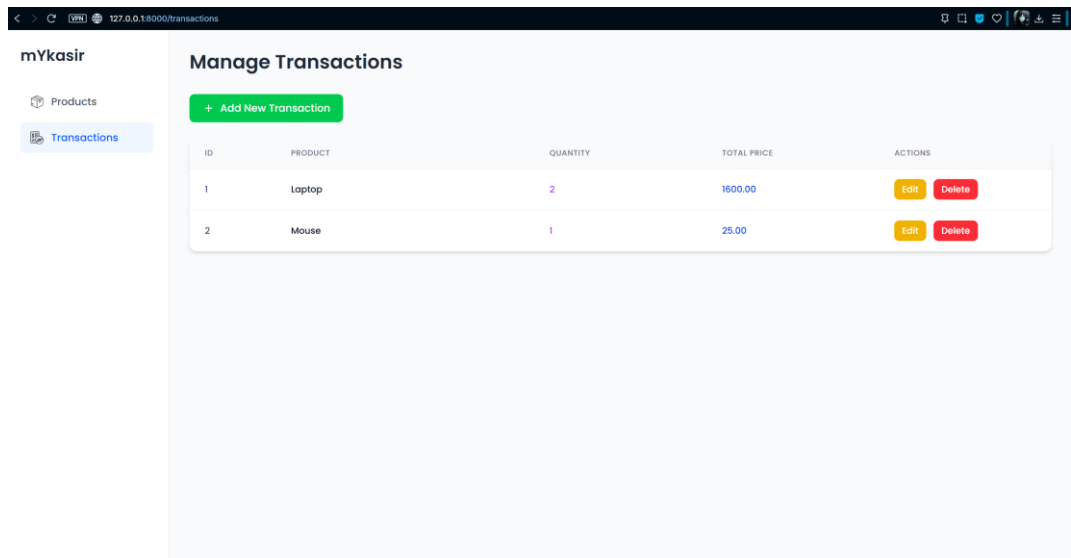
2. Tampilan awal project, berisi Implementasi kode yang sudah dibuat (baik View/Blade, Controller, atau Desain Model / DB yang sudah dibuat)

note: untuk kodingannya, hanya saya ss beberapa saja, selebihnya ada di:

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

- a. Tampilan awal project





b. Blade.php

```

<body class="bg-gray-50 antialiased h-screen flex overflow-hidden">
  <aside class="bg-white w-64 flex-shrink-0 border-r border-gray-200">
    <div class="p-6">
      <h2 class="text-2xl font-semibold text-gray-800 mb-8">mYkasir</h2>
      <nav class="flex flex-col space-y-2">
        <a href="/products" class="bg-blue-50 hover:bg-blue-100 text-blue-600 rounded-lg py-3 px-4 font-medium transition duration-150 ease">
          
          Products
        </a>
        <a href="/transactions" class="text-gray-600 hover:bg-gray-100 rounded-lg py-3 px-4 font-medium transition duration-150 ease-in-out">
          
          Transactions
        </a>
      </nav>
    </div>
  </aside>

  <div class="flex-1 overflow-auto">
    <div class="max-w-full mx-auto p-8">
      <h1 class="text-3xl font-semibold text-gray-800 mb-8">Manage Products</h1>

      <div id="addProductModal" class="fixed z-10 inset-0 overflow-y-auto bg-black/50 backdrop-blur-md hidden modal">
        <div class="flex items-center justify-center min-h-screen p-4">
          <div class="bg-white rounded-xl shadow-xl w-11/12 md:w-3/4 lg:w-1/2 max-w-lg">
            <div class="p-6 border-b border-gray-200">
              <div class="flex justify-between items-center">
                <h2 class="text-xl font-semibold text-gray-800">Add New Product</h2>
                <span class="close text-gray-400 hover:text-gray-600 cursor-pointer">

```

```

$(document).ready(function() {
  // Fetch and display products
  function fetchProducts() {
    $.ajax({
      url: '/api/products',
      type: 'GET',
      dataType: 'json',
      success: function(response) {
        if (response.status === 'success') {
          let products = response.data;
          let tableBody = $('#productsTable').find('tbody');
          tableBody.empty();

          products.forEach(product => {
            tableBody.append(
              <tr class="hover:bg-gray-50">
                <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-800">${product.id}</td>
                <td class="px-6 py-4 text-sm font-medium text-gray-900">${product.name}</td>
                <td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-blue-700">${product.price}</td>
                <td class="px-6 py-4 whitespace-nowrap text-sm text-purple-600">${product.stock}</td>
                <td class="px-6 py-4 whitespace-nowrap text-sm">
                  <button data-id="${product.id}" class="edit-btn bg-yellow-500 hover:bg-yellow-600 text-white font-medium py">
                  <button data-id="${product.id}" class="delete-btn bg-red-500 hover:bg-red-600 text-white font-medium py-1.5">
                </td>
              </tr>
            );
          });
        } else {
          showErrorModal('Failed to fetch products: ' + response.message);

```

c. Controller

```
class ProductController extends Controller
{
  /**
   * Display a listing of the resource.
   */
  public function index(): JsonResponse
  {
    try {
      $products = Product::all();
      return response()->json([
        'status' => 'success',
        'data' => $products,
      ], 200);
    } catch (Exception $e) {
      return response()->json([
        'status' => 'error',
        'message' => 'Failed to retrieve products: ' . $e->getMessage(),
      ], 500);
    }
  }

  /**
   * Store a newly created resource in storage.
   */
  public function store(Request $request): JsonResponse
  {
    $validator = Validator::make($request->all(), [
      'name' => 'required|string|max:255',
      'price' => 'required|numeric|min:0',
      'stock' => 'required|integer|min:0',
    ]);

class TransactionController extends Controller
{
  /**
   * Display a listing of the resource.
   */
  public function index(): JsonResponse
  {
    try {
      $transactions = Transaction::with('product')->get(); // Eager load product details
      return response()->json([
        'status' => 'success',
        'data' => $transactions,
      ], 200);
    } catch (Exception $e) {
      return response()->json([
        'status' => 'error',
        'message' => 'Failed to retrieve transactions: ' . $e->getMessage(),
      ], 500);
    }
  }

  /**
   * Store a newly created resource in storage.
   */
  public function store(Request $request): JsonResponse
  {
    $validator = Validator::make($request->all(), [
      'product_id' => 'required|exists:products,id',
      'quantity' => 'required|integer|min:1',
    ]);
  }
}
```

d. Model

```
class Product extends Model
{
    use HasFactory;

    protected $fillable = ['name', 'price', 'stock'];
}

class Transaction extends Model
{
    use HasFactory;

    protected $fillable = ['product_id', 'quantity', 'total_price'];

    public function product(): BelongsTo
    {
        return $this->belongsTo(Product::class);
    }
}
```

e. DB

