Dokumen R&D Lengkap - Sistem Inventori Toko Kelontong

# 1. Deskripsi Proyek

Sistem ini digunakan oleh toko kelontong untuk mencatat inventaris barang secara digital, menggantikan pencatatan manual. Sistem mencakup fitur pengelolaan data barang, kategori barang, stok masuk dan keluar, serta laporan barang.  
  
Framework: Laravel 12  
Database: MySQLi (MySQL)  
Frontend: Blade Template (Bootstrap)  
Metodologi: Agile (Scrum) - sprint mingguan

# 2. Struktur dan Relasi Database

Diagram Entitas Relasi (ERD):  
- users (1) --- (n) stock\_in / stock\_out  
- items (1) --- (n) stock\_in / stock\_out  
- categories (1) --- (n) items  
  
Tabel:  
- users(id, name, email, password, created\_at, updated\_at)  
- categories(id, name, created\_at, updated\_at)  
- items(id, name, category\_id, stock, unit, price, created\_at, updated\_at)  
- stock\_in(id, item\_id, quantity, date, user\_id)  
- stock\_out(id, item\_id, quantity, date, user\_id)

# 3. Dokumentasi API

Autentikasi:  
POST /api/login  
  
Barang:  
GET /api/items  
POST /api/items  
PUT /api/items/{id}  
DELETE /api/items/{id}  
  
Kategori:  
GET /api/categories  
POST /api/categories  
  
Stok Masuk:  
GET /api/stock-in  
POST /api/stock-in  
  
Stok Keluar:  
GET /api/stock-out  
POST /api/stock-out  
  
Laporan:  
GET /api/reports/stock

# 4. Fungsi Menu dan Modul

- Dashboard: Statistik barang dan grafik stok.  
- Data Barang: CRUD barang toko.  
- Kategori: Pengelolaan kategori.  
- Stok Masuk: Pencatatan barang masuk.  
- Stok Keluar: Pencatatan barang keluar.  
- Laporan: Laporan rekap bulanan/tahunan.  
- Pengguna: Manajemen akun admin.

# 5. Rancangan Tampilan Antarmuka

Sidebar Navigasi:  
- Dashboard  
- Barang  
- Kategori  
- Stok Masuk / Keluar  
- Laporan  
  
Halaman Barang:  
- Tabel daftar barang  
- Tombol Tambah/Edit/Hapus  
- Form input barang  
  
Halaman Laporan:  
- Filter berdasarkan tanggal/kategori  
- Export ke PDF/Excel

# 6. Konvensi Penamaan Variabel

- Nama variabel: camelCase (example: itemName, totalStock)  
- Nama API endpoint: kebab-case (example: /api/stock-in)  
- Nama model: PascalCase (User, Item, StockIn)  
- Penamaan tabel: snake\_case (stock\_out, stock\_in)

# 7. Rencana Pengembangan Agile (Sprint)

Sprint 1:  
- Setup project Laravel  
- Desain database  
- Fitur login dan autentikasi  
  
Sprint 2:  
- CRUD kategori & barang  
- Tampilan dashboard awal  
  
Sprint 3:  
- Fitur stok masuk/keluar  
- Validasi & notifikasi  
  
Sprint 4:  
- Fitur laporan dan export  
- Pengujian & deployment

# 8. Strategi Pengujian

- Unit testing: Controller, Model  
- API testing: Postman  
- UI testing: Manual klik dan uji input form  
- Blackbox testing: Simulasi penggunaan sistem oleh user  
  
Pengujian dilakukan pada akhir setiap sprint.

1. Sprint1
2. Instalasi: composer create-project laravel/laravel tokoKu

Composer create-project laravel/laravel tokoKu “10.\*”

1. Buat database:

php artisan make:migration create\_users\_table:

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('email')->unique();

$table->string('password');

$table->rememberToken();

$table->timestamps();

});

1. Buat Login
   1. Modifikasiapp/Model/User.php:

<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class User extends Authenticatable

{

use Notifiable;

protected $fillable = [

'name', 'email', 'password',

];

protected $hidden = [

'password', 'remember\_token',

];

// Relasi dengan stock\_in dan stock\_out

public function stockIns()

{

return $this->hasMany(StockIn::class);

}

public function stockOuts()

{

return $this->hasMany(StockOut::class);

}

}

b. Buat controller : php artisan make:controller Auth/AuthController

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class AuthController extends Controller

{

public function showLoginForm()

{

return view('auth.login');

}

public function login(Request $request)

{

$credentials = $request->validate([

'email' => 'required|email',

'password' => 'required',

]);

if (Auth::attempt($credentials)) {

$request->session()->regenerate();

return redirect()->intended('dashboard');

}

return back()->withErrors([

'email' => 'Email atau password salah.',

]);

}

public function logout(Request $request)

{

Auth::logout();

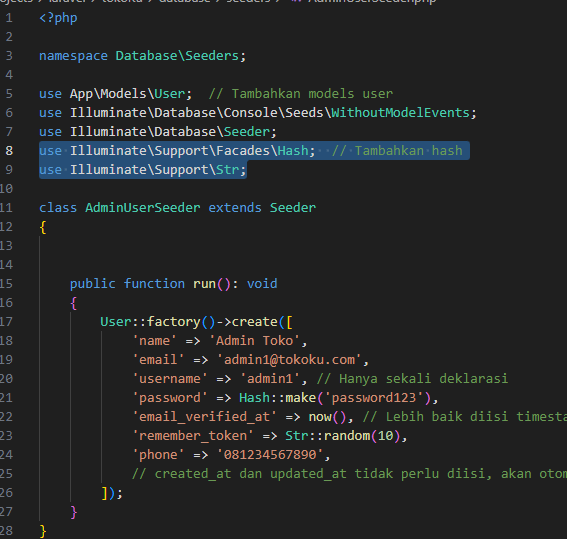
$request->session()->invalidate();

$request->session()->regenerateToken();

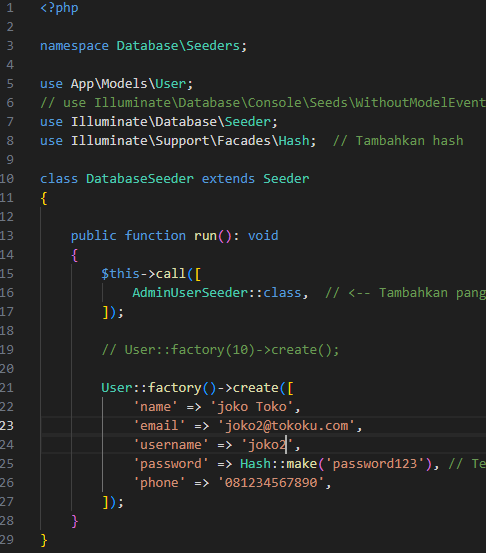
return redirect('/');

}

}

Admin seeder databse: 

Defaul seeder:



Route:



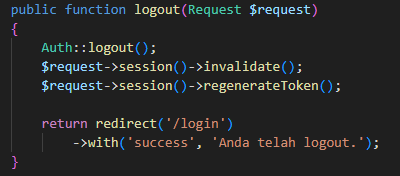
Controller login:



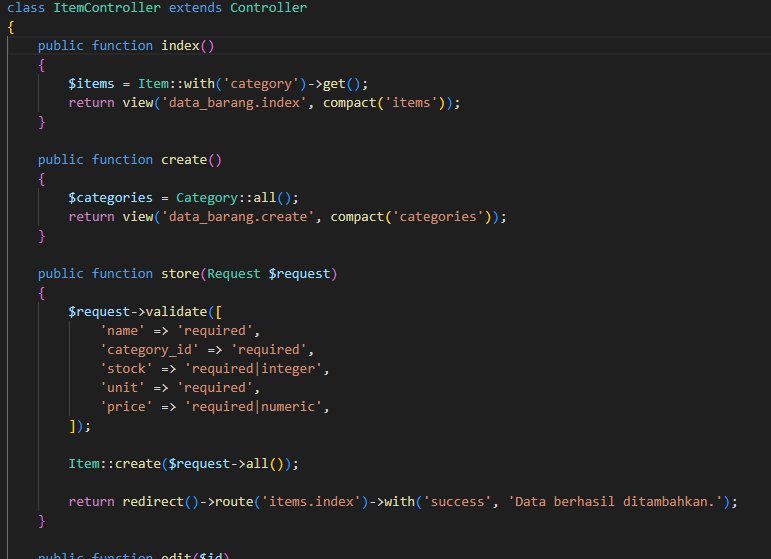
View login:

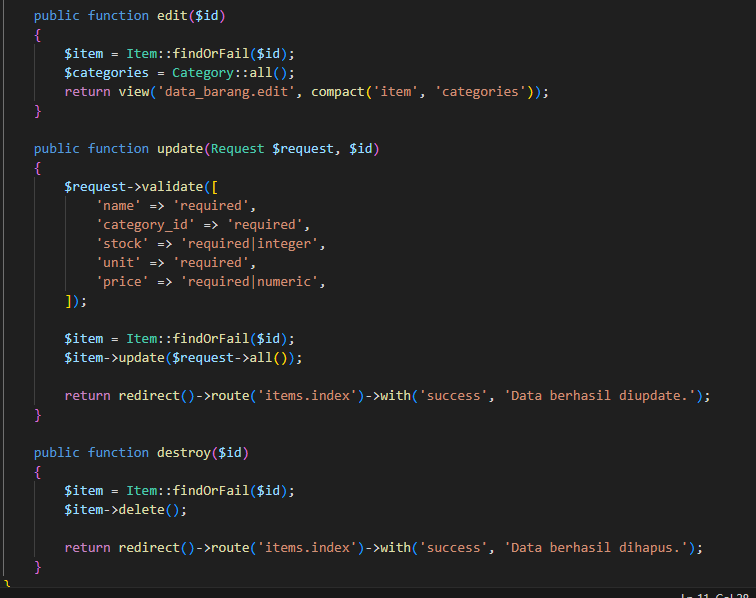


Logout:



Controller item:





View item:

@extends('layouts.app')

@section('content')

<div class="card">

    <div class="card-header d-flex justify-content-between align-items-center">

        <h5>Data Barang</h5>

        <a href="{{ route('items.create') }}" class="btn btn-primary btn-sm">+ Tambah Data</a>

    </div>

    <div class="card-body">

        <div class="table-responsive">

            <table id="myTable" class="table table-striped">

                <thead>

                    <tr>

                        <th>Nama Barang</th>

                        <th>Kategori</th>

                        <th>Stok</th>

                        <th>Satuan</th>

                        <th>Harga</th>

                        <th>Aksi</th> <!-- Tambahkan kolom Aksi -->

                    </tr>

                </thead>

                <tbody>

                    @foreach($items as $item)

                    <tr>

                        <td>{{ $item->name }}</td>

                        <td>{{ $item->category->name }}</td>

                        <td>{{ $item->stock }}</td>

                        <td>{{ $item->unit }}</td>

                        <td>Rp {{ number\_format($item->price, 2, ',', '.') }}</td>

                        <td>

                            <a href="{{ route('items.edit', $item->id) }}" class="btn btn-warning btn-sm">Edit</a>

                            <form action="{{ route('items.destroy', $item->id) }}" method="POST" style="display:inline-block;"  class="delete-form">

                                @csrf

                                @method('DELETE')

                                <button class="btn btn-danger btn-sm" type="submit">Hapus</button>

                            </form>

                        </td>

                    </tr>

                    @endforeach

                </tbody>

            </table>

        </div>

    </div>

</div>

@endsection

@push('scripts')

<script>

$(document).ready(function() {

    $('#myTable').DataTable({

        responsive: true

    });

    // Auto hide alert success

    setTimeout(function() {

        $("#success-alert").fadeOut('slow');

    }, 1500);

    // SweetAlert untuk delete

    $('.delete-form').on('submit', function(e) {

        e.preventDefault();

        const form = this;

        Swal.fire({

            title: 'Yakin mau hapus?',

            text: "Data yang dihapus tidak bisa dikembalikan!",

            icon: 'warning',

            showCancelButton: true,

            confirmButtonColor: '#3085d6',

            cancelButtonColor: '#d33',

            confirmButtonText: 'Ya, hapus!',

            cancelButtonText: 'Batal'

        }).then((result) => {

            if (result.isConfirmed) {

                form.submit();

            }

        });

    });

    // SweetAlert setelah berhasil tambah/edit/delete

    @if (session('success'))

        Swal.fire({

            icon: 'success',

            title: '{{ session('success') }}',

            showConfirmButton: false,

            timer: 1500

        });

    @endif

});

</script>

@endpush

Create itm:@extends('layouts.app')

@section('content')

<div class="card">

    <div class="card-header">

        Tambah Data Barang

    </div>

    <div class="card-body">

        <form action="{{ route('items.store') }}" method="POST">

            @csrf

            <div class="mb-3">

                <label>Nama Barang</label>

                <input type="text" name="name" class="form-control" required>

            </div>

            <div class="mb-3">

                <label>Kategori</label>

                <select name="category\_id" class="form-control" required>

                    <option value="">-- Pilih Kategori --</option>

                    @foreach($categories as $category)

                    <option value="{{ $category->id }}">{{ $category->name }}</option>

                    @endforeach

                </select>

            </div>

            <div class="mb-3">

                <label>Stok</label>

                <input type="number" name="stock" class="form-control" required>

            </div>

            <div class="mb-3">

                <label>Satuan</label>

                <input type="text" name="unit" class="form-control" required>

            </div>

            <div class="mb-3">

                <label>Harga</label>

                <input type="number" name="price" class="form-control" required>

            </div>

            <button type="submit" class="btn btn-primary">Simpan</button>

            <a href="{{ route('items.index') }}" class="btn btn-secondary">Batal</a>

        </form>

    </div>

</div>

@endsection

Edit item:@extends('layouts.app')

@section('content')

<div class="card">

    <div class="card-header">

        Edit Data Barang

    </div>

    <div class="card-body">

        <form action="{{ route('items.update', $item->id) }}" method="POST">

            @csrf

            @method('PUT')

            <div class="mb-3">

                <label>Nama Barang</label>

                <input type="text" name="name" class="form-control" value="{{ $item->name }}" required>

            </div>

            <div class="mb-3">

                <label>Kategori</label>

                <select name="category\_id" class="form-control" required>

                    @foreach($categories as $category)

                    <option value="{{ $category->id }}" {{ $item->category\_id == $category->id ? 'selected' : '' }}>

                        {{ $category->name }}

                    </option>

                    @endforeach

                </select>

            </div>

            <div class="mb-3">

                <label>Stok</label>

                <input type="number" name="stock" class="form-control" value="{{ $item->stock }}" required>

            </div>

            <div class="mb-3">

                <label>Satuan</label>

                <input type="text" name="unit" class="form-control" value="{{ $item->unit }}" required>

            </div>

            <div class="mb-3">

                <label>Harga</label>

                <input type="number" name="price" class="form-control" value="{{ $item->price }}" required>

            </div>

            <button type="submit" class="btn btn-primary">Update</button>

            <a href="{{ route('items.index') }}" class="btn btn-secondary">Batal</a>

        </form>

    </div>

</div>

@endsection