Jum'at, 28 Juni 2024

LAPORAN UJIAN AKHIR SEMESTER
PEMROGRAMAN WEB II

Panky Bintang Pradana Yosua//H1D022077

Repositori Github aplikasi: https://github.com/discernix/simi
Framework yang digunakan: Laravel
Library CSS: Tailwindcss, PicoCSS (minimalist css)
Autentikasi: Laravel Breeze

A. Hasil Coding

1. Migration User

```php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
     * Reverse the migrations.
```

2. Migration Kategori

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function up(): void
    {
        Schema::create('kategori', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function down(): void
    {
        Schema::dropIfExists('kategori');
    }
};
```

3. Migration Suppliers

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function up(): void
    {
        Schema::create('suppliers', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->text('alamat');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function down(): void
    {
        Schema::dropIfExists('suppliers');
    }
};
```

4. Migration Barang

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function up(): void
    {
        Schema::create('barang', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->text('deskripsi');
            $table->bigInteger('harga');
            $table->unsignedBigInteger('id_kategori');
            $table->unsignedBigInteger('id_supplier');
            $table->foreign('id_kategori')->references('id')->on('kategori')->onDelete('cascade');
            $table->foreign('id_supplier')->references('id')->on('suppliers')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function down(): void
    {
        Schema::dropIfExists('barang');
    }
};
```

5. Migration Penerimaan

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function up(): void
    {
        Schema::create('penerimaan', function (Blueprint $table) {
            $table->id();
            $table->date('tanggal_terima');
            $table->integer('jumlah');
            $table->unsignedBigInteger('id_barang');
            $table->foreign('id_barang')->references('id')->on('barang')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    Codeium: Refactor | Explain | ×
    public function down(): void
    {
        Schema::dropIfExists('penerimaan');
    }
};
```

6. Routing Web

```php
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/dashboard', function () {
    return view('dashboard', [
        'menu' => [
            [
                'title' => 'Kategori',
                'description' => 'Manajemen Kategori merupakan halaman untuk mengelola kategori'
            ],
            [
                'title' => 'Suppliers',
                'description' => 'Manajemen Supplier merupakan halaman untuk mengelola supplier'
            ],
            [
                'title' => 'Barang',
                'description' => 'Manajemen Barang merupakan halaman untuk mengelola barang'
            ],
            [
                'title' => 'Penerimaan',
                'description' => 'Manajemen Penerimaan merupakan halaman untuk mengelola penerimaan'
            ]
        ]
    ]);
})->middleware(['auth', 'verified'])->name('dashboard');

Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
    Route::prefix('dashboard')->group(function () {
        Route::resource('/kategori', KategoriController::class);
        Route::resource('/suppliers', SupplierController::class);
        Route::resource('/barang', BarangController::class);
        Route::resource('/penerimaan', PenerimaanController::class);
    });
});
```

7. Login

```php
You, 8 minutes ago | 1 author (You) | Codeium: Refactor | Explain
class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     */
    Codeium: Refactor | Explain | ×
    public function create(): View
    {       You, 8 minutes ago • first commit
        return view('auth.login');
    }

    /**
     * Handle an incoming authentication request.
     */
    Codeium: Refactor | Explain | ×
    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

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

        return redirect()->intended(route('dashboard', absolute: false));
    }

    /**
     * Destroy an authenticated session.
     */
    Codeium: Refactor | Explain | ×
    public function destroy(Request $request): RedirectResponse
    {
        Auth::guard('web')->logout();

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

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

        return redirect('/');
    }
}
```

8. Registrasi

```php
You, 7 minutes ago | 1 author (You) | Codeium: Refactor | Explain
class RegisteredUserController extends Controller
{
    /**
     * Display the registration view.
     */
    Codeium: Refactor | Explain | ×
    public function create(): View      You, 7 minutes ago • first commit
    {
        return view('auth.register');
    }

    /**
     * Handle an incoming registration request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    Codeium: Refactor | Explain | ×
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:'.User::class],
            'password' => ['required', 'confirmed', Rules\Password::defaults()],
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);

        event(new Registered($user));

        Auth::login($user);

        return redirect(route('dashboard', absolute: false));
    }
}
```

9. CRUD Kategori

```php
<?php

namespace App\Http\Controllers;

use App\Models\Kategori;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

Codeium: Refactor | Explain
class KategoriController extends Controller
{
    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function index()
    {
        return view('kategori.index', [
            'title' => 'Kategori',
            'data' => Kategori::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function store(Request $request)
    {
        $request->validate([
            'nama' => 'min:3'
        ]);

        Kategori::create($request->input());
        return Redirect::back();
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function edit(Kategori $kategori)
    {
        return view('kategori.edit', [
            'title' => 'Kategori',
            'data' => $kategori
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function update(Request $request, Kategori $kategori)
    {
        $request->validate([
            'nama' => 'min:3'
        ]);

        $kategori->update($request->input());
        return Redirect::to(route('kategori.index'));
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function destroy(Kategori $kategori)
    {
        $kategori->delete();
        return Redirect::back();
    }
}
```

10. CRUD Suppliers

```php
<?php

namespace App\Http\Controllers;

use App\Models\Supplier;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

Codeium: Refactor | Explain
class SupplierController extends Controller
{
    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function index()
    {
        return view('suppliers.index', [
            'title' => 'Suppliers',
            'data' => Supplier::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function store(Request $request)
    {
        $request->validate([
            'nama' => 'min:3',
            'alamat' => 'min:5'
        ]);

        Supplier::create($request->input());
        return Redirect::back();
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function edit(Supplier $supplier)
    {
        return view('suppliers.edit', [
            'title' => 'Suppliers',
            'data' => $supplier
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function update(Request $request, Supplier $supplier)
    {
        $request->validate([
            'nama' => 'min:3',
            'alamat' => 'min:5'
        ]);

        $supplier->update($request->input());
        return Redirect::to(route('suppliers.index'));
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function destroy(Supplier $supplier)
    {
        $supplier->delete();
        return Redirect::back();
    }
}
```

11. CRUD Barang

```php
<?php

namespace App\Http\Controllers;

use App\Models\Barang;
use App\Models\Kategori;
use App\Models\Supplier;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

Codeium: Refactor | Explain
class BarangController extends Controller
{
    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function index()
    {
        return view('barang.index', [
            'title' => 'Barang',
            'data' => Barang::with(['kategori', 'suppliers'])->get(),
            'kategori' => Kategori::all(),
            'suppliers' => Supplier::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function store(Request $request)
    {
        $request->validate([
            'nama' => 'min:3',
            'deskripsi' => 'min:10',
            'harga' => 'required',
        ]);

        Barang::create($request->input());
        return Redirect::back();
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function edit(Barang $barang)
    {
        return view('barang.edit', [
            'title' => 'Barang',
            'data' => $barang,
            'kategori' => Kategori::all(),
            'suppliers' => Supplier::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function update(Request $request, Barang $barang)
    {
        $request->validate([
            'nama' => 'min:3',
            'deskripsi' => 'min:10',
            'harga' => 'required'
        ]);

        $barang->update($request->input());
        return Redirect::to(route('barang.index'));
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function destroy(Barang $barang)
    {
        $barang->delete();
        return Redirect::back();
    }
}
```

12. CRUD Penerimaan

```php
namespace App\Http\Controllers;

use App\Models\Barang;
use App\Models\Penerimaan;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

Codeium: Refactor | Explain
class PenerimaanController extends Controller
{
    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function index()
    {
        return view('penerimaan.index', [
            'title' => 'Penerimaan',
            'data' => Penerimaan::all(),
            'barang' => Barang::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function store(Request $request)
    {
        $request->validate([
            'jumlah' => 'required',
        ]);

        Penerimaan::create($request->input());
        return Redirect::back();
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function edit(Penerimaan $penerimaan)
    {
        return view('penerimaan.edit', [
            'title' => 'Penerimaan',
            'data' => $penerimaan,
            'barang' => Barang::all()
        ]);
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function update(Request $request, Penerimaan $penerimaan)
    {
        $request->validate([
            'jumlah' => 'required',
        ]);

        $penerimaan->update($request->input());
        return Redirect::to(route('penerimaan.index'));
    }

    Codeium: Refactor | Explain | Generate Function Comment | ×
    public function destroy(Penerimaan $penerimaan)
    {
        $penerimaan->delete();
        return Redirect::back();
    }
}
```

13. Model Kategori

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

Codeium: Refactor | Explain
class Kategori extends Model
{
    use HasFactory;

    protected $guarded = [];

    protected $table = 'kategori';
}
```

14. Model Suppliers

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Supplier extends Model
{
    use HasFactory;

    protected $guarded = [];

    protected $table = 'suppliers';
}
```

15. Model Barang

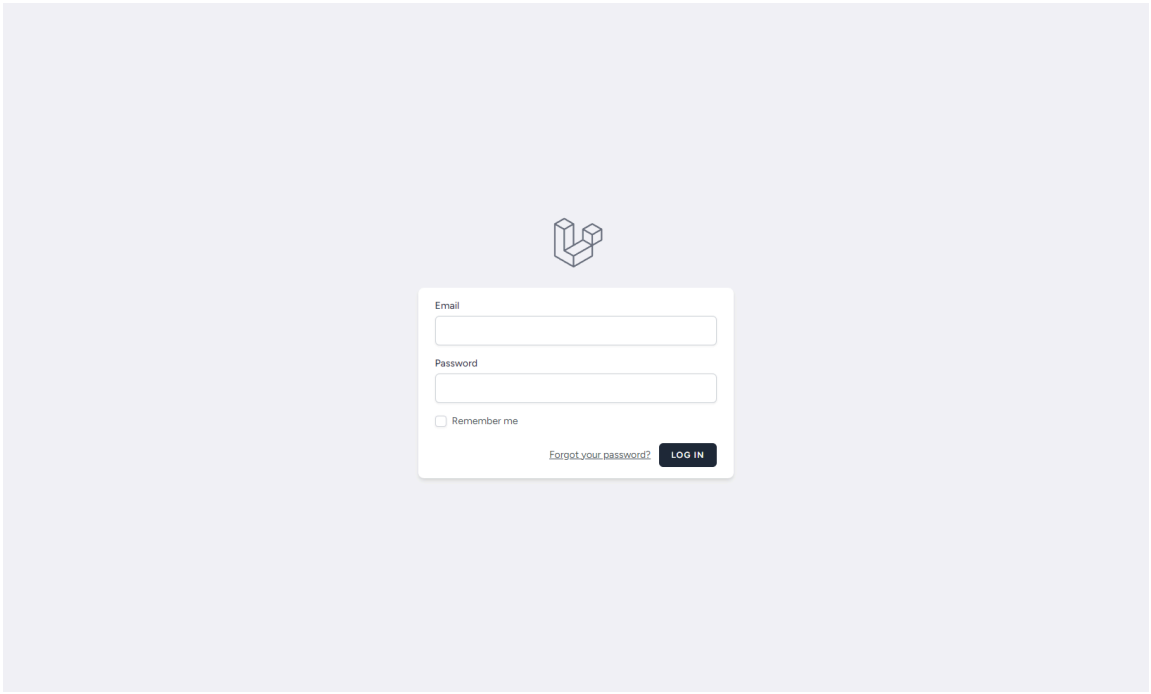
```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Barang extends Model
{
    use HasFactory;

    protected $guarded = [];

    protected $table = 'barang';

    public function kategori(): BelongsTo
    {
        return $this->belongsTo(Kategori::class, 'id_kategori');
    }

    public function suppliers(): BelongsTo
    {
        return $this->belongsTo(Supplier::class, 'id_supplier');
    }
}
```

16. Model Penerimaan

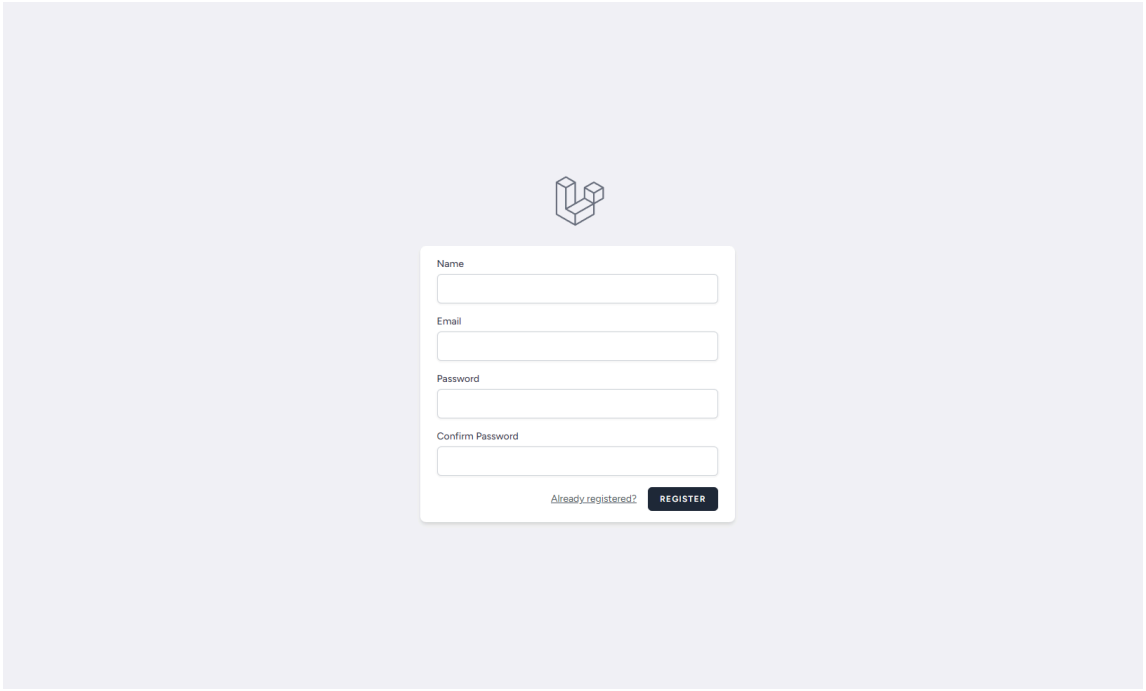
```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Penerimaan extends Model
{
    use HasFactory;

    protected $guarded = [];

    protected $table = 'penerimaan';

    public function barang(): BelongsTo
    {
        return $this->belongsTo(Barang::class, 'id_barang');
    }
}
```

B. Tampilan Aplikasi Web Manajemen Inventaris

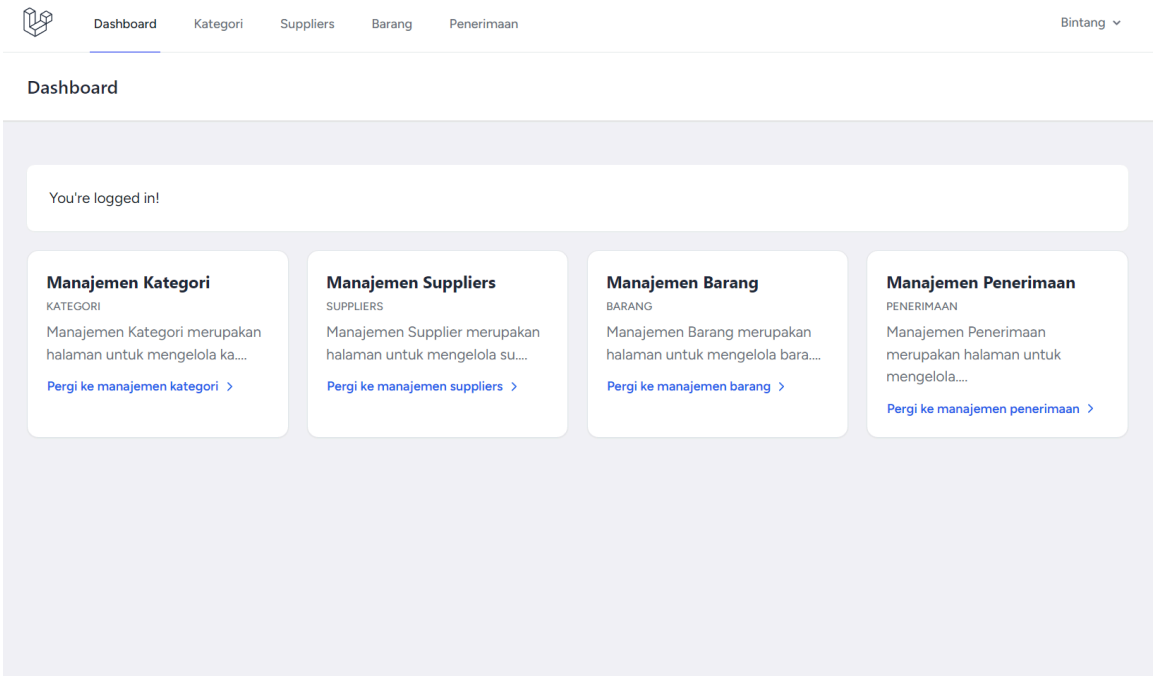
1. Tampilan Login



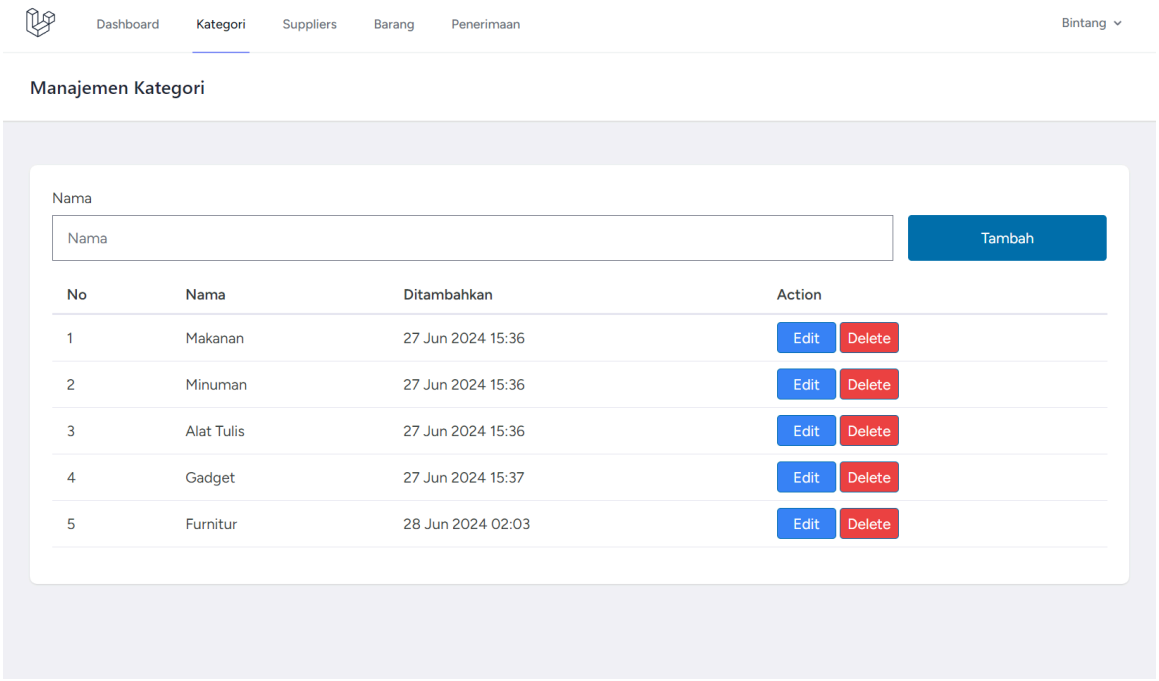
2. Tampilan Registrasi



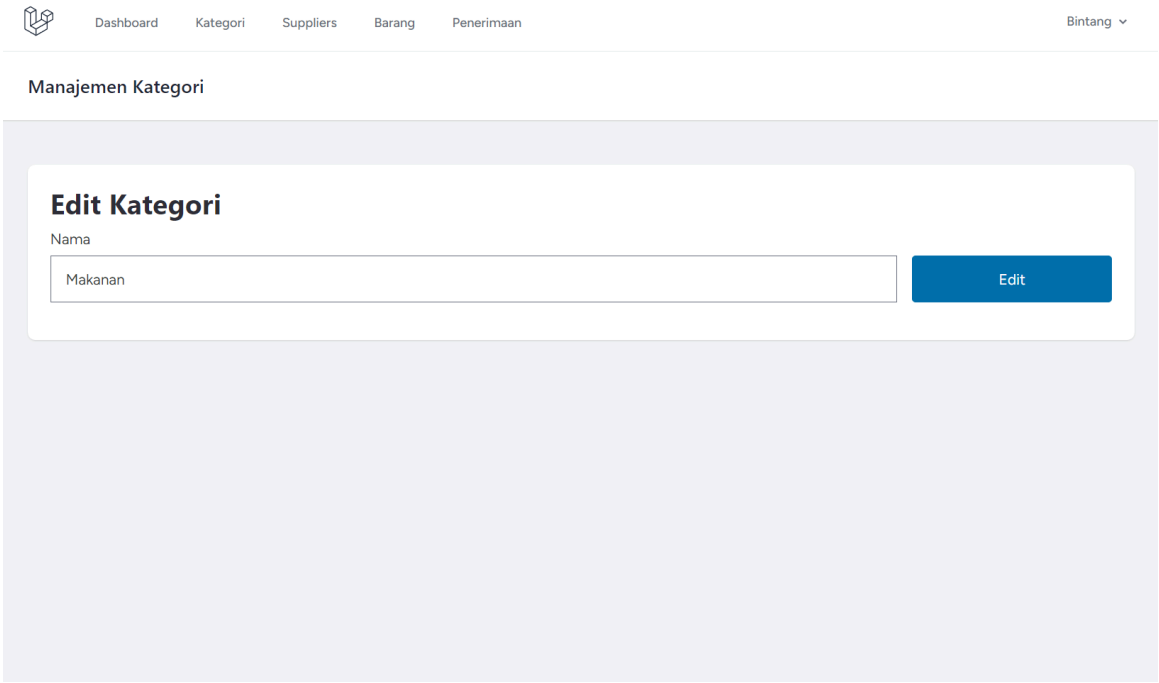
3. Halaman Dashboard



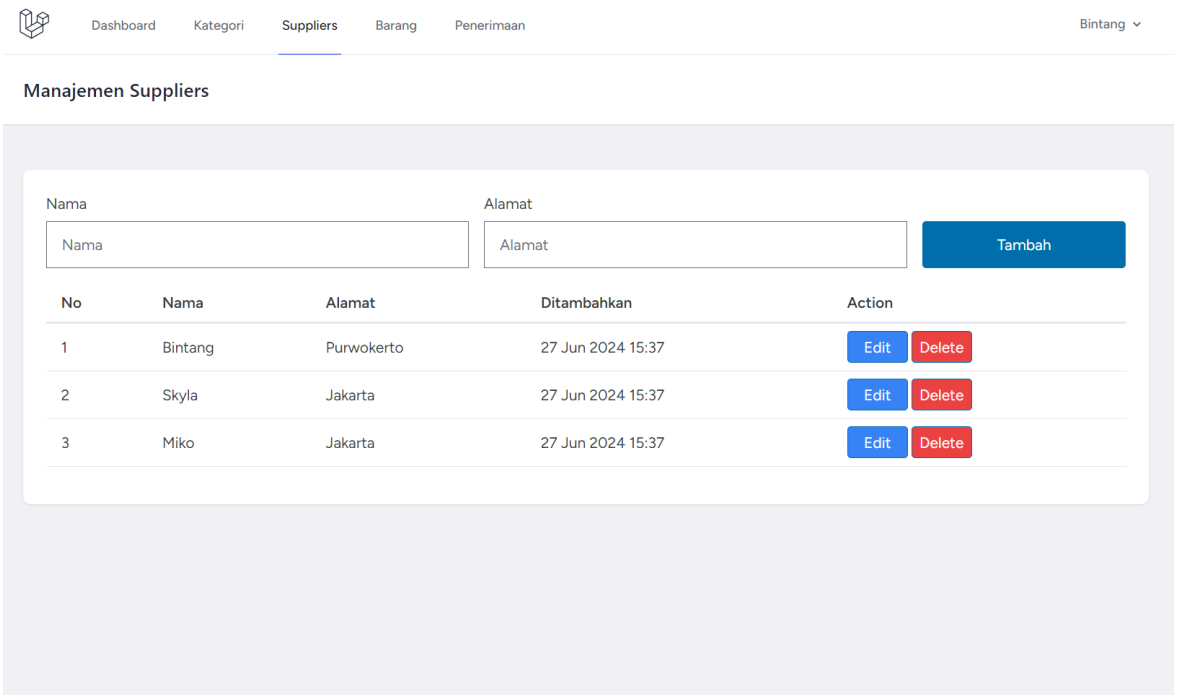
4. Halaman melihat, menambah, dan menghapus Kategori



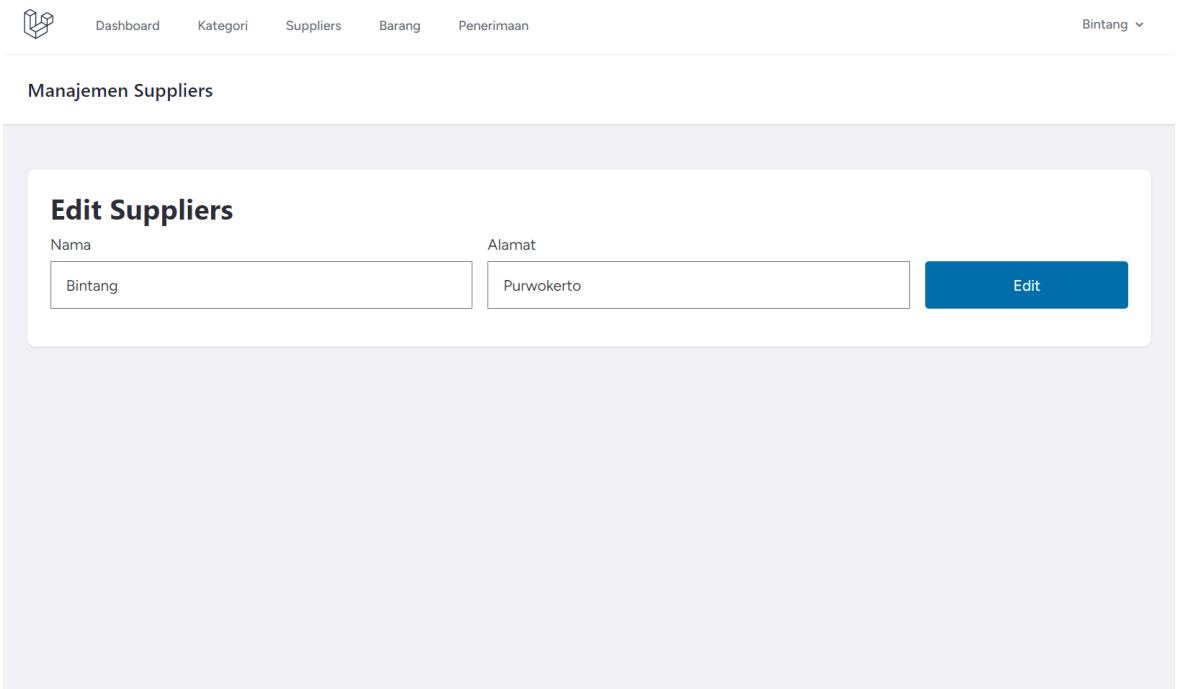
5. Halaman mengedit Kategori



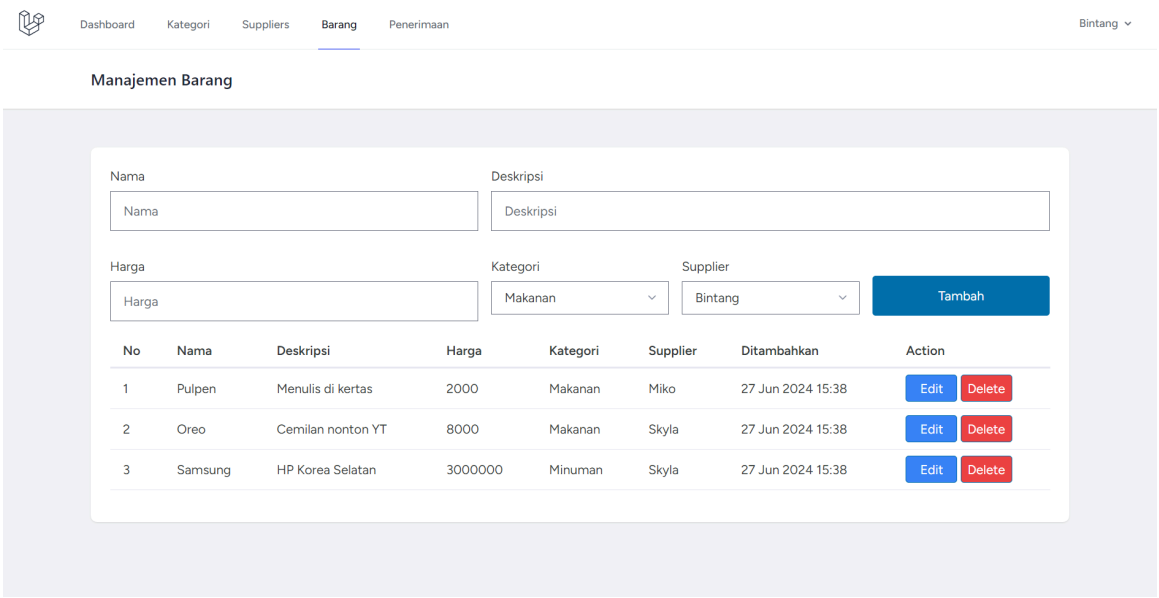
6. Halaman melihat, menambah, dan menghapus Suppliers



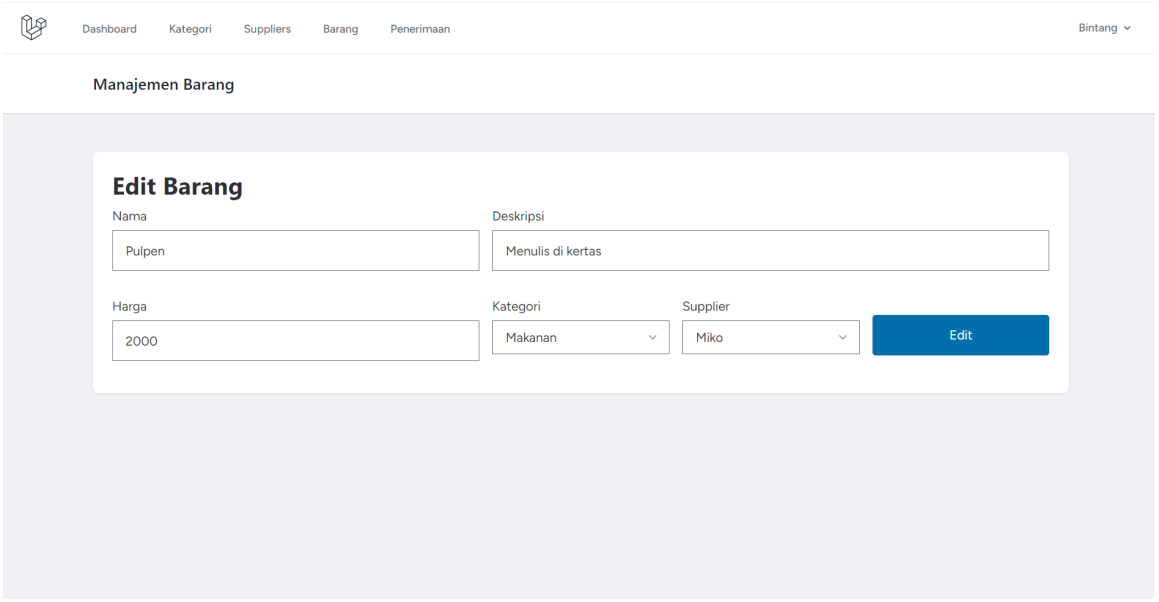
7. Halaman mengedit Supplier



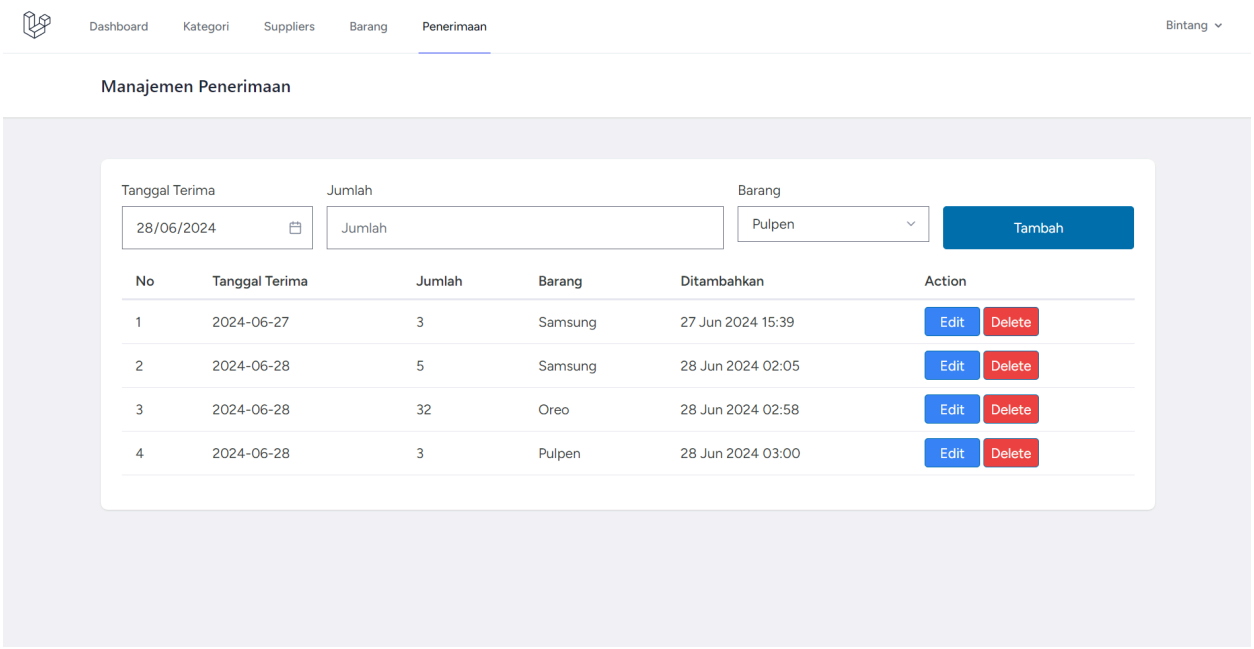
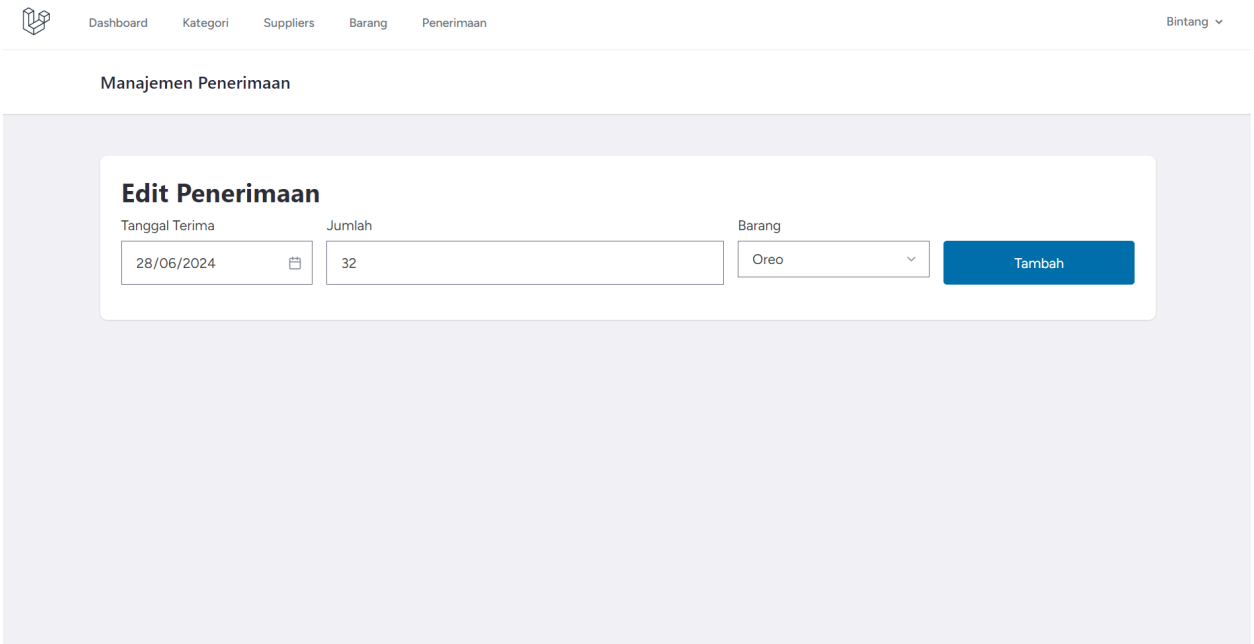
8. Halaman melihat, menambah, dan menghapus Barang



9. Halaman mengedit Barang



10. Halaman melihat, menambah, dan menghapus Penerimaan



11. Halaman mengedit penerimaan