LAPORAN AKHIR
PENGEMBANGAN SISTEM BERORIENTASI OBJEK
DEPARTEMEN ILMU KOMPUTER IPB
SEMESTER GENAP 2018/2019

NAMA APLIKASI : PS Store
NAMA DEVELOPER : Tim bel
ANGGOTA DEVELOPER : Mochamad Rhayfa - G64160054
: Ivan Mahinza - G64160066
: Suryo Hamukti S - G64160095

Deskripsi Singkat Aplikasi

PS Store merupakan sistem yang dibangun untuk jual-beli barang pada suatu toko, dimana terdapat 3 jenis user di dalamnya; pembeli, supplier, dan admin. Transaksi dapat dilakukan di sistem, mulai dari melihat barang yang tersedia, pembayaran, hingga melihat status pemesanan. Namun proses verifiksi pembayaran masih manual dengan cara mengunggah bukti pembayaran ke sistem untuk kemudian diverifikasi oleh admin secara manual. Aplikasi ini lebih cenderung seperti toko online (B2C) dibanding e-commerce (C2C).

User analysis

Aplikasi ini memiliki user sebanyak tiga jenis user, yaitu ada supplier, admin, dan pembeli. Target Pengguna aplikasi ini tentunya adalah supplier dan pembeli, sementara admin hanya untuk pengelola bisnisnya.

Supplier dapat melakukan :

- Meminta verifikasi admin penambahan / penghilangan barang yang ingin dijual
- Melihat produk dan detailnya

Admin dapat melakukan :

- Mengubah status pembayaran
- Mengatur kategori dan subkategori jenis barang
- Mengubah status user
- Mengontrol transaksi

Pembeli dapat melakukan :

- CRUD identitas
- Melihat semua produk
- Melihat detail produk
- Melihat data penjual

- CRUD bagian transaksi (meliputi add to cart sampai pemberitahuan transaksi telah selesai)

User Story

Sebagai Pembeli, saya ingin melihat lihat produk apa saja yang dijual oleh PS Store.
Keterangan : Terdapat Category dan Sub-Category, dan fitur search untuk memudahkan surfing

Sebagai Supplier, saya ingin mendapat keuntungan melalui menyuplai barang di PS Store, dan juga mengetahui seberapa laku barang dagangan yang saya suplai.
Keterangan : Saya bisa melakukan suplai apabila admin telah melakukan verifikasi / authorization terhadap suplai yang ingin saya salurkan, dan saya bisa melihat detail produk yang dijualnya

Sebagai Pembeli, saya ingin membeli produk yang telah saya lihat di PS Store.
Keterangan : Pembelian dilakukan dengan cara saya mencari dahulu apa saja yang saya ingin beli, kemudian dimasukkan kedalam keranjang, lalu saya melakukan checkout dan melakukan pembayaran di luar sistem, namun mengirim bukti pembayarannya kedalam sistem.

Sebagai Admin, saya ingin mengelola website PS Store dimana website tersebut menjadi sarana barang barang yang didapat dari supplier ditawarkan kepada khalayak umum (pembeli).
Tentunya saya perlu mengelola setiap adanya perubahan terkait konten website.
Keterangan : Setiap checkout adalah tanggungan admin, jadi admin perlu memverifikasi seluruh transaksi yang ada didalam sistem.

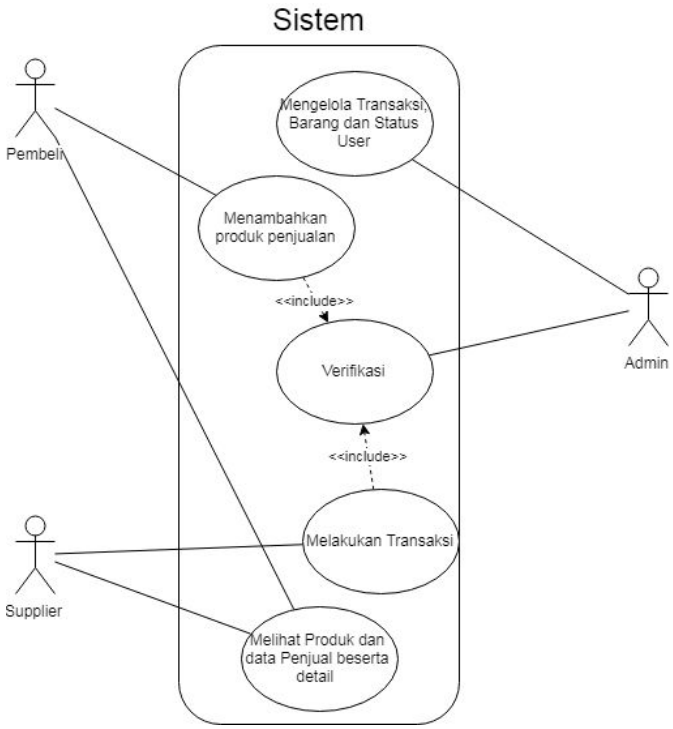
Spesifikasi teknis lingkungan pengembangan

Lingkungan Pengembangan PS Store adalah sebagai berikut :

- Laravel Framework version 5.8.16
- php version 7.1.3
- uxweb sweet-alert version 2.0 or greater
- SQLYog as GUI database version 13.1.1
- xampp for windows version 7.2.3
- Sublime text editor version 3.2.1
- Visual Studio Code version 1.33

Hasil dan pembahasan

Use Case Diagram



Keterangan terlampir dalam use case description.

Use Case Description

1. UC1 : Mengelola Transaksi, Barang, dan Status User
    - Importance Level : High
    - Use Case Type : Essential, Temporal
    - Primary Actor : Admin
    - Stakeholder and Interest : Supplier dan Pembeli
    - Brief Description : Di dalam use case ini, Admin bertugas untuk maintenance kondisi website.
    - Pre-Condition : Melakukan pengelolaan bila ada request dari user, dan pengelolaan bulanan
    - Trigger :
        - Ada pembeli yang ingin jadi supplier (External)
        - Ada barang yang sudah lama tidak ada transaksi dan tidak ada kunjungan (External)
        - Ada supplier yang tak bertanggung jawab (External)
        - Pengelolaan perbulan (Temporal)

- Events :
    - Menghapus / Mengubah User
    - Mengelola Jenis Barang
    - Menambahkan barang
    - Dan Lainnya
- Flow Events :
    - CRUD pada umumnya
- Alternate/Exceptional Flows : -
- Post-Condition : Tampilan dan Izin akses terbaharui

2. UC2 : Menambah Produk Penjualan
    - Importance Level : Medium
    - Use Case Type : Essential, Temporal
    - Primary Actor : Supplier
    - Stakeholder and Interest : Pembeli
    - Brief Description : Di dalam use case ini, Supplier dapat menambah produk beserta stocknya
    - Pre-Condition : Telah login sebagai Supplier
    - Trigger :
        - Supplier ingin menambahkan jenis barang baru yang disuplainya kedalam sistem (External)
        - Memasukan suplai bulanan (Temporal)
    - Events :
        - Menambahkan jenis produk pada database
        - Menambahkan stock produk pada database
        - Menghilangkan produk yang telah berhenti disuplai
    - Flow Events :
        - Login sebagai Supplier
        - Memilih produk mana yang ingin ditambah atau dihilangkan dan
        - Memberi pilihan yang akan diverifikasi admin
        - Tampilan akan terbaharui apabila verifikasi telah dilakukan oleh admin
    - Alternate/Exceptional Flows :
        - Admin men-decline verifikasi
    - Post-Condition : Data dalam sistem terbaharui

3. UC3 : Verifikasi
    - Importance Level : High
    - Use Case Type : Essential
    - Primary Actor : Admin

- Stakeholder and Interest : Pembeli dan Supplier
- Brief Description : Di dalam use case ini, Admin bertugas untuk memverifikasi seluruh transaksi dan barang transaksinya
- Pre-Condition : Ada sebuah rangkaian kegiatan dari end-user yang perlu diverifikasi
- Trigger :
    - Mendapat keranjang yang ingin dibayar oleh pembeli
    - Mendapat request penambahan / penghilangan produk dari supplier
- Events :
    - Melakukan verifikasi / Memberikan akses
- Flow Events :
    - Pembelian :
        - Admin mendapatkan keranjang yang ingin dibayar oleh pembeli
        - Pembeli melakukan pembayaran dan mengupload bukti pembayaran ke sistem
        - Admin melakukan verifikasi pembayaran di sistem dan mengirimkan barang ke pembeli dengan kurir pilihan pembeli
        - Admin memberikan notifikasi transaksi sukses berhasil
    - Penambahan / Penghilangan suplai :
        - Admin mendapatkan request dari supplier
        - Admin memastikan barang ada dan didapat dari supplier yang tepat, lalu melakukan verifikasi
        - Admin mengupdate database sistem
        - Tampilan akan terbaharui apabila verifikasi telah dilakukan oleh admin
- Alternate/Exceptional Flows :
    - Admin men-decline verifikasi
- Post-Condition : Data dalam sistem terbaharui, Status Keranjang berubah menjadi Transaksi Berhasil, Barang sampai pada tangan pembeli.

4. UC4 : Melakukan Transaksi
    - Importance Level : High
    - Use Case Type : Essential
    - Primary Actor : Pembeli
    - Stakeholder and Interest : Admin dan supplier
    - Brief Description : Di dalam use case ini, Pembeli melakukan inti proses bisnis sebuah jual beli
    - Pre-Condition : Telah login sebagai pembeli, sedang melihat lihat produk
    - Trigger :
        - Ada pembeli yang tertarik ingin membeli produk (External)

- Events :
    - Melakukan proses jual beli
- Flow Events :
    - Mencari / Melihat lihat produk
    - Melihat detail produk yang ingin dibeli
    - Menambahkan kedalam keranjang
    - Mencari produk lain yang ingin ditambahkan kedalam keranjang (opsional)
    - Melakukan Checkout
    - Mengupload bukti pembayaran
    - Terverifikasi oleh admin dan barang dikirim
    - Pembeli menerima barang beliannya.
- Alternate/Exceptional Flows : Tidak ada kurir yang tersedia, transaksi gagal secara terpaksa
- Post-Condition : Mendapatkan barang beliannya, riwayat pembelian terbaharui

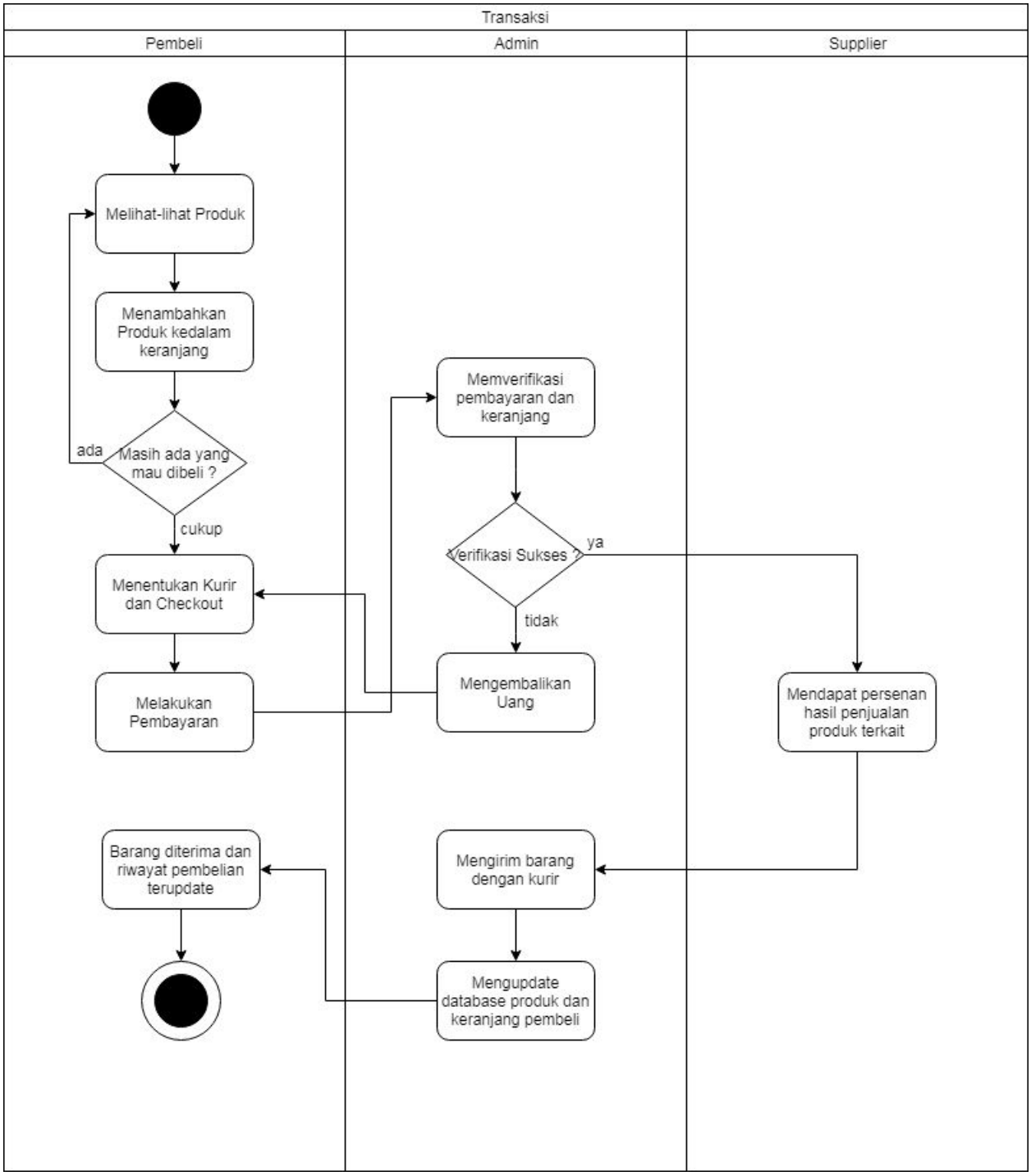
5. UC5 : Melakukan Transaksi
    - Importance Level : High
    - Use Case Type : Essential
    - Primary Actor : Pembeli
    - Stakeholder and Interest : Admin dan supplier
    - Brief Description : Di dalam use case ini, Pembeli melakukan inti proses bisnis sebuah jual beli toko online
    - Pre-Condition : Telah login sebagai pembeli, sedang melihat lihat produk
    - Trigger :
        - Ada pembeli yang tertarik ingin membeli produk (External)
    - Events :
        - Melakukan proses jual beli
    - Flow Events :
        - Mencari / Melihat lihat produk
        - Melihat detail produk yang ingin dibeli
        - Menambahkan kedalam keranjang
        - Mencari produk lain yang ingin ditambahkan kedalam keranjang (opsional)
        - Melakukan Checkout
        - Mengupload bukti pembayaran
        - Terverifikasi oleh admin dan barang dikirim
        - Pembeli menerima barang beliannya.
    - Alternate/Exceptional Flows : Tidak ada kurir yang tersedia, transaksi gagal secara terpaksa

- Post-Condition : Mendapatkan barang beliannya, riwayat pembelian terbaharui

6. UC6: Melihat produk, supplier, beserta detailnya
    - Importance Level : Medium
    - Use Case Type : Essential, Temporal
    - Primary Actor : Pembeli, Supplier
    - Stakeholder and Interest : Admin, Supplier dan Pembeli
    - Brief Description : Di dalam use case ini, end-user dapat melihat apa saja yang ditampilkan oleh sistem untuk diperjual belikan dan siapa supplier barangnya
    - Pre-Condition : Telah melakukan login
    - Trigger :
        - Ingin melihat detail spesifikasi produk (External)
        - Ingin melihat detail latar belakang suppliernya (External)
        - Supplier ingin melihat seberapa laku produk suplainya (Temporal)
    - Events :
        - Melihat lihat produk yang dijual di sistem
    - Flow Events :
        - Mencari dengan fitur search
        - Melakukan surfing berdasarkan kategori-subkategori
        - Menambahkan barang yang dilihat ke keranjang (akan lanjut ke proses transaksi)
    - Alternate/Exceptional Flows : -
    - Post-Condition : -

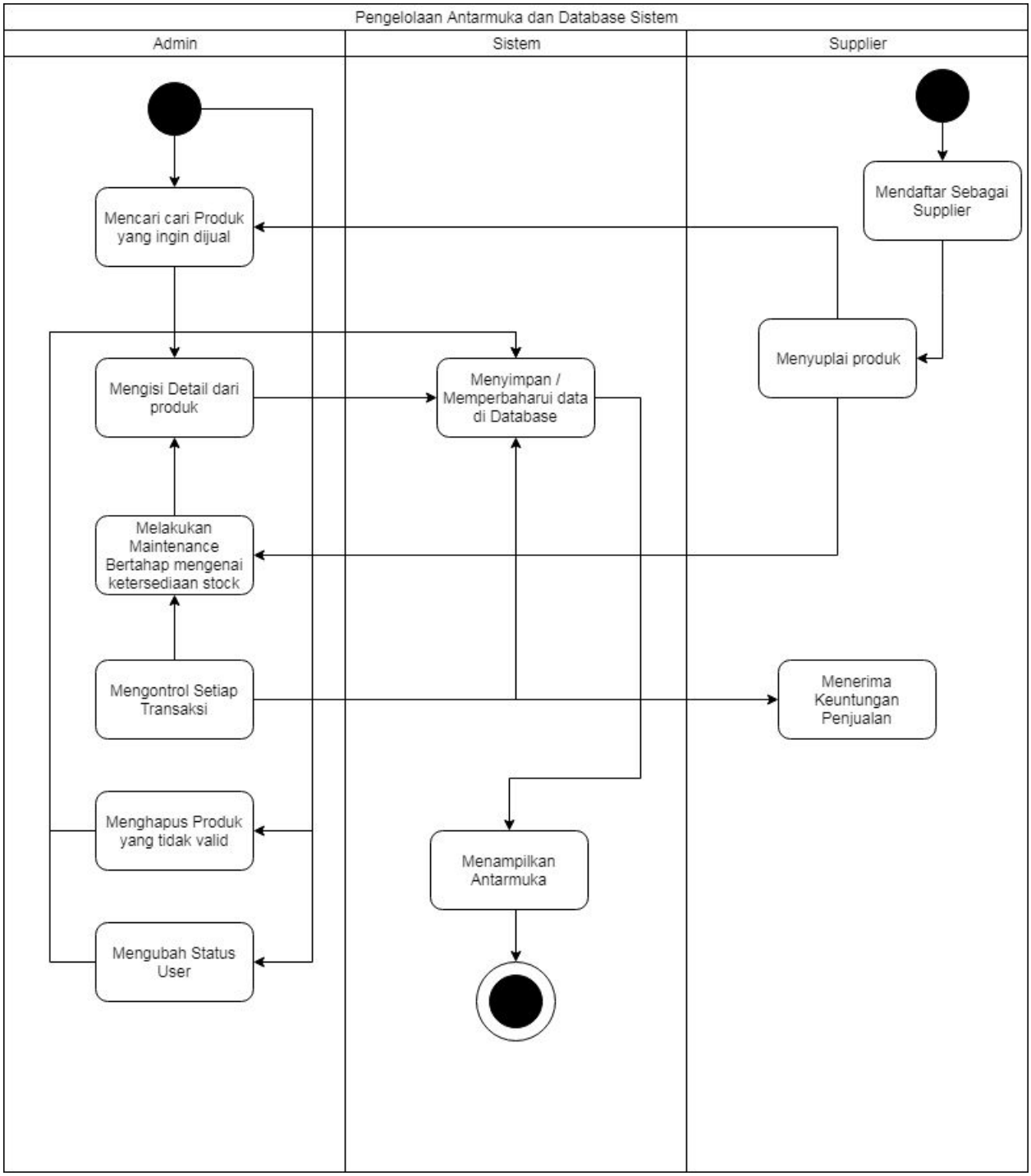
Activity Diagram

Proses bisnis intinya (Proses Transaksi) sebagai berikut :

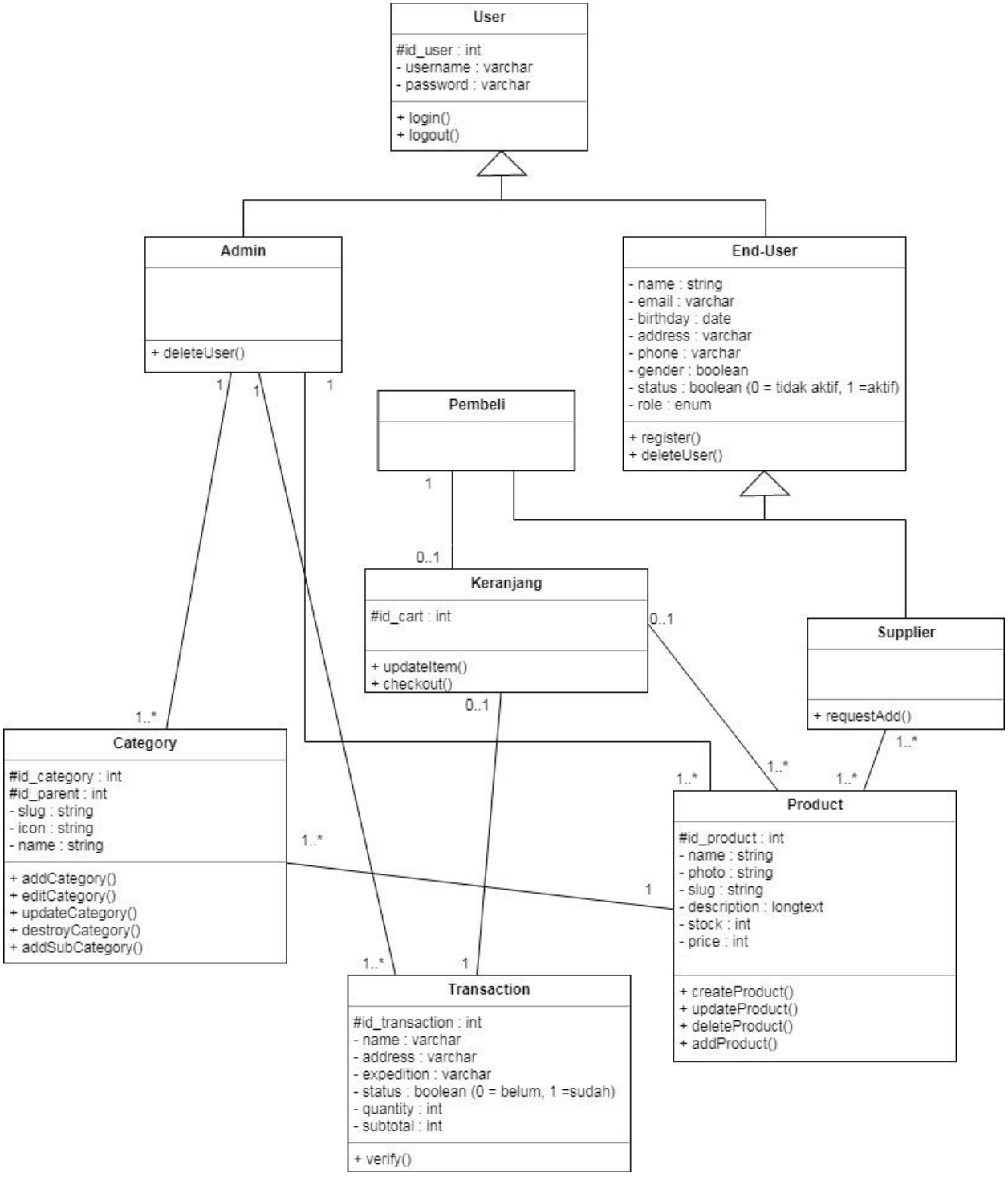


Proses transaksi dimulai saat pembeli melihat-lihat produk yang tersedia, kemudian pembayaran diverifikasi oleh admin, dan transaksi dinyatakan selesai jika barang telah diterima oleh pembeli.

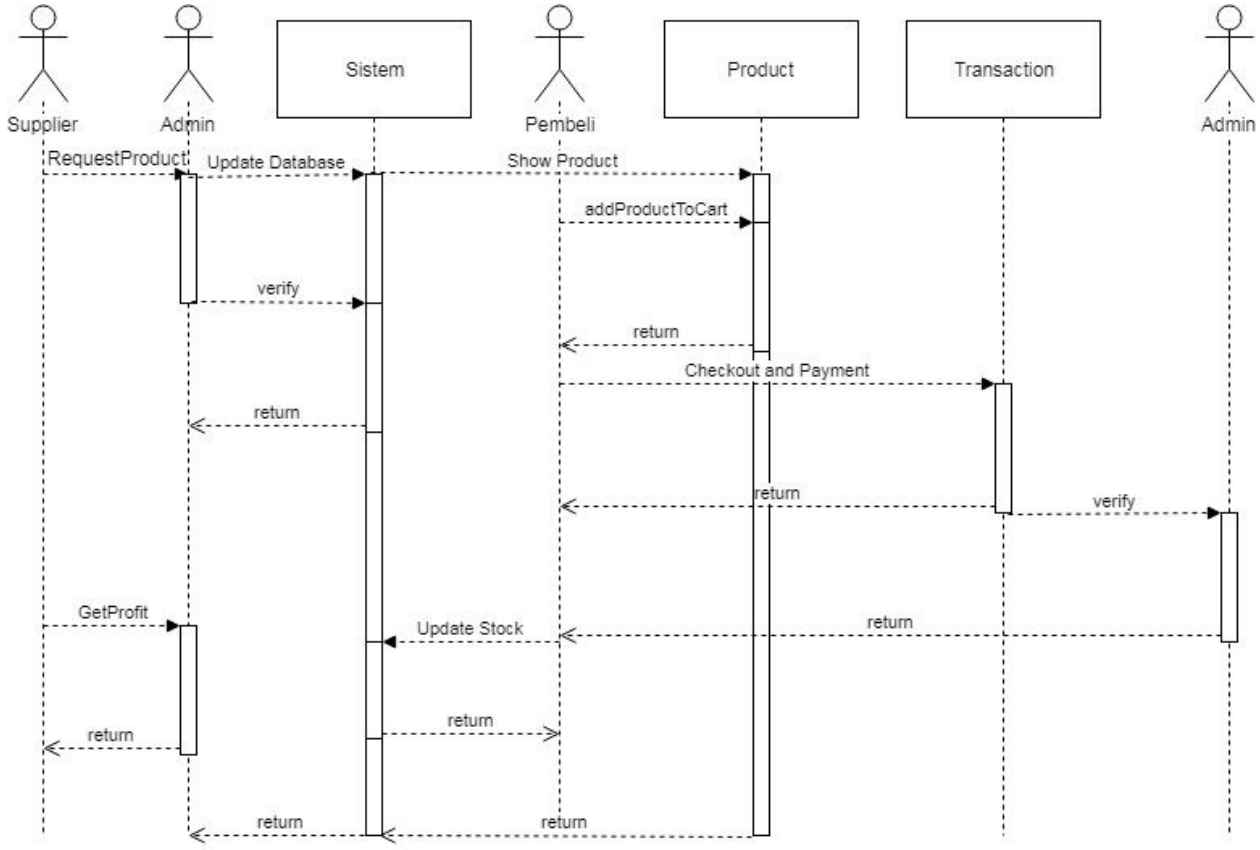
Sementara itu, untuk pengelolaan tampilan dan database dari sistemnya adalah sebagai berikut :



Class Diagram :



Sequence Diagram :



Implementasi Program

Salah satu penerapan OOP pada aplikasi ini adalah ORM, Karena Framework Laravel menyediakan suatu fitur bernama Eloquent yang menerapkan desain ORM. ORM sendiri adalah suatu metode/teknik pemrograman yang digunakan untuk mengkonversi data dari lingkungan bahasa pemrograman berorientasi objek (OOP) dengan lingkungan database relasional, ORM tersebut bertugas sebagai penghubung dan sekaligus mempermudah kita dalam membuat aplikasi yang menggunakan database relasional agar menjadikan tugas kita lebih efisien.

Penerapan tersebut terletak pada controllernya.

```php
class CategoryController extends Controller
{
    public function index() {
        $categorys = Category::where('parent_id',null)->first();
        return view('admin.category.index',compact('categorys'));
    }

    // input category
    public function store(Request $request) {
        $category = new Category;
        $category->slug = $request->slug;
        $category->name = $request->name;
        $category->parent_id = $request->parent_id;
        $category->icon = $request->icon;
        $category->user_id = Auth::user()->id;
        $category->save();
        Alert::success('Success Message', 'Data berhasil ditambah');
        return redirect('admin/category');
    }

    // edit category
    public function edit(Request $request,$id) {
        $category = Category::find($id);
        $categorys = Category::where('parent_id',null)->get();
        return view('admin.category.edit', compact('category','categorys'));
    }

    public function update(Request $request,$id) {
        $category = Category::find($id);
        $category->slug = $request->slug;
        $category->name = $request->name;
        $category->parent_id = $request->parent_id;
        $category->icon = $request->icon;
        $category->save();
        Alert::success('Success Message', 'Data berhasil diperbaharui');
        return redirect(route('category.index'));
    }

    public function destroy($id) {
        $category = Category::find($id);
        $category->delete();
        Alert::success('Success Message', 'Data berhasil dihapus');
        return redirect(route('category.index'));
    }
}
```

Diterapkan pula abstraksi kelas category dan enkapsulasi pada potongan kode tersebut.

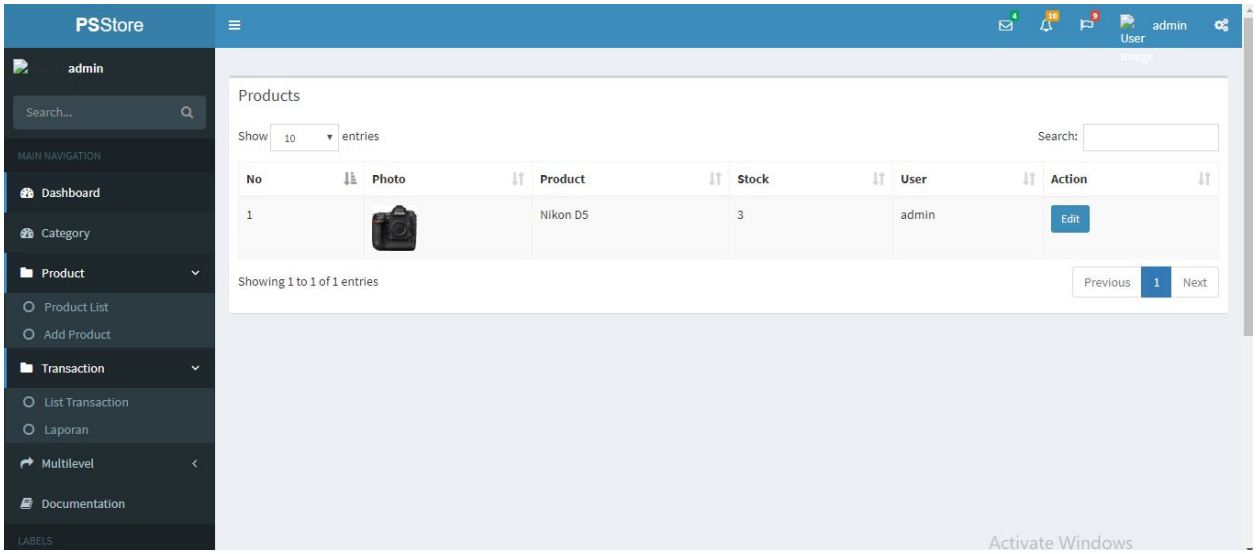
Ada juga penerapan inheritance pada bagian front endnya yaitu menggunakan fitur *extends* dan *section* yang berguna untuk menggunakan template ke tempat lain (Penggunaan sebuah objek untuk objek turunannya)

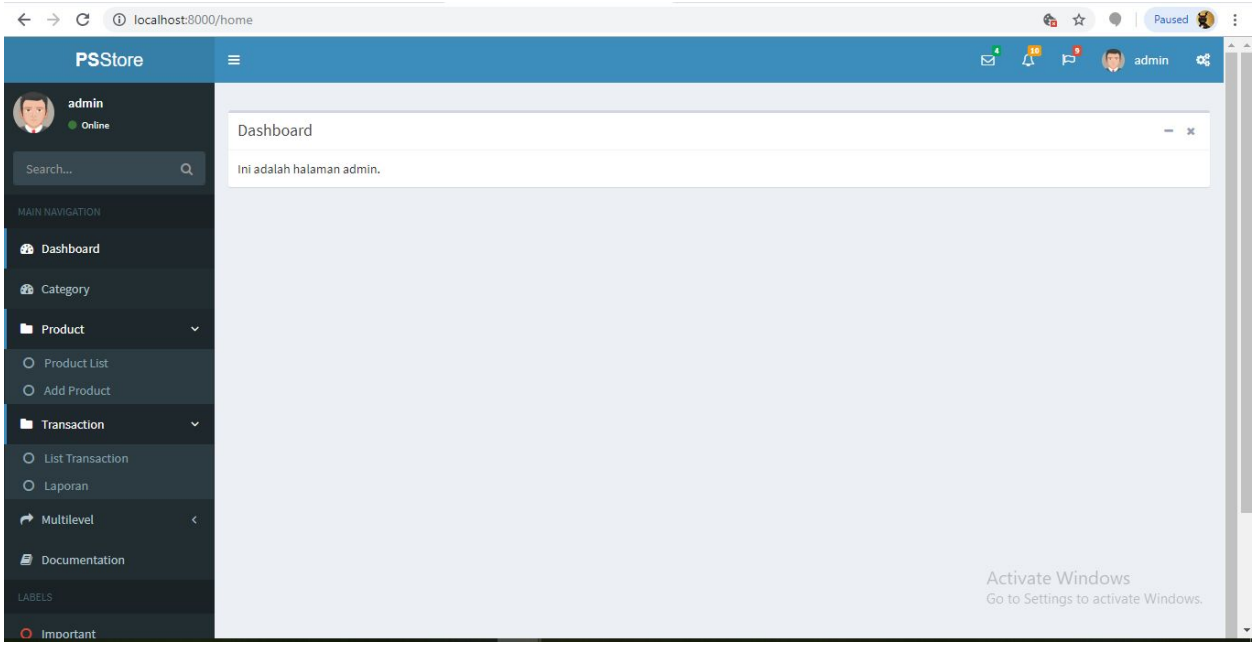
```
@extends('admin.layout.master')
    @section('header')
        <!-- DataTables -->
  <link rel="stylesheet" href="{{ asset('static/bower_components/datatables.net-bs/css/dataTables.bootstrap.min.css') }}">
    <style type="text/css">
    .table_list {
      list-style: none;
      padding: 3px;
      margin-left: -30px;
    }
  </style>
  @endsection
    @section('body')
  <div class="row">

    <div class="col-md-12">
      <div class="box">
       <div class="box-header">
            <h3 class="box-title">Products</h3>
        </div>
            <!-- /.box-header -->
       <div class="box-body">
           <table id="example1" class="table table-bordered table-striped">
             <thead>
                <tr>
                  <th>No</th>
                  <th>Photo</th>
                  <th>Product</th>
                  <th>Stock</th>
                  <th>User</th>
                  <th>Action</th>
                </tr>
              </thead>
```

Potongan kode lengkap dapat diakses di : https://github.com/suryojimbo/psbo

Proses Pengujian

Karena aplikasi kami belum rampung, maka yang kami bisa tampilkan hanya pada bagian admin saja. Mohon maaf. Berikut adalah cuplikannya


PSStore
admin
admin
Search...
MAIN NAVIGATION
Dashboard
Category
Product
Product List
Add Product
Transaction
List Transaction
Laporan
Multilevel
Documentation
LABELS
Products
Show 10 entries
Search:
No
Photo
Product
Stock
User
Action
1
Nikon D5
3
admin
Edit
Showing 1 to 1 of 1 entries
Previous
1
Next
Activate Windows


localhost:8000/home
Paused
PSStore
admin
admin
Online
Search...
MAIN NAVIGATION
Dashboard
Category
Product
Product List
Add Product
Transaction
List Transaction
Laporan
Multilevel
Documentation
LABELS
Important
Dashboard
Ini adalah halaman admin.
Activate Windows
Go to Settings to activate Windows.

Saran

Sebagai evaluasi dari sistem yang telah dibuat, ke depannya seluruh proses pembayaran sebaiknya dapat dilakukan dalam sistem tanpa harus verifikasi manual oleh admin toko. Namun terasa hambar berkata demikian jika aplikasinya saja belum rampung, jadi saran dari kami adalah bulatkan niat dan tekad ketika mengembangkan sebuah aplikasi, jangan mudah menyerah ketika bertemu pesan error. Di luar sana masih banyak developer yang terombang-ambing menghadapi error juga, dan lagi pula ada yang namanya *stackoverflow* yang menjadi teman baik anda ketika berhadapan dengan kode yang error (walaupun metode ini rentan menyebabkan anti-pattern namun tidak apa karena *learning can be on the hard ways*).

Selain itu, terapkan juga *clean code* pada kodingannya agar terdokumentasi dengan baik dan rapi. Disarankan juga untuk tidak menerapkan sistem kebut semalam layaknya membuat candi, karena kami sudah mencobanya dan gagal. Tidak baik juga untuk kesehatan.

Lampiran

Design Pattern

1. Data Access Pattern yang diterapkan adalah **ORM Patterns**. Sebuah fitur bawaan dari laravel yang sudah dijelaskan seperti diatas.
2. Terdapat **Singleton** atau adanya suatu kelas yang berisi hanya satu instances saja, yaitu admin

Anti Pattern

1. Ketiga dari kami mungkin merasakan **Death March**, adalah perasaan yang dirasa ditakdirkan untuk gagal, atau yang membutuhkan kerja keras yang tidak berkelanjutan, atau perasaan “Everyone knows that the project is going to be a disaster”. Dalam manejemen proyek, perasaan tipikal ini biasanya merupakan hasil dari harapan yang tidak realistis atau terlalu optimis dalam scheduling, membuat fitur mewah, atau keduanya, dan sering kali termasuk kurangnya dokumentasi yang sesuai atau pelatihan yang relevan dan keahlian dari luar yang akan diperlukan untuk menyelesaikan tugasnya secara baik. Pengetahuan tentang sifat terkutuk dari proyek ini sangat membebani jiwa para pesertanya, seolah-olah mereka tidak berdaya menyaksikan diri mereka sendiri dan rekan kerja mereka dipaksa untuk menyiksa diri mereka sendiri dan berbaris menuju kematian. Seringkali, pawai kematian akan melibatkan upaya putus asa untuk memperbaiki jalannya proyek dengan meminta anggota tim untuk bekerja terutama jam yang melelahkan (14 jam sehari, 7 hari minggu, dll) atau dengan mencoba "melempar (cukup) mayat ke masalah ", sering menyebabkan kelelahan (*burnout*). (dikutip dari wikipedia**)**
2. Karena terjadi banyak pemangkasan pada fitur, terjadi **Smoke and Mirrors** yaitu sebuah fitur yang dijanjikan dari awal namun tidak jadi pada akhirnya.