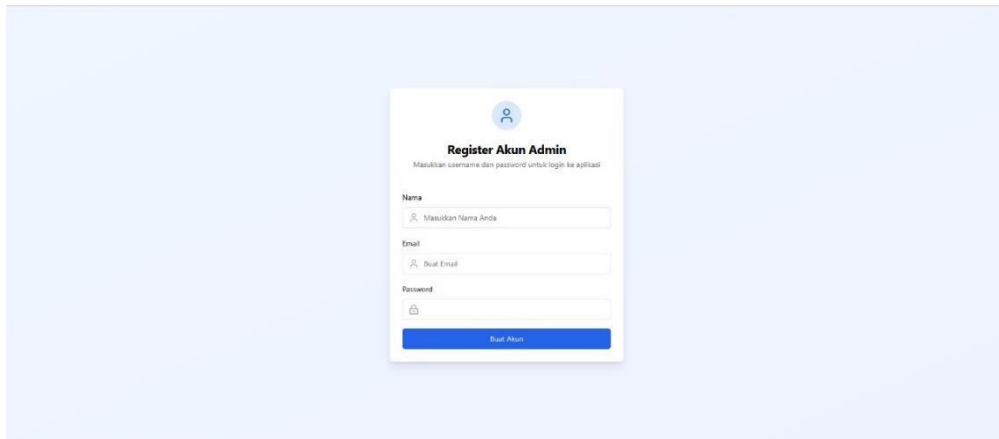


LAMPIRAN

A. Lampiran Pendukung Penelitian

1. Hasil Aplikasi Website

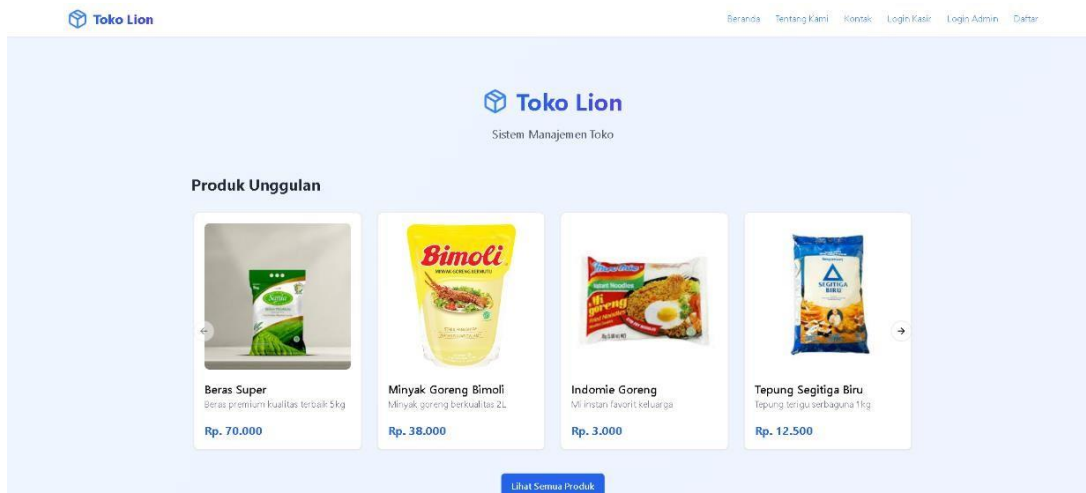
Untuk mengunjungi website Toko Lion dapat menggunakan link berikut ini:
<https://tokolion.my.id/> .



Lampiran 1

Register Akun Admin

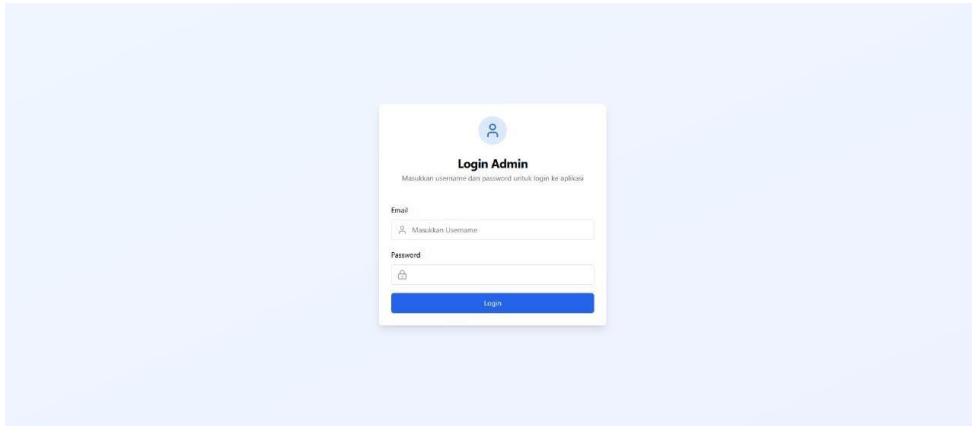
Sumber: Rancangan Penulis.



Lampiran 2

Tampilan Landing Page Atau Halaman Utama

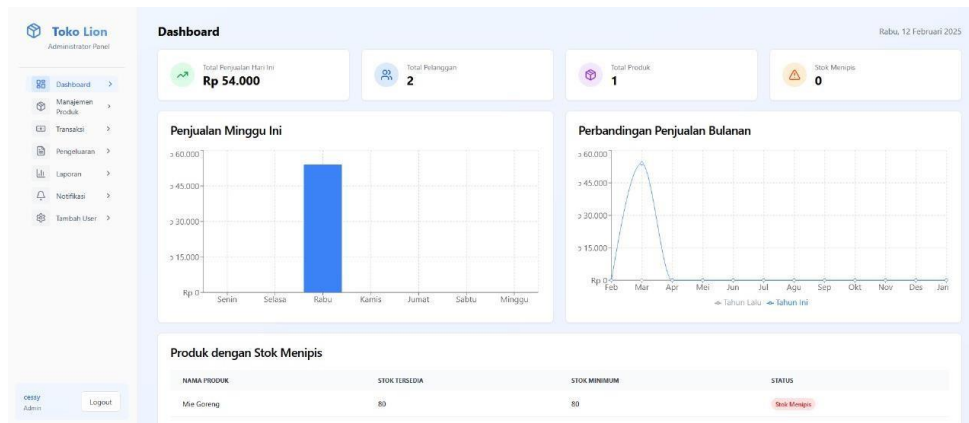
Sumber: Rancangan Penulis.



Lampiran 3

Tampilan *Login Admin*

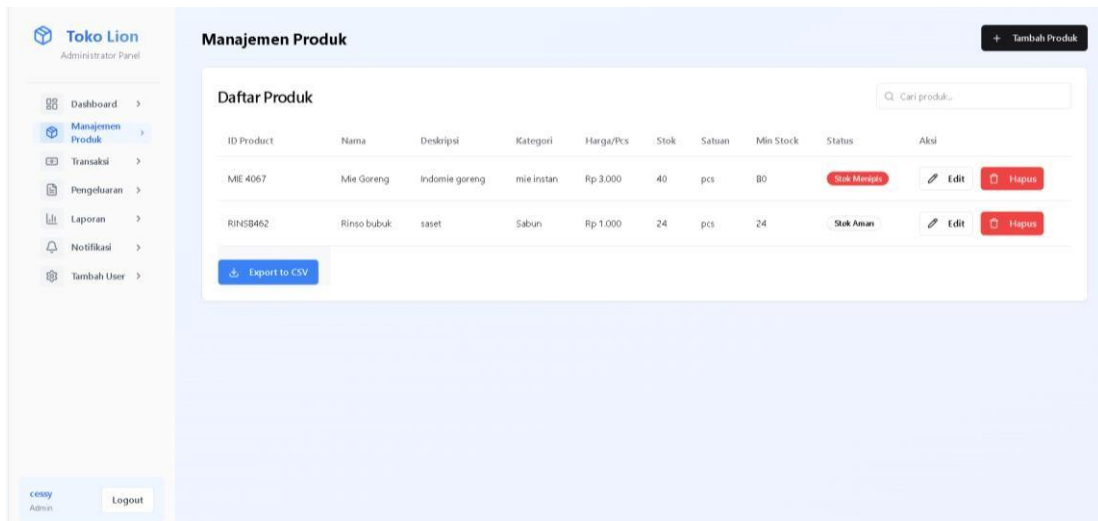
Sumber: Rancangan Penulis.



Lampiran 4

Tampilan Dashboard Admin

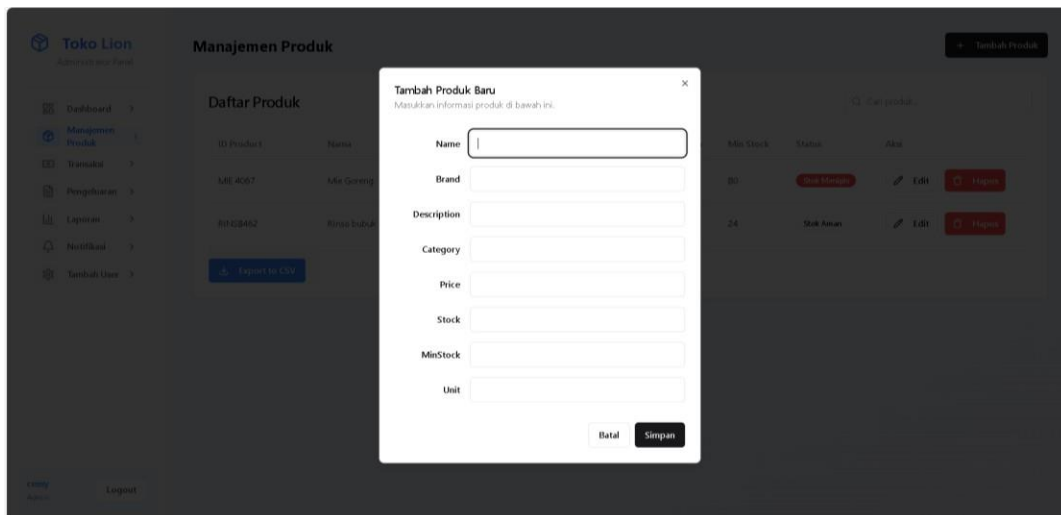
Sumber: Rancangan Penulis.



Lampiran 5

Tampilan Manajemen Produk yang Dikelola Admin

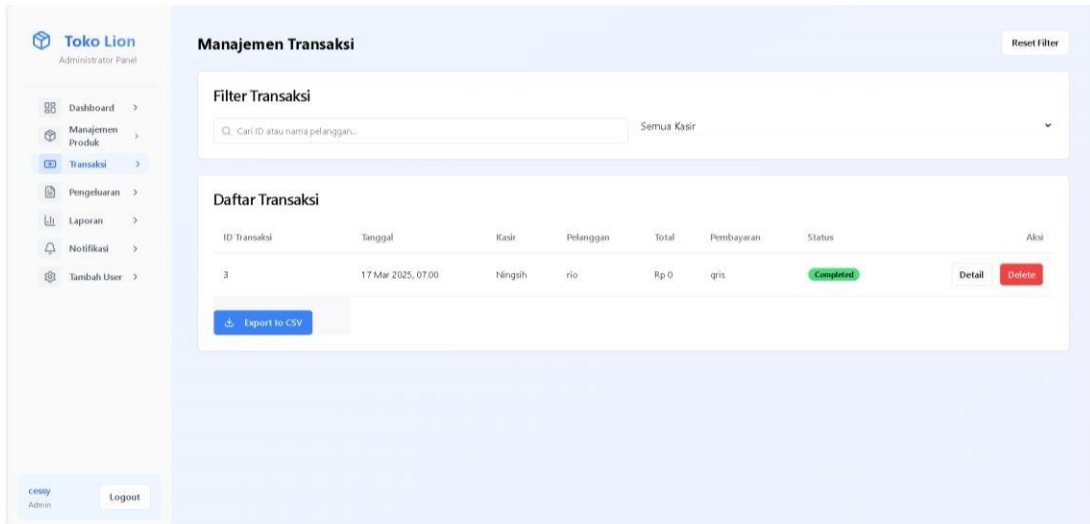
Sumber: Rancangan Penulis.



Lampiran 6

Tampilan Penambahan Produk yang Dikelola Admin

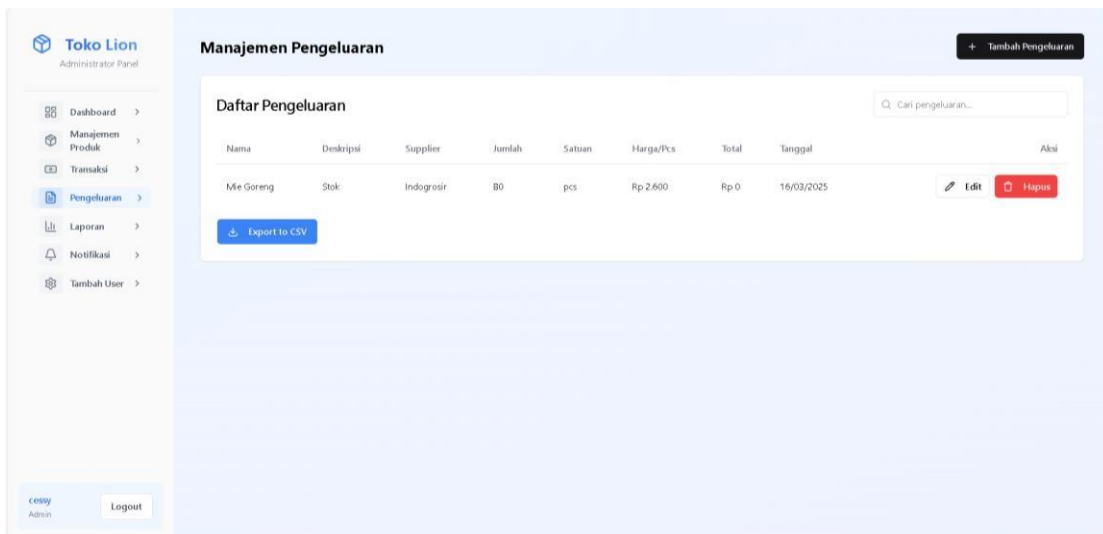
Sumber: Rancangan Penulis.



Lampiran 7

Tampilan Manajemen Transaksi yang Dikelola Admin

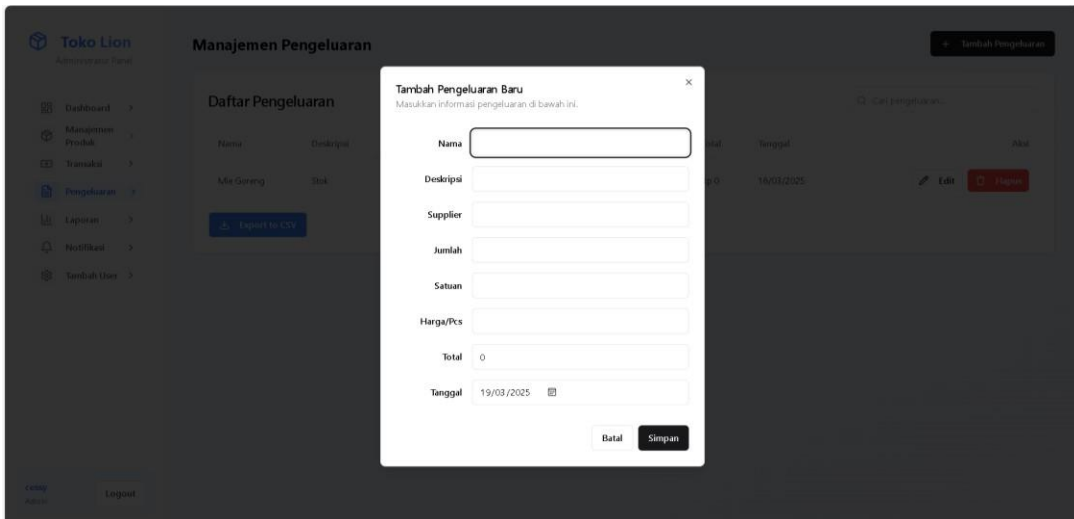
Sumber: Rancangan Penulis.



Lampiran 8

Tampilan Manajemen Pengerluan yang Dikelola Admin

Sumber: Rancangan Penulis.



Lampiran 9

Tampilan Pencatatan Pengeluaran yang Dikelola Admin

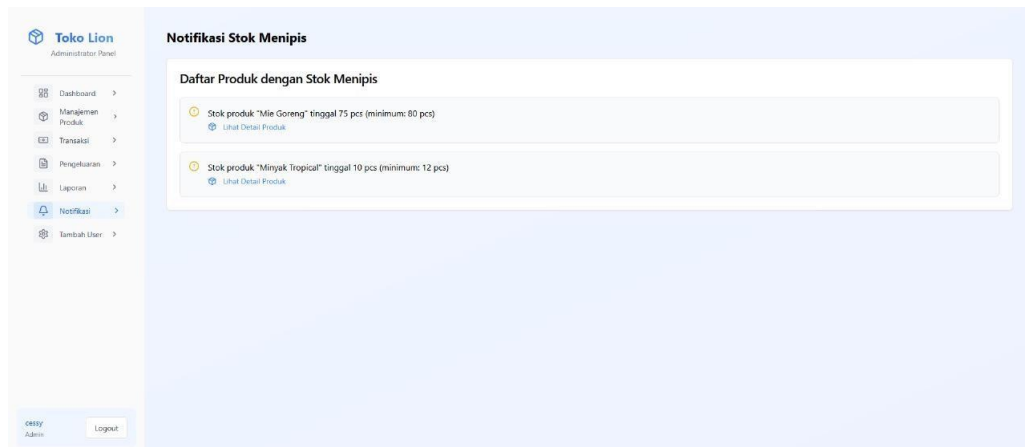
Sumber: Rancangan Penulis.



Lampiran 10

Tampilan Laporan Dalam Bentuk Bar Chart dan Pie Chart

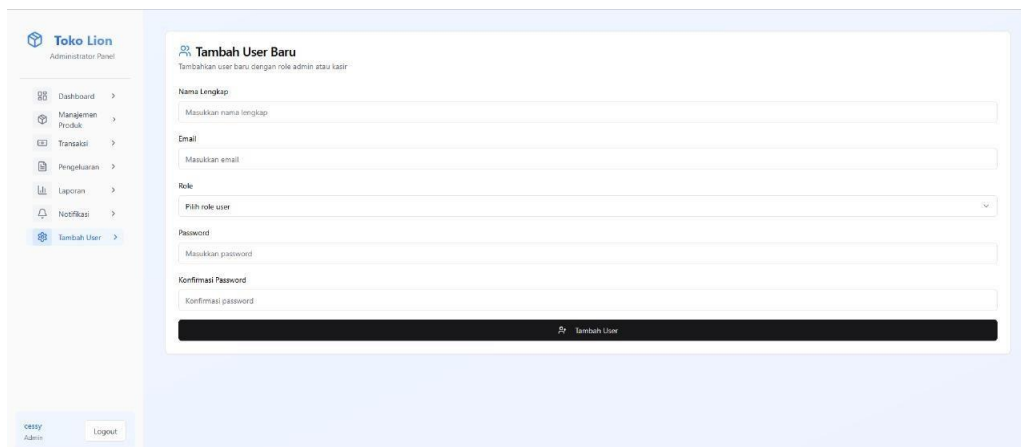
Sumber: Rancangan Penulis.



Lampiran 11

Tampilan Menu Notifikasi Stok Menipis yang Dikelola Admin

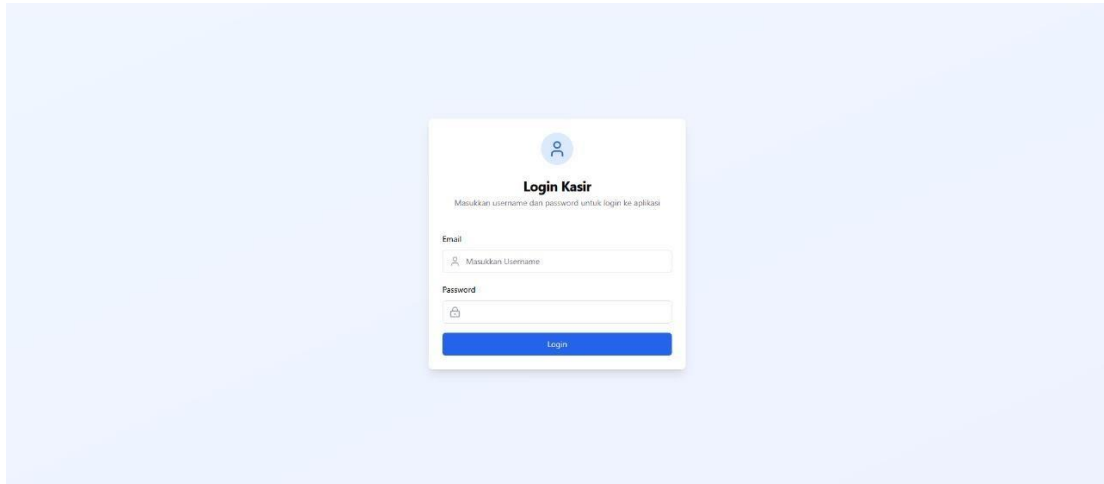
Sumber: Rancangan Penulis.



Lampiran 12

Tampilan Menambahkan User Baru yang Dikelola Admin

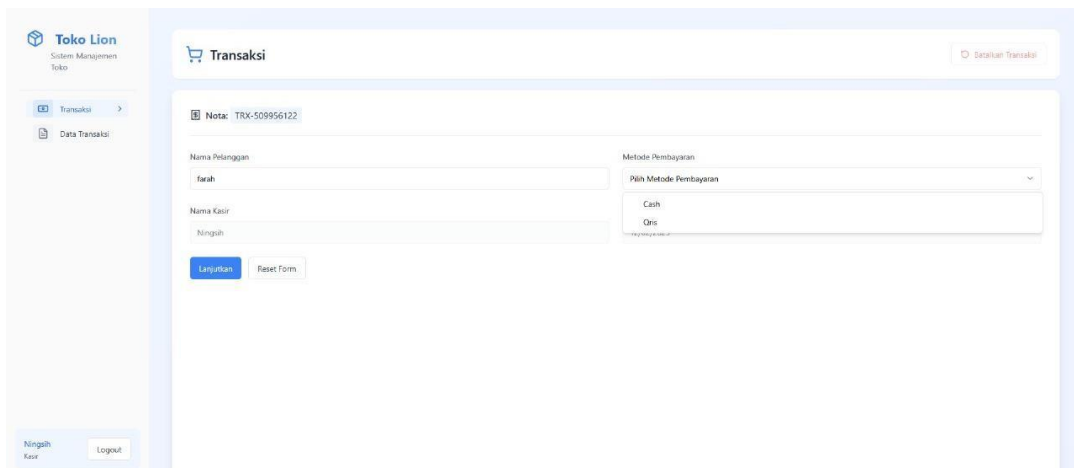
Sumber: Rancangan Penulis.



Lampiran 13

Tampilan Login Kasir

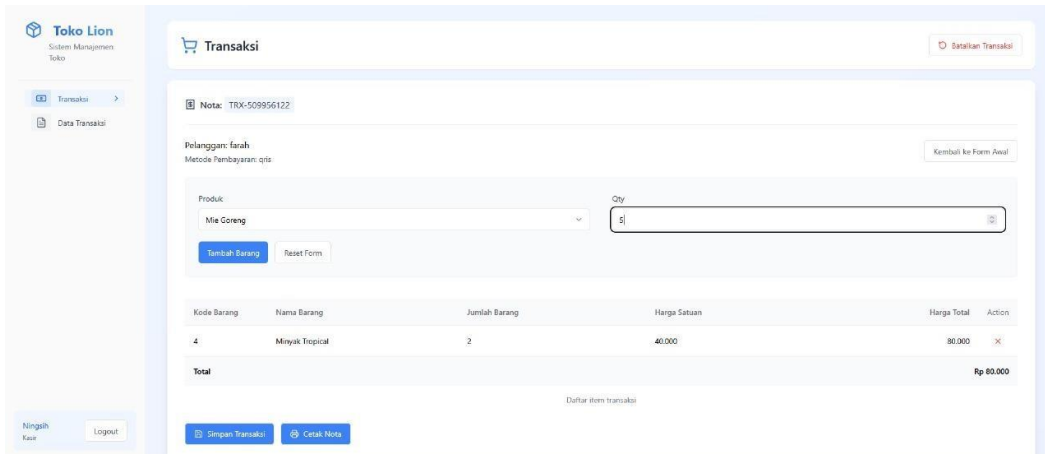
Sumber: Rancangan Penulis.



Lampiran 14

Tampilan Menu Transaksi yang Dikelola Kasir

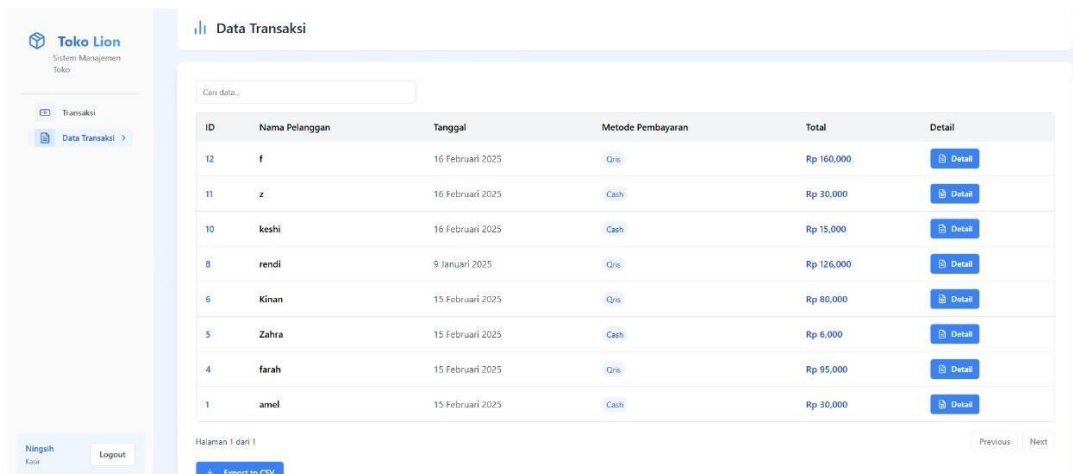
Sumber: Rancangan Penulis.



Lampiran 15

Tampilan Pemasukan Data Transaksi yang Dikelola Kasir

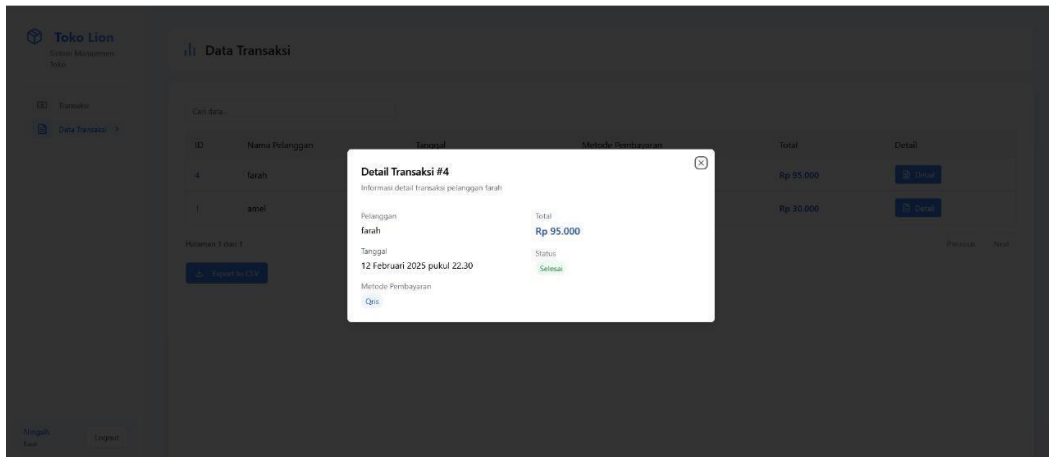
Sumber: Rancangan Penulis.



Lampiran 16

Tampilan Menu Riwayat Transaksi yang Dikelola Kasir

Sumber: Rancangan Penulis.



Lampiran 17

Tampilan Detail Riwayat Transaksi yang Dikelola Kasir

Sumber: Rancangan Penulis.

2. Database

a. Code Prisma Pada Visual Studio Code

```
// This is your Prisma schema file,
// learn more about it in the docs: https://pris.ly/d/prisma-schema

// Looking for ways to speed up your queries, or scale easily with your
// serverless or edge functions?
// Try Prisma Accelerate: https://pris.ly/cli/accelerate-init

generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}

model User {
  id      Int      @id @default(autoincrement())
  name    String
  email   String   @unique
  password String
  role    Role     @default(KASIR)
  createdAt DateTime @default(now())
  Expenses Expense[]
  Sales   Sale[]
}
```

```

model Product {
  id      Int    @id @default(autoincrement())
  name    String
  description String?
  category String
  price   Float
  stock   Int
  minStock Int    @default(10)
  unit    String
  createdAt DateTime @default(now())

  Sales SaleDetail[]
}

model Customer {
  id      Int    @id @default(autoincrement())
  name    String
  contact String?
  email   String
  address String
  createdAt DateTime @default(now())
}

model Sale {
  id      Int    @id @default(autoincrement())
  userId   Int
  customer String
  totalAmount Int?
  paymentMethod String
  date     DateTime @default(now())
  createdAt DateTime @default(now())

  user User    @relation(fields: [userId], references: [id])
  details SaleDetail[]
}

model SaleDetail {
  id      Int    @id @default(autoincrement())
  quantity Int
  totalPrice Int
  productId Int
  saleId   Int
  createdAt DateTime @default(now())

  product Product @relation(fields: [productId], references: [id])
  sale Sale    @relation(fields: [saleId], references: [id])
}

model Expense {
  id      Int    @id @default(autoincrement())

```

```

description String
amount Float
date DateTime
userId Int
createdAt DateTime @default(now())

```

```

user User @relation(fields: [userId], references: [id])
}

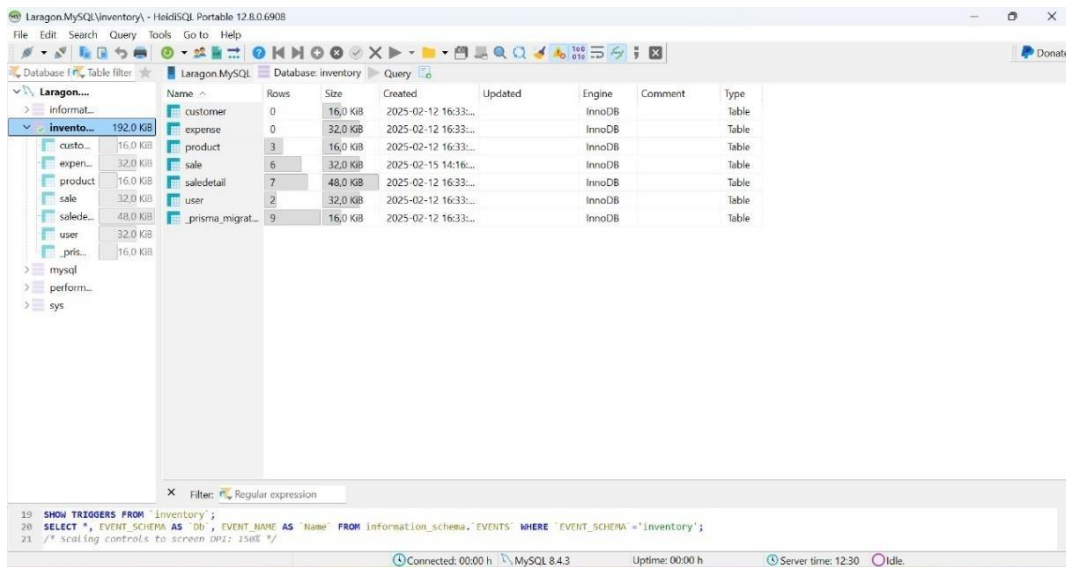
```

```

enum Role {
ADMIN
KASIR
}

```

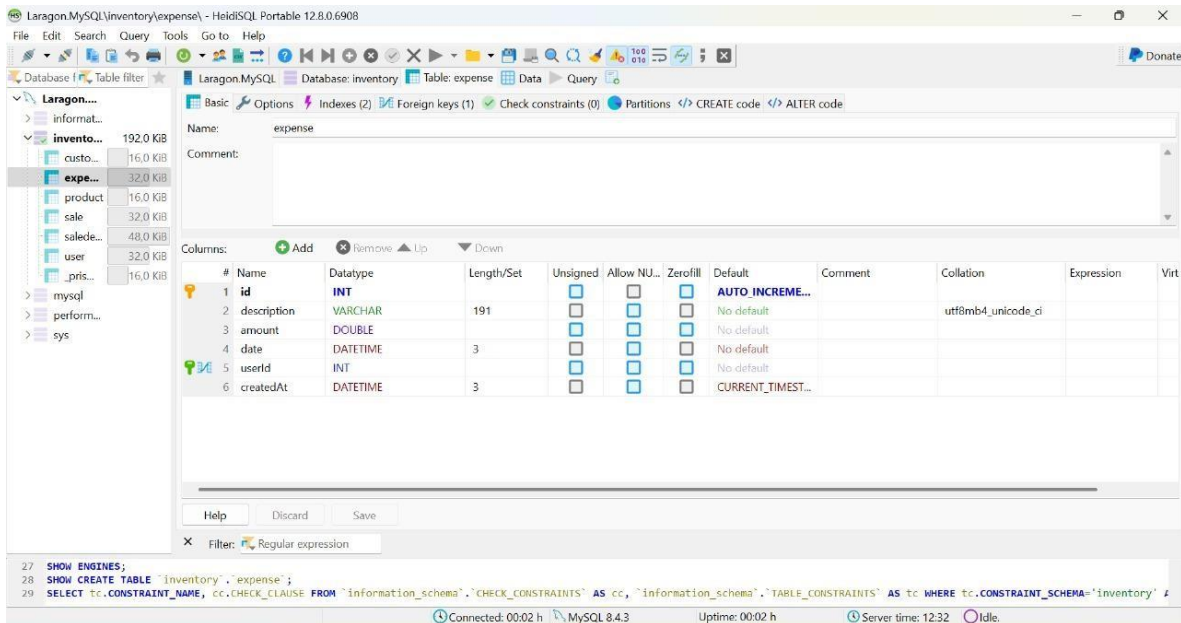
b. Gambar Database Pada Laragon



Lampiran 18

Tampilan Database yang Digunakan atau Inventory

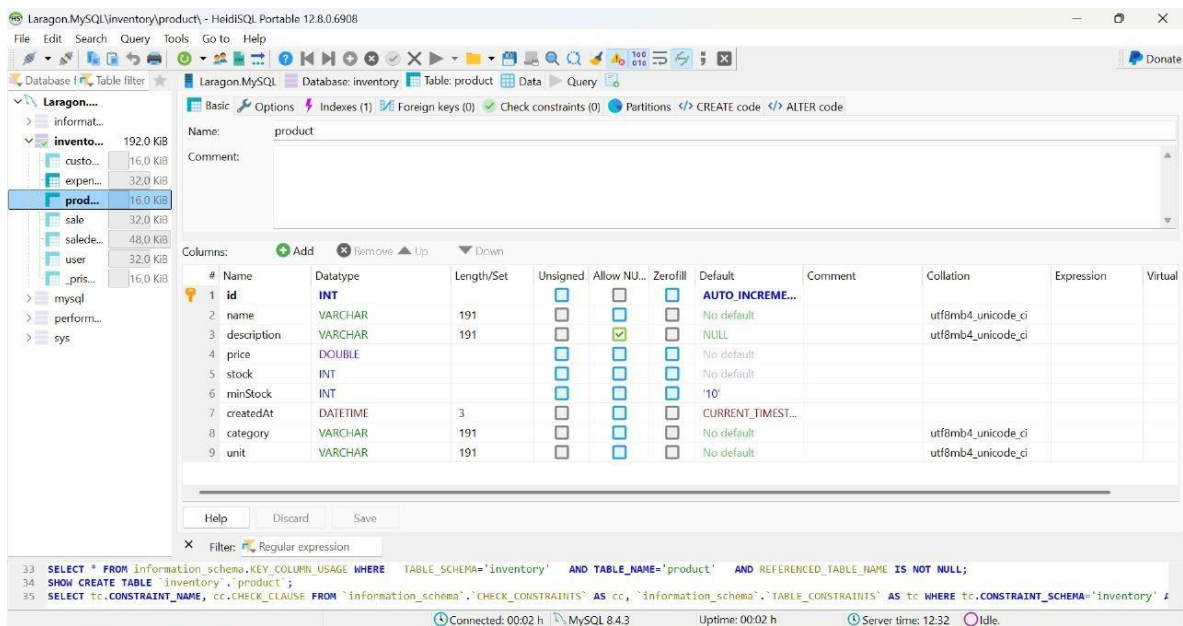
Sumber: Rancangan Penulis.



Lampiran 19

Tampilan Database Pengeluaran

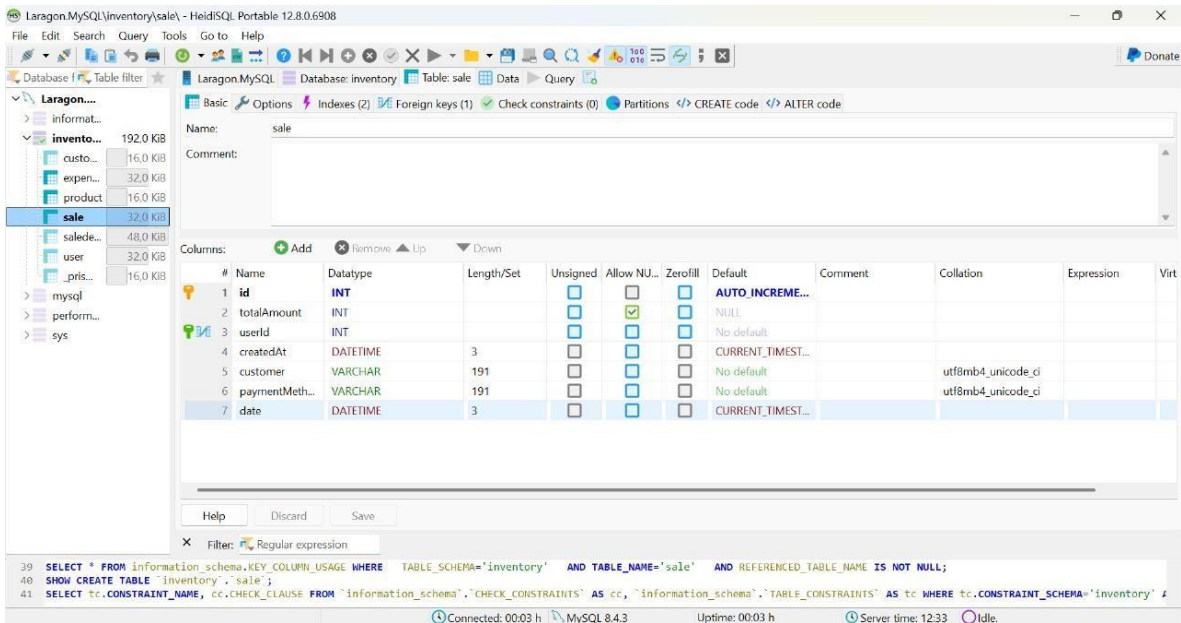
Sumber: Rancangan Penulis.



Lampiran 20

Tampilan Database Produk

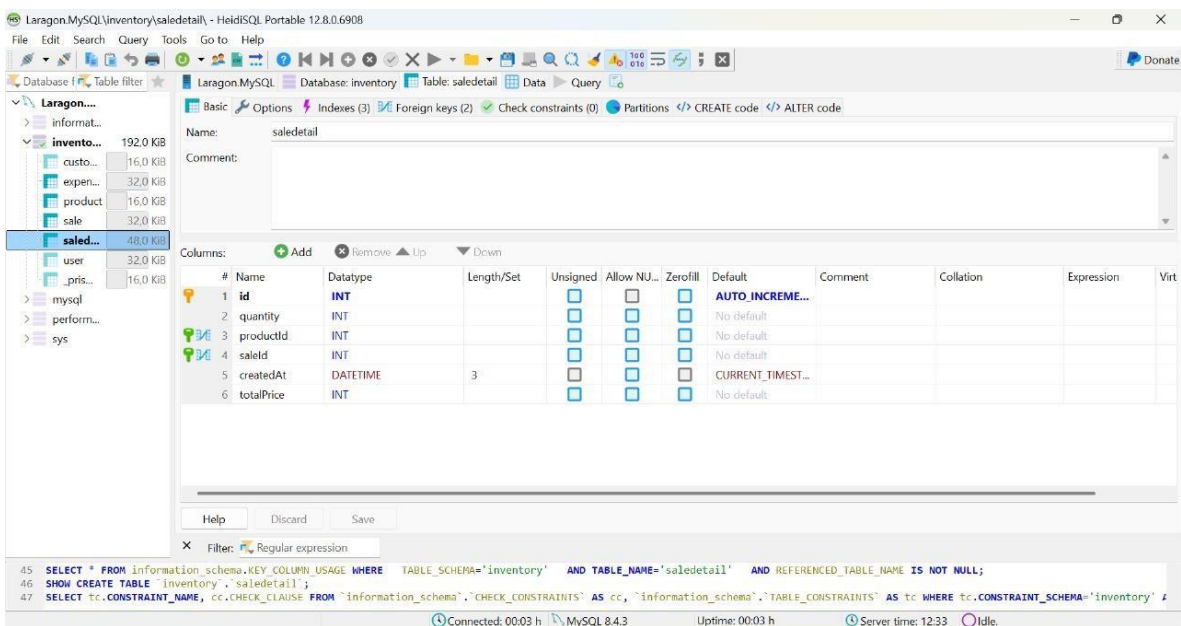
Sumber: Rancangan Penulis.



Lampiran 21

Tampilan Database Penjualan

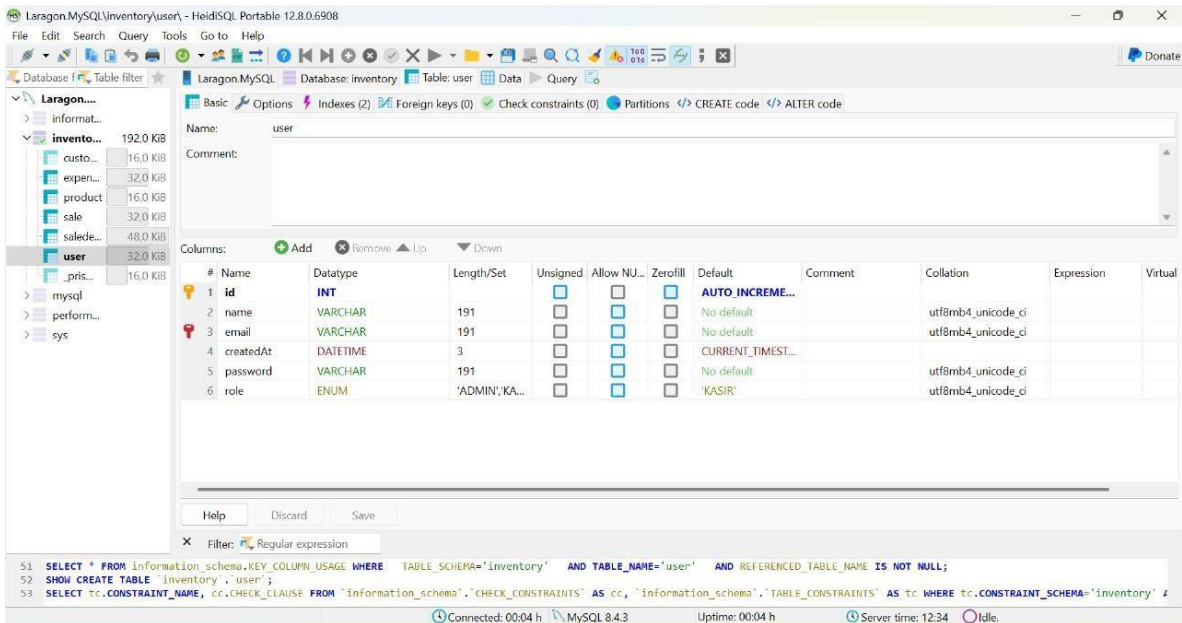
Sumber: Rancangan Penulis.



Lampiran 22

Tampilan Database Detail Penjualan

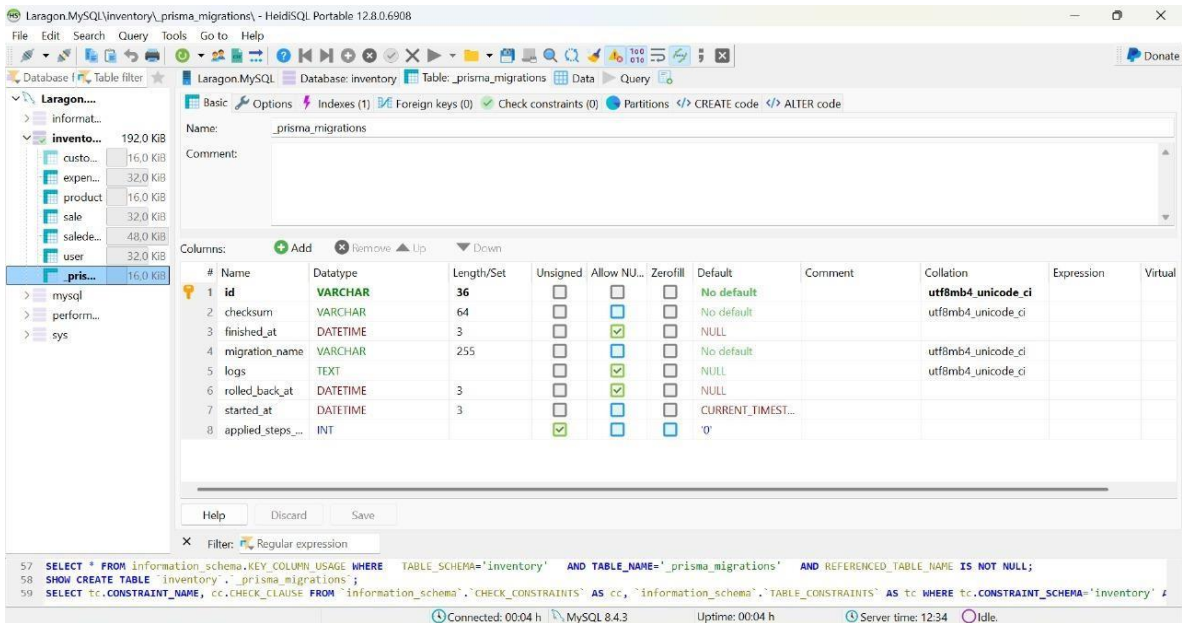
Sumber: Rancangan Penulis.



Lampiran 23

Tampilan Database Pengguna

Sumber: Rancangan Penulis.



Lampiran 24

Tampilan Database Prisma

Sumber: Rancangan Penulis.

3. Code Program

a. Frontend

(1) Landing.Jsx

```

import React from "react";
import { Package, ShoppingCart, Users } from "lucide-react";
import { Card } from "@components/ui/card";
import { Button } from "@components/ui/button";
import { useNavigate } from "react-router-dom";

export default function Landing() {
  const navigate = useNavigate();

  return (
    <div className="min-h-screen bg-gradient-to-br from-blue-50 to-indigo-50 flex flex-col items-center justify-center p-6">
      <div className="text-center mb-12">
        <div className="flex items-center justify-center gap-3 mb-4">
          <Package className="w-10 h-10 text-blue-500" />
          <h1 className="text-4xl font-bold bg-gradient-to-r from-blue-600 to-indigo-600 bg-clip-text text-transparent">
            Toko Lion
          </h1>
        </div>
        <p className="text-gray-600 text-lg">Sistem
          Manajemen Toko</p>
      </div>

      <div className="flex flex-col md:flex-row gap-6 w-full max-w-4xl justify-center items-stretch">
        <Card className="flex-1 group hover:shadow-xl transition-all duration-300 overflow-hidden">
          <div className="p-8 flex flex-col items-center h-full border-2 border-transparent hover:border-blue-500 rounded-lg">
            <div className="p-4 bg-blue-100 rounded-full mb-6 group-hover:bg-blue-200 transition-colors">
              <ShoppingCart className="w-8 h-8 text-blue-600" />
            </div>
            <h2 className="text-2xl font-semibold mb-4 text-gray-800">
              Login Sebagai Kasir
            </h2>
            <p className="text-gray-600 mb-8 text-center">
              Akses sistem kasir untuk mengelola transaksi dan
              penjualan harian
            </p>
            <Button
              className="w-full mt-auto bg-blue-600 hover:bg-blue-700"
              onClick={() => navigate("/user/login")}
            >
              Masuk sebagai Kasir
          </div>
        </Card>
      </div>
    </div>
  );
}

```

```

        </Button>
      </div>
    </Card>

    <Card className="flex-1 group hover:shadow-xl
transition-all duration-300 overflow-hidden">
      <div className="p-8 flex flex-col items-center h-full
border-2 border-transparent hover:border-indigo-500 rounded-
lg">
        <div className="p-4 bg-indigo-100 rounded-full mb-6
group-hover:bg-indigo-200 transition-colors">
          <Users className="w-8 h-8 text-indigo-600" />
        </div>
        <h2 className="text-2xl font-semibold mb-4 text-
gray-800">
          Login Sebagai Admin
        </h2>
        <p className="text-gray-600 mb-8 text-center">
          Kelola inventori, laporan, dan pengaturan sistem toko
        </p>
        <Button
          className="w-full mt-auto bg-indigo-600 hover:bg-
indigo-700"
          onClick={() => navigate("/admin/login")}
        >
          Masuk sebagai Admin
        </Button>
      </div>
    </Card>
  </div>

  <footer className="mt-16 text-center text-gray-500">
    <p className="text-sm">© 2025 Toko Lion. All rights
reserved.</p>
  </footer>
</div>
);
}

```

(2) *Register.Jsx*

```

import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardHeader,
  CardTitle,
} from "@components/ui/card";
import { Input } from "@components/ui/input";
import { Label } from "@components/ui/label";

```

```

import { useNavigate } from "react-router-dom";
import { LockKeyhole, User } from "lucide-react";
import axios from "axios";
import Swal from "sweetalert2";
import { useState } from "react";
import Table from "@components/component/Table.jsx";

export default function Register() {
  const navigate = useNavigate();
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleRegister = async (e) => {
    e.preventDefault();

    // Validation
    if (!name || !email || !password) {
      Swal.fire({
        icon: "error",
        title: "Oops...",
        text: "Semua field harus diisi!",
      });
      return;
    }

    // Email validation
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(email)) {
      Swal.fire({
        icon: "error",
        title: "Format Email Salah",
        text: "Mohon masukkan email yang valid",
      });
      return;
    }

    try {
      const response = await
      axios.post("http://localhost:3000/api/users/", {
        name,
        email,
        password,
        role: "ADMIN", // Role is always set to ADMIN
      });

      if (response.data) {
        await Swal.fire({
          icon: "success",
          title: "Registrasi Berhasil!",
        });
      }
    }
  };
}

```

```

        text: "Akun admin berhasil dibuat",
    });
    navigate("/"); // Redirect to login page after successful
registration
    }
} catch (error) {
    console.error("Registration error:", error);

    let errorMessage = "Terjadi kesalahan saat registrasi";
    if (error.response) {
        // Use error message from API if available
        errorMessage = error.response.data.message ||
errorMessage;
    }

    Swal.fire({
        icon: "error",
        title: "Registrasi Gagal",
        text: errorMessage,
    });
}
};

return (
    <div className="flex justify-center min-h-screen items-
center bg-gradient-to-br from-blue-50 to-indigo-50 p-4">
        <div className="w-full max-w-md">
            <Card className="border-none shadow-lg">
                <CardHeader className="space-y-1">
                    <div className="flex justify-center mb-4">
                        <div className="p-3 rounded-full bg-blue-100">
                            <User size={32} className="text-blue-600" />
                        </div>
                    </div>
                    <CardTitle className="text-2xl text-center font-bold">
                        Register Akun Admin
                    </CardTitle>
                    <CardDescription className="text-center">
                        Masukkan username dan password untuk login ke
aplikasi
                    </CardDescription>
                </CardHeader>
                <CardContent>
                    <form
                        className="mt-4"
onSubmit={handleRegister}>
                        <div className="flex flex-col gap-4">
                            <div className="space-y-2">
                                <Label htmlFor="name" className="text-sm font-
medium">
                                    Nama

```

```

        </Label>
        <div className="relative">
          <User className="absolute left-3 top-2.5 h-5 w-5
text-gray-400" />
          <Input
            id="name"
            type="text"
            placeholder="Masukkan Nama Anda"
            className="pl-10"
            required
            value={name}
            onChange={(e) => setName(e.target.value)}
          />
        </div>
      </div>
    <div className="space-y-2">
      <Label htmlFor="email" className="text-sm font-
medium">
        Email
      </Label>
      <div className="relative">
        <User className="absolute left-3 top-2.5 h-5 w-5
text-gray-400" />
        <Input
          id="email"
          type="email"
          placeholder="Buat Email"
          className="pl-10"
          required
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
      </div>
    </div>
    <div className="space-y-2">
      <Label htmlFor="password" className="text-sm
font-medium">
        Password
      </Label>
      <div className="relative">
        <LockKeyhole className="absolute left-3 top-2.5
h-5 w-5 text-gray-400" />
        <Input
          id="password"
          type="password"
          className="pl-10"
          required
          value={password}
          onChange={(e) => setPassword(e.target.value)}
        />
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
    <Button
      type="submit"
      className="w-full bg-blue-600 hover:bg-blue-700
text-white transition-colors"
    >
      Buat Akun
    </Button>
  </div>
</form>
</CardContent>
</Card>
</div>
</div>
);
}

```

b. *Backend*

(1) *App.Js*

```

const express = require("express");
const app = express();
const cors = require("cors");
app.use(express.json());

const apiRoutes = require("./routes");
app.use(cors());
app.get("/hello", (req, res) => res.send("hello world"));
app.use("/api", apiRoutes);

module.exports = app;

```

(2) *Server.Js*

```

const PORT = process.env.PORT || 3000;
const app = require("./app.js");
const router = require("./routes");

require("dotenv").config();

// app.use("/api", router);

app.listen(PORT, () => {
  console.log("Server is running on port ", PORT);
});

```

4. Foto Toko Lion



Lampiran 25

Foto Toko Lion Dari Luar

Sumber: Dokumentasi Langsung Penulis.



Lampiran 26

Foto Toko Lion Dari Luar

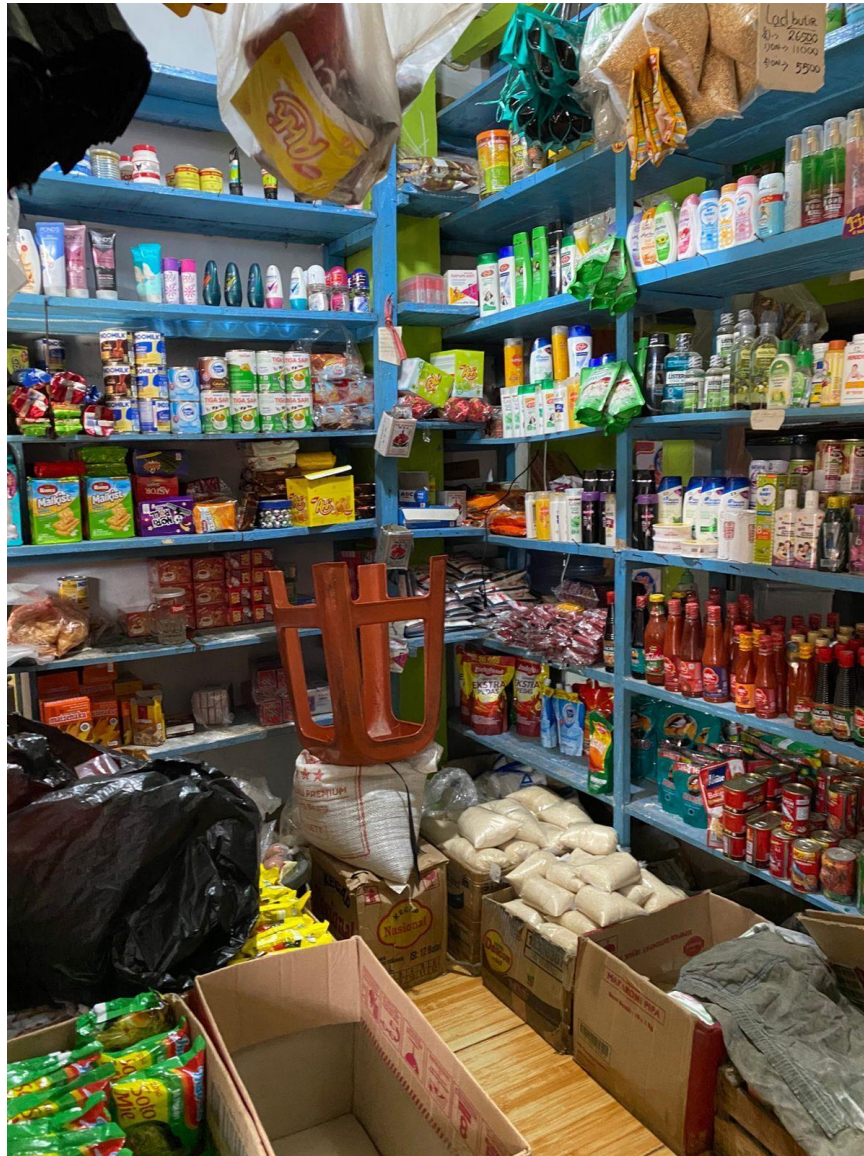
Sumber: Dokumentasi Langsung Penulis.



Lampiran 27

Foto Toko Lion Dari Sisi Pembeli

Sumber: Dokumentasi Langsung Penulis.



Lampiran 28

Foto Bagian Dalam Toko Lion

Sumber: Dokumentasi Langsung Penulis.



Lampiran 29

Foto Bagian Dalam Toko Lion

Sumber: Dokumentasi Langsung Penulis.