

(Kelompok 2)

(Project Aplikasi Toko Lia)

Software Design Document v.1.0

Nama : Alvon Tritara
 David Dwiputranto
 Hasbie Muhammad Irsyad
 Nuramalia Putri Hapsari

Bagian : Designer

Date: (21/11/2023)

TABLE OF CONTENTS

1.	PENDAHULUAN	4
1.1	<i>Tujuan.....</i>	<i>4</i>
1.2	<i>Ruang Lingkup.....</i>	<i>4</i>
1.3	<i>Ikhtisar.....</i>	<i>4</i>
1.4	<i>Referensi</i>	<i>5</i>
1.5	<i>Definisi dan Singkatan.....</i>	<i>5</i>
2.	GAMBARAN UMUM SISTEM	6
3.	ARSITEKTUR SISTEM.....	7
3.1	<i>Rancangan Arsitektur</i>	<i>7</i>
3.2	<i>Deskripsi Dekomposisi</i>	<i>10</i>
3.3	<i>Alasan Perancangan.....</i>	<i>11</i>
4.	RANCANGAN DATA	12
4.1	<i>Deskripsi Data.....</i>	<i>12</i>
4.2	<i>Kamus Data</i>	<i>15</i>
5.	RANCANGAN KOMPONEN	20
6.	RANCANGAN ANTARMUKA	23
6.1	<i>Gambaran Umum Rancangan Antarmuka</i>	<i>23</i>
6.2	<i>Tampilan Layar</i>	<i>24</i>
6.3	<i>Tampilan Layar dan Tindakan</i>	<i>31</i>
7.	MATRIKS PERSYARATAN	34
8.	LAMPIRAN	35

1. PENDAHULUAN

1.1 Tujuan

Dokumen ini dibuat untuk menggambarkan arsitektur dari sistem pada aplikasi Toko Lia yang rinci sehingga dapat digunakan untuk menjadi panduan untuk programmer dalam membangun aplikasi. Pengguna utama dari dokumen ini adalah anggota Kelompok 2, khususnya untuk PIC Programmer.

1.2 Ruang Lingkup

Aplikasi Toko Lia adalah aplikasi yang dibuat dengan tujuan membantu proses pencatatan transaksi harian di Toko Lia. Aplikasi menggunakan dasar dari sistem sebuah aplikasi kasir, dimana transaksi akan disimpan dalam basis data. Kumpulan transaksi yang disimpan kemudian dapat digunakan untuk membuat laporan laba rugi harian/bulanan, pencatatan hutang, hingga informasi perubahan stok barang. Setiap desain yang dibuat pada project ini bersifat khusus dan terpersonalisasi pada kegiatan rutin yang terjadi di Toko Lia, sehingga beberapa desain mungkin hanya dapat berlaku pada project ini dan berlandaskan kemudahan penggunaan untuk rutinitas yang dilakukan pada Toko Lia

1.3 Ikhtisar

Software Design Document (SDD) ini adalah dokumen yang dibuat sebagai panduan untuk anggota kelompok 2 dalam membangun software aplikasi Toko Lia, sehingga setiap proses pembuatan komponen bisa sesuai dengan kebutuhan dan tidak terlewat.

Section	Isi
1 Pendahuluan	1.1 Tujuan 1.2 Ruang Lingkup 1.3 Ikhtisar 1.4 Referensi 1.5 Definisi dan Singkatan
2 Gambaran Umum Sistem	Fitur fungsional, lingkungan implementasi
3 Arsitektur Sistem	3.1 Rancangan Arsitektur 3.2 Deskripsi Dekomposisi 3.3 Alasan Perancangan
4 Rancangan Data	4.1 Deskripsi Data 4.2 Kamus Data
5 Rancangan Komponen	
6 Rancangan Antarmuka	6.1 Gambaran Umum Antarmuka 6.2 Tampilan Layar

	6.3 Objek Layar dan Tindakan
7 Matriks Persyaratan	
8 Lampiran	

Susunan tim pengembang “Kelompok 2” :

Nama	Posisi
Nuramalia Putri Hapsari	Project Manager, System Analyst
Alvon Tritara	UI / UX Designer
Hasbie Muhammad Irsyad	Programmer
David Dwiputranto	Programmer, Documenter

1.4 Referensi

- SDD Template by Dr. Lulu Chaerani M.
- SRS v.1.0 Aplikasi Toko Lia by Kelompok 2
- SDD v.1.9 Aplikasi Busayu 15 April 2018 on github : firstiaulyaa/RPL-D-5

1.5 Definisi dan Singkatan

Istilah	Definisi
DBMS (Database Management System)	Perangkat lunak pengelola basis data
SDK (Software Development Kit)	Package yang berisi fungsi-fungsi pre-programmed yang dapat digunakan untuk membangun sebuah perangkat lunak
SDD (Software Design Document)	Dokumen yang berisi rincian rancangan pembuatan software
SRS (Software Requirement Specification)	Dokumen yang berisi rincian kebutuhan dan spesifikasi yang akan dipenuhi dari software yang akan dibangun
ERD (Entity Relationship Diagram)	Diagram yang menjelaskan hubungan antar entitas (abstraksi dari objek yang memiliki sifat yang dapat dijelaskan), agar dapat dibangun rancangan basis datanya
Use Case Diagram	Diagram yang menggambarkan fungsionalitas / fungsi yang dapat dijalankan oleh aktor (yang berinteraksi dengan software/sistem)

Class Diagram	Diagram yang menjelaskan relasi antara class / objek
BPMN (Business Process Model & Notation)	Representasi grafis dari proses bisnis dalam bentuk notasi yang menjelaskan langkah-langkah
Enkapsulasi	Sifat pemrograman berbasis objek yang dapat menjaga akses ke dalam class/objek dari class/objek yang hanya diberikan hak akses

2. GAMBARAN UMUM SISTEM

Proyek sistem informasi ini akan menghasilkan produk berupa aplikasi berbasis android bernama Aplikasi Toko Lia, dimana aplikasi ini akan difungsikan untuk menggantikan kegiatan bisnis pada Toko Lia yang selama ini pencatatan dilakukan menggunakan tulis tangan dan tidak ada perhitungan keuntungan dari penjualan. Produk aplikasi ini akan difungsikan untuk menggantikan kegiatan-kegiatan berulang yang dapat dilakukan lebih akurat dan cepat dengan dibantu aplikasi sistem informasi.

Fitur fungsional yang dapat digunakan oleh class user pada sistem ini berupa :

No.	Fungsi	Keterangan
1.	Setup Produk	Membuat produk-produk dagangan dengan informasi lengkap harga modal dan harga jual beserta tanggal kadaluwarsa hingga jumlah stok yang tersedia
2.	Update Stok	Merubah stok dari setiap produk (menambahkan atau mengurangi), sebagai bentuk penanggulangan kesalahan input data ataupun kegiatan restok rutin
3.	Pencarian	Fitur pencarian produk dengan kata kunci berdasarkan nama
4.	Fitur Kasir	Fitur untuk membuat transaksi dan menyatakan transaksi tersebut sebagai pembelian, hutang, penggunaan pribadi, ataupun dibuang karena melewati masa kadaluwarsa
5.	Kasbon	Fitur untuk menambahkan transaksi ke tab kasbon pelanggan untuk mencatat total transaksi yang belum terbayarkan oleh pelanggan

Dalam pembuatan perangkat lunak ini, dibuat menggunakan :

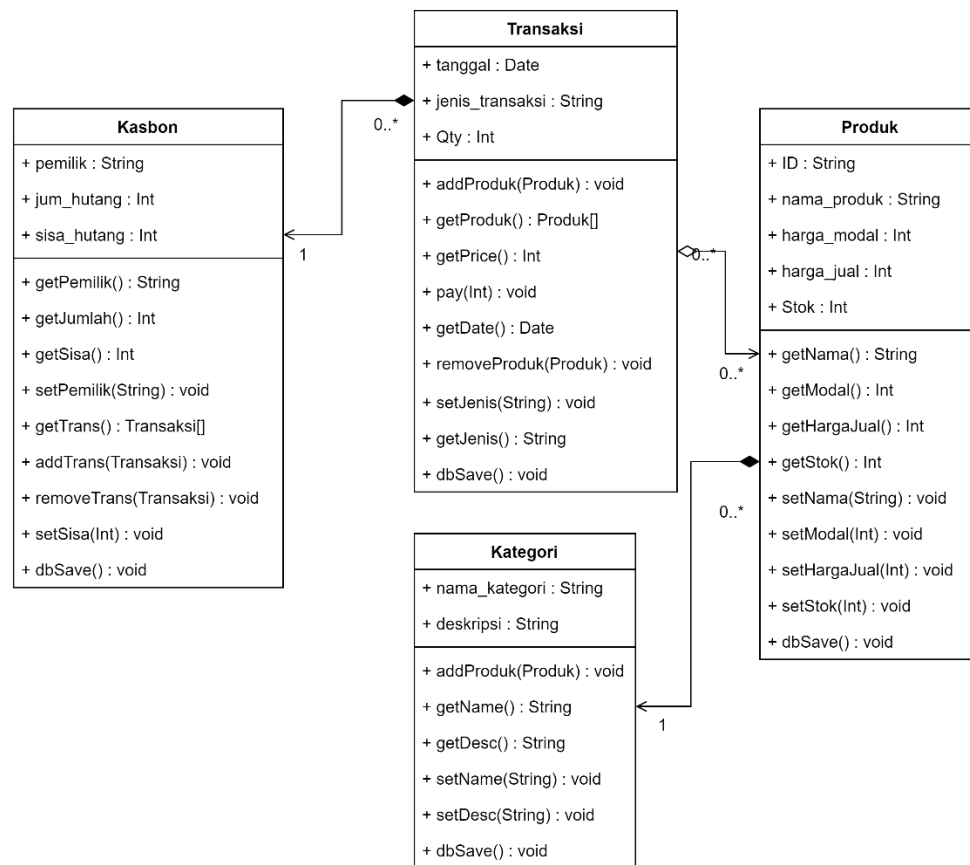
- Development Tool : Android Studio (SDK Platform : Android 12.0 API Level 31)
- DBMS : SQLite (kemungkinan berubah)

Kelompok pengguna dari aplikasi hanya ada satu yaitu Owner, dimana Owner memiliki hak fungsionalitas penuh pada perangkat lunak. Karena hanya ada satu class user, maka tidak dibuat basis data yang mengatur macam-macam pengguna dan tidak dibuat pengaturan privileges.

3. ARSITEKTUR SISTEM

3.1 Rancangan Arsitektur

○ Class Diagram



Gambar 1 : Class Diagram

Class Diagram di atas menjelaskan class / objek dan relasinya dengan class / objek lainnya. Enkapsulasi dibuat dengan akses sesuai dengan relasi pada gambar. Method dan lokal variabel dasar yang diperlukan dapat dilihat beserta dengan tipe datanya pada Gambar 1.

Kasbon hanya terhubung dengan Transaksi, dengan kebutuhan method untuk mendapatkan daftar Transaksi yang terdapat di dalam Kasbon.

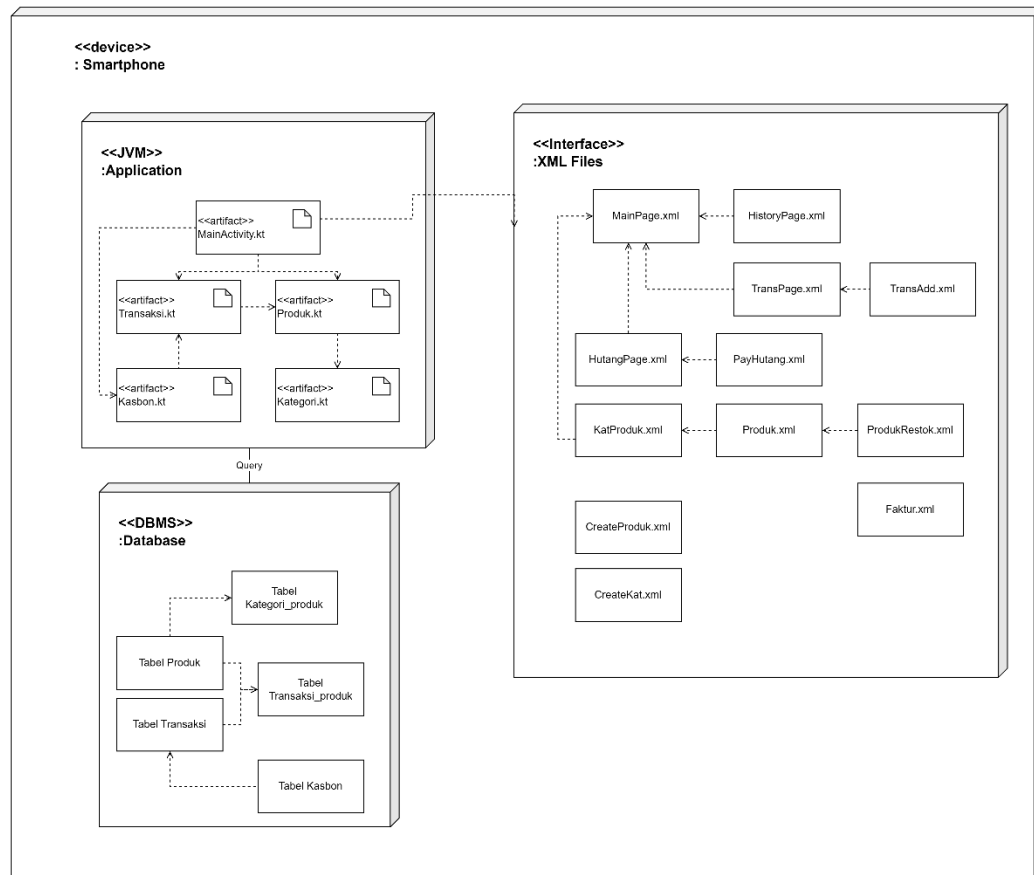
Transaksi terhubung dengan Produk, dimana banyak produk perlu dimasukkan ke dalam transaksi begitu pula sebaliknya banyak transaksi yang dapat diisi dengan class Produk.

Kategori membuat pengelompokkan Produk berdasarkan keinginan dari pengguna, dan hanya perlu

terhubung dengan Produk.

Setiap class memiliki method get dan set untuk mengakses variabel lokal sesuai dengan batasan yang diberikan.

○ Deployment Diagram



Gambar 2 : Deployment Diagram

Aplikasi adalah offline dengan semua bagian berada dalam 1 node device, yaitu smartphone dari pemilik Toko. Aplikasi dibuat dengan desain berbasis object, dimana setiap class akan dibuat dalam file .kt.

File fisik dari tampilan interface berupa XML digambarkan sebagai artifacts yang berada di dalam <<interface>>, dan code program beserta fungsi yang dibuat berbasis object, akan terisi dalam class file masing-masing.

Basis data dan aplikasi akan terhubung berdasarkan query yang dijalankan dari layer aplikasi ke bagian dbms lokal dari android.

○ Use Case Diagram



Gambar 3 : Use Case Diagram

Secara umum fitur-fitur fungsional pada aplikasi dijelaskan pada use case yang dilakukan oleh actor (Owner : Pemilik Toko Lia) pada Gambar 3.

Fungsi dalam hubungan generalisasi → spesifikasi :

- Setup list produk → membuat produk baru
- Setup list produk → memodifikasi field produk
- Update stok → menambah stok
- Update stok → mengurangi stok
- Fitur kasir → membuat transaksi
- Fitur kasir → menambahkan transaksi ke penggunaan pribadi
- Fitur kasir → menambahkan transaksi ke tab kasbon

Fungsi yang bergantung (dependent) :

- Menambahkan transaksi ke penggunaan pribadi → membuat transaksi
- Menambahkan transaksi ke kasbon → membuat transaksi

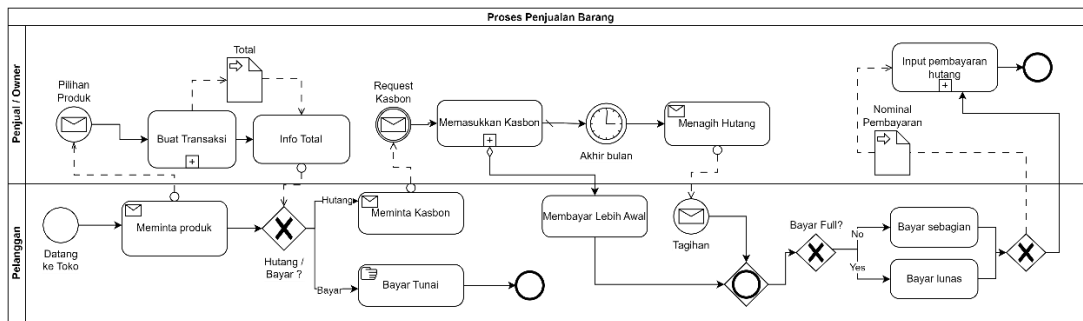
Fungsi yang melengkapi use case utama :

- Mencari produk → update stok & membuat transaksi
- Melihat tab penggunaan pribadi → menambahkan transaksi ke penggunaan pribadi
- Melihat tab kasbon → menambahkan transaksi ke tab kasbon
- Melihat riwayat transaksi → menambahkan transaksi ke tab kasbon & penggunaan pribadi

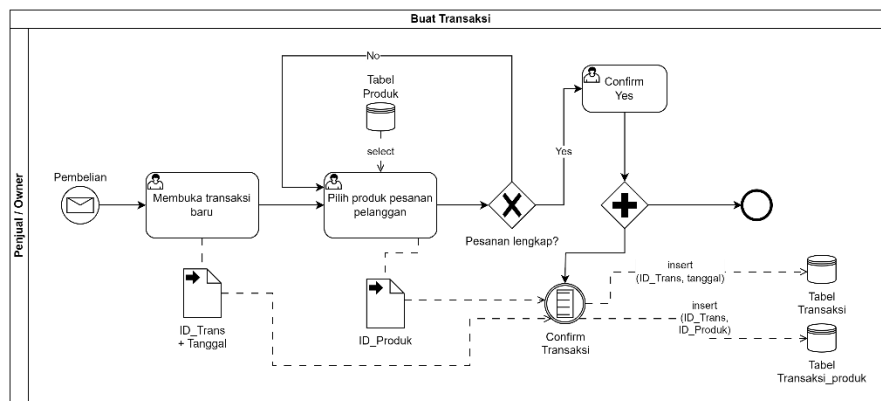
3.2 Deskripsi Dekomposisi

Dekomposisi menggunakan BPMN, dengan bisnis proses utama adalah penjualan produk dan proses restok barang. Proses penjualan barang memiliki 3 sub proses yang perlu dirincikan yaitu :

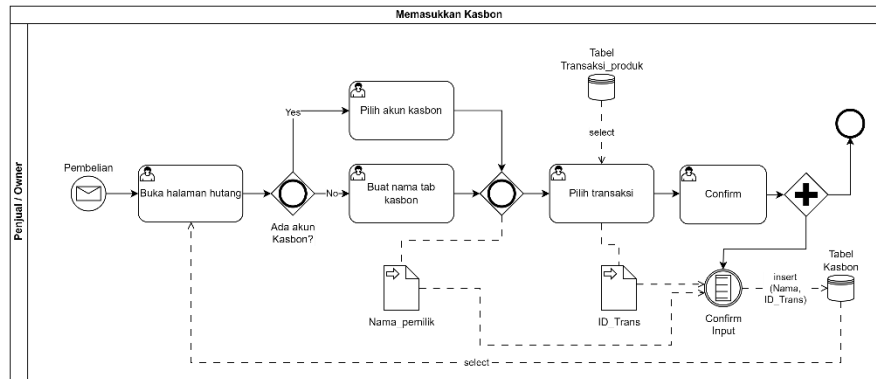
- Buat Transaksi
- Memasukkan Kasbon
- Input pembayaran hutang



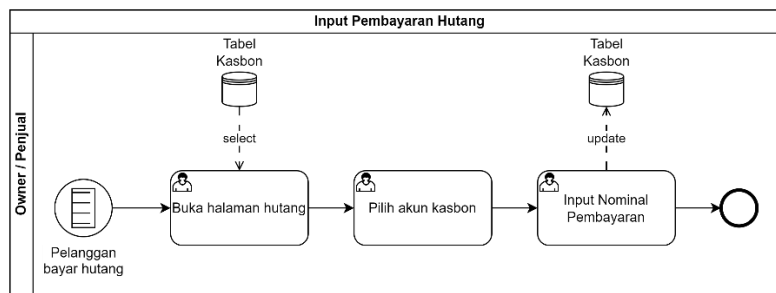
Gambar 4 : BPMN Penjualan Produk



Gambar 4.a : Sub Proses Membuat Transaksi

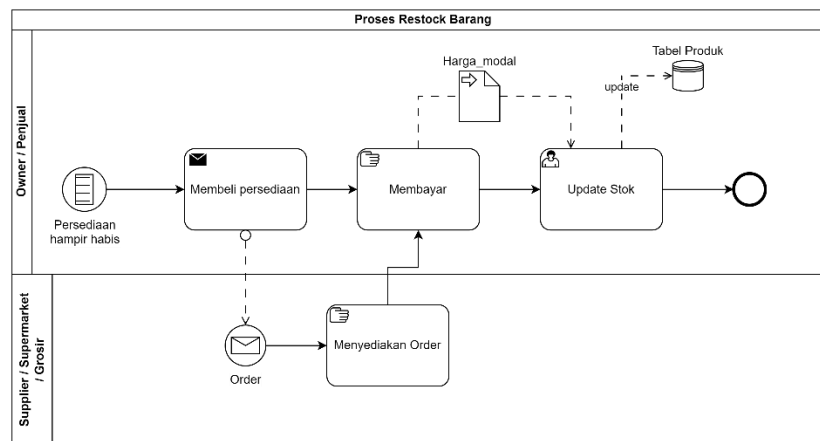


Gambar 4.b : Sub Proses Memasukkan Kasbon



Gambar 4.c : Sub Proses Input Pembayaran Hutang

BPMN untuk proses restock barang :



Gambar 5 : BPMN Proses Restok Barang

3.3 Alasan Perancangan

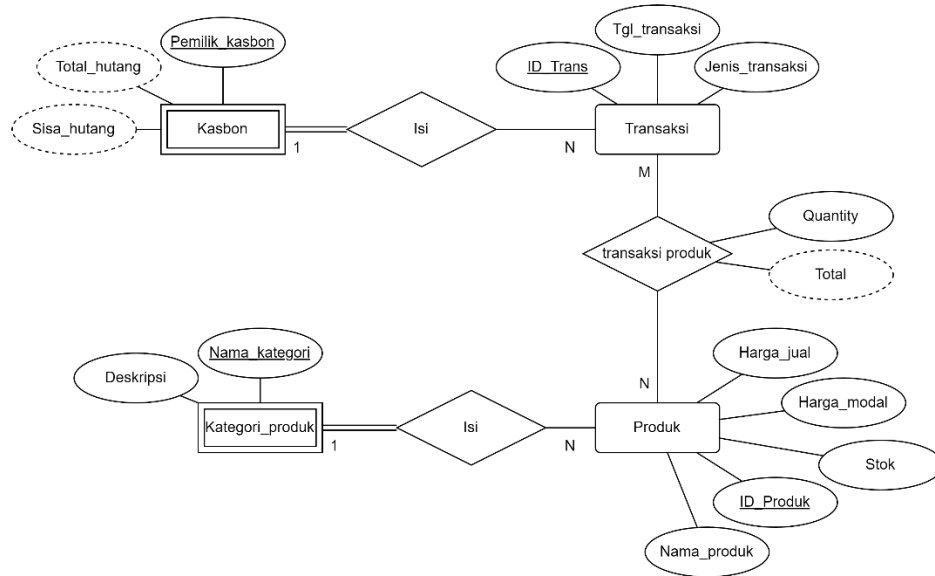
Arsitektur yang dibangun pada 3.1 merupakan Object Oriented Design (OOD) Software karena developer program sudah cukup familiar dengan konsep pemrograman tersebut, dimana juga aplikasi memiliki kebutuhan fitur enkapsulasi yang bisa didapatkan dari pemrograman berbasis objek, sehingga akses perubahan pada variabel pada masing-masing class dapat diatur.

4. RANCANGAN DATA

4.1 Deskripsi Data

Sumber data pada aplikasi berasal dari pengaturan dan input oleh pengguna yang akan disimpan ke dalam basis data yang dirincikan ke dalam :

- Entity Relationship Diagram (ERD) :



Merubah ERD tersebut menjadi tabel relasional sebelum di normalisasi :

- Setiap entitas dibuat menjadi relasi dengan semua atribut simple

Transaksi	
PK	ID_Trans
	Tanggal
	Jenis_transaksi

Kasbon	
PK	Pemilik_kasbon
	Total_hutang
	Sisa_hutang

Produk	
PK	ID_Produk
	Nama
	Harga_jual
	Harga_modal
	Stok

Kategori_produk	
PK	Nama_kategori
	Deskripsi

- Setiap binary relationship 1:N, dimana tipe entity bersisi N mempunyai partial participation constraint, buat relasi baru dimana primary keynya gabungan dari primary key kedua entity
➔ Menambahkan relasi :

Isi_Kategori	
PK,FK1	<u>ID_Kategori</u>
PK,FK2	<u>ID_Produk</u>

Isi_Kasbon	
PK,FK1	<u>Pemilik_kasbon</u>
PK,FK2	<u>ID_Trans</u>

3. Setiap binary relationship M:N, buat relasi baru dimana primary keynya gabungan primary key kedua tipe entity yang berelasi

➔ Menambahkan relasi :

Transaksi_produk	
PK,FK1	<u>ID_Trans</u>
PK,FK2	<u>ID_Produk</u>
	Quantity
	Total

4. Setiap weak entity, dibuat relasi baru yang memuat semua atribut weak entity dan primary keynya gabungan partial key dan primary key dari relasi induk

➔ Menambahkan relasi :

Kasbon	
PK	<u>Pemilik_kasbon</u>
PK,FK	<u>ID_Trans</u>
	Total_hutang
	Sisa_hutang

Kategori_produk	
PK	<u>Nama_kategori</u>
PK,FK	<u>ID_Produk</u>
	Deskripsi

Akhirnya memiliki kumpulan tabel :

Kasbon	
PK	<u>Pemilik_kasbon</u>
PK,FK	<u>ID_Trans</u>
	Total_hutang
	Sisa_hutang

Transaksi_produk	
PK,FK1	<u>ID_Trans</u>
PK,FK2	<u>ID_Produk</u>
	Quantity
	Total

Transaksi	
PK	<u>ID_Trans</u>
	Tanggal
	Jenis_transaksi

Kasbon	
PK	<u>Pemilik_kasbon</u>
	Total_hutang
	Sisa_hutang

Kategori_produk	
PK	<u>Nama_kategori</u>
PK,FK	<u>ID_Produk</u>
	Deskripsi

Isi_Kategori	
PK,FK1	<u>ID_Kategori</u>
PK,FK2	<u>ID_Produk</u>

Isi_Kasbon	
PK,FK1	<u>Pemilik_kasbon</u>
PK,FK2	<u>ID_Trans</u>

Produk	
PK	<u>ID_Produk</u>
	Nama
	Harga_jual
	Harga_modal
	Stok

Kategori_produk	
PK	<u>Nama_kategori</u>
	Deskripsi

Kemudian dilakukan normalisasi :

1. menghilangkan ketergantungan fungsional

Pemilik_kasbon → Total_hutang

Pemilik_kasbon → Sisa_hutang

Isi_Kasbon	
PK,FK1	<u>Pemilik_kasbon</u>
PK,FK2	<u>ID_Trans</u>

Kasbon	
PK	<u>Pemilik_kasbon</u>
PK,FK	<u>ID_Trans</u>
	Total_hutang
	Sisa_hutang

Kasbon	
PK	<u>Pemilik_kasbon</u>
	Total_hutang
	Sisa_hutang

Nama_kategori → Deskripsi

Isi_Kategori	
PK,FK1	<u>ID_Kategori</u>
PK,FK2	<u>ID_Produk</u>

Kategori_produk	
PK	<u>Nama_kategori</u>
PK,FK	<u>ID_Produk</u>
	Deskripsi

Kategori_produk	
PK	<u>Nama_kategori</u>
	Deskripsi

2. menghilangkan ketergantungan transitif

Kategori_produk	
PK	<u>Nama_kategori</u>
PK,FK	<u>ID_Produk</u>

Isi_Kategori	
PK,FK1	<u>ID_Kategori</u>
PK,FK2	<u>ID_Produk</u>

Transaksi_produk	
PK,FK1	<u>ID_Trans</u>
PK,FK2	<u>ID_Produk</u>
	Quantity
	Total

Transaksi	
PK	<u>ID_Trans</u>
	Tanggal
	Jenis_transaksi

Produk	
PK	<u>ID_Produk</u>
	Nama
	Harga_jual
	Harga_modal
	Stok

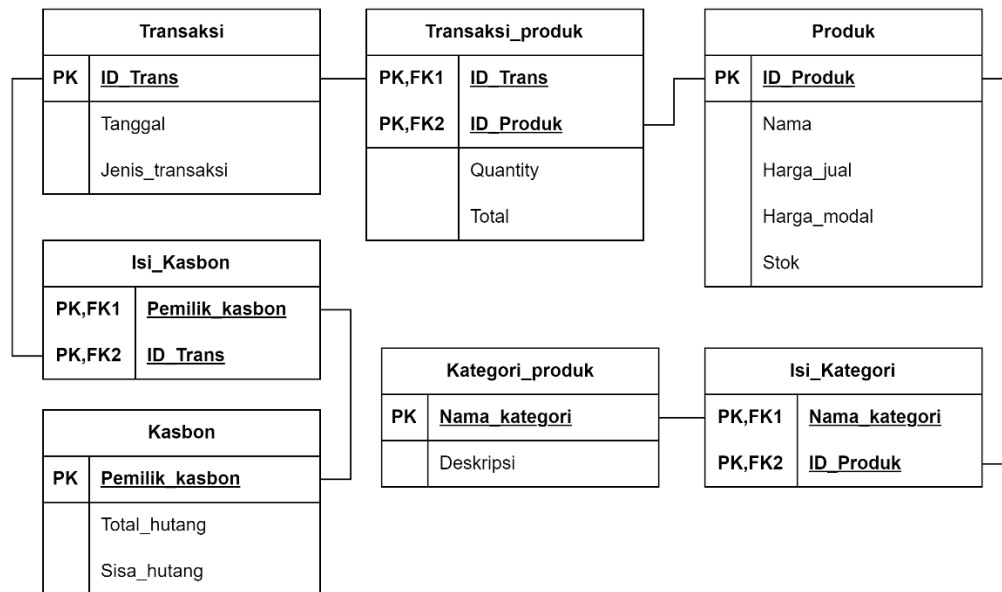
Kasbon	
PK	<u>Pemilik_kasbon</u>
PK,FK	<u>ID_Trans</u>

Isi_Kasbon	
PK,FK1	<u>Pemilik_kasbon</u>
PK,FK2	<u>ID_Trans</u>

Kasbon	
PK	<u>Pemilik_kasbon</u>
	Total_hutang
	Sisa_hutang

Kategori_produk	
PK	<u>Nama_kategori</u>
	Deskripsi

Tidak ada ketergantungan transitif, tapi ada beberapa tabel yang memiliki tabel yang sama. Seperti tabel Kategori_produk dengan kunci primer gabungan dan tabel Isi_Kategori berikut juga tabel Kasbon dengan kunci primer gabungan dan tabel Isi_Kasbon. Maka hanya perlu diambil salah satu, dan menyisakan 7 Tabel akhir :



Setiap input dari pengguna aplikasi, akan disimpan ke dalam basis data yang terdiri dari 7 (tujuh) tabel : Tabel Kasbon, Tabel Isi_Kasbon, Tabel Transaksi, Tabel Transaksi_produk, Tabel Produk, Tabel Isi_Kategori, dan Tabel Kategori_produk. Kecuali ID_Trans dan ID_Produk, setiap isi field berasal dari input pengguna di dalam aplikasi. ID_Trans dan ID_Produk dibuat di dalam program secara otomatis menggunakan rules yang tidak perlu campur tangan pengguna.

4.2 Kamus Data

Nama Tabel	:	Produk
Keterangan	:	Tabel untuk menyimpan record tentang produk yang dijual di Toko Lia
Primary Key	:	ID_Produk
Foreign Key	:	-
Jumlah Field	:	5

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	ID_Produk*	Char(10)	Kode identifikasi produk
2.	Nama	Char(50)	Nama produk yang dijual di toko
3.	Harga_jual	Integer	Harga jual produk
4.	Harga_modal	Integer	Harga perolehan dari produk
5.	Stok	Integer	Jumlah barang yang tersedia di toko

Nama Tabel : Transaksi
 Keterangan : Tabel untuk menyimpan identifier dari transaksi tanpa isi produk
 Primary Key : ID_Trans
 Foreign Key : -
 Jumlah Field : 3

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	ID_Trans*	Char(10)	Kode identifikasi transaksi
2.	Tanggal	Date	Tanggal dibuatnya transaksi
3.	Jenis_transaksi	Char(20)	Jenis transaksi kategorikal (akan diset dalam program : “pembelian”, “penggunaan sendiri”, “hutang”

Nama Tabel : Transaksi_produk
 Keterangan : Tabel untuk menyimpan isian produk yang ada di dalam transaksi
 Primary Key : ID_Trans, ID_Produk (composite)
 Foreign Key : ID_Trans, ID_Produk
 Jumlah Field : 4

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	ID_Trans*	Char(10), FK	Kode identifikasi transaksi
2.	ID_Produk*	Char(10), FK	Kode identifikasi produk
3.	Quantity	Integer	Jumlah produk yang ada di dalam transaksi
4.	Total	Integer	Total harga transaksi

Nama Tabel : Kategori_produk
 Keterangan : Tabel untuk menyimpan kategori, dimana produk akan dikategorikan
 Primary Key : Nama_kategori
 Foreign Key : -
 Jumlah Field : 2

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	Nama_kategori*	Char(20)	Nama dari kategori yang dibuat
2.	Deskripsi	Char(50)	Deskripsi singkat menjelaskan kategori

Nama Tabel : Isi_Kategori
 Keterangan : Tabel untuk menyimpan produk berdasarkan kategori
 Primary Key : Nama_kategori, ID_Produk (composite)
 Foreign Key : Nama_kategori, ID Produk
 Jumlah Field : 2

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	Nama_kategori*	Char(20), FK	Nama dari kategori yang dibuat
2.	ID_Produk*	Char(10), FK	Kode identifikasi produk

Nama Tabel : Kasbon
 Keterangan : Tabel yang menyimpan akun kasbon beserta total hutang yang dimiliki oleh akun
 Primary Key : Pemilik_kasbon
 Foreign Key : -
 Jumlah Field : 3

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	Pemilik_kasbon*	Char(20)	Nama dari pemilik kasbon
2.	Total_hutang	Integer	Total nilai transaksi yang ada di akun kasbon
3.	Sisa_hutang	Integer	Sisa nilai transaksi yang belum terbayar di akun kasbon

Nama Tabel : Isi_Kasbon
 Keterangan : Tabel yang menyimpan akun kasbon beserta total hutang yang dimiliki oleh akun
 Primary Key : Pemilik_kasbon, ID_Trans (composite)
 Foreign Key : Pemilik_kasbon, ID_Trans
 Jumlah Field : 2

No.	Nama Field	Tipe Data (Size)	Deskripsi
1.	Pemilik_kasbon*	Char(20), FK	Nama dari pemilik kasbon
2.	ID_Trans*	Char(10), FK	Kode identifikasi transaksi

Class : Kasbon
 Keterangan : Class pembentuk objek kasbon untuk menjalankan fungsi-fungsi yang hanya perlu dilakukan seputar akun hutang
 Jumlah Variabel : 3
 Jumlah Method : 9

No.	Nama Variabel	Tipe Data	Mutable
1.	Pemilik	String	No
2.	Jum_hutang	Integer	Yes
3.	Sisa_hutang	Integer	Yes

No.	Nama Method	Return Type	Parameter
1.	getPemilik()	String	-
2.	getJumlah()	Integer	-
3.	getSisa()	Integer	-
4.	setSisa(int)	Void	Satu parameter bertipe int
5.	setPemilik(String)	Void	Satu parameter bertipe String
6.	addTrans()	Void	Satu parameter bertipe data Transaksi
7.	getTrans()	Transaksi[]	Satu parameter bertipe data String
8.	removeTrans(Transaksi)	Void	Satu parameter bertipe data Transaksi
9.	dbSave()	Void	-

Class : Transaksi
 Keterangan : Class pembentuk objek Transaksi untuk menghitung total nominal, jumlah barang, dan jenis transaksi yang akan disimpan
 Jumlah Variabel : 3
 Jumlah Method : 9

No.	Nama Variabel	Tipe Data	Mutable
1.	Tanggal	Date	No
2.	Jenis_transaksi	String	Yes
3.	Quantity	Integer	Yes

No.	Nama Method	Return Type	Parameter
1.	addProduk(Produk)	Void	Satu parameter bertipe Produk
2.	getProduk()	Produk[]	-
3.	getPrice()	Integer	-
4.	pay(int)	Void	Satu parameter bertipe int
5.	getDate()	Date	-
6.	removeProduk(Produk)	Void	Satu parameter bertipe Produk
7.	setJenis(String)	Void	Satu parameter bertipe String
8.	getJenis()	String	-
9.	dbSave()	Void	-

Class : Kategori

Keterangan : Class pembentuk kategori dari produk

Jumlah Variabel : 2

Jumlah Method : 6

No.	Nama Variabel	Tipe Data	Mutable
1.	Nama_kategori	String	Yes
2.	Deskripsi	String	Yes

No.	Nama Method	Return Type	Parameter
1.	addProduk(Produk)	Void	Satu parameter bertipe Produk
2.	getName()	String	-
3.	getDesc()	String	-
4.	setName()	Void	Satu parameter bertipe String
5.	setDesc()	Void	Satu parameter bertipe String
6.	dbSave()	Void	-

Class : Produk
 Keterangan : Class pembentuk objek produk yang dijual di toko
 Jumlah Variabel : 5
 Jumlah Method : 9

No.	Nama Variabel	Tipe Data	Mutable
1.	ID	String	No
2.	Nama_produk	String	Yes
3.	Harga_modal	Integer	Yes
4.	Harga_jual	Integer	Yes
5.	Stok	Integer	Yes

No.	Nama Method	Return Type	Parameter
1.	getNama()	String	-
2.	getModal()	Integer	-
3.	getHargaJual()	Integer	-
4.	getStok()	Integer	-
5.	setNama(String)	Void	Satu parameter bertipe String
6.	setModal(Int)	Void	Satu parameter bertipe Int (Integer)
7.	setHargaJual(Int)	Void	Satu parameter bertipe Int (Integer)
8.	setStok(Int)	Void	Satu parameter bertipe Int (Integer)
9.	dbSave()	Void	-

5. RANCANGAN KOMPONEN

Mengacu pada 3.1 dan 3.2, komponen utama pada sistem dibagi menjadi 3 :

- XML Files (View Interface)
 - o MainPage.xml (halaman utama)
 - o KatProduk.xml (halaman kategori produk)
 - o Produk.xml (halaman rincian produk)
 - o HistoryPage.xml (halaman riwayat transaksi)
 - o TransPage.xml (halaman pengisian transaksi)
 - o TransAdd.xml (halaman penambahan produk ke transaksi)
 - o HutangPage.xml (halaman hutang)
 - o PayHutang.xml (halaman proses pembayaran hutang)
 - o dst.

- Kotlin Files (Algoritma Program)
 - MainActivity.kt (fungsi main)
 - Fungsi :
 - main(){
....}
 - Produk.kt
 - Constructor :
Produk(String nama, Int modal, Int jual){
this.nama_produk = nama
this.harga_modal = modal
this.harga_jual = jual
}
Fungsi :
 - getNama(){
return this.Nama}
 - getModal(){
return this.harga_modal}
 - getHargaJual(){
return this.harga_jual}
 - getStok(){
return this.Stok}
 - setNama(String nama){
this.Nama = nama}
 - setModal(int modal){
this.harga_modal = modal}
 - setHargaJual(int harga){
this.harga_jual = harga}
 - setStok(int qty){
this.Stok = qty}
 - dbSave(){
send query = insert into tabel Produk (ID, getNama(), getModal(), getJual(),
getStok())}
 - Kategori.kt
 - Constructor:
Kategori(String nama, String desc){
this.nama_kategori = nama
this.deskripsi = desc}
Fungsi :
 - addProduk(Produk){
[array].append(Produk)}
 - getName(){
return this.nama_kategori}
 - getDesc(){
return this.deskripsi}
 - setName(String nama){
this.nama_kategori = nama}
 - setDesc(String desc){
this.deskripsi = desc}
 - dbSave(){
send query = insert into tabel Kategori (getName(), getDesc())}

- Transaksi.kt
 - Constructor:
 - Transaksi(Date tanggal, String jenis, Int qty){
 - This.tanggal = tanggal
 - This.jenis_transaksi = jenis
 - This.Qty = qty }
 - Fungsi :
 - addProduk(Produk id){
 - [Produk array].append(id)}
 - getProduk(){
 - return [Produk array]}
 - getPrice(){
 - for each item in [Produk array]{
 - total = total + item.getHargaJual * Qty
 - }
 - return total }
 - pay(Int nominal){
 - total = nominal – total }
 - getDate(){
 - return this.tanggal }
 - removeProduk(Produk id){
 - [Produk array].pop(id)}
 - setJenis(String jenis){
 - this.jenis_transaksi = jenis }
 - getJenis(){
 - return this.jenis_transaksi }
 - dbSave(){
 - send query = insert into tabel Transaksi (ID, getDate(), getJenis(), Qty) }
- Kasbon.kt
 - Constructor :
 - Kasbon(String nama, int hutang, int sisa){
 - This.pemilik = nama
 - This.jum_hutang = hutang
 - This.sisa_hutang = sisa }
 - Fungsi :
 - getPemilik(){
 - return this.pemilik }
 - getJumlah(){
 - return this.jum_hutang }
 - getSisa(){
 - return this.sisa_hutang }
 - setPemilik(String nama){
 - this.pemilik = nama }
 - getTrans(){
 - return [Transaksi array]}
 - addTrans(Transaksi id){
 - [Transaksi array].append(id)}
 - removeTrans(Transaksi id){
 - [Transaksi array].pop(id)}
 - setSisa(Int nominal){
 - this.sisa_hutang = nominal }

- dbSave(){
send query = insert into tabel Kasbon (getPemilik(), getJumlah(), getSisa())}
- Basis Data Relasional
 - Tabel Produk
Create table Produk (id char 10, nama char 50, harga_jual int, harga_modal int, stok int) set primary key id
 - Tabel Transaksi
Create table Transaksi (id char 10, tanggal date, jenis_transaksi char 20) set primary key id
 - Tabel Kasbon
Create table Kasbon (pemilik char 20, total_hutang int, sisa_hutang int) set primary key pemilik
 - Tabel Transaksi_produk
Create table Transaksi_produk (id_trans char 10, id_produk char 10, quantity int, total int) set primary key id_trans, id_produk
 - Tabel Isi_Kasbon
Create table Isi_Kasbon (pemilik char 20, id_trans char 10) set primary key pemilik, id_trans
 - Tabel Kategori_produk
Create table Kategori_produk (nama_kat char 20, deskripsi char 50) set primary key nama_kat
 - Tabel Isi_Kategori
Create table Isi_Kategori (nama_kat char 20, id_produk char 10) set primary key nama_kat, id_produk

6. RANCANGAN ANTARMUKA

6.1 Gambaran Umum Rancangan Antarmuka

Saat membuka aplikasi, tampilan pertama yang muncul adalah tampilan dasbor yang menunjukkan nominal pemasukkan, pengeluaran, dan hutang. Melalui navigasi bar melalui tombol garis tiga, halaman lain dapat diakses yaitu : halaman Transaksi, Produk, Restok, Hutang, dan History.

Halaman Produk digunakan untuk membuat produk yang dapat diinputkan ke dalam transaksi ataupun untuk dimodifikasi jumlahnya untuk halaman Restok. Produk dibuat dengan memilih tombol membuat kategori, kemudian memasukkan nama kategori beserta deskripsi singkat. Setelah itu, dilanjutkan dengan pengisian field nama produk, harga modal dan harga jual dari produk. Setelah disimpan, produk yang dibuat dapat dilihat di dalam kategori yang dibuat.

Pada halaman Transaksi, pengguna dapat mencari barang untuk ditambahkan ke halaman transaksi dan kemudian kuantitasnya dapat diatur sesuai jumlah transaksi. Pada akhir transaksi, konfirmasi akan berupa pemilihan kategori yaitu : Hutang, Pemakaian Sendiri, dan Pembelian. Hutang artinya transaksi akan disimpan sebagai hutang ke dalam akun kasbon, pemakaian sendiri artinya barang di dalam transaksi digunakan oleh pemilik toko, dan pembelian artinya transaksi pembelian dengan pelanggan seperti biasa.

Pada halaman Restok, pengguna akan mengakses halaman kategori dan mencari produk yang baru


dibeli, kemudian menekan tombol restok untuk memasukkan inputan baru harga modal sesuai harga yang dibayarkan saat pembelian dan jumlah stok barang. Input harga jual diterakan sebagai inputan tidak wajib apabila pengguna ingin meningkatkan harga jual dari barang yang distok.




Halaman Hutang akan memunculkan daftar akun kasbon, dimana ada tombol detail untuk menunjukkan rincian transaksi yang dimasukkan ke dalam kasbon. Pada halaman ini dapat dilakukan modifikasi nominal hutang berdasarkan pembayaran yang dilakukan oleh penghutang, artinya ada pembayaran sebagian dan ada pembayaran penuh yang merubah nilai hutang menjadi nol.

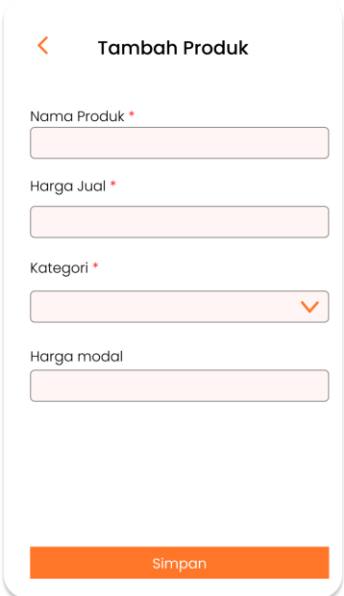
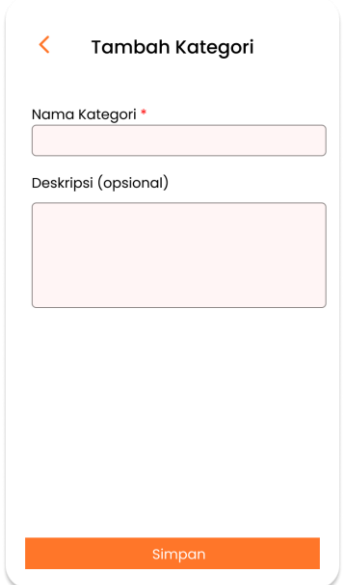
Pada halaman History akan ditampilkan daftar transaksi yang sudah dilakukan di dalam aplikasi. Untuk melihat beragam jenis transaksi, akan ada filter berupa dropdown list yang dapat menampilkan transaksi sesuai kategori filter.



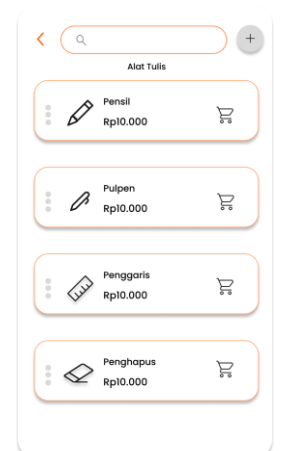
Untuk menghindari misoperasi dalam penginputan data terutama bagian transaksi, akan ditampilkan dialog box berupa konfirmasi Yes/No.


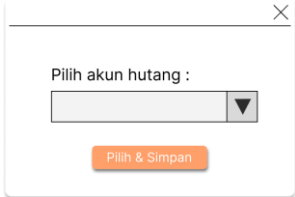
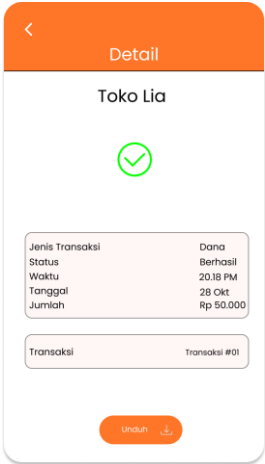
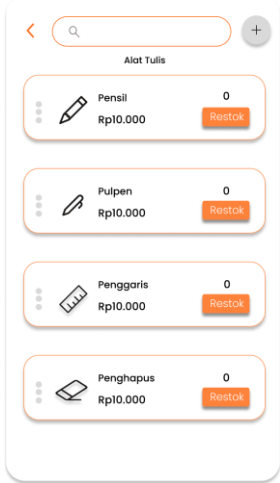
6.2 Tampilan Layar

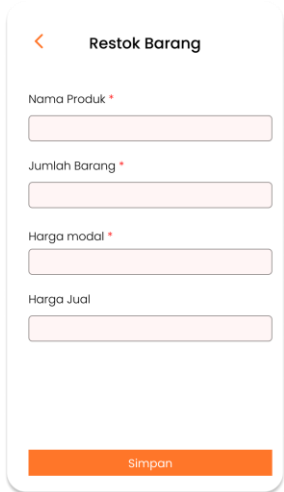
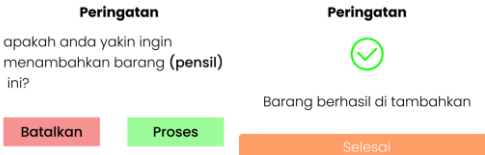
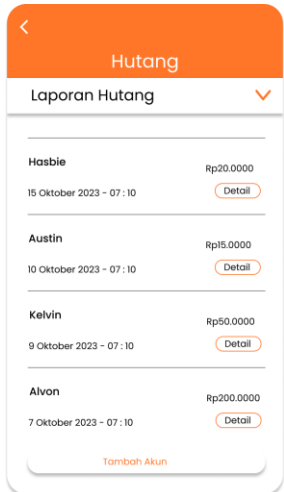
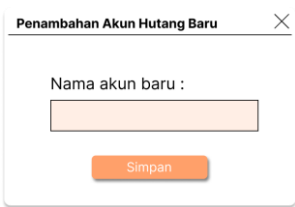
Tampilan	Nama Halaman
	Halaman load





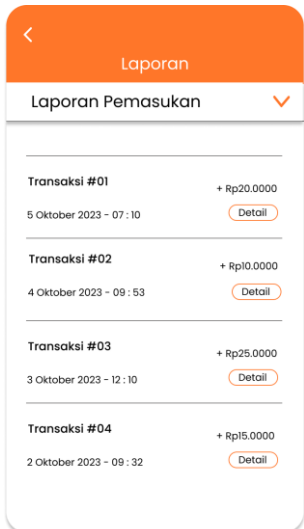
Tampilan	Nama Halaman
	Halaman Utama / Dasbor
	Side navbar
	Halaman produk

Tampilan	Nama Halaman
	Halaman Tambah Produk
	Halaman Tambah Kategori





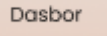
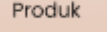
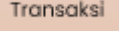
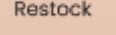
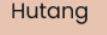
Tampilan	Nama Halaman
	Halaman Transaksi
	Halaman Pemilihan Kategori Produk tanpa akses modifikasi
	Halaman Pemilihan Produk untuk masuk ke dalam Transaksi

Tampilan	Nama Halaman
	Dialog Box Setelah Pilihan Proses
	Dialog Box Setelah Pilihan Hutang
	Halaman Faktur Transaksi
	Halaman Restok per Produk dalam Kategori



Tampilan	Nama Halaman
	Halaman Pengisian Field Restok
	Dialog Box Konfirmasi Restok
	Halaman Hutang
	Dialog Box input Nama Akun


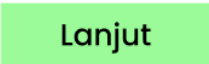








Tampilan	Nama Halaman
	Halaman Detail Hutang
	Dialog Box setelah Proses Hutang
	Dialog Box Pengisian Nominal Cicil
	Dialog Box Konfirmasi Pembayaran Hutang
	Halaman History

6.3 Tampilan Layar dan Tindakan

Gambar	Lokasi Halaman	Nama Fitur	Deskripsi
	Halaman Utama	Garis tiga	Digunakan untuk membuka list navigation bar
	Halaman Utama	Pemasukan	Menampilkan jumlah pemasukan yang berasal dari pembelian oleh pembeli dari Toko Lia
	Halaman Utama	Pengeluaran	Menampilkan jumlah pengeluaran yang digunakan untuk kebutuhan dari Toko Lia
	Halaman Utama	Hutang	Menampilkan jumlah hutang yang belum dibayar oleh para pembeli dari Toko Lia
	Side navbar	Dasbor	Mengarahkan pengguna aplikasi menuju halaman awal aplikasi Toko Lia
	Side navbar	Produk	Mengarahkan pengguna aplikasi menuju halaman daftar produk aplikasi Toko Lia
	Side navbar	Transaksi	Mengarahkan pengguna aplikasi menuju halaman transaksi aplikasi Toko Lia
	Side navbar	Restock	Mengarahkan pengguna aplikasi menuju halaman restock barang aplikasi Toko Lia
	Side navbar	Hutang	Mengarahkan pengguna aplikasi menuju halaman hutang

	Side navbar	History	Mengarahkan pengguna aplikasi menuju halaman riwayat transaksi
	Side navbar	Keluar Apl	Menutup aplikasi Toko Lia
	Halaman Tambah Kategori, Halaman Tambah Produk	Simpan	Menyimpan produk ke dalam daftar barang aplikasi Toko Lia
	Semua halaman kecuali Halaman Utama, Halaman Kategori dan Dialog Box	Kembali	Mengarahkan pengguna Kembali ke halaman produk
	Halaman produk, Halaman Pemilihan Kategori Produk tanpa akses modifikasi	Kategori produk	Menampilkan secara singkat produk – produk yang terdapat dalam kategori produk. Pengguna dapat mengubah daftar produk yang terdapat di dalam kategori produk dengan menekan salah satu kategori.
	Halaman produk	Tambah Produk	Menambah produk ke dalam salah satu kategori produk Toko Lia
	Halaman produk	Tambah Kategori	Menambah Kategori yang ada di aplikasi Toko Lia
 Pensil Rp10.000	Semua halaman dengan tampilan produk	Barang	Nama barang beserta harganya
	Halaman konfirmasi transaksi	Kuantitas	Mengatur kuantitas barang
Transaksi #01 5 Oktober 2023 - 07 : 10	Halaman History, Halaman Detail Hutang	Identitas Transaksi	Menampilakn kode identifikasi transaksi beserta waktu transaksi
+ Rp20.0000	Halaman History	Uang	Keterangan nominal uang (+) berarti pemasukan, (-) berarti pengeluaran

Detail	Halaman History	Detail	Melihat detail transaksi
Laporan Pemasukan 	Halaman History	Filter History	Dropdown list yang memuat jenis history/transaksi yang dapat dilihat pada halaman
Tambah Pesanan	Halaman Konfirmasi Transaksi	Tambah pesanan	Menambahkan pesanan yang dipesan oleh pembeli
Total harga Rp. 2.515.000 Bayar Rp. 2.600.000 Kembalian Rp. 85.000	Halaman Detail Hutang, Halaman Konfirmasi Transaksi	Detail pembayaran	Menampilkan detail pembayaran yang dilakukan oleh pembeli meliputi total harga, uang yang diberikan ke pengguna, dan uang kembalian
Batal	Halaman Detail Hutang, Halaman Konfirmasi Transaksi	Batal	Membatalkan pesanan yang dipesan oleh pembeli
Proses	Halaman Detail Hutang, Halaman Konfirmasi Transaksi	Proses	Memproses pembelian yang dipesan oleh pembeli
Jenis Transaksi Dana Status Berhasil Waktu 20.18 PM Tanggal 28 Okt Jumlah Rp 50.000	Halaman Faktur	Detail faktur	Menampilkan detail – detail faktur transaksi yang dilakukan oleh pembeli, seperti jenis transaksi, status, waktu, tanggal, dan jumlah
Transaksi Transaksi #01	Halaman Faktur	Identitas transaksi	Menampilkan kode identitas transaksi
Unduh 	Halaman Faktur	Unduh faktur	Mengunduh faktur dengan format .pdf
Cicil	Dialog Box setelah Proses Hutang	Tombol Cicil	Melanjutkan pembayaran hutang dengan nominal input tertentu
Penuh	Dialog Box setelah Proses Hutang	Tombol Bayar Penuh	Melanjutkan pembayaran hutang dengan nominal penuh dari sisa hutang

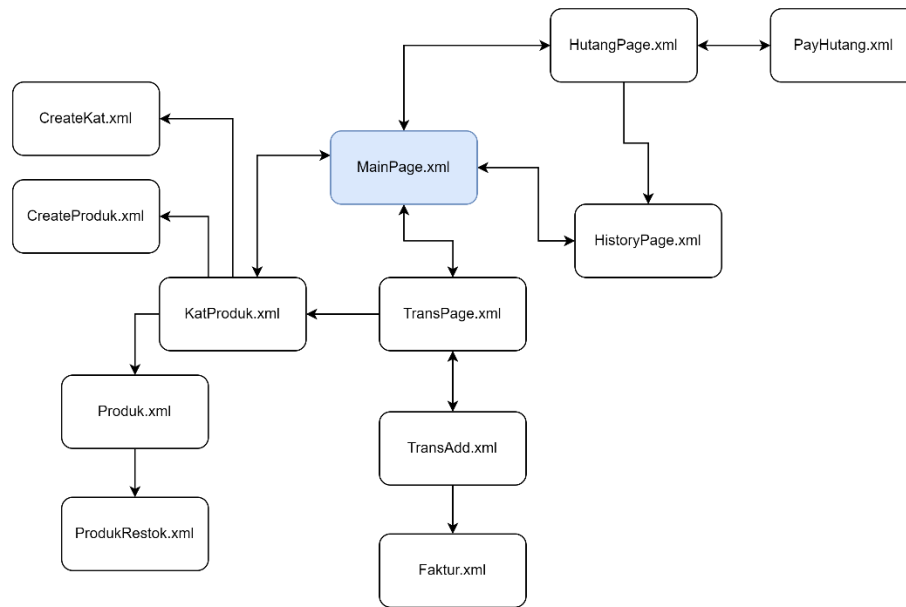
	Dialog Box Pengisian Nominal Cicil	Tombol Kembali	Menuju ke halaman sebelumnya
	Dialog Box Pengisian Nominal Cicil	Tombol Lanjut	Melanjutkan pembayaran / konfirmasi nominal pembayaran
	Halaman Hutang	Tombol Tambah Akun	Melanjutkan ke membuat nama baru untuk akun kasbon
	Halaman Konfirmasi Transaksi	Input Bayar	Memasukkan nominal pembayaran agar otomatis menghitung kembalian
	Dialog Box Setelah Pilihan Proses	Tombol Pakai Sendiri	Menandakan / mengirim transaksi sebagai jenis penggunaan sendiri
	Dialog Box Setelah Pilihan Proses	Tombol Penjualan	Menandakan transaksi sebagai penjualan normal dengan pembayaran utuh
	Dialog Box Setelah Pilihan Proses	Tombol Hutang	Menandakan transaksi sebagai hutang dari salah satu akun yang akan dipilih
	Halaman Restok per Produk dalam Kategori	Tombol Restok	Menyimpan input jumlah restok dari barang yang dipilih
	Dialog Box Menambahkan Akun Hutang	Tombol Simpan	Menyimpan input nama dari pengguna
	Dialog Box Setelah Pilihan Hutang	Tombol Pilih	Memilih akun hutang yang setelah proses pembayaran

7. MATRIKS PERSYARATAN

Requirement	Komponen	Jenis Kebutuhan
Fungsi Setup List Produk	KatProduk.xml Produk.xml CreateProduk.xml CreateKat.xml Produk.kt	Sesuai SRS v.1.1

	Tabel Produk Tabel Kategori Tabel Isi_kategori	
Fungsi Update Stok	Produk.xml ProdukRestok.xml Tabel Produk Produk.kt	Sesuai SRS v.1.1
Fungsi Pencarian Produk	KatProduk.xml Produk.xml Produk.kt Kategori.kt Tabel Isi_kategori Tabel Kategori Tabel Produk	Sesuai SRS v.1.1
Fungsi Fitur Kasir	TransPage.xml TransAdd.xml KatProduk.xml Produk.xml Produk.kt Transaksi.kt Tabel Produk Tabel Transaksi	Sesuai SRS v.1.1
Fungsi Kasbon	HutangPage.xml PayHutang.xml Kasbon.kt Transaksi.kt Tabel Kasbon Tabel Isi_Kasbon Tabel Transaksi	Sesuai SRS v.1.1
Fungsi Melihat History	HistoryPage.xml	Mengikuti Hasil Analisa
Fungsi Pelaporan	Faktur.xml MainPage.xml	Sesuai Project Charter awal

8. LAMPIRAN



Gambar 6 : Navigation Diagram