BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Database

Untuk mengimplementasikan SQLite, peneliti telah menggunakan database Room atau *room persistence library*. Ini adalah lapisan abstraksi yang berada di atas SQLite. Di Android, Room merupakan ORM (Object Relational Mapper) untuk SQLite, dan juga sebagai komponen arsitektur.

Dengan menerapkan anotasi, Room memudahkan pengembang menggunakan SQLite. Salah satu keuntungannya adalah menghilangkan kebutuhan pengembang untuk menulis banyak kode boilerplate untuk membangun dan mengelola database. Itu juga memvalidasi kueri SQL pada waktu kompilasi. Artinya, jika ada kesalahan kueri SQL, aplikasi tidak dapat dikompilasi. Ini mencegah pengembang mengalami masalah run-time[24]. Database ini didesain untuk membuat sistem seringan mungkin dengan fitur-fitur yang digunakannya.

4.1.1 Implementasi Database

Kode program diatas merupakan kelas abstrak yang digunakan untuk mengatur dan menginisialisasi database dalam aplikasi menggunakan framework Room pada Android.

Anotasi @TypeConverters(Converters::class) menunjukkan bahwa kelas ini menggunakan kelas Converters sebagai konverter tipe data untuk Room Database.

Anotasi @Database(entities = [....], version = 2) menandakan bahwa kelas ini adalah database dengan versi 2, dan memiliki beberapa tabel seperti Users, Product, Cart, Transaction, Restock, Return, Balance, BalanceReport, Accounting, dan Services.

Kelas ini juga memiliki beberapa objek Migration yang digunakan untuk melakukan migrasi database ketika ada perubahan pada skema database. Fungsi getDatabase() digunakan untuk mendapatkan instance database dan menggunakannya dalam aplikasi. Synchronized block digunakan untuk mencegah terjadinya race condition ketika memanggil instance database dalam thread yang berbeda.

Secara keseluruhan, kelas ini adalah komponen penting dalam pengembangan aplikasi Android yang membutuhkan penggunaan database, dan membantu memudahkan pengembang dalam pengaturan dan migrasi database pada aplikasi.

4.1.2 Implementasi Tabel

A. Tabel Pembukuan (Accounting)

```
@Entity(tableName = "accounting")
data class Accounting(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idAccounting")
    var idAccounting: Int = 0,

    @ColumnInfo(name = "dateAccounting")
    var dateAccounting: String? = null,

    @ColumnInfo(name = "initDigital")
    var initDigital: Int? = null,

    @ColumnInfo(name = "initCash")
    var initCash: Int? = null,

    @ColumnInfo(name = "initStock")
    var initStock: Int? = null,

    @ColumnInfo(name = "initStockWorth")
    var initStockWorth: Int? = null,

    @ColumnInfo(name = "capitalInvest")
    var capitalInvest: Int? = null,
```

Kode ini menggunakan anotasi @Entity untuk menunjukkan bahwa ini adalah tabel dalam database dan memberikan nama tabel sebagai "accounting". Setiap properti dalam kelas ini mewakili kolom dalam tabel.

Ada juga anotasi @PrimaryKey yang menunjukkan bahwa properti "idAccounting" digunakan sebagai kunci utama, dan anotasi @ColumnInfo digunakan untuk memberikan nama kolom yang sesuai di database. Setiap properti dapat memiliki nilai null yang menunjukkan bahwa kolom tersebut dapat kosong. Selain itu, setter dan getter untuk properti tersebut telah dibuatkan oleh Kotlin secara otomatis.

Tabel ini berfungsi untuk menyimpan data pembukuan yang akan dibantu oleh sistem dengan mengambil data-data yang diperlukan dari tabel Balance, BalanceReport, Cart, Return, Restock, dan transaction.

B. Tabel Saldo (Balance)

Kode program di atas adalah definisi dari sebuah kelas bernama Balance dengan anotasi @Entity yang menunjukkan bahwa kelas ini adalah entitas database.

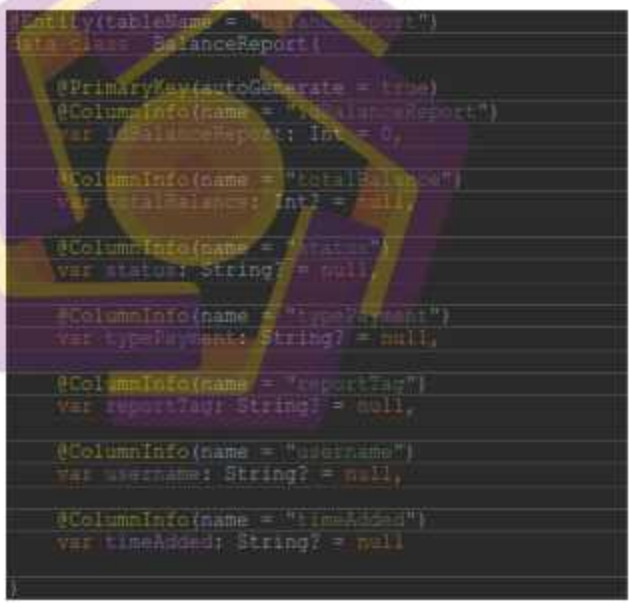
Kelas ini memiliki tiga properti yaitu idBalance, digitalBalance, dan CashBalance yang semuanya merupakan variabel nullable dengan tipe data Int?, serta Profit yang merupakan variabel nullable dengan tipe data Int?.

Properti idBalance memiliki anotasi @PrimaryKey yang menunjukkan bahwa properti ini adalah kunci utama dari entitas ini dan akan secara otomatis di-generate oleh sistem database.

Selain itu, setiap properti memiliki anotasi @ColumnInfo yang menunjukkan nama kolom dari properti tersebut di dalam tabel database.

Tabel ini berfungsi untuk mempermudah pengguna untuk memantau jumlah saldo yang akan selalu diperbarui ketika ada transaksi, restock, return, dsb.

C. Tabel Laporan Keuangan (Balance Report)



Ini adalah kelas Kotlin dengan anotasi @Entity. Kelas ini merepresentasikan entitas "Laporan Saldo" yang akan disimpan dalam tabel bernama "balanceReport".

Tabel ini akan memiliki beberapa kolom, yang didefinisikan sebagai properti dalam kelas ini, seperti "idBalanceReport" sebagai kolom primary key, "totalBalance" sebagai kolom nilai saldo total, "status" sebagai kolom status, "typePayment" sebagai kolom tipe pembayaran, "reportTag" sebagai kolom tag laporan, "username" sebagai kolom nama pengguna, dan "timeAdded" sebagai kolom waktu ditambahkan. Properti properti ini juga diberi anotasi @ColumnInfo untuk memberi tahu Room tentang nama kolom yang sesuai dengan properti tersebut.

Tabel ini berfungsi untuk merekam keuangan yang masuk dan keluar dari sistem kasir. Data tersebut dapat berupa investasi atau modal dari pemilik usaha, proses restock ataupun return. Untuk laporan keunangan pada proses transaksi akan direkam secara mandiri pada tabel transaksi.

D. Tabel Keranjang (Cart)

```
@Entity(tableName = "cart")
data class Cart(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idItem")
    var idItem: Int = 0,

    @ColumnInfo(name = "idProduct")
    var idProduct: Int? = null,

    @ColumnInfo(name = "nameItem")
    var nameItem: String? = null,

    @ColumnInfo(name = "priceItem")
    var priceItem: Int? = 0,

    @ColumnInfo(name = "totalItem")
    var totalItem: Int? = null,

    @ColumnInfo(name = "profitItem")
    var itemProfit: Int? = 0,

    @ColumnInfo(name = "totalProfit ")
    var totalProfit: Int? = 0,
```

```
    @ColumnInfo(name = "totalpayment")
    var totalpayment: Int? = 0,

    @ColumnInfo(name = "username")
    var username: String? = null,

    @ColumnInfo(name = "status")
    var status: String? = null,

    @ColumnInfo(name = "idTransaction")
    var idTransaction: Int? = 0,

    @ColumnInfo(name = "productPhoto")
    val productPhoto: Bitmap

)
```

Kode program di atas adalah sebuah kelas data (data class) bernama "Cart" yang memiliki anotasi (@Entity) yang menunjukkan bahwa kelas tersebut merupakan sebuah tabel dalam basis data. Tabel ini memiliki idItem, idProduct, nameItem, priceItem, totalItem, itemProfit, totalProfit, totalpayment, username, status, idTransaction, dan productPhoto.

Kolom idItem dan idProduct bertipe data integer dan diatur sebagai primary key dan foreign key, sedangkan kolom nameItem, itemProfit, dan status bertipe data string. Kolom priceItem, totalItem, totalProfit, totalpayment, dan idTransaction bertipe data integer.

Selain itu, kolom productPhoto bertipe data Bitmap, yang dapat digunakan untuk menyimpan gambar produk. Ada juga beberapa anotasi seperti @PrimaryKey, @ColumnInfo, dan @NonNull, yang digunakan untuk mengkonfigurasi kolom tabel dan mendefinisikan aturan validasi data pada setiap kolom.

Tabel ini berfungsi untuk memberikan kemudahan bagi pengguna untuk mengetahui total harga dan seluruh data item barang ataupun jasa yang di-inputkan oleh pengguna. Ketika proses transaksi telah selesai, maka proses transaksi akan berjalan dan data yang ada pada keranjang akan di-update dan direkam untuk keperluan laporan barang keluar.

E. Tabel Barang (Product)

```
@Entity(tableName = "product")
data class  Product(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idProduct")
    var idProduct: Int = 0,

    @ColumnInfo(name = "nameProduct")
    var nameProduct: String? = null,

    @ColumnInfo(name = "brandProduct")
    var brandProduct: String? = null,

    @ColumnInfo(name = "priceProduct")
    var priceProduct: Int? = null,

    @ColumnInfo(name = "stockProduct")
    var stockProduct: Int? = null,

    @ColumnInfo(name = "size_product")
    var size_product: String? = null,

    @ColumnInfo(name = "totalPriceProduct")
    var totalPriceProduct: Int? = null,

    @ColumnInfo(name = "totalPurchases")
    var totalPurchases: Int? = null,

    @ColumnInfo(name = "profitProduct")
    var profitProduct: Int? = 0,

    @ColumnInfo(name = "productPhoto")
    val productPhoto: Bitmap,

    @ColumnInfo(name = "username")
    var username: String? = null,

    @ColumnInfo(name = "timeAdded")
    var timeAdded: String? = null

)
```

Kelas Product adalah representasi dari sebuah produk dalam sistem. Kelas ini memiliki beberapa atribut seperti idProduct yang merupakan identifier dari sebuah produk, nameProduct dan brandProduct yang masing-masing merepresentasikan nama dan merek dari produk.

PriceProduct yang menyimpan harga jual dari produk, stockProduct yang menyimpan jumlah stok produk, sizeProduct yang menyimpan ukuran dari produk, realPriceProduct yang menyimpan harga asli produk, totalPurchases yang menyimpan total pembelian produk, profitProduct yang menyimpan profit dari produk.

ProductPhoto yang menyimpan gambar dari produk, username yang menyimpan username dari pengguna yang menambahkan produk, dan timeAdded yang menyimpan waktu ditambahkannya produk ke sistem.

Kelas ini diimplementasikan sebagai entitas Room, sehingga dapat disimpan ke database lokal. Tabel ini berfungsi untuk menampung data barang atau produk yang ada pada toko kedalam sistem.

F. Tabel Restock (Restok)

```
@Entity(tableName = "restock")
data class Restock(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idRestock")
    var idRestock: Int = 0,

    @ColumnInfo(name = "idProduct")
    var idProduct: Int = 0,

    @ColumnInfo(name = "nameProduct")
    var nameProduct: String? = null,

    @ColumnInfo(name = "stockAdded")
    var stockAdded: Int? = null,

    @ColumnInfo(name = "totalPurchases")
    var totalPurchases: Int? = null,

    @ColumnInfo(name = "supplier")
    var supplier: String? = null,

    @ColumnInfo(name = "username")
    var username: String? = null,
```

```
    @ColumnInfo(name = "restockDate")
    var restockDate: String? = null

)
```

Kelas Restock merepresentasikan entitas restok barang pada aplikasi. Terdapat beberapa properti yang didefinisikan pada kelas ini, di antaranya adalah idRestock sebagai primary key, idProduct yang mereferensikan id produk yang di-restok, nameProduct sebagai nama produk, stockAdded sebagai jumlah stok tambahan yang di-restok, totalPurchases sebagai total pembelian produk tersebut, supplier sebagai nama supplier yang mensuplai produk, username sebagai username pengguna yang melakukan restok, dan restockDate sebagai tanggal restok dilakukan.

Kelas ini akan digunakan untuk melakukan operasi CRUD (create, read, update, delete) pada data restok di dalam database aplikasi.

Tabel ini berfungsi untuk merekam proses restock atau penambahan stok barang menurut tabel barang. Sehingga selain menambahkan data pada tabel, sistem akan secara otomatis melakukan update pada jumlah stok di tabel barang (Product).

G. Tabel Retur (Return)

```
@Entity(tableName = "return")
data class  Return(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idReturn")
    var idReturn: Int = 0,

    @ColumnInfo(name = "idCart")
    var idCart: Int? = null,

    @ColumnInfo(name = "nameItem")
    var nameItem: String? = null,

    @ColumnInfo(name = "totalItem")
    var totalItem: Int? = null,
```

```
    @ColumnInfo(name = "totalRefund")
    var totalRefund: Int? = null,

    @ColumnInfo(name = "note")
    var note: String? = null,

    @ColumnInfo(name = "username")
    var username: String? = null,

    @ColumnInfo(name = "returnDate")
    var returnDate: String? = null

)
```

Kode program tersebut merupakan definisi dari sebuah kelas Kotlin dengan nama "Return". Kelas ini memiliki beberapa properti seperti "idReturn", "idCart", "nameItem", dan sebagainya. Properti-properti tersebut ditandai dengan anotasi "@ColumnInfo" yang menunjukkan bahwa properti tersebut dapat disimpan di database. Kelas ini juga diberi anotasi "@Entity" dan disertai dengan nama tabel di dalam database.

Kelas "Return" ini mewakili entitas dalam aplikasi yang berhubungan dengan pengembalian produk. Setiap kali sebuah produk dikembalikan oleh pelanggan, data pengembalian tersebut dapat disimpan dalam bentuk objek "Return" di dalam database. Properti-properti pada kelas ini menggambarkan informasi-informasi penting yang perlu disimpan terkait pengembalian produk tersebut, seperti id keranjang (cart), nama produk, jumlah yang dikembalikan, catatan, dan sebagainya.

Tabel ini berfungsi untuk merekam proses return menurut permintaan kostumer. Sistem akan merespon dengan menyimpan data dan melakukan update pada data transaksi secara spesifik menurut id dari item.

H. Tabel Jasa Pelayanan (Services)

Kode program ini adalah sebuah kelas Kotlin dengan nama Services yang ditandai dengan anotasi @Entity, yang digunakan untuk mewakili tabel "services" pada database.

Kelas ini memiliki beberapa atribut seperti idServices, serviceName, serviceMaterialPrice, serviceFinalPrice, serviceProfit, estimatedTime, username, dan timeAdded, masing-masing ditandai dengan anotasi @ColumnInfo yang menunjukkan nama kolom di tabel database yang sesuai dengan atribut tersebut. Atribut idServices ditandai dengan anotasi @PrimaryKey dan autoGenerate=true, yang menunjukkan bahwa atribut tersebut akan menjadi primary key pada tabel dan nilainya akan secara otomatis di-generate oleh sistem.

Tabel "services" pada database akan menyimpan informasi tentang layanan yang ditawarkan, termasuk nama layanan, harga material, harga final, keuntungan, estimasi waktu pengerjaan, nama pengguna, dan waktu ditambahkan. Tabel ini berfungsi untuk menampung data jasa pelayanan yang di-inputkan user kedalam sistem.

I. Tabel Transaksi (Transaction)

```
@Entity(tableName = "transaction")
data class Transaction(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idTransaction")
    var idTransaction: Int = 0,

    @ColumnInfo(name = "totalTransaction")
    var totalTransaction: Int? = null,

    @ColumnInfo(name = "profitTransaction")
    var profitTransaction: Int? = null,

    @ColumnInfo(name = "moneyReceived")
    var moneyReceived: Int? = null,

    @ColumnInfo(name = "moneyChange")
    var moneyChange: Int? = null,

    @ColumnInfo(name = "totalItem")
    var totalItem: Int? = null,

    @ColumnInfo(name = "username")
    var username: String? = null,

    @ColumnInfo(name = "typePayment")
    var typePayment: String? = null,

    @ColumnInfo(name = "transactionDate")
    var transactionDate: String? = null,

    @ColumnInfo(name = "proofPhoto")
    val proofPhoto: ByteArray,

    @ColumnInfo(name = "transactionType")
    val transactionType: String? = null

)
```

Ini adalah kode program Kotlin untuk membuat model entitas "Transaction" yang akan digunakan dalam aplikasi. Entitas ini merepresentasikan sebuah transaksi yang terjadi dalam bisnis.

Setiap transaksi memiliki ID, total transaksi, keuntungan, jumlah uang yang diterima, jumlah kembalian, jumlah barang yang dibeli, nama pengguna, jenis pembayaran, tanggal transaksi, foto bukti, dan jenis transaksi. ID adalah nilai unik untuk setiap transaksi dan diberikan secara otomatis dengan menggunakan opsi "autoGenerate". Total transaksi dan keuntungan adalah nilai numerik yang merepresentasikan jumlah uang yang terlibat dalam transaksi tersebut.

Jumlah uang yang diterima dan kembalian juga disimpan sebagai nilai numerik. Jumlah barang dan jenis pembayaran juga disimpan sebagai data numerik dan teks. Tanggal transaksi dan foto bukti disimpan sebagai data teks dan gambar. Jenis transaksi juga disimpan sebagai data teks.

Tabel transaksi ini berfungsi untuk merekam proses transaksi dalam bentuk record tabel. Dengan penambahan record pada tabel ini, sistem akan mengupdate beberapa tabel sekaligus seperti saldo pada tabel Balance, update data status pada tabel Cart, dan menjadi data utama yang akan ada pada struk.

J. Tabel User

```
@Entity(tableName = "users")
data class Users(

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "idUser")
    var idUser: Int = 0,

    @ColumnInfo(name = "fullname")
    var fullname: String? = null,

    @ColumnInfo(name = "username")
    var username: String? = null,
```

```
    @ColumnInfo(name = "email")
    var email: String? = null,

    @ColumnInfo(name = "password")
    var password: String? = null,

    @ColumnInfo(name = "role")
    var role: String? = null,

    @ColumnInfo(name = "photoUser")
    var photoUser: Bitmap

)
```

Kode program yang diberikan adalah sebuah data class dengan nama Users yang didekorasi dengan anotasi @Entity dan memiliki tabel dengan nama "users" di database.

Tabel ini memiliki beberapa kolom yang dijelaskan oleh beberapa anotasi @ColumnInfo seperti "idUser" yang menjadi primary key, "fullname" bertipe data String yang menyimpan nama lengkap pengguna, "username" bertipe data String yang menyimpan username pengguna, "email" bertipe data String yang menyimpan alamat email pengguna, "password" bertipe data String yang menyimpan password pengguna, "role" bertipe data String yang menyimpan peran pengguna dalam aplikasi, dan "photoUser" sebagai Bitmap yang menyimpan foto pengguna.

Tabel ini digunakan oleh user untuk dapat melakukan *login session* pada sistem ini, serta diperlukan untuk beberapa pendataan seperti proses transaksi, restock, dsb.

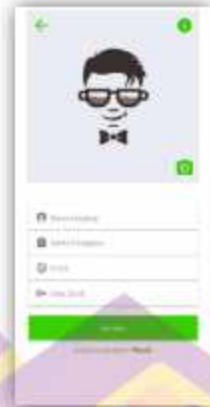
4.2 Implementasi Tampilan

4.2.1 Tampilan Halaman *Application Introduction Slider*

Gambar 4.1 Halaman *Application Introduction Slider*

Dalam halaman Application Introduction Slider ini berisi pengenalan 3 fitur utama aplikasi Manajemen Barang dan Jasa, Manajemen Transaksi, serta Manajemen Laporan. Pengguna bisa mengusap atau *swipe* layer ke kanan atau ke kiri untuk melihat pengenalan fitur tersebut. Terakhir ada tombol *Next* untuk melakukan skip pada halaman Application Introduction Tersebut. Tampilan dari halaman ini ada pada gambar 4.1.

4.2.2 Tampilan Halaman *Registration*



Gambar 4.2 Halaman *Registration*

Pada halaman registrasi yang dapat dilihat pada gambar 4.2, calon pengguna dapat melakukan pendaftaran, data yang diperlukan berupa nama lengkap, username, email, password, dan foto profil. Setelah pengguna mendaftarakan mencari data identik untuk menghindari penumpukan data pada database. Jika ditemukan data identik atau kesalahan data, maka sistem akan mengeluarkan notifikasi seperti gambar 4.3.

Gambar 4.3 Halaman *Invalid Registration*

4.2.3 Tampilan Halaman *Login*

Gambar 4.4 Halaman *Login*

Pada Halaman *Login* seperti gambar 4.4 diatas, pengguna harus memasukan email serta password yang terdaftar pada sistem. Setelah data dimasukan dengan benar dan pengguna menekan tombol *Sign In*, maka sistem secara otomatis akan memulai *session login*. Adapun jika terjadi kesalahan pada saat proses *login*, maka akan muncul notifikasi seperti gambar 4.5 berikut ini.

Gambar 4.5 Halaman *Invalid Login*

4.2.4 Tampilan Halaman *Dashboard*

Gambar 4.6 Halaman *Dashboard*

Pada halaman *Dashboard* ini seperti gambar 4.6 diatas, pengguna dapat me-*monitoring* seluruh data yang ada pada database seperti informasi keuangan, stok, transaksi, keuntungan, serta tombol untuk navigasi ke fitur-fitur utama sistem ini.

Gambar 4.7 Halaman *Dashboard (Balance Method Dialog)*

Fitur tersebut berupa manajemen produk atau barang, jasa pelayanan, transaksi, dsb. Pengguna dapat memantau data keuangan dan transaksi yang terjadi per-hari sistem ini dibuka. Fitur untuk menambah atau menarik saldo bisa diakses dengan menekan icon withdraw atau plus di layar dan akan muncul dialog seperti pada gambar 4.7 diatas.

Pengguna juga dapat melakukan manajemen keuangan dengan menanam saldo untuk modal dan penarikan saldo. Sistem akan secara otomatis merekam laporan keuangan tersebut dan dapat dipantau langsung oleh pengguna. Perbedaan tampilan fitur withdraw dan tambah saldo atau investasi dapat dilihat pada gambar 4.8 dibawah..

Gambar 4.9 Halaman *Dashboard (Withdraw & Invest)*

4.2.5 Tampilan Halaman *Product List*

Gambar 4.8 Halaman *Product List*

Pada halaman *Product list*, pengguna dapat melihat data barang yang diinputkan pada sistem. Halaman ini dilengkapi fitur *searchView* atau pencarian, dimana pengguna dapat mencari data barang dengan kata kunci nama barang. Pengguna juga dapat melakukan filter terhadap data barang tersebut sesuai dengan urutan alfabet, waktu penginputan, harga, dsb.

Data barang yang tampil berupa nama barang, waktu penginputan, harga, perhitungan keuntungan per-item, jumlah stok, merk, ukuran, dan foto. Pengguna dapat melihat informasi lebih lanjut dengan menekan ikon panah kebawah dan ikon informasi yang ada disebelah jumlah stok. Tampilan dari halaman *product list* dapat dilihat dari gambar 4.9 diatas.

Beberapa dialog akan muncul dengan menekan tombol tertentu, dialog tersebut akan tampak seperti gambar 4.10 dibawah ini. Terdapat fitur usap untuk menghapus juga pada halaman ini, serta terdapat informasi detail untuk stok.

Gambar 4.10 Halaman *Product List (Dialog)*

Data barang tersebut juga dapat dihapus oleh pengguna dengan melakukan swipe atau usap pada layer kea rah kanan yang akan diikuti dengan munculnya sebuah dialog konfirmasi hapus data. Dialpg tersebut akan mengurangi *human error* yang kemungkinan terjadi seperti layer terusap atau tertekan secara tidak sengaja.

Halaman Product List ini juga sudah menerapkan *Bottom Navigation Menu* untuk membantu pengguna untuk bermavigasi atau berpindah dari halaman satu ke halaman yang lain.

4.2.6 Tampilan Halaman *Add Product*

Gambar 4.11 Halaman *Add Product*

Pada halaman *Add Product* seperti gambar 4.11 diatas, pengguna dapat menambahkan data barang dengan melakukan beberapa input data melalui form yang disediakan oleh sistem. Beberapa data yang harus diinputkan seperti nama barang, ukuran, merk, dsb. Ada beberapa data yang akan secara otomatis diproses oleh sistem setelah pengguna menyelesaikan input data pada form. Seperti harga asli per-item dan keuntungan per-item.

Saat pengguna menekan tombol simpan, sistem akan secara otomatis melakukan validasi data yang diinputkan untuk mengurangi data yang tidak akurat masuk kedalam database, dan ketika ada data yang kosong atau kurang tepat, maka akan keluar sebuah dialog error yang menginformasikan pada pengguna tentang kesalahan data yang terjadi.

Dialog untuk pop up keberhasilan proses dan metode pembayaran akan muncul ketika proses sudah divalidasi oleh sistem. Dialog tersebut akan ditampilkan seperti pada gambar 4.12.

Gambar 4.12 Halaman *Add Product (Dialog)*

Setelah data yang diinputkan telah memenuhi syarat, maka akan muncul sebuah dialog pilihan metode pembayaran yang akan secara otomatis mengurangi saldo dari tipe pembayaran yang dipilih.

4.2.7 Tampilan Halaman *Edit Product*

Gambar 4.13 Halaman *Edit Product*

Pada halaman *Edit Product* seperti pada gambar 4.13 diatas, pengguna dapat melakukan pembaruan detail data barang seperti nama barang, ukuran, merk, foto, dan harga jual. Sistem juga akan melakukan kalkulasi secara otomatis sesaat ketika pengguna melakukan penginputan data pada form, seperti harga asli baru per-item dan keuntungan baru per-item.

Halaman ini juga dilengkapi dengan validasi form, sistem akan melakukan cek data pada form untuk menghindari data tidak akurat atau kosong yang dapat masuk kedalam database.

Gambar 4.14 Halaman *Edit Product (Dialog)*

Dialog untuk *pop up* keberhasilan proses dan metode pembayaran akan muncul ketika proses sudah divalidasi oleh sistem. Dialog tersebut akan ditampilkan seperti pada gambar 4.14.

4.2.8 Tampilan Halaman *Restock Product*

Gambar 4.15 Halaman *Restock Product*

Pada halaman *Restock Product* seperti pada gambar 4.15 diatas, pengguna dapat melakukan penambahan jumlah stok baru pada barang tertentu yang ada pada database. Pengguna diharuskan untuk mengisi form seperti jumlah stok barang, total harga beli stok, dan supplier.

Gambar 4.16 Halaman *Restock Product (Invalid)*

Sistem akan melakukan kalkulasi sesaat ketika pengguna melakukan input data, seperti memperhitungkan harga asli baru per-item dan keuntungan baru per-item.

Halaman ini juga dilengkapi validasi form, setelah pengguna menekan tombol simpan maka sistem akan melakukan cek pada form untuk menghindari data tidak akurat atau kosong masuk kedalam database. Sistem akan menampilkan dialog seperti pada gambar 4.16 diatas.

Setelah data yang diinputkan telah memenuhi syarat, maka akan muncul sebuah dialog pilihan metode pembayaran yang akan secara otomatis mengurangi saldo dari tipe pembayaran yang dipilih.

4.2.9 Tampilan Halaman *Services List*

Gambar 4.17 Halaman *Service List*

Pada halaman *Services List* seperti pada gambar 4.17 diatas, pengguna dapat melihat data jasa pelayanan yang ada pada database sistem, data jasa tersebut berupa nama jasa, harga, keuntungan, dan estimasi waktu.

Pengguna dapat melakukan perubahan data dengan menekan ikon pensil atau edit pada data pelayanan tersebut, seperti pada gambar 4.18 dibawah. Halaman ini juga sudah dilengkapi *Bottom Navigation Menu* yang memudahkan pengguna untuk berpindah dari halaman satu ke halaman yang lain.

Gambar 4.18 Halaman *Service List (Dialog)*

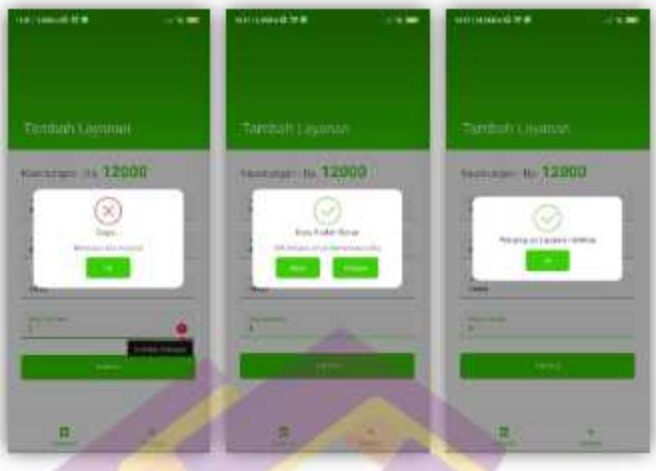
Pengguna juga dapat melakukan hapus data dengan melakukan swipe atau usap layer ke kanan yang akan diikuti oleh sebuah dialog konfirmasi hapus data. Hal tersebut akan mengurangi *human error* seperti layer tertekan atau terusap secara tidak sengaja. Dialog yang akan muncul ketika pengguna telah mengusap adalah seperti pada gambar 4.18 diatas.

4.2.10 Tampilan Halaman *Add Services*

Gambar 4.19 Halaman *Add Services*

Pada halaman *Add Service* seperti pada gambar 4.19 diatas, pengguna dapat menambahkan data jasa pelayanan dengan mengisi form yang disediakan sistem. Pengguna dapat mengisi data seperti data nama, harga material, harga jual dan estimasi waktu. Sistem juga akan melakukan kalkulasi secara otomatis sesaat setelah pengguna menginputkan data jasa pelayanan. Seperti harga asli, dan keuntungan per-jasa.

Saat pengguna menekan tombol simpan, sistem akan secara otomatis melakukan validasi data yang diinputkan untuk mengurangi data yang tidak akurat masuk kedalam database, dan ketika ada data yang kosong atau kurang tepat, maka akan keluar sebuah dialog error seperti pada gambar 4.20 dibawah yang menginformasikan pada pengguna tentang kesalahan data yang terjadi.



Gambar 4.20 Halaman *Add Services (Dialog)*

Setelah data yang diinputkan telah memenuhi syarat, maka akan muncul sebuah dialog pilihan metode pembayaran yang akan secara otomatis mengurangi saldo dari tipe pembayaran yang dipilih.

4.2.11 Tampilan Halaman *Edit Services*

Gambar 4.21 Halaman *Edit Services*

Pada halaman *Edit Service* seperti pada gambar 4.21 diatas, pengguna dapat melakukan perubahan data pada jasa pelayanan seperti nama jasa, harga material, harga jual dan estimasi waktu.

Sistem juga akan melakukan *update* secara otomatis pada beberapa data seperti harga asli baru per-item, dan keuntungan baru per-item. Halaman ini juga dilengkapi dengan validasi form seperti pada gambar 4.22, sistem akan melakukan cek data pada form untuk menghindari data tidak akurat atau kosong yang dapat masuk kedalam database.

Gambar 4.22 Halaman *Edit Services (Dialog)*

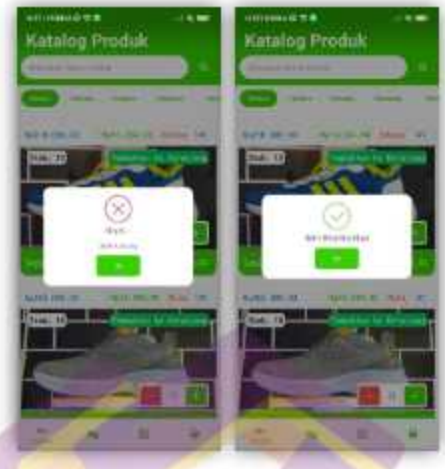
4.2.12 Tampilan Halaman *Transaction Product*

Gambar 4.23 Halaman *Transaction Product*

Halaman ini termasuk pada fitur utama sistem yaitu yang memudahkan pengguna atau kasir dalam melakukan proses transaksi. Halaman tersebut ditampilkan seperti pada gambar 4.23 diatas, *bottom navigation menu* juga sudah diterapkan pada halaman ini.

Pada halaman *Transaction Product*, berisi list data product atau data barang yang mirip dengan halaman Product List. Hanya saja ikon pensil atau fitur edit pada tiap data dihilangkan. Pada data tersebut, dilengkapi sebuah ikon *plus* dan *minus* serta QTY yang berguna untuk menambahkan jumlah stok pada data tertentu. Seiring dengan bertambahnya QTY tersebut, maka harga jual akan dikali jumlah QTY yang akan membuat sebuah variabel baru yaitu total harga.

Halaman ini digunakan untuk menambahkan data barang dengan jumlah tertentu kedalam keranjang. Sehingga setelah menentukan jumlah QTY atau jumlah barang, pengguna dapat menekan tombol "*Add to Cart*" untuk menyimpan data tersebut pada keranjang.



Gambar 4.24 Halaman *Transaction Product (Dialog)*

Fitur validasi data melengkapi halaman ini, seperti saat pengguna menekan tombol "*Add to Cart*" tanpa menyesuaikan jumlah barang pada QTY, maka akan muncul dialog error pada layar pengguna. Pengguna juga tidak dapat berpindah ke halaman Payment ketika keranjang kosong. Dialog tersebut bisa dilihat pada gambar 4.24 diatas.

4.2.13 Tampilan Halaman *Transaction Services*

Gambar 4.25 Halaman *Transaction Services*

Pada halaman *Transaction Service* seperti pada gambar 4.25 diatas, berisi list data jasa pelayanan yang mirip dengan halaman Services List. Hanya saja ikon pensil atau fitur edit pada tiap data dihilangkan.

Gambar 4.26 Halaman *Transaction Services (Dialog)*

Pada data tersebut, dilengkapi sebuah ikon *plus* dan *minus* serta QTY yang berguna untuk menambahkan jumlah jasa yang diperlukan pada data tertentu. Seiring dengan bertambahnya QTY tersebut, maka harga jual akan dikali jumlah QTY yang akan membuat sebuah variable harga total baru.

Halaman ini digunakan untuk menambahkan data jasa pelayanan dengan jumlah tertentu kedalam keranjang. Sehingga setelah menentukan jumlah QTY atau jumlah barang, sebuah form baru akan muncul yaitu pengguna harus menginputkan nama kostumer, kemudian dilanjutkan dengan menginputkan data tersebut menggunakan ikon *Add*.

Fitur validasi data melengkapi halaman ini, seperti pada saat pengguna menekan ikon *Add* tanpa menyesuaikan jumlah barang pada QTY, maka akan muncul dialog error pada layar pengguna. Pengguna juga tidak dapat berpindah ke halaman Payment ketika keranjang kosong. Dialog untuk validasi ini sama dengan yang ada pada halaman Transaction Product, seperti pada gambar 4.26 diatas.

4.2.14 Tampilan Halaman *Payment Product*

Gambar 4.27 Halaman *Payment Product*

Tampilan halaman ini dapat dilihat pada gambar 4.27, pengguna dapat melihat data keranjang seperti total pembayaran, dan list data barang. Ini merupakan halaman pembayaran barang, sehingga setelah kostumer membayar total pembayaran dengan sejumlah uang, uang yang diterima harus diinputkan ke form penerimaan uang yang tersedia, kemudian sistem akan melakukan kalkulasi uang kembalian secara otomatis.

Gambar 4.28 Halaman *Payment Product (Dialog)*

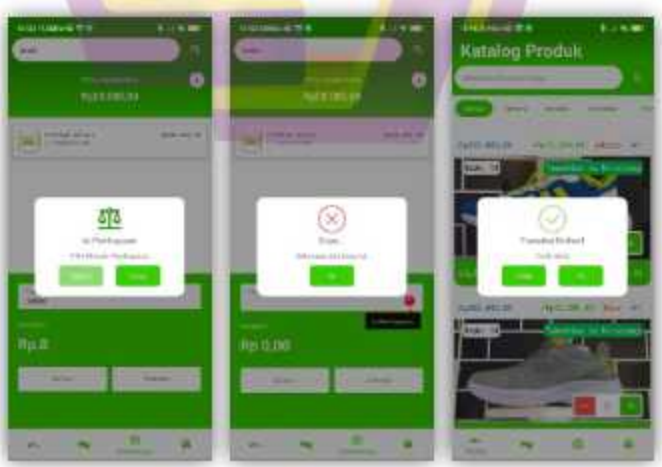
Fitur validasi juga melengkapi halaman ini, fitur ini akan berjalan ketika tombol simpan ditekan. Sistem akan mengambil data uang penerimaan, jika uang tersebut lebih kecil dari total pembayaran maka akan keluar sebuah dialog *error* seperti pada gambar 4.28.

Setelah data sudah valid, maka akan keluar tombol metode pembayaran digital atau tunai. Ketika proses transaksi sudah selesai, maka sistem akan secara otomatis mencetak struk dengan mengambil data-data pada tabel transaksi dan keranjang tersebut.

4.2.15 Tampilan Halaman *Payment Services*

Gambar 4.29 Halaman *Payment Services*

Pada halaman *Payment Services* yang dapat dilihat pada gambar 4.29 diatas ini, pengguna harus menginputkan nama kostumer terlebih dahulu untuk melihat data jasa pelayanan yang dipesan pada *searchView* atau kolom pencarian.



Gambar 4.30 Halaman *Payment Services (Dialog)*

Pengguna dapat mencetak struk pra-jasa dengan menekan tombol *Print,* dan apabila jasa pelayanan sudah selesai maka pengguna bisa menekan tombol simpan untuk mengakhiri proses transaksi dan mengubah status data pelayanan pada database.

Fitur validasi juga melengkapi halaman ini, fitur ini akan berjalan ketika tombol simpan ditekan. Sistem akan mengambil data uang penerimaan, jika uang tersebut lebih kecil dari total pembayaran maka akan keluar sebuah dialog *error* yang sama seperti pada halaman Payment Product sebelumnya.

Setelah data sudah valid, maka akan keluar tombol metode pembayaran digital atau tunai. Ketika proses transaksi sudah selesai, maka sistem akan secara otomatis mencetak struk dengan mengambil data-data pada tabel transaksi dan keranjang tersebut. Dialog tersebut dapat dilihat pada gambar 4.30 diatas.

4.2.16 Tampilan Halaman *Return Product*

Gambar 4.31 Halaman *Return Product*

Pada halaman *Return Product* seoertu gambar 4.31 diatas, pengguna dapat memproses permintaan retur dari kostumer untuk mendata pengembalian barang dengan cara memasukkan id transaksi pada *searchView* atau kolom pencarian. Setelah data ditemukan, maka akan muncul list transaksi, data tersebut berupa data barang yang sudah dibayar.

Data tersebut juga dilengkapi ikon *plus* dan *minus* serta QTY untuk membantu pengguna untuk melakukan kalkulasi jumlah *refund* atau pengembalian dana yang diambil dari total harga dari jumlah QTY yang disesuaikan. Ketika ikon tersebut ditekan, maka akan keluar sebuah form baru yaitu catatan retur. Pengguna dapat menginputkan alasan kostumer tersebut melakukan retur. Ketika tombol *Return* ditekan, maka sistem akan melakukan proses perhitungan kerugian dan melakukan pembaruan status pada barang yang di-retur. Jika terdapat kekurangan data atau data tidak valid, akan muncul dialog seperti dibawah, akan tetapi jika validasi sistem sukses maka akan muncul dialog sukses, seperti gambar 4.32 dibawah.

Gambar 4.32 Halaman *Services Product (Dialog)*

4.2.17 Tampilan Halaman *Transaction Report*

Gambar 4.33 Halaman *Transaction Report*

Pada halaman *Transaction Report* yang dapat dilihat pada gambar 4.33 ini, pengguna dapat melihat seluruh data transaksi yang dilakukan dengan sistem ini. List yang ada pada halaman ini akan memperlihatkan laporan transaksi secara sederhana dengan menampilkan jumlah pembayaram, total keuntungan, dan waktu transaksi. Pengguna dapat melihat detail transaksi dengan menekan ikon mata pada data tersebut.

Pengguna juga dapat melakukan filter pada laporan transaksi, filter tersebut berupa laporan dengan rentan semua waktu, per-bulan, atau per-hari dimana sistem ini dibuka. Sehingga memudahkan pengguna untuk melakukan *Monitoring Data*.

Halaman ini juga sudah dilengkapi *Bottom Navigation Menu* yang memudahkan pengguna untuk bernavigasi atau berpindah dari halaman satu ke halaman yang lain. Pada gambar 4.34 dibawah, terdapat perbedaan tampilan laporan transaksi antara transaksi dengan pembayaran digital dan tunai. yaitu pada tombol bukti untuk menampilkan bukti pembayaran digital

Gambar 4.34 Halaman *Detail Transaction Report*

Pada gambar 4.35 dibawah ini, terdapat filter yang hanya bisa dipergunakan oleh admin. Filter tersebut berguna untuk melihat rekaman transaksi yang dilakukan oleh seluruh pengguna yang terdaftar pada sistem.

Gambar 4.35 Halaman Transaction Report

4.2.18 Tampilan Halaman *Accounting*

Gambar 4.36 Halaman *Accounting*

Pada halaman *Accounting* atau Pembukuan yang dapat dilihat pada gambar 4.36 diatas, pengguna dapat melihat rekap pembukuan bulanan dalam wujud list per-bulan. Pada tiap pembukuan berisikan keterangan bulan, username dari pengguna yang membuat rekap data, id pembukuan, data saldo awal, stok awal, keuangan, transaksi, retur, restock, keperluan, saldo akhir, dan stok akhir. Pengguna dapat melihat data secara detail dengan menekan ikon panah kebawah.

Pengguna juga dapat melakukan *update* atau pembaruan data pada pembukuan dengan menekan tombol *Add* yang akan mengarahkan pengguna ke halaman Generating Monthly Accounting yang sebelumnya diikuti dialog konfirmasi, dialog tersebut dapat dilihat seperti pada gambar 4.37 dibawah ini.

Gambar 4.37 Halaman *Accounting (Dialog)*

4.2.19 Tampilan Halaman *Generating Monthly Accounting*

Gambar 4.38 Halaman *Generating Monthly Accounting*

Pada halaman *Generating Monthly Accounting* yang dapat dilihat pada gambar 4.38 diatas, pengguna akan diperlihatkan data rekap pembukuan sementara yang dibuat secara otomatis oleh system.

Gambar 4.39 Format *Accounting Report*

Pembaruan otomatis tersebut dilakukan tiap bulannya atau data pembukuan yang telah diubah atau dilanjutkan oleh pengguna per-bulan dimana sistem ini dibuka. Untuk melakukan generate data, pengguna cukup menekan ikon *New Folder*. Pengguna juga dapat meneruskan laporan tersebut dalam format chat dengan menekan ikon *whatsapp* dan file pdf dengan menekan ikon pdf pada gambar 4.38 diatas. Hasil dari format kedua file seperti pada gambar 4.39 diatas.

4.2.20 Tampilan Halaman *Balance Report*

Gambar 4.40 Halaman *Balance Report*

Pada halaman *Balance Report* yang dapat dilihat pada gambar 4.40 diatas, pengguna dapat melihat laporan saldo masuk dan keluar yang tipe saldonya dibedakan oleh warna. Data tersebut berisi total saldo, keterangan masuk atau keluar, dan tag dari laporan. Halaman ini tidak menampung rekaman data saldo masuk dari proses transaksi karena pengguna dapat melihatnya pada halaman *Transaction Report*.

4.2.21 Tampilan Halaman *Product Report*

Gambar 4.41 Halaman *Product Report*

Pada halaman *Product Report* ini, berisi laporan barang masuk atau keluar dari sistem. Laporan barang masuk berisikan laporan dari Restok barang, sedangkan barang keluar merupakan laporan dari barang yang sudah dibeli oleh kostumer. Kedua laporan tersebut dapat diakses dengan menekan filter yang ada di bagian atas halaman.

Laporan barang keluar diambil dari rekaman transaksi dari tiap-tiap keranjang. Sedangkan laporan restok diambil dari rekaman restok pada database. Tampilan dari halaman ini dapat dilihat pada gambar 4.41 diatas,

4.2.22 Tampilan Halaman *Settings*

Gambar 4.42 Halaman *Settings*

Halaman *settings* seperti gambar 4.41 diatas, memuat pengaturan untuk setelah bahasa aplikasi dan koneksi printer thermal via bluetooth. Bagian tentang juga memuat informasi mengenai aplikasi dan pembuat aplikasi serta informasi utnuk kontaknya.

4.2.23 Tampilan Halaman *Profile*

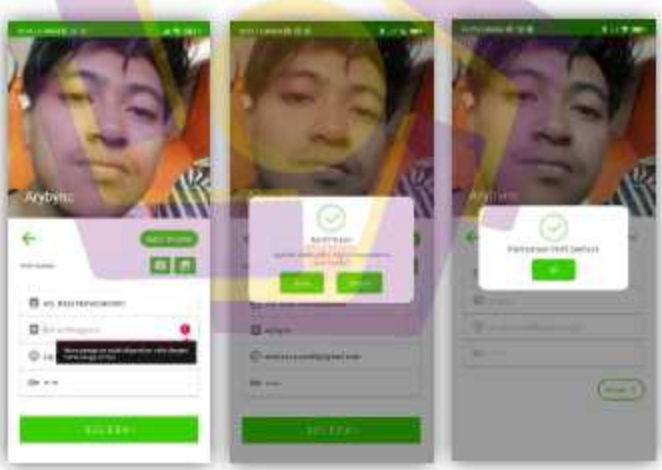
Gambar 4.43 Halaman *Profile*

Pada halaman *Profile* yang dapat dilihat pada gambar 4.43, berisikan data akun dari pengguna. Yaitu foto profil, username, nama lengkap, email, password. Terdapat fitur edit yang dapat diakses pengguna dengan menekan tombol *edit*. Untuk mengakhiri sesi login, pengguna dapat melakukan *logout* atau *sign out* dengan menekan tombol *logout*.

Untuk bagian *Edit Profile* seperti pada gambar 4.44. pengguna dapat mengubah seluruh data akun. Dengan menekan tombol edit sebelumya, sistem akan menyembunyikan seluruh informasi pada tampilan profil dan mengubah halaman menjadi form edit profil.

Untuk validasi sistem, ketika pengguna telah selesai melakukan update dan menekan tombol selesai, maka sistem akan melakukan koreksi seperti pada gambar 4.45 dibawah ini. Dialog akan muncul sesuai hasil validasi dari sistem.

Gambar 4.44 Halaman *Profile (Edit)*



Gambar 4.45 Halaman *Profile (Dialog)*

4.3 Pengujian Menggunakan Blackbox Testing

Pengujian dalam sebuah sistem memiliki tujuan untuk membuktikan bahwa sistem telah berfungsi dengan baik berdasarkan uji coba kepada sistem dan program. Pengujian yang dilakukan dalam sistem ini yaitu dengan menggunakan metode *blackbox*. Pengujian dengan metode *blackbox* merupakan pengujian yang lebih memfokuskan terhadap uji fungsionalitasn suatu sistem dalam menjalankan seluruh sistem manajemen surat yang ada. Selain itu, pengujian ini juga mampu mengetahui kemampuan sebuah sistem dalam menangani kekeliruan yang dilakukan oleh pengguna. Hasil dari pengujian *blackbox* tersebut dapat dilihat pada tabel 4.1 hingga tabel 4.17.

4.3.1 Kerangka pengujian *registrasi* pengguna

Tabel 4.1 Kerangka pengujian *registrasi* pengguna

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Menginputkan data akun dengan benar pada form pendaftaran di halaman registrasi	Pengguna mampu mendaftarkan akun dengan sukses dan akun dapat digunakan untuk login kedalam sistem	Pendaftaran berhasil dan pengguna diarahkan ke halaman login	Berhasil
Menginputkan data akun yang menduplikasi data yang sudah ada dalam sistem	Pengguna tidak bisa mendaftarkan akun dengan data yang terduplikasi atau sudah digunakan oleh pengguna lain	*Pop up* error yang menginformasikan bahwa data yang di-inputkan sudah digunakan oleh pengguna lain	Berhasil
Tidak menginputkan data akun secara lengkap atau tidak valid	Pengguna tidak memasukan data yang tidak valid kedalam sistem	Tampil sebuah dialog error yang menginformasikan bahwa data tidak lengkap	Berhasil

4.3.2 Kerangka pengujian *login* pengguna

Tabel 4.2 Kerangka pengujian *login* pengguna

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Menginputkan email dan password dengan benar	Pengguna dapat memasuki halaman *dashboard* utama sistem dan sesi login dimulai	Terdapat *Pop up* login sukses dan pengguna diarahkan ke halaman *dashboard*	Berhasil
Menginputkan email dan password dengan salah	Mengurangi pihak tidak bertanggung jawab masuk kedalam sistem	Terdapat *Pop up login* gagal dan pengguna diharuskan untuk menginputkan data dengan benar	Berhasil

4.3.3 Kerangka pengujian manajemen data saldo

Tabel 4.3 Kerangka pengujian manajemen data saldo

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Menginputkan total saldo yang di-investasikan atau di tarik dengan benar	Data saldo yang inputkan tercatat dalam sistem dan *update* tampilan secara otomatis	*Cardview* bagian data yang terkait saldo akan ter-*update*	Berhasil
Menginputkan total saldo yang di-investasikan atau di tarik dengan kosong atau tidak valid	Mengurangi kesalahan data saldo yang ditambahkan kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada penginputan data dan pengguna diharuskan untuk memperbaiki data yang ingin diinputkan	Berhasil

4.3.4 Kerangka pengujian manajemen data barang

Tabel 4.4 Kerangka pengujian manajemen data barang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat detail data barang yang tersimpan menurut filter yang telah ditentukan atau secara *default*.	Data barang akan tampil sesuai kriteria atau secara *default* dan pengguna dapat melihat detail data tersebut dengan menekan tombol panah bawah	List data barang muncul secara default atau menurut filter yang dipilih pengguna.. Monitoring data barang pada sistem tampil melalui *cardView*	Berhasil
Menginputkan data barang baru dengan benar	Data barang, dan restok dapat disimpan oleh sistem, serta pembaruan data saldo untuk pembelian stok dan halaman barang akan melakukan *refresh* secara otomatis	Terdapat dialog sukses yang menginformasikan bahwa proses penambahan data barang berhasil dan sistem akan mengarahkan pengguna ke halaman utama barang	Berhasil
Menginputkan data barang baru dengan data tidak lengkap atau format data yang salah	Mengurangi kesalahan data barang baru yang ditambahkan kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada penginputan data dan pengguna diharuskan untuk memperbaiki data yang ingin ditambahkan	Berhasil

Mengubah detail data pada barang tertentu dengan benar	Data barang yang dipilih dapat ter-*update* dan halaman barang akan melakukan *refresh* secara otomatis	Terdapat dialog sukses yang menginformasikan bahwa proses pengubahan detail data barang berhasil dan sistem akan mengarahkan pengguna ke halaman utama barang	Berhasil
Mengubah detail data pada barang tertentu dengan salah atau data tidak lengkap	Mengurangi kesalahan data barang yang diubah kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada pengubahan data dan pengguna diharuskan untuk memperbaiki data yang ingin diubah	Berhasil
Menambahkan stok barang (restok) pada barang tertentu dengan benar	Data restok barang yang ditambahkan dapat disimpan oleh sistem, data saldo dan barang akan ter-*update* untuk keperluan pembelian dan penambahan stok dan halaman barang akan melakukan *refresh* secara otomatis	Terdapat dialog sukses yang menginformasikan bahwa proses penambahan data restok barang berhasil dan sistem akan mengarahkan pengguna ke halaman utama barang	Berhasil

Menambahkan stok barang (restok) pada barang tertentu dengan salah atau data tidak lengkap	Mengurangi kesalahan data restok barang yang diubah kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada penambahan data dan pengguna diharuskan untuk memperbaiki data yang ingin diubah	Berhasil
Menghapus data barang tertentu dengan melakukan usap barang ke arah kanan dengan konfirmasi penghapusan	Data yang dipilih akan terhapus dengan konfirmasi penghapusan data yang disetujui oleh pengguna dan halaman barang akan melakukan *refresh* secara otomatis	Data barang dapat terhapus setelah pengguna mengkonfirmasi penghapusan pada dialog yang muncul setelah melakukan usap barang ke kanan	Berhasil
Melakukan membatalkan penghapusan data tertentu yang sudah terlanjur di usap ke kanan	Mengurangi *human error* yang kemungkinan terjadi ketika pengguna tidak sengaja melakukan usap barang ke kanan sehingga data barang batal dihapus	Data barang batal terhapus setelah pengguna membatalkan penghapusan pada dialog yang muncul setelah melakukan usap barang ke kanan	Berhasil
Melakukan pencarian dengan mengisi searchView yang disediakan dengan nama barang dengan benar	Membantu pengguna untuk menemukan data barang tertentu serta tersimpan dalam sistem	Data barang tampil dalam wujud list yang berisi data yang dimasukan pengguna pada searchView	Berhasil

Melakukan pencarian dengan mengisi searchView dengan nama barang dengan salah atau kosong	Dapat dan meminimalisir kesalahan data yang tampil pada layar	Muncul Dialog error yang menginformasikan ke pengguna bahwa tidak ada data barang yang sesuai dengan data yang dimasukkan pada searchView pada sistem	Berhasil

4.3.5 Kerangka pengujian manajemen data pelayanan

Tabel 4.5 Kerangka pengujian manajemen data pelayanan

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat detail data pelayanan yang tersimpan menurut filter yang telah ditentukan atau secara *default*.	Data pelayanan akan tampil secara lengkap sesuai kriteria atau secara *default*	List data pelayanan muncul secara default atau menurut filter yang dipilih pengguna. Monitoring data pelayanan pada sistem tampil melalui *cardView*	Berhasil
Menginputkan data pelayanan baru dengan benar	Data pelayanan dapat disimpan oleh sistem dan halaman pelayanan akan melakukan *refresh* secara otomatis	Terdapat dialog sukses yang menginformasikan bahwa proses penambahan data pelayanan berhasil dan sistem akan mengarahkan pengguna ke halaman pelayanan	Berhasil

Menginputkan data pelayanan baru dengan data tidak lengkap atau format data yang salah	Mengurangi kesalahan data pelayanan baru yang ditambahkan kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada penginputan data dan pengguna diharuskan untuk memperbaiki data yang ingin ditambahkan	Berhasil
Mengubah detail data pada pelayanan tertentu dengan benar	Data pelayanan yang dipilih dapat ter-*update* dan halaman pelayanan akan melakukan *refresh* secara otomatis	Terdapat dialog sukses yang menginformasikan bahwa proses pengubahan detail data pelayanan berhasil dan sistem akan mengarahkan pengguna ke halaman pelayanan	Berhasil
Mengubah detail data pada pelayanan tertentu dengan salah atau data tidak lengkap	Mengurangi kesalahan data pelayanan yang diubah kedalam sistem	Terdapat dialog error yang menginformasikan kesalahan pada pengubahan data dan pengguna diharuskan untuk memperbaiki data yang ingin diubah	Berhasil

Menghapus data pelayanan tertentu dengan melakukan usap pelayanan ke arah kanan dengan konfirmasi penghapusan	Data yang dipilih akan terhapus dengan konfirmasi penghapusan data yang disetujui oleh pengguna	Data pelayanan dapat terhapus setelah pengguna mengkonfirmasi penghapusan pada dialog yang muncul setelah melakukan usap pelayanan ke kanan	Berhasil
Melakukan membatalkan penghapusan data tertentu yang sudah terlanjur di usap ke kanan	Mengurangi *human error* yang kemungkinan terjadi ketika pengguna tidak sengaja melakukan usap pelayanan ke kanan sehingga data pelayanan batal dihapus	Data pelayanan batal terhapus setelah pengguna membatalkan penghapusan pada dialog yang muncul setelah melakukan usap pelayanan ke kanan	Berhasil
Melakukan pencarian dengan mengisi searchView yang disediakan dengan nama pelayanan dengan benar	Membantu pengguna untuk menemukan data pelayanan tertentu serta tersimpan dalam sistem	Data pelayanan tampil dalam wujud list yang berisi data yang dimasukan pengguna pada searchView	Berhasil
Melakukan pencarian dengan mengisi searchView dengan nama pelayanan dengan salah atau kosong	Dapat dan meminimalisir kesalahan data yang tampil pada layar	Muncul Dialog error yang menginformasikan ke pengguna bahwa tidak ada data pelayanan yang sesuai dengan data yang dimasukkan pada searchView pada sistem	Berhasil

4.3.6 Kerangka pengujian penambahan barang pada keranjang

Tabel 4.6 Kerangka pengujian penambahan barang pada keranjang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat detail data barang yang tersimpan menurut filter yang telah ditentukan atau secara *default*.	Data barang akan tampil sesuai kriteria atau secara *default* dan pengguna dapat memulai proses penambahan data barang pada keranjang	List data barang muncul secara default atau menurut filter yang dipilih pengguna.	Berhasil
Melakukan pencarian dengan mengisi searchView yang disediakan dengan nama barang dengan benar	Membantu pengguna untuk menemukan data barang tertentu serta tersimpan dalam sistem	Data barang tampil dalam wujud list yang berisi data yang dimasukan pengguna pada searchView	Berhasil
Melakukan pencarian dengan mengisi searchView dengan nama barang dengan salah atau kosong	Dapat dan meminimalisir kesalahan data yang tampil pada layar	Muncul Dialog error yang menginformasikan ke pengguna bahwa tidak ada data barang yang sesuai dengan data yang dimasukkan pada searchView pada sistem	Berhasil

Menambahkan barang dan jumlah tertentu kedalam keranjang dengan benar	Data barang dan jumlah berhasil masuk kedalam keranjang	Muncul Dialog notifikasi sukses dan data masuk kedalam keranjang sistem serta jumlah stok dari barang terpilih akan berkurang secara otomatis	Berhasil
Menambahkan barang dan jumlah tertentu kedalam keranjang dengan jumlah barang dikosongi	Mengurangi kesalahan data barang yang dimasukkan kedalam keranjang sistem	Muncul dialog notifikasi error yang menginformasikan bahwa jumlah barang tidak boleh dikosongi dan pengguna harus mengulang proses penambahan barang kedalam keranjang	Berhasil
Menambahkan barang dan jumlah tertentu kedalam keranjang dengan jumlah barang melebihi jumlah stok	Mengurangi kesalahan data barang yang dimasukkan kedalam keranjang sistem	Muncul dialog notifikasi error yang menginformasikan bahwa jumlah barang tidak boleh melebihi stok yang ada dan pengguna diharuskan untuk memperbaiki jumlah barang yang ingin ditambahkan kedalam keranjang	Berhasil

4.3.7 Kerangka pengujian penambahan pelayanan pada keranjang

Tabel 4.7 Kerangka pengujian penambahan pelayanan pada keranjang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat detail data pelayanan yang tersimpan menurut filter yang telah ditentukan atau secara *default*.	Data pelayanan akan tampil sesuai kriteria atau secara *default* dan pengguna dapat memulai proses penambahan data pelayanan pada keranjang pelayanan	List data pelayanan muncul secara default atau menurut filter yang dipilih pengguna.	Berhasil
Melakukan pencarian dengan mengisi searchView yang disediakan dengan nama pelayanan dengan benar	Membantu pengguna untuk menemukan data pelayanan tertentu serta tersimpan dalam sistem	Data pelayanan tampil dalam wujud list yang berisi data yang dimasukan pengguna pada searchView	Berhasil
Melakukan pencarian dengan mengisi searchView dengan nama pelayanan dengan salah atau kosong	Dapat dan meminimalisir kesalahan data yang tampil pada layar	Muncul Dialog error yang menginformasikan ke pengguna bahwa tidak ada data pelayanan yang sesuai dengan data yang dimasukkan pada searchView pada sistem	Berhasil

Menambahkan pelayanan, nama kostumer dan jumlah tertentu kedalam keranjang dengan benar	Data pelayanan berhasil masuk kedalam keranjang	Muncul Dialog notifikasi sukses dan data masuk kedalam keranjang sistem	Berhasil
Menambahkan pelayanan dan jumlah tertentu kedalam keranjang dengan jumlah barang dikosongi atau nama kostumer tidak diisi	Mengurangi kesalahan data pelayanan yang dimasukkan kedalam keranjang sistem	Muncul dialog notifikasi error yang menginformasikan bahwa data tidak boleh dikosongi, pengguna harus mengulang proses input pelayanan ke keranjang	Berhasil

4.3.8 Kerangka pengujian pembayaran transaksi barang

Tabel 4.8 Kerangka pengujian pembayaran transaksi barang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Mencoba memasuki halaman pembayaran dengan menekan tombol pembayaran dan memilih keranjang barang untuk melihat data keranjang barang dan informasi keranjang yang sudah terisi	Pengguna dapat memantau barang-barang yang ada dalam keranjang dan informasi terkait pembayaran	Pengguna akan diarahkan menuju halaman pembayaran yang berisi data keranjang barang dan informasi pembayaran serta fitur untuk melanjutkan proses transaksi	Berhasil

Mencoba memasuki halaman pembayaran dengan menekan tombol pembayaran dan memilih keranjang barang untuk melihat data keranjang barang dan informasi keranjang yang belum terisi	Meminimalisir kesalahan data transaksi yang kosong dapat dimasukkan kedalam sistem dengan memberi pemberitahuan error	Muncul dialog notifikasi error yang memberitahukan bahwa keranjang barang kosong dan pengguna harus menambahkan barang terlebih dahulu kedalam keranjang untuk memulai proses pembayaran dan melihat keranjang	Berhasil
Menghapus data barang yang ditambahkan dalam keranjang barang dengan melakukan usap barang yang dipilih ke kanan dan konfirmasi penghapusan	Data barang pada keranjang sistem akan terhapus dan jumlah pembayaran dan barang akan secara otomatis ter-*update*	Muncul dialog notifikasi penghapusan telah sukses dan ketika keranjang barang kosong, pengguna akan diarahkan untuk kembali ke halaman pemilihan barang	Berhasil
Membatalkan penghapusan data barang yang ditambahkan dalam keranjang barang yang sudah di-usap ke kanan	Mengurangi human error yang mengusap barang tertentu secara tidak sengaja dan membatalkan penghapusan barang	Data barang pada keranjang batal terhapus setelah pengguna membatalkan penghapusan pada dialog yang muncul setelah melakukan usap pelayanan ke kanan	Berhasil

Memasukan uang pembayaran yang diterima dari kostumer dengan jumlah sama atau melebihi total yang harus dibayarkan lalu menekan tombol simpan	Jumlah kembalian akan terkalkulasi secara otomatis dan pengguna dapat melanjutkan proses pembayaran	Jumlah kembalian dapat tampil dengan lancar dan muncul dialog pemilihan metode pembayaran	Berhasil
Memasukan uang pembayaran yang diterima dari kostumer dengan jumlah kurang dari total yang harus dibayarkan atau mengosongi form uang diterima lalu menekan tombol simpan	Dapat mengurangi kesalahan data transaksi yang dimasukkan dalam sistem dan mengurangi kerugian secara keuangan	Muncul dialog notifikasi error yang menginformasikan bahwa tidak ada uang yang diterima atau error muncul pada form uang yang diterima kurang dan pengguna harus mengulangi proses pembayaran	Berhasil
Memilih metode pembayaran tunai untuk transaksi barang	Data transaksi akan tersimpan, saldo keuangan akan secara otomatis ter-update dan pengguna akan diarahkan ke halaman utama transaksi	Muncul dialog notifikasi sukses yang berisi opsi pemilihan cetak struk dan pengguna akan diarahkan ke halaman utama transaksi	Berhasil

Memilih metode pembayaran digital untuk transaksi barang	Pengguna diharuskan untuk mengisi bukti pembayaran digital sebelum menyimpan data transaksi dan meng-update saldo keuangan	Muncul dialog notifikai pemilihan opsi pengisian foto bukti pembayaran digital. Setelah pengguna mengisi bukti, akan muncul dialog notifikasi sukses yang berisi opsi pemilihan cetak struk dan pengguna akan diarahkan ke halaman utama transaksi	Berhasil

4.3.9 Kerangka pengujian pembayaran transaksi pelayanan

Tabel 4.9 Kerangka pengujian pembayaran transaksi barang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memasuki halaman pembayaran dengan menekan tombol pembayaran dan memilih keranjang pelayanan untuk melihat data keranjang pelayanan dan informasi keranjang yang sudah terisi	Pengguna dapat memantau pelayanan yang ada dalam keranjang dan informasi terkait pembayaran	Pengguna akan diarahkan menuju halaman pembayaran yang berisi searchView yang nantinya dapat diisi dengan nama kostumer yang memesan pelayanan	Berhasil

Memasuki halaman pembayaran dengan menekan tombol pembayaran dan memilih keranjang pelayanan untuk melihat data keranjang pelayanan dan informasi keranjang yang belum terisi	Meminimalisir kesalahan data transaksi yang kosong dapat dimasukkan kedalam sistem dengan memberi pemberitahuan error	Muncul dialog notifikasi error yang memberitahukan bahwa keranjang pelayanan kosong dan pengguna harus menambahkan pelayanan terlebih dahulu kedalam keranjang untuk memulai proses pembayaran dan melihat keranjang	Berhasil
Mencari data pelayanan pada keranjang dengan memasukkan nama kostumer pada searchView yang tersedia dengan benar	Data pelayanan akan muncul secara spesifik dan pengguna dapat melanjutkan proses pembayaran	Data pelayanan pada keranjang akan muncul sesuai dengan data menurut nama kostumer tertentu dan tampilan pembayaran akan muncul seperti pada tampilan pembayaran barang	Berhasil
Mengosongi nama kostumer atau memasukkan nama yang salah serta tidak terdaftar pada sistem di searchView	Mengurangi kesalahan penambahan data transaksi saat menampilkan data pelayanan pada keranjang	Muncul dialog notifikasi error yang menginformasikan nama tidak terdaftar dalam sistem pelayanan dan pengguna diharuskan untuk mengulangi proses pencarian data pelayanan	Berhasil

Menghapus data pelayanan yang ditambahkan dalam keranjang pelayanan dengan melakukan usap pelayanan yang dipilih ke kanan dan konfirmasi penghapusan	Data pelayanan pada keranjang sistem akan terhapus dan jumlah pembayaran secara otomatis ter-*update*	Muncul dialog notifikasi penghapusan telah sukses dan ketika keranjang pelayanan kosong, pengguna akan diarahkan untuk kembali ke halaman pemilihan pelayanan	Berhasil
Membatalkan penghapusan data pelayanan yang ditambahkan dalam keranjang pelayanan yang sudah di-usap ke kanan	Mengurangi human error yang mengusap pelayanan tertentu secara tidak sengaja dan membatalkan penghapusan pelayanan	Data pelayanan pada keranjang batal terhapus setelah pengguna membatalkan penghapusan pada dialog yang muncul setelah melakukan usap pelayanan ke kanan	Berhasil
Memasukan uang pembayaran yang diterima dari kostumer dengan jumlah sama atau melebihi total yang harus dibayarkan lalu menekan tombol simpan	Jumlah kembalian akan terkalkulasi secara otomatis dan pengguna dapat melanjutkan proses pembayaran	Jumlah kembalian dapat tampil dengan lancar dan muncul dialog pemilihan metode pembayaran	Berhasil

Memasukan uang pembayaran yang diterima dari kostumer dengan jumlah kurang dari total yang harus dibayarkan atau mengosongi form uang diterima lalu menekan tombol simpan	Dapat mengurangi kesalahan data transaksi yang dimasukkan dalam sistem dan mengurangi kerugian secara keuangan	Muncul dialog notifikasi error yang menginformasikan bahwa tidak ada uang yang diterima atau error muncul pada form uang yang diterima kurang dan pengguna harus mengulangi proses pembayaran	Berhasil
Memilih metode pembayaran tunai untuk transaksi pelayanan	Data transaksi akan tersimpan, saldo keuangan akan secara otomatis ter-update dan akan diarahkan ke halaman utama transaksi	Muncul dialog notifikasi sukses yang berisi opsi pemilihan cetak struk dan pengguna akan diarahkan ke halaman utama transaksi	Berhasil
Memilih metode pembayaran digital untuk transaksi pelayanan	Pengguna diharuskan untuk mengisi bukti pembayaran digital sebelum menyimpan data transaksi dan meng-update saldo keuangan	Muncul dialog notifikai pemilihan opsi pengisian foto bukti pembayaran digital. Setelah pengguna mengisi bukti, akan muncul dialog notifikasi sukses yang berisi opsi pemilihan cetak struk dan pengguna akan diarahkan ke halaman utama transaksi	Berhasil

4.3.10 Kerangka pengujian pencetakan struk

Tabel 4.10 Kerangka pengujian pembayaran transaksi barang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Melakukan proses transaksi sampai selesai, setelah dialog informasi transaksi sukses muncul bersama konfirmasi cetak struk, pengguna menekan tombol cetak	Sistem akan secara otomatis mengambil data transaksi kemudian membuat format struk berisi data tersebut. Kemudian sistem akan secara otomatis mencetak struk dengan printer yang terkoneksi	Struk yang berisi data lengkap transaksi berhasil tercetak dan siap diberikan kepada kostumer	Berhasil
Melakukan proses penambahan data pelayanan, kemudian masuk ke halaman pembayaran pelayanan. Pengguna mencari nama kostumer pada searchView, setelah data tampil, pengguna menekan tombol cetak serta melakukan mengkonfirmasi pencetakan melalui dialog yang muncul	Sistem akan mengambil data transaksi pelayanan menurut nama kostumer yang ditentukan dan membuat format struk berisi data pelayanan tersebut. Kemudian sistem akan secara otomatis mencetak struk dengan printer yang terkoneksi	Struk yang berisi data lengkap transaksi berhasil tercetak dan siap diberikan kepada kostumer	Berhasil

4.3.11 Kerangka pengujian pengembalian barang (retur)

Tabel 4.11 Kerangka pengujian pengembalian barang (retur)

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Mencari data transaksi dengan memasukkan id transaksi yang didapat dari struk transaksi barang pada searchView yang tersedia dengan benar	Pengguna dapat melihat data transaksi berupa barang yang telah dibeli dalam wujud list	Data transaksi akan muncul sesuai id transaksi yang diinputkan dan proses retur dapat dilanjutkan	Berhasil
Mengosongi searchView atau mengisinya dengan id yang tidak sesuai atau tidak tersimpan pada sistem	Meminimalisir kesalahan data transaksi yang muncul pada proses retur	Muncul dialog notifikasi error yang memberitahukan bahwa data transaksi yang diminta tidak ada dan pengguna diharuskan untuk mengulang proses pencarian data transaksi	Berhasil
Menambahkan data retur dengan menyesuaikan jumlah barang yang ingin diretur menurut barang yang ada pada data transaksi dan menginputkan catatan retur dengan benar lalu menekan ikon tambah	Data retur dapat ditambahkan pada sistem, dan saldo keluar akan direkam oleh sistem untuk keperluan *refund*.	Muncul dialog notifikasi pengembalian barang sukses dan pengguna akan diarahkan ke halaman utama transaksi	Berhasil

Mengosongi catatan retur jumlah barang melebihi jumlah yang ada pada data transaksi lalu menekan ikon tambah	Mengurangi kesalahan data retur yang ditambahkan kedalam sistem	Muncul dialog notifikasi error yang menginformasikan bahwa data retur tidak lengkap dan jumlah barang tidak boleh melebihi yang dibeli saat transaksi dan pengguna diharuskan untuk mengulang proses retur	Berhasil

4.3.12 Kerangka pengujian laporan transaksi

Tabel 4.12 Kerangka pengujian tranksaksi

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat list data laporan transaksi dan detail yang tersimpan menurut filter yang telah ditentukan atau secara *default.*	Data laporan transaksi akan tampil secara lengkap sesuai kriteria atau secara *default*	List data laporan transaksi muncul secara *default* atau menurut filter yang dipilih pengguna. Monitoring data laporan transaksi pada sistem tampil melalui *cardView*	Berhasil

Melihat detail dengan menekan ikon mata pada data laporan transaksi tertentu dengan pembayaran tunai	Data laporan transaksi akan tampil secara detail menurut data yang dipilih	Pengguna akan diarahkan kedalam detail transaksi yang berisi informasi tanggal, jumlah pembayaran, keuntungan, jumlah barang, pengguna yang merekam proses transaksi, barang yang dijual, dsb	Berhasil
Melihat detail dengan menekan ikon mata pada data laporan transaksi tertentu dengan pembayaran digital	Data laporan transaksi akan tampil secara detail menurut data yang dipilih	Pengguna akan diarahkan kedalam detail transaksi yang berisi informasi yang hampir sama dengan transaksi metode tunai, yang berbeda adalah munculnya tombol bukti untuk melihat bukti pembayaran transaksi	Berhasil
Login sebagai admin dan mencoba masuk ke filter pengguna untuk melihat data transaksi menurut pengguna yang terdaftar di sistem dan memilih pengguna yang memiliki rekaman transaksi	Membantu admin memonitoring hasil kerja dari pengguna yang terdaftar pada sistem	Pengguna dapat melihat pengguna yang terdaftar pada sistem dan informasi detail perekaman transaksi pada sistem yang dilakukan oleh pengguna tertentu	Berhasil

Login sebagai admin dan mencoba masuk ke filter pengguna untuk melihat data transaksi yang dikelompokkan menurut pengguna yang terdaftar pada sistem dan memilih pengguna yang belum merekam transaksi dengan kostumer	Meminimalisir kesalahan data transaksi yang muncul ke layar pengguna	Muncul dialog notifikasi error yang menginformasikan bahwa pengguna yang dipilih belum memiliki rekaman transaksi	Berhasil
Login sebagai pengguna biasa dan mencoba masuk ke filter pengguna untuk melihat data transaksi yang dikelompokkan menurut pengguna yang terdaftar pada sistem dan memilih pengguna yang belum merekam transaksi dengan kostumer	Melindungi keamanan data agar hanya dapat dilihat oleh admin	Filter pengguna tidak muncul pada halaman laporan transaksi	Berhasil

4.3.13 Kerangka pengujian laporan pembukuan

Tabel 4.13 Kerangka pengujian laporan pembukuan

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat list data laporan pembukuan dan detail yang tersimpan	Data laporan pembukuan akan tampil secara lengkap dan pengguna dapat melihat detail data pembukuan dengan menekan ikon panah bawah	List data laporan pembukuan akan tampil pada layar, beserta tombol "Buat" untuk membuat atau melakukan *generate data* pembukuan	Berhasil
Membuat laporan pembukuan secara otomatis dengan menekan tombol buat pada halaman laporan pembukuan lalu menekan ikon *new folder*	Sistem akan secara otomatis melakukan kalkulasi, mengambil beberapa data yang diperlukan dari database, seperti transaksi, restok, retur, dsb	Halaman *generate* pembukuan akan secara otomatis terisi oleh data-data yang diperlukan untuk perekaman pembukuan	Berhasil
Membagikan data pembukuan setelah melakukan *generate data* pembukuan ke salah satu kontak pada aplikasi *whatsapp* dengan menekan ikon *whatsapp*	Sistem akan membuatkan format chat berupa data pembukuan yang diambil dari proses *generate data* sebelumnya	Pengguna akan diarahkan ke aplikasi *whatsapp*, khususnya pada halaman pemilihan chat baru berisi list kontak dari ponsel pengguna untuk kemudian meneruskan format chat yang sudah dibuatkan oleh sistem ke kontak tertentu	Berhasil

Mengunduh data pembukuan dengan menekan ikon pdf	Sistem akan membuatkan sebuah format file pdf berisi tabel data pembukuan yang diambil dari proses generate data sebelumnya	Pengguna akan diarahkan untuk memilih opsi aplikasi pdf Viewer untuk melihat hasil unduhan file pdf pembukuan dan file tersebut dapat diunduh dan disimpan pada direktori ponsel pengguna	Berhasil

4.3.14 Kerangka pengujian laporan keuangan

Tabel 4.14 Kerangka pengujian laporan keuangan

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat list data laporan keuangan dan detail yang tersimpan	Data laporan keuangan akan tampil secara lengkap	List data laporan keuangan, monitoring data laporan keuangan pada sistem tampil melalui *cardView*	Berhasil

4.3.15 Kerangka pengujian laporan barang

Tabel 4.15 Kerangka pengujian laporan barang

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Memilih, melihat list data laporan barang dan detail yang tersimpan serta menurut filter yang dipilih atau secara *default*	Data laporan barang akan tampil secara lengkap	List data laporan barang akan muncul sesuai filter yang dipilih atau secara *default.* monitoring data laporan keuangan pada sistem tampil melalui *cardView*	Berhasil

4.3.16 Kerangka pengujian monitoring data sistem pada dashboard

Tabel 4.16 Kerangka pengujian laporan keuangan

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Melihat data transaksi, stok, keuangan yang berisi keuntungan, kerugian dan saldo pada halaman dashboard	Sistem mengambil data lalu menampilkannya pada halaman dashboard, data tersebut diambil secara *realtime* atau akan selalu ter-update apabila ada perubahan atau penambahan data pada sistem	Halaman dashboard muncul dengan menampilkan data-data transaksi, stok, keuangan dengan sederhana pada sistem sehingga pengguna dapat memonitoring pembaruan data	Berhasil

4.3.17 Kerangka pengujian profil akun pengguna

Tabel 4.17 Kerangka pengujian profil akun pengguna

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Melihat data profil akun pengguna yang sedang login pada halaman profil	Sistem akan menampilkan data profil pengguna yang dicatat oleh *preference* sebagai pengguna yang login	Muncul data pengguna yang berisi username, nama lengkap, email dan password serta foto profil.	Berhasil

Melakukan *update* data profil pengguna dengan menekan tombol edit profil dan mengubah form data akun yang tersedia lalu menekan simpan dengan benar	Sistem dapat melakukan update data profil dan pengguna dapat melakukan login dengan data akun yang baru	Muncul form edit data profil yang berisi data akun, setelah data diubah dan pengguna menekan tombol selesai, data akan diperbarui setelah pengguna melakukan konfirmasi untuk menyimpan perubahan data pada akun di dialog notifikasi yang muncul	Berhasil
Melakukan *update* data profil pengguna dengan menekan tombol edit profil dan mengubah form data akun yang tersedia lalu menekan simpan dengan salah atau mengosongi beberapa form yang diperlukan	Menghentikan proses edit dengan maksud meminimalisir kesalahan data yang diubah pada sistem	Muncul dialog notifikasi error yang menginformasikan bahwa data tidak lengkap atau data tidak sesuai format yang mengharuskan pengguna mengulangi proses pembaruan data akun	Berhasil
Melakukan logout sistem dengan menekan tombol logout pada halaman profil	Sistem akan mengakhiri sesi login dan mengarahkan pengguna ke halaman login	Halaman berpindah ke halaman login dan pengguna siap untuk login kembali	Berhasil

4.3.18 Kerangka pengujian setelan pengaturan

Tabel 4.18 Kerangka pengujian profil akun pengguna

Aktivitas Pengujian	Realisasi yang Diharapkan	Hasil Pengujian	Kesimpulan
Melakukan konfigurasi konektifitas printer thermal dengan menekan ikon printer dan mengkoneksikan ponsel dengan printer tertentu	Sistem dapat melakukan proses cetak struk dengan printer yang ditentukan oleh pengguna	Bluetooth akan dihidupkan secara otomatis dan pengguna akan diarahkan pada halaman list koneksi bluetooth yang tersedia, setelah pengguna memilih printer yang diinginkan, maka akan muncul pop up yang menginformasikan koneksi printer tersambung dan siap untuk mencetak	Berhasil
Mengganti setelan bahasa aplikasi dengan memasuki halaman pemilihan bahasa setelah menekan ikon *language*	Sistem akan mengganti bahasa aplikasi sesuai dengan bahasa yang dipilih oleh pengguna	Pengguna akan diarahkan pada halaman pemilihan bahasa. Setelah pengguna memilih bahasa yang ditentukan, pengguna akan diarahkan kembali pada halaman setelan dan bahasa aplikasi sudah otomatis berganti dengan bahasa yang ditentukan	Berhasil