

**RANCANG BANGUN APLIKASI *INVENTORY MANAGEMENT* BERBASIS  
ANDROID MENGGUNAKAN METODE *SCRUM***

**Oleh:  
JULIO ADI PUTRA**

**Skripsi**

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar  
SARJANA TEKNIK**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**



**LEMBAR PENGESAHAN**

## **KATA PENGANTAR**

**ABSTRAK**

## DAFTAR ISI

<b>JUDUL .....</b>	<b>Error! Bookmark not defined.</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR GAMBAR.....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Tujuan.....	2
1.3. Rumusan Masalah .....	2
1.4. Batasan Masalah.....	2
1.5. Sistematika Penulisan.....	2
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>4</b>
2.1. Scrum .....	4
2.2. Flutter .....	5
2.3. Dart.....	6
2.4. Android.....	6
2.5. Android Studio .....	6
2.6. Firebase .....	7
2.7. Management Inventory .....	8
2.8. Penelitian Terdahulu .....	8
2.8.1. Aplikasi Manajemen <i>Inventory</i> Berbasis Mobile .....	8
2.8.2. Penerapan Inventory Management Dalam Meningkatkan Profitabilitas di Toko X Kupang.....	9

2.8.3. <i>Automatic Management Inventory System (AMIS): Efektifitas Pengawasan dan Pengelolaan Persediaan Terintegrasi</i> .....	10
2.8.4. <i>IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG)</i> .....	10
2.8.5. <i>PERANCANGAN APLIKASI MANAJEMEN PERSEDIAAN GUDANG BERBASIS WEBSITE PADA UMKM BATIK SINUWUN DENGAN AGILE SCRUM DEVELOPMENT METHOD</i> .....	11
2.8.6. <i>Application of Database Technology to Improve the Efficiency of Inventory Management for Small Businesses</i> .....	12
2.8.7. <i>DESIGN AND BUILD INVENTORY MANAGEMENT INFORMATION SYSTEM USING THE SCRUM METHOD</i> .....	13
<b>BAB III METODE PENELITIAN</b> .....	14
3.1. Waktu dan Tempat .....	14
3.2. Jadwal Penelitian.....	14
3.3. Alat Penelitian .....	14
3.4. Tahapan Penelitian .....	15
3.4.1. Studi Literatur .....	<b>Error! Bookmark not defined.</b>
3.4.2. Pengembangan Sistem .....	<b>Error! Bookmark not defined.</b>
3.4.3. Penulisan Laporan.....	18
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	19
<b>BAB V KESIMPULAN</b> .....	19
<b>DAFTAR PUSTAKA</b> .....	56

**DAFTAR GAMBAR**

Gambar 1 Tahapan metode scrum .....	4
Gambar 2 Tahap penelitian.....	<b>Error! Bookmark not defined.</b>



**DAFTAR TABEL**

Tabel 1 Tabel Sprint Backlog .....	<b>Error! Bookmark not defined.</b>
------------------------------------	-------------------------------------

## **BAB I PENDAHULUAN**

### **1.1. Latar Belakang**

Pada zaman sekarang teknologi menjadi suatu hal yang penting dan tidak bisa lepas dari manusia. Kehadiran teknologi memiliki banyak manfaat bagi manusia, salah satunya yaitu mempermudah pekerjaan manusia. Perkembangan teknologi mendorong manusia untuk melakukan sebagian besar penggunaan komputer agar hasil yang dibutuhkan dapat diproses dengan cepat, salah satunya adalah mengelola persediaan dengan menggunakan sistem informasi [1]. Manajemen adalah penggunaan sumber daya organisasi untuk mencapai sasaran dan kinerja yang tinggi dalam berbagai tipe organisasi profit maupun nonprofit. Sedangkan *Inventory* adalah suatu aset yang ada dalam bentuk barang-barang yang dimiliki untuk dijual dalam toko maupun barang-barang yang sedang dalam proses pembuatan. Manajemen *inventory* merupakan pengaturan persediaan barang dengan tujuan kelancaran usahanya [2].

Manajemen *inventory* merupakan hal yang penting bagi pemilik toko, hal tersebut bisa mempermudah pemilik toko dalam mendata stok barang yang ada di gudang maupun memantau tanggal kadaluarsa barang yang ada di gudang. Dalam hal ini, teknologi yang umumnya digunakan untuk mencatat dan memanajemen barang adalah menuliskan secara manual, baik itu stok barang, tanggal kadaluarsa, maupun riwayat barang masuk dan keluar. Dalam kegiatan pendataan barang dapat terjadi kesalahan dalam mendata barang secara manual serta memakan waktu yang cukup banyak tergantung dari banyaknya barang yang dimiliki oleh gudang dan toko tersebut. Maka dari itu manajemen *inventory* perlu dilakukan komputerisasi, guna untuk meminimalisir kesalahan dalam mendata sebuah barang serta mempersingkat waktu dalam pendataan

Dengan adanya permasalahan manajemen barang manual tersebut, dilakukanlah penelitian guna untuk menyelesaikan permasalahan tersebut. Logitory merupakan sebuah aplikasi berbasis android yang bertujuan untuk memudahkan para pemilik toko untuk memanajemen barangnya yang ada di gudang. Aplikasi ini memiliki beberapa fitur yang bisa menjawab permasalahan diatas, seperti bisa melihat stok

barang yang ada di gudang, melihat tanggal kadaluarsa barang, dan mencatat barang masuk dan keluar.

## **1.2. Tujuan**

Tujuan dari penelitian ini adalah untuk melakukan perancangan sekaligus pembangunan sebuah aplikasi berbasis android yang dapat memanajemen barang serta melakukan testing aplikasi manajemen barang sesuai fungsinya.

## **1.3. Rumusan Masalah**

Bagaimana merancang dan membangun sebuah aplikasi manajemen barang menggunakan bahasa pemrograman flutter menggunakan metode scrum.

## **1.4. Batasan Masalah**

Pada perancangan dan pembangunan aplikasi ini, terdapat beberapa batasan masalah supaya penelitian ini lebih terarah serta agar memudahkan dalam penyelesaian penelitian ini. Beberapa batasan masalah sebagai berikut:

1. Sistem yang dirancang merupakan aplikasi berbasis android.
2. Sistem dibuat menggunakan bahasa pemrograman flutter.
3. Sistem ini menggunakan firebase untuk databasenya.
4. User interface sistem dibuat dengan menggunakan aplikasi Figma

## **1.5. Sistematika Penulisan**

Sistematika penulisan memuat inti dari tiap-tiap bab. Adapun sistematika penulisan dari penelitian ini sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab pendahuluan menjelaskan tentang latar belakang, tujuan, rumusan masalah, serta batasan masalah dari penelitian ini.

### **BAB II TINJAUAN PUSTAKA**

Bab tinjauan pustaka meliputi landasan teori-teori yang mendasari pembahasan pada penelitian dan penjelasan mengenai alat-alat yang digunakan dalam penyelesaian penelitian ini serta kajian terkait penelitian terdahulu.

### BAB III METODE PENELITIAN

Bab metode penelitian menguraikan gambaran obyek penelitian dan analisis permasalahan yang ada, meliputi:

- a. Waktu dan tempat
- b. Alat Penelitian
- c. Tahapan penelitian

### BAB IV HASIL DAN PEMBAHASAN

Pada bab ini memaparkan hasil dan pembahasan yang telah dilakukan pada penelitian ini.

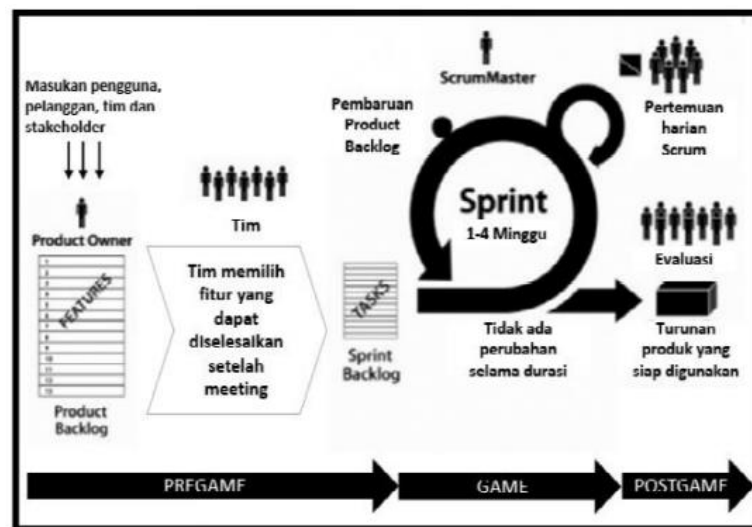
### BAB V KESIMPULAN

Kesimpulan dan penutup.

## BAB II TINJAUAN PUSTAKA

### 2.1. Scrum

Menurut schwaber & sutherland scrum adalah sebuah kerangka kerja yang dapat mengatasi suatu masalah kompleks yang selalu berubah, dan juga dinilai dapat memberikan kualitas produk yang baik sesuai dengan keinginan pengguna secara kreatif dan produktif [3].



*Gambar 1 Tahapan metode scrum*

#### a. Tim scrum

Tim scrum terdiri dari pemilik produk, tim pengembang, dan scrum master. Dalam menyelesaikan pekerjaannya tim scrum mengatur dirinya sendiri dengan cara terbaik dan tim harus memiliki kompetensi yang dibutuhkan untuk menyelesaikan pekerjaan, tanpa diatur oleh pihak lain yang berada di luar anggota tim [3].

#### b. Scrum artefak

Scrum artefak adalah sebuah artefak yang dibuat agar informasi-informasi penting dapat ditingkatkan transparansinya sehingga semua pihak dapat memahami informasi pada artefak dengan pemahaman yang sama [3].

c. Scrum event

Scrum events adalah kegiatan yang harus dihadiri dalam scrum guna mendukung berjalanya sprint, memberikan suatu keteraturan dan mengurangi adanya acara lain yang tidak tercakup pada scrum. Semua scrum events dijadikan satu kedalam batasan waktu yang disebut sprint [3].

d. Sprint

Sprint merupakan batasan waktu selama satu bulan atau kurang, dimana produk jadi inkremental yang berpotensi “Selesai” didalam sprint harus berfungsi. Setiap sprint memuat scrum event yang terdiri atas sprint planning, daily scrum, sprint review, dan sprint retrospective [3].

e. Sprint planning

Tujuan sprint planning ini adalah untuk merencanakan pekerjaan apa saja yang nantinya dilakukan dalam sprint. Hal-hal yang akan dilakukan dalam perencanaan sprint adalah :

- 1) Menentukan tujuan sprint
- 2) Menentukan panjang sprint
- 3) Memutuskan item backlog yang akan diikuti dalam sprint dengan melihat hasil perhitungan perkiraan kecepatan tim [3].

## 2.2. Flutter

Flutter merupakan sebuah SDK (Software Development Kit) yang digunakan untuk mengembangkan aplikasi mobile yang dibuat oleh Google. Flutter dikembangkan untuk membuat aplikasi yang mempunyai performance yang tinggi dan dapat dipublikasikan untuk platform Android dan iOS dari codebase tunggal. Flutter menggunakan bahasa pemrograman dart sehingga dapat dengan mudah untuk dipelajari. Bahasa pemrograman dart dianggap mudah apabila telah terbiasa dan familiar menggunakan bahasa pemrograman Java atau Javascript. Selain itu, Flutter

juga menyediakan kerangka reactive-functional, mesin render 2D, widget yang siap untuk digunakan, dan tools yang digunakan untuk membantu dalam melakukan pengembangan aplikasi [4].

### **2.3. Dart**

Dart merupakan bahasa pemrograman yang dibuat oleh Google dan didesain oleh Lars Bak dan Kasper Lund. Bahasa pemrograman Dart dapat digunakan untuk membangun aplikasi server atau dalam bentuk command line interface, web, ataupun mobile (Android dan iOS). Dart merupakan Bahasa pemrograman yang mendukung adanya pendefinisian fungsi di luar kelas atau sering disebut dengan top level function. Dalam Dart. Kode program utama disimpan di dalam fungsi main() sama halnya seperti C/C++ [4].

Dart merupakan bahasa pemrograman yang bersifat open source. Dart merupakan bahasa pemrograman yang menggunakan konsep berorientasi objek dengan sintaks gaya C. Hal tersebut mendukung konsep pemrograman seperti antarmuka, class, tidak seperti bahasa pemrograman lainnya, Dart tidak mendukung array. Dart dapat mereplikasi struktur data seperti array, generic dan pengetikan opsional [4].

### **2.4. Android**

Android merupakan salah satu platform dari perangkat smartphone. Salah satu keuntungan dari Android yaitu lisensinya yang bersifat terbuka (open source) dan gratis (free) sehingga bebas untuk dikembangkan karena tidak ada biaya royalti maupun didistribusikan dalam bentuk apapun. Hal ini memudahkan para programmer untuk membuat aplikasi baru di dalamnya [5].

### **2.5. Android Studio**

Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android [6].

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android studio memiliki fitur :

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan [6].

## **2.6. Firebase**

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan Mobile Apps Developer. Dengan adanya Firebase, apps developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend [7].

Dengan pengertian di atas penulis akan menggunakan Firebase untuk menghubungkan data-data informasi dari aplikasi yang di buat dan di hubungkan ke Firebase. Berikut adalah beberapa fitur dari Firebase dan penjelasannya:

- a. Firebase Authentication Sebagian besar aplikasi perlu mengetahui identitas pengguna. Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman di cloud dan memberikan pengalaman personal yang sama di setiap perangkat pengguna [7].
- b. Cloud Storage Cloud Storage for Firebase adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala Google. Firebase SDK untuk Cloud Storage menambahkan keamanan Google pada upload dan download file untuk aplikasi yang akan di buat [7].



- c. **Firestore Realtime Database** Firestore Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, dengan menggunakan ini semua klien akan berbagi sebuah instance realtime database dan menerima update data terbaru secara otomatis [7].

## **2.7. Management Inventory**

Manajemen adalah penggunaan sumber daya organisasi untuk mencapai sasaran dan kinerja yang tinggi dalam berbagai tipe organisasi profit maupun nonprofit. Sedangkan Inventory adalah suatu aset yang ada dalam bentuk barang-barang yang dimiliki untuk dijual dalam toko maupun barang-barang yang sedang dalam proses pembuatan. Manajemen inventory merupakan pengaturan persediaan barang dengan tujuan kelancaran usahanya [2].

## **2.8. Black Box Testing**

Metode *Black Box Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai. Proses Black Box Testing dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan. Salah satu jenis pengujian *Black Box Testing* adalah dengan menggunakan teknik *Equivalence Partitioning* (EP) yang penulis gunakan untuk menguji masukan serta membagi masukan ke dalam kelompok-kelompok berdasarkan fungsinya [8].

## **2.9. Penelitian Terdahulu**

### **2.9.1. Aplikasi Manajemen Inventory Berbasis Mobile**

Penelitian yang dilakukan oleh Angga Eka Syaputra yang berjudul “Aplikasi Manajemen *Inventory* Berbasis Mobile”, penelitian ini bertujuan untuk membuat sebuah aplikasi manajemen *inventory* berbasis *mobile*, yang mana sebelumnya

manajemen *inventory* dilakukan secara manual menggunakan buku-buku tebal yang berisiko kesalahan dalam mendata barang yang dimiliki gudang maupun toko. Aplikasi yang dikembangkan menggunakan *Eclips Mars 2* dan *Balsamiq Mockup* untuk bagian *software design*. *Database* aplikasi dibangun menggunakan *MYSQL* dan *XAMPP* sehingga aplikasi dapat berjalan sesuai yang dengan kebutuhan untuk memudahkan dalam pendataan barang yang ada di gudang, serta memudahkan pengguna dalam mencari barang yang telah dilakukan pendataan [2].

### **2.9.2. Penerapan Inventory Management Dalam Meningkatkan Profitabilitas di Toko X Kupang**

Penelitian yang dilakukan oleh Rosina Jappi dan Dianne Frisko Koan yang berjudul “Penerapan Inventory Management Dalam Meningkatkan Profitabilitas Di Toko X Kupang”, penelitian ini dilakukan pada Toko X yang berada di Kupang yang bertujuan untuk meningkatkan profitabilitas dari toko tersebut. Toko X memiliki permasalahan dalam *inventory management* sehingga mengakibatkan kurangnya profit dalam toko tersebut. Hal ini disebabkan karena ketidaktepatan kuantitas dan waktu pemesanan, ditambah lagi dengan adanya penumpukan stok sehingga terjadi penurunan harga kain. Dengan demikian, Toko X mengalami kerugian dari biaya-biaya *non-value-added* tersebut. Metode yang digunakan dalam penelitian ini yaitu *explanatory research* yang bertujuan mengatasi masalah-masalah *inventory* dan studi penelitian ini termasuk *applied research* yang mencoba memberikan solusi untuk meningkatkan profitabilitas bisnis yang ada. Hasil dari penelitian ini menunjukkan bahwa penerapan *inventory management* ini dapat mewujudkan efisiensi biaya dengan adanya *EOQ*, *ROP*, dan *Safety Stock* agar pengelolaan *inventory* baik dari aktivitas pembelian, penyimpanan, dan penjualan lebih terorganisir. Selain itu, adanya pengembangan bisnis yang lebih *profitable* dengan memproduksi produk dari kain yang menumpuk sehingga meningkatkan margin laba [9].

### **2.9.3. *Automatic Management Inventory System* (AMIS): Efektifitas Pengawasan dan Pengelolaan Persediaan Terintegrasi**

Penelitian yang dilakukan oleh A.A Gde Satia Utama dkk yang berjudul “*Automatic Management Inventory System* (AMIS): Efektifitas Pengawasan dan Pengelolaan Persediaan Terintegrasi bertujuan untuk memberikan informasi kinerja aplikasi AMIS yang dapat digunakan secara tepat dan pasti untuk mengetahui berapa total persediaan yang masih ada di gudang atau persediaan yang masih menjadi safety stock dan berapa persediaan yang harus dilakukan penambahan atau pembelian. Hasil dari penelitian ini, AMIS memiliki tiga proses pokok, yaitu *input*, transaksi dan *output*. Pada proses *input*, dilakukan setting awal mengenai EEQ, dan semua informasi persediaan. Pada proses transaksi, AMIS secara otomatis memberitahu bahwa diperlukannya pembelian persediaan barang kepada bagian pembelian, manajer persediaan, bagian gudang, dan bagian penjualan. Pada proses *output*, AMIS mengeluarkan laporan periodik mengenai siklus persediaan yang terjadi selama periode berjalan [10].

### **2.9.4. IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG)**

Penelitian yang dilakukan oleh Shon Hadji dkk yang berjudul “Implementasi Metode Scrum Pada Pengembangan Aplikasi Delivery Order Berbasis Website (Studi Kasus Pada Rumah Makan Lombok Idjo Semarang)” memiliki tujuan membangun sebuah aplikasi *delivery order* berbasis *website* dengan menggunakan metode Scrum. Penelitian ini menggunakan Scrum sebagai metode yang diambil dalam pengembangan aplikasi. Dalam pengembangan aplikasi ini, ada beberapa tahap yang dilakukan oleh Shon Hadji dkk, antara lain tahap pengumpulan data, adapun tahap pengumpulan data ini dilakukan secara observasi dan juga studi pustaka. Selanjutnya yaitu Analisa Sistem Berjalan, tahap ini dilakukan evaluasi terhadap data yang telah dikumpulkan pada tahap yang sebelumnya. Dan yang terakhir yaitu tahap pengembangan sistem. Pada tahapan ini dibentuk *team scrum*, membuat *product backlog*, dan fase *sprint* (*sprint planning*, *sprint backlog*, *daily scrum*, *sprint review*, dan *sprint retrospective*). Hasil dari pengembangan aplikasi

*delivery order* berbasis *website* studi kasus di Rumah Makan Lombok Idjo telah dibangun dengan menggunakan metode *Scrum* dimana dengan metode ini dapat mengatasi perubahan *requirements* pada saat *fase* pengembangan sistem dan *scrum* memiliki tahapan yang bersifat perulangan dimana jika produk pada *sprint* pertama belum cukup memenuhi kebutuhan, maka pada *sprint* berikutnya dapat dikembangkan sistem yang sesuai dengan evaluasi pengguna.

#### **2.9.5. PERANCANGAN APLIKASI MANAJEMEN PERSEDIAAN GUDANG BERBASIS WEBSITE PADA UMKM BATIK SINUWUN DENGAN AGILE SCRUM DEVELOPMENT METHOD**

Penelitian yang dilakukan oleh Muhammad Yumna Majdina dkk yang berjudul “Perancangan Aplikasi Manajemen Persediaan Gudang Berbasis *Website* Pada UMKM Batik Sinuwun Dengan *Agile Scrum Development*” dilatarbelakangi dengan sebuah UMKM Batik yang berpusat di Sukoharjo, Jawa Tengah. UMKM Batik tersebut memiliki barang yang beraneka ragam, mulai dari kain batik hingga beraneka pakaian batik. UMKM tersebut sempat mengalami penurunan pendapatan dikarenakan tidak terpenuhinya pesanan bulanan pada bulan tertentu yang disebabkan oleh ketidaksamaan data pada pergudangan, yaitu antara data stok dan data aktual yang mengakibatkan tidak terdapatnya kain batik sebagai bahan dasar utama. Selain itu, UMKM Batik Sinuwun juga menghadapi permasalahan lain, yaitu tidak terdapatnya proses pencatatan data barang gudang baik barang masuk maupun barang keluar serta stok barang jadi. Sehingga mengakibatkan apabila toko mendapatkan pesanan barang maka akan menghubungi pihak pergudangan untuk menanyakan ketersediaan stok barang jadi. Tujuan dari penelitian ini yaitu merancang aplikasi manajemen berbasis *website* untuk bisa memperbaiki manajemen inventory barang pada usaha UMKM Batik Sinuwun. Hal tersebut bertujuan untuk mempermudah pemilik usaha dalam manajemen inventory barang di gudang agar tersusun secara rapih. Metode yang digunakan untuk pengembangan aplikasi ini yaitu *Agile Development Methode* dengan Model *Scrum*. Hasil dari penelitian ini yaitu aplikasi manajemen persediaan pergudangan berbasis *website* telah terbuat dengan metode *Agile Scrum Development* dari proses pengumpulan *user requirement* hingga aplikasi selesai dibuat serta telah lolos pengujian *Black*

*Box testing* serta ISO:9126 sehingga aplikasi dapat berjalan tanpa kendala dan sesuai dengan kebutuhan *user* [11].

#### ***2.9.6. Application of Database Technology to Improve the Efficiency of Inventory Management for Small Businesses***

Penelitian yang dilakukan oleh Oksana Y. Iliashenko dkk yang berjudul “*Application of Database Technology to Improve the Efficiency of Inventory Management for Small Businesses*” dilatarbelakangi oleh ketidakefesiensian sebuah pengelolaan inventory management pada bisnis-bisnis kecil. Dalam penelitian tersebut, manajemen inventory merupakan hal yang penting bagi bisnis-bisnis kecil. Akan tetapi bisnis-bisnis kecil tersebut lebih suka menggunakan Microsoft Office daripada sistem informasi khusus. Konsep OLAP (On-Line Analytical Processing) merupakan salah satu solusi daripada permasalahan tersebut. OLAP memungkinkan pencarian informasi yang fleksibel dan cepat, berbagai operasi analitik termasuk pelaporan data, dan analisis perbandingan data dari waktu ke waktu. Perangkat lunak berkonsep OLAP cukup mahal bahkan bagi perusahaan besar sekalipun. Oleh karena itu menggunakan Aplikasi OLAP untuk usaha kecil sangat tidak dianjurkan. Usaha kecil banyak menggunakan software seperti Microsoft Excel. Alternatif MS Excel dapat menjadi solusi sederhana menggunakan DBMS. DBMS tujuan umum menggunakan kueri SQL sebagai kueri untuk tampilan data yang kompleks dan aktivitas pendukung pengambilan keputusan. Jadi, dalam database tujuan umum, alat OLAP diimplementasikan dengan kueri SQL. Makalah ini mengusulkan solusi untuk masalah saldo akuntansi dan penggunaan pendukung keputusan dengan alat sederhana. Permintaan yang disarankan, menyediakan pendaftaran saldo secara real time, menampilkan hasil dalam bentuk grafik, yang memudahkan pengambilan keputusan pengadaan. Contoh yang dipertimbangkan menunjukkan kepatuhan penuh parameter dasar antara karakteristik proses manajemen inventaris, yang dijelaskan menggunakan model matematika, dan implementasi praktis dari suatu proses melalui database. Ini membuktikan kebenaran penggunaan model matematis ini dalam tugas manajemen inventaris, dalam batas area subjek [12].

### **2.9.7. DESIGN AND BUILD INVENTORY MANAGEMENT INFORMATION SYSTEM USING THE SCRUM METHOD**

Penelitian yang dilakukan oleh Ilham Firman Ashari dkk yang berjudul “*Design and Build Inventory Management Information System Using the Scrum Method*” dilakukan di PT Telkom Akses. Penelitian ini dilatarbelakangi oleh PT Telkom Akses yang masih melakukan pendataan secara manual, hal tersebut bisa mengakibatkan lambatnya pengelolaan data dan akan berdampak pada *workflow* PT Telkom Akses. Banyak data yang dimiliki akan rusak atau terduplikasi, dan pencarian data akan sulit. Maka dari itu perlu dilakukan digitalisasi dalam pendataan barang agar lebih efisien. Sehingga data tidak akan mudah rusak atau terduplikasi. Penelitian ini menggunakan metode *Scrum* sebagai pengembangan sistem aplikasinya. Dalam pengembangannya, digunakan *sprint scrum* seperti *sprint planning*, *sprint backlog*, *sprint review* dan *daily scrum*. Hasil dari penelitian ini adalah sistem inventarisasi berbasis web yang dapat digunakan oleh pengguna untuk mendata barang masuk dan keluar secara real time. Berdasarkan hasil evaluasi pengujian dengan menggunakan metode pengujian *black box* pada sistem sesuai dengan desain awal dan kebutuhan yang diinginkan. Pengembangan sistem inventory ini menggunakan framework CodeIgniter 3 dengan bahasa pemrograman PHP dan menggunakan metodologi *Scrum* [1].

### BAB III METODE PENELITIAN

#### 3.1. Waktu dan Tempat

Waktu dan tempat penelitian dilaksanakan pada:

- a. Waktu Penelitian : Februari sampai dengan Juni 2023
- b. Tempat Penelitian : Universitas Lampung

#### 3.2. Jadwal Penelitian

Penelitian dilakukan selama lima bulan dengan rinciana sebagai berikut:

Kegiatan	Pelaksanaan				
	Februari	Maret	April	Mei	Juni
Studi Literatur					
Penulisan Laporan					
Development					

#### 3.3. Alat Penelitian

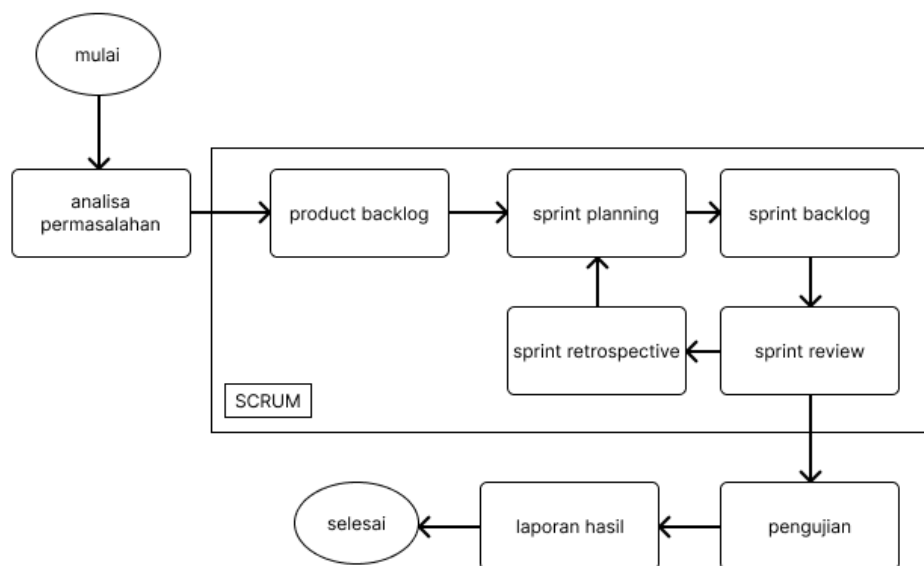
Alat-alat yang digunakan dalam penelitian rancang bangun aplikasi manajemen *inventory* sebagai berikut:

No.	Nama	Spesifikasi
1	Laptop Lenovo V14-IKB	<i>Intel@Core I3-8130U@2.2 Ghz, RAM 12GB, Sistem Operasi Windows 11</i>
2	<i>Figma</i>	<i>Desktop Version</i>
3	<i>Android Studio</i>	Versi 2022.1.1

4	<i>Flutter</i>	Versi 3.3.9
5	<i>Dart</i>	Versi 2.18.5
6	<i>Microsoft Edge</i>	Versi 111.0.1661.62
7	<i>Firebase</i>	Versi 8.3.2
8	Android Emulator	Android 10.0

### 3.4. Tahapan Penelitian

Terdapat beberapa tahapan yang dilakukan dalam penelitian ini, dapat dilihat pada gambar di bawah ini.



Gambar 2 Diagram alir metode scrum

Gambar 2 di atas merupakan tahapan yang dilakukan ketika melaksanakan penelitian, dimulai dengan menganalisis permasalahan yang dihadapi oleh *user* yang nantinya akan menjadi acuan dalam pembuatan *product backlog*, lalu setelah



*product backlog* dibuat, dilanjutkan dengan membuat sebuah *sprint planning* yang memuat waktu yang dibutuhkan dalam pembuatan *backlog*. Tahap selanjutnya yaitu melakukan *sprint backlog*, yang diambil dari di *sprint planning* telah ditentukan. Dilanjutkan dengan *sprint review* yang akan melihat apakah *sprint backlog* sudah sesuai dengan apa yang dibutuhkan atau belum, jika belum memenuhi maka akan dilakukan *sprint retrospective* untuk bahan evaluasi, jika sesuai dilanjutkan dengan pengujian aplikasi hingga penulisan laporan hasil.

#### **3.4.1. Analisa Permasalahan**

Pada tahap ini dilakukan analisis permasalahan yang ada di pergudangan. Saat ini manajemen pergudangan hanya menggunakan sebuah buku untuk melakukan pencatatan barang keluar, barang masuk, hingga pendataan stok barang. Data barang-barang tersebut memiliki risiko yang tinggi untuk terjadinya kesalahan dalam pendataan sebuah barang. Sehingga diperlukan sebuah alternatif baru untuk melakukan pendataan barang yang meminimalisir terjadinya kesalahan dalam pencatatan barang, baik barang masuk maupun barang keluar. Aplikasi manajemen barang ini dibuat dalam bentuk *mobile app* yang dibuat menggunakan *framework flutter* dan terintegrasi dengan *database*. Dalam hubungan aplikasi dengan *database* dibutuhkan sebuah koneksi internet, baik dari data seluler maupun *wifi* agar aplikasi dapat mengambil data yang telah terintegrasi dengan *database*.

#### **3.4.2. Product Backlog**

*Product backlog* adalah langkah pertama dalam metodologi Scrum. *Product backlog* terdiri dari tugas-tugas (*tasks*) yang akan dikembangkan pada sebuah sistem. Tugas-tugas ini dapat berupa fungsional, non-fungsional, peningkatan, perbaikan, tambalan, ide, atau persyaratan lain yang diperlukan oleh sistem. Hasil analisis yang telah dilakukan akan dijadikan *product backlog* yang kemudian dikerjakan pada bagian *sprint*.

### **3.4.3. *Sprint***

Sprint adalah tahap kedua setelah product backlog. Tahapan sprint menentukan batas waktu atau estimasi pengerjaan dengan durasi produk satu bulan atau kurang, dimana ada proses increment “selesai”, yang dapat digunakan dan berpotensi untuk dirilis. Sprint juga memiliki durasi yang konsisten sepanjang siklus hidup selama pengembangan produk. Sprint baru dimulai segera setelah sprint sebelumnya selesai. Pada tahap ini terdapat dua kegiatan yaitu sebagai berikut:

#### **3.4.3.1. *Sprint Planning***

*Sprint planning* merupakan tahapan yang dilakukan setelah penentuan *product backlog* yang telah disusun, yang akan membahas tujuan dan misi dari setiap fitur, dan juga menentukan waktu yang dibutuhkan untuk menyelesaikan setiap fitur yang ingin dikembangkan.

#### **3.4.3.2. *Sprint Backlog***

Pada tahap pembuatan *sprint backlog* yang membuat daftar kebutuhan yang menjadi prioritas dengan waktu pengerjaan yang telah ditentukan oleh *stakeholder* yang terlibat. Tahapan ini juga melengkapi fitur-fitur yang telah ditentukan pada *product backlog* dan *sprint planning*.

#### **3.4.3.3. *Sprint Review***

*Sprint review* dilaksanakan di akhir sprint untuk menginspeksi *increment* dan meninjau apa saja *product backlog* yang telah dan belum diselesaikan. Inspeksi bertujuan untuk melakukan perbaikan apa saja yang perlu dilakukan untuk proses pengembangan di *sprint* berikutnya

#### **3.4.3.4. *Sprint Retrospective***

*Sprint retrospective* dilakukan setelah *sprint review* dan sebelum rencana *sprint* berikutnya. Aktivitas ini bertujuan untuk agar kendala dan kekurangan pada sprint sebelumnya bisa diminimalisir.

#### **3.4.4. Pengujian Aplikasi**

Tahap selanjutnya yaitu pengujian aplikasi, pada tahap pengujian ini digunakan metode *Black Box Testing*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap *form*-nya.

#### **3.4.5. Penulisan Laporan**

Tahapan selanjutnya yaitu penulisan laporan. Tujuan dari penulisan laporan ini yaitu untuk mendokumentasikan penelitian yang telah dilakukan, agar bisa bermanfaat bagi pembaca.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1. Hasil

Penelitian ini menghasilkan sebuah aplikasi manajemen *inventory* berbasis android, dimana pengguna dapat melakukan pendataan barang dan pemantauan barang secara *online* yang terintegrasi dengan *database*.

### 4.2. Implementasi pembuatan aplikasi menggunakan Scrum

#### 4.2.1. *User Story*

Dalam pelaksanaan penelitian dilakukan analisis dan didapatkan sebuah hasil, yang pertama yaitu *user story*. *User story* memuat hal sebagai berikut:

1. Pengguna aplikasi ingin melakukan pendataan barang.
2. Pengguna aplikasi ingin melihat semua data barang yang tersedia.
3. Pengguna aplikasi ingin melihat data barang yang memuat data stok barang, dan tanggal kadaluwarsa.
4. Pengguna aplikasi ingin data barang dapat diubah.

#### 4.2.2. *Product Backlog*

Dalam pengerjaan penelitian mendapatkan hasil *product backlog* yang memiliki kaitan dengan *user story*. *Product backlog* sendiri mengacu pada daftar fungsionalitas yang diprioritaskan yang harus dimiliki oleh sebuah sistem, dalam hal ini adalah manajemen *inventory*. *Product backlog* yang diperoleh dapat dijabarkan sebagai berikut:

1. Sistem mampu melakukan *login* menggunakan email dan *password*.
2. Sistem mampu menampilkan daftar barang yang telah dimasukkan ke dalam *database*
3. Sistem mampu membuat, mengedit, dan menghapus barang.
4. Sistem mampu memfilter barang yang ada di *database*.
5. Sistem mampu mendeteksi barang yang memiliki stok kosong.
6. Sistem mampu mendeteksi barang yang kadaluwarsa.
7. Sistem mampu mendeteksi barang yang stoknya menipis.

8. Sistem mampu melakukan *logout*.

#### 4.2.3. *Sprint Planning*

Berdasarkan *backlog* yang telah dibuat maka dilakukan tahapan selanjutnya yaitu perencanaan sprint atau *sprint planning*. Di dalam *sprint planning* memuat informasi terkait *backlog* yang telah dibuat, lalu dijabarkan dalam sebuah tabel dengan informasi yang lebih detail sekaligus memuat informasi durasi pelaksanaan sprint.

Tabel 1 *Backlog Sprint Planning*

<i>Sprint ID</i>	<i>Backlog Item</i>	<b>Estimasi (hari)</b>
1	Pembuatan <i>user interface</i>	1
2	Perancangan <i>UML</i> untuk kebutuhan sistem aplikasi	2
3	<i>Slicing UI</i> bagian <i>splash screen</i> , <i>onboarding</i> , dan menu utama	2
4	<i>Slicing UI</i> halaman <i>login</i>	2
5	<i>Slicing UI</i> halaman <i>dashboard</i>	2
6	<i>Slicing UI</i> halaman <i>menu item</i> , halaman <i>detail item</i> , dan halaman <i>edit item</i>	3
7	<i>Slicing UI</i> halaman <i>incoming item</i> dan halaman <i>outcoming item</i>	3
8	<i>Slicing UI</i> halaman <i>profile</i>	2
9	Pembuatan dan pemasangan <i>database</i> menggunakan <i>firebase</i> pada aplikasi	1
10	Implementasi <i>RestAPI</i> untuk <i>login</i> dan <i>logout</i>	3
11	Implementasi <i>RestAPI</i> untuk pengambilan data barang pada halaman <i>menu item</i>	3
12	Implementasi <i>RestAPI</i> untuk penambahan barang pada halaman <i>incoming item</i>	3

13	Implementasi <i>RestAPI</i> untuk pengurangan barang pada halaman <i>outcoming item</i>	3
14	Implementasi <i>RestAPI</i> untuk pengeditan barang pada halaman <i>edit item</i>	3
15	Implementasi <i>RestAPI</i> untuk penghapusan barang pada halaman <i>detail item</i>	3

Berdasarkan pada *backlog sprint planning* yang telah direncanakan maka dilakukan tahapan selanjutnya yaitu merencanakan pengerjaan *sprint* yang dijabarkan ke dalam *sprint planning*.

Berikut penjabaran *sprint planning* yang dirancang dan akan dilakukan:

1. *Sprint planning* pertama dilakukan *sprint* pengembangan yang bertujuan untuk melakukan pembuatan *user interface*, dan pengonsepan sistem kebutuhan perangkat lunak dari aplikasi manajemen *inventory*. Dalam *Sprint planning* pertama ini dilakukan pengerjaan *Sprint ID 1* pembuatan *user interface* dan *Sprint ID 2* perancangan *UML* untuk kebutuhan sistem aplikasi.
2. *Sprint planning* kedua dilakukan *sprint slicing user interface* aplikasi untuk tampilan seluruh fitur aplikasi menggunakan android studio. Dalam *sprint planning* kedua ini dilakukan pengerjaan *Sprint ID 3* pembuatan tampilan *splash screen*, *onboarding*, dan menu utama, *Sprint ID 4* pembuatan tampilan halaman *login*, *Sprint ID 5* pembuatan tampilan halaman *dashboard*, *Sprint ID 6* pembuatan tampilan halaman *menu item*, halaman *detail item*, dan halaman *edit item*, *Sprint ID 7* pembuatan tampilan halaman *incoming item* dan halaman *outcoming item*, *Sprint ID 8* pembuatan tampilan halaman *profile*.
3. *Sprint planning* ketiga akan dilakukan *sprint* implementasi *RestAPI* untuk penghubungan aplikasi *Android* dengan *database* menggunakan *Firebase*. Dalam *Sprint planning* ketiga ini dilakukan pengerjaan *Sprint ID 9* melakukan pembuatan *RestAPI Firebase* dan pemasangan ke dalam aplikasi, *Sprint ID 10* melakukan implementasi *RestAPI* untuk *login* dan *logout*, *Sprint ID 11* melakukan implementasi *RestAPI* untuk pengambilan

data barang, *Sprint ID* 12 melakukan implementasi *RestAPI* untuk penambahan data barang, *Sprint ID* 13 melakukan implementasi *RestAPI* untuk pengurangan data barang, *Sprint ID* 14 melakukan implementasi *RestAPI* untuk pengeditan data barang, *Sprint ID* 15 melakukan implementasi *RestAPI* untuk penghapusan data barang.

#### 4.2.4. *Sprint Backlog*

##### 4.2.4.1. *Sprint Backlog Pertama*

###### a. **Product Backlog Item Sprint 1**

*Product Backlog Item* disusun berdasarkan dari sprint planning yang telah dilakukan. Berikut merupakan list Product Backlog Item yang dilakukan pada sprint pertama:

Tabel 2 *Sprint Backlog Item Pertama*

<i>Sprint ID</i>	<i>Backlog Item</i>
1	Pembuatan <i>user interface</i>
2	Perancangan <i>UML</i> untuk kebutuhan sistem aplikasi

###### b. **Hasil Sprint**

Sprint pertama dilakukan selama 1 minggu. Sprint pertama terbagi menjadi dua Sprint ID sama seperti yang tertera pada tabel 2, dimana Sprint ID 1 merupakan pembuatan user interface dan Sprint ID 2 merupakan perancangan Unified Markup Language (UML) untuk kebutuhan sistem aplikasi. Dalam pembuatan user interface digunakan aplikasi Figma desktop version. Perancangan UML mencakup use case diagram dan activity diagram.

###### a) **User Interface**

User interface merupakan sebuah tampilan yang nantinya akan menjadi acuan dalam pembuatan aplikasi. Berikut merupakan user interface yang telah dibuat:



Gambar 3 halaman *splash screen* aplikasi



Gambar 4 halaman *onboarding* aplikasi

9:41

LOGITORY

**LOGIN**

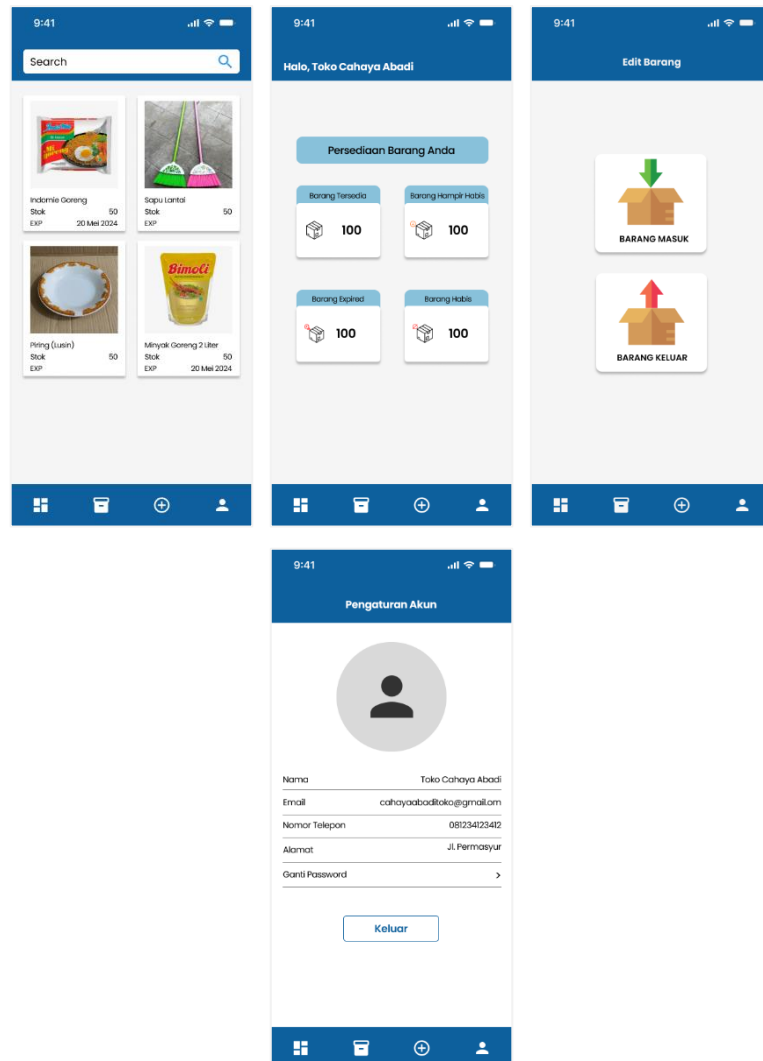
**Email**  
Masukkan username Anda

**Password**  
Masukkan password Anda

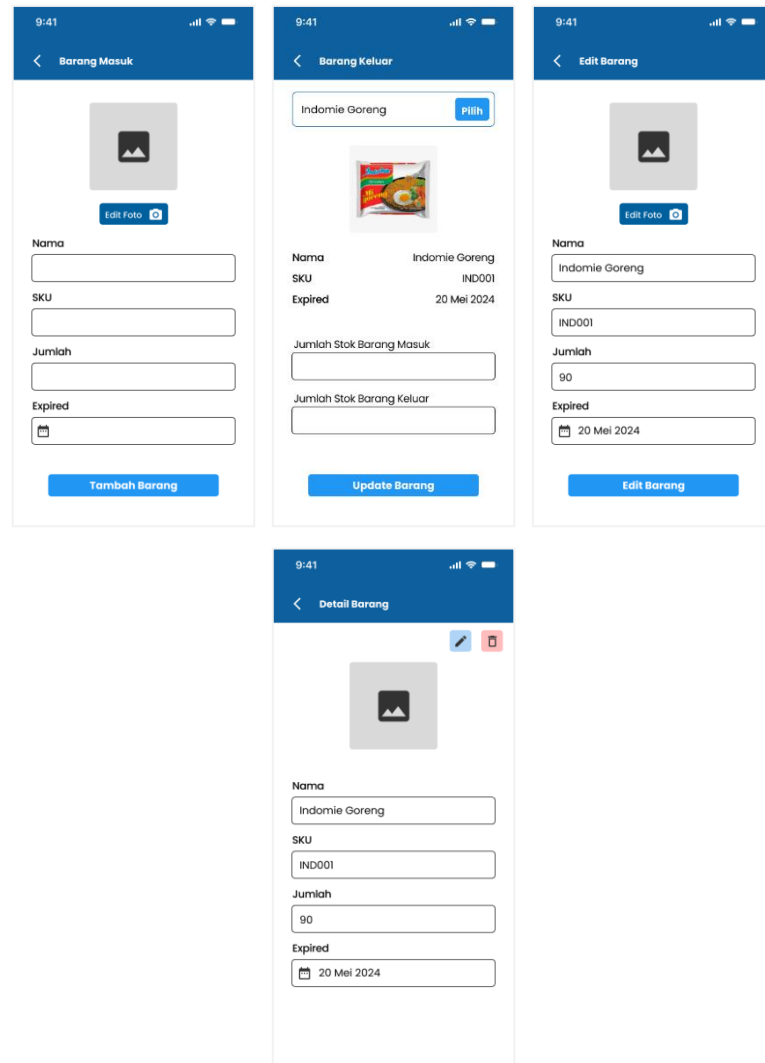
Masuk

Gambar 5 halaman *login* aplikasi





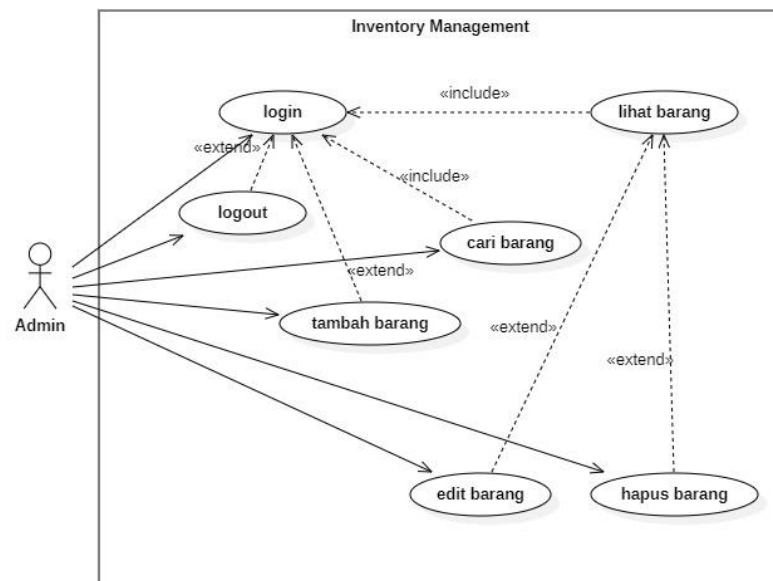
Gambar 6 halaman menu utama aplikasi



Gambar 7 halaman manajemen barang

### b) Use Case Diagram

Use case diagram adalah satu dari jenis diagram UML (Unified Modeling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Berikut ini merupakan use case diagram pada sistem inventory management:



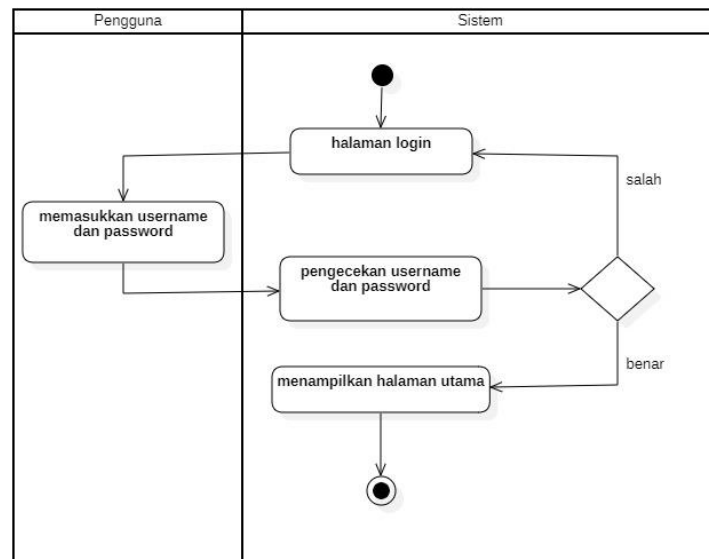
Gambar 8 use case diagram intentry management

Gambar 8 menjelaskan tentang use case diagram pada sistem management inventory. Sistem dikembangkan dengan hanya memiliki satu aktor, yaitu admin. Admin dapat melakukan login ke dalam sistem, menambahkan barang, mengedit barang, menghapus barang, dan mencari barang ke dalam sistem, serta admin juga dapat melakukan logout dari sistem.

### c) Activity Diagram

Aplikasi inventory management berbasis android memiliki beberapa fitur yang akan dijelaskan dalam activity diagram berikut:

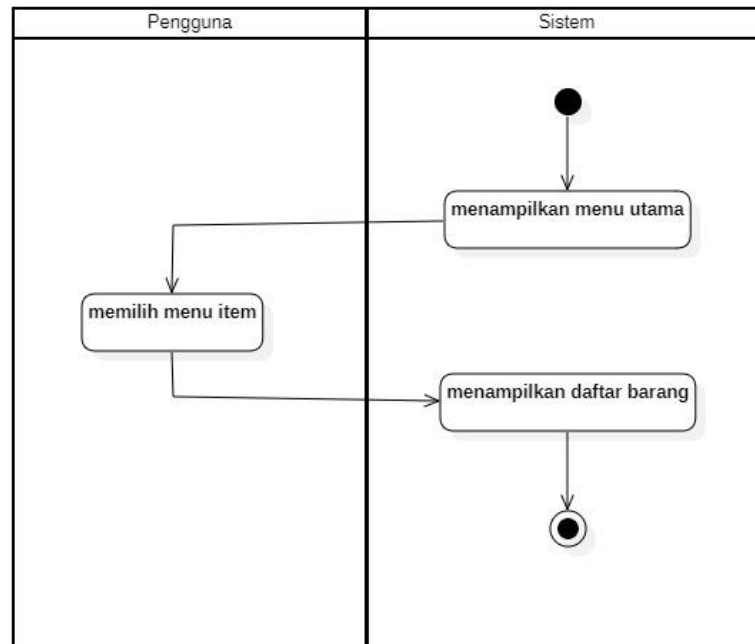
- a. Activity diagram login user



Gambar 9 activity diagram login user

Gambar 9 di atas menjelaskan tentang activity diagram ketika pengguna ingin melakukan login ke dalam aplikasi. Dimulai dari sistem menampilkan sebuah halaman login, lalu pengguna diminta untuk memasukkan sebuah username dan password, lalu sistem akan melakukan pengecekan, apabila username dan password yang dimasukkan benar, maka sistem akan mengarahkan pengguna ke halaman utama, tetapi jika username atau password salah, maka sistem akan kembali menampilkan halaman login.

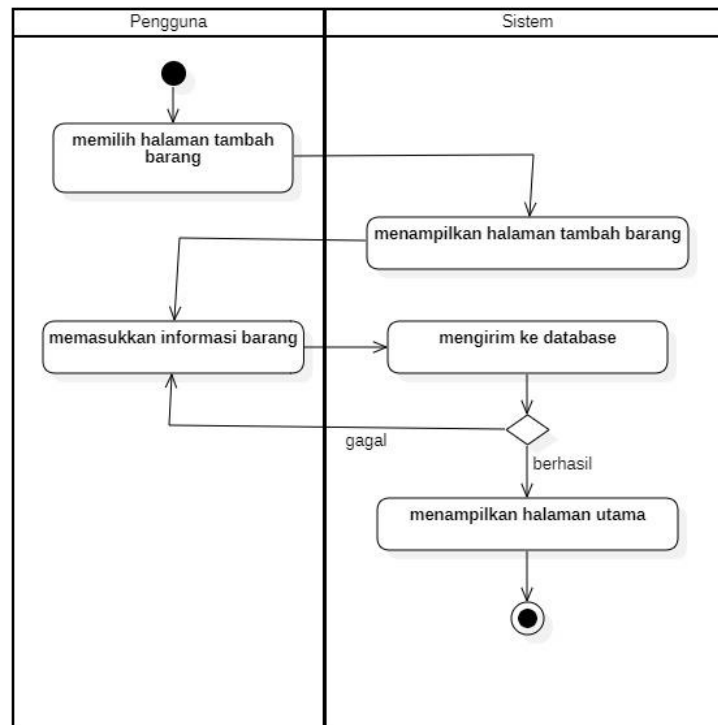
## b. Activity diagram melihat barang



Gambar 10 activity diagram melihat barang

Gambar 10 di atas menjelaskan tentang activity diagram ketika pengguna ingin melihat barang yang ada di dalam aplikasi. Dimulai saat pengguna berada di halaman menu utama, lalu pengguna memilih halaman item, lalu sistem akan otomatis menampilkan seluruh barang yang tersedia di dalam sistem.

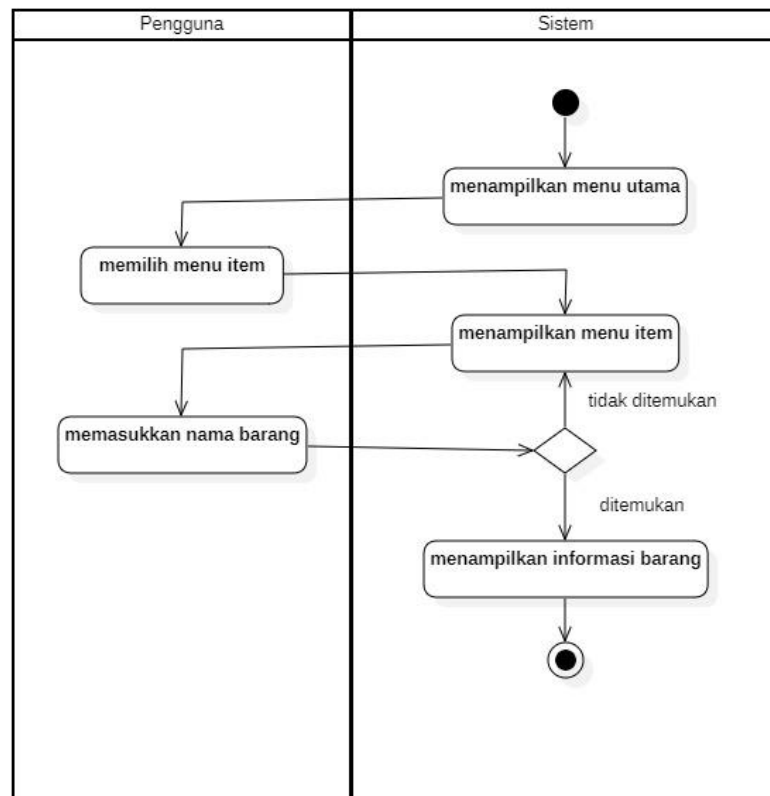
## c. Activity diagram menambahkan barang



Gambar 11 activity diagram menambahkan barang

Gambar 11 di atas menjelaskan tentang activity diagram ketika pengguna ingin menambahkan sebuah barang ke dalam sistem. Dimulai dari pengguna memilih halaman tambah barang, lalu sistem akan menampilkan halaman tambah barang, selanjutnya pengguna diminta untuk memasukkan informasi barang yang akan dimasukkan. Ketika selesai memasukkan informasi barang, data barang akan dikirim ke database, jika barang berhasil ditambahkan ke dalam sistem, maka sistem akan mengarahkan pengguna ke halaman utama, jika gagal maka user akan diminta untuk memasukkan informasi barang lagi.

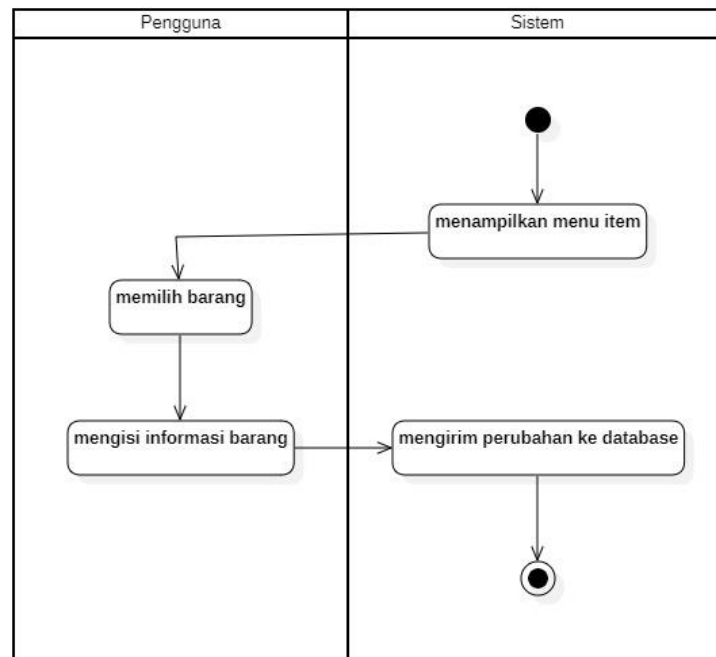
## d. Activity diagram mencari barang



Gambar 12 activity diagram mencari barang

Gambar 12 di atas menjelaskan tentang activity diagram ketika pengguna ingin mencari sebuah barang di dalam sistem. Dimulai dari sistem yang menampilkan menu utama, lalu pengguna akan memilih menu item, selanjutnya sistem akan menampilkan menu item beserta barang-barang yang ada di sistem. Lalu pengguna memasukkan nama barang yang akan dicari, jika ditemukan sistem akan menampilkan informasi barang tersebut, jika tidak ditemukan sistem akan menampilkan menu item kembali.

## e. Activity diagram mengedit barang

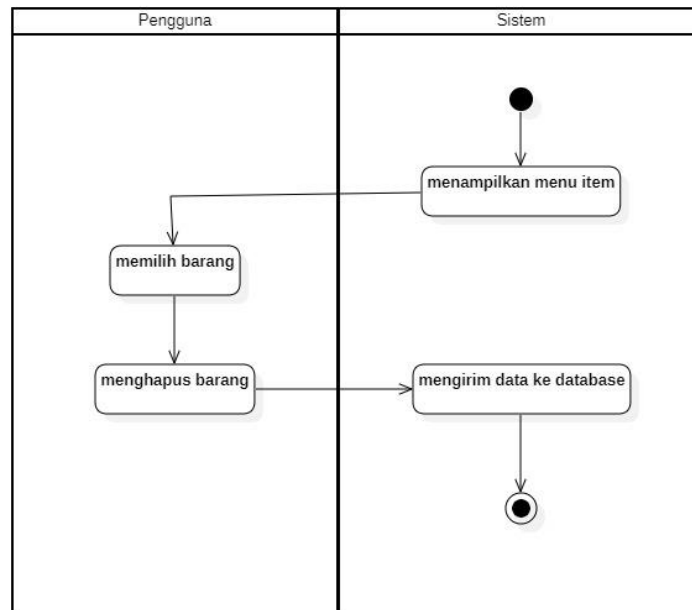


Gambar 13 activity diagram mengedit barang

Gambar 13 di atas menjelaskan tentang activity diagram ketika pengguna ingin melakukan pengeditan terhadap sebuah barang yang ada di sistem. Dimulai dari sistem yang menampilkan menu item, lalu pengguna memilih sebuah barang yang akan diedit, selanjutnya pengguna akan mengisikan informasi barang yang akan diedit. Setelah itu data perubahan yang telah dilakukan selanjutnya akan dikirim ke database.



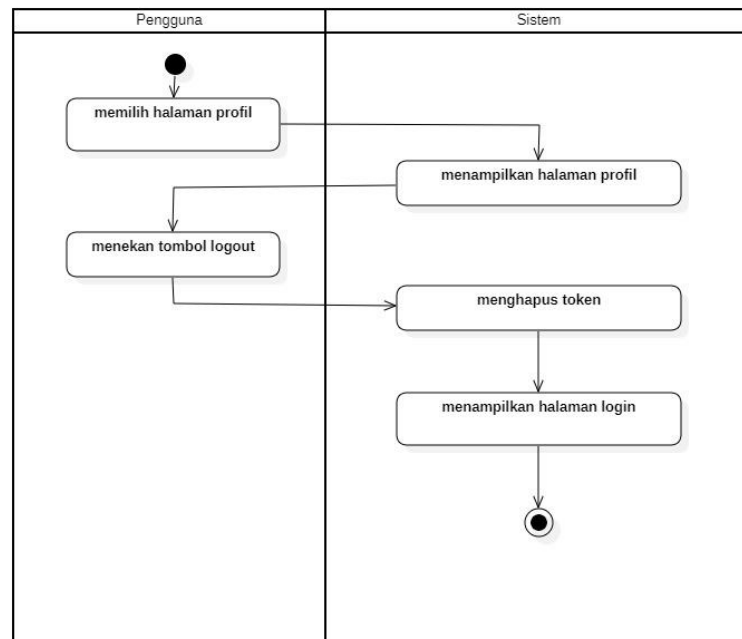
## f. Activity diagram menghapus barang



Gambar 14 activity diagram menghapus barang

Gambar 14 di atas merupakan sebuah activity diagram ketika pengguna ingin menghapus sebuah barang dari sistem. Dimulai dari sistem yang menampilkan menu item, lalu pengguna memilih sebuah barang yang akan dihapus, selanjutnya pengguna akan menghapus barang yang telah dipilihnya. Setelah itu penghapusan data yang telah dilakukan selanjutnya akan dikirim ke database.

## g. Activity diagram logout user



Gambar 15 activity diagram logout

Gambar 15 di atas merupakan sebuah activity diagram ketika pengguna ingin melakukan logout dan keluar dari aplikasi. Dimulai dari pengguna memilih halaman profil, lalu sistem akan menampilkan halaman profil. Selanjutnya pengguna menekan tombol logout, lalu sistem akan menghapus token yang ada, lalu sistem akan mengarahkan pengguna ke halaman login.

4.2.4.2. *Sprint Backlog Kedua*a. **Product Backlog Item Sprint 2**

Berikut merupakan list dari backlog item yang akan dilakukan pada sprint kedua:

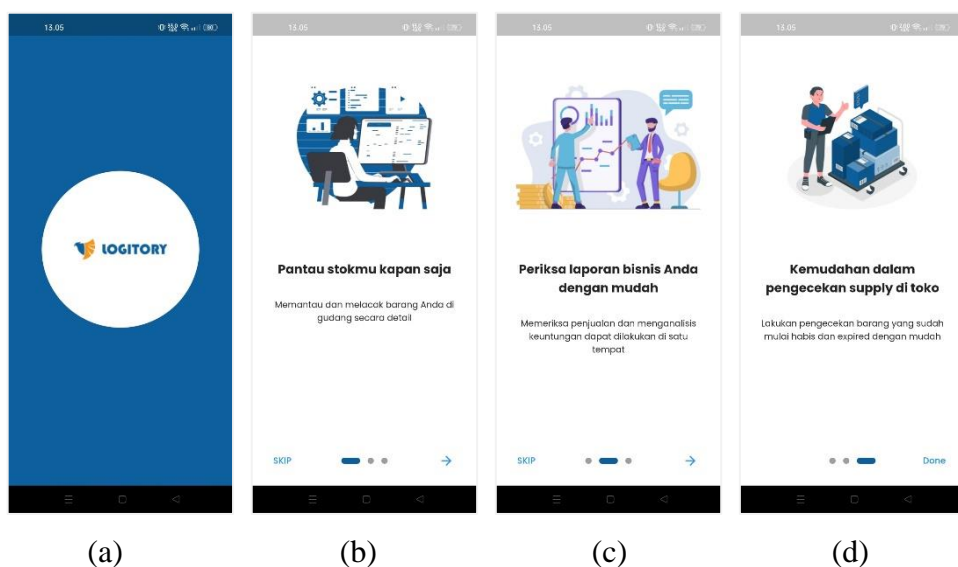
Tabel 3 *Sprint Backlog Item Kedua*

<i>Sprint ID</i>	<i>Backlog Item</i>
3	<i>Slicing UI bagian splash screen, onboarding dan menu utama</i>
4	<i>Slicing UI halaman login</i>

5	<i>Slicing UI halaman dashboard</i>
6	<i>Slicing UI halaman menu item, halaman detail item, dan halaman edit item</i>
7	<i>Slicing UI halaman incoming item dan halaman outcoming item</i>
8	<i>Slicing UI halaman profile</i>

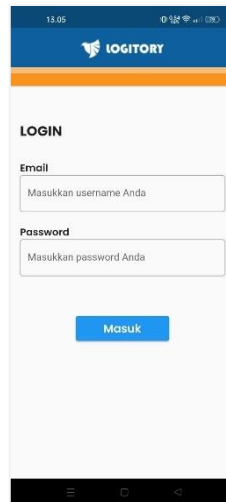
### b. Hasil Sprint

Sprint kedua merupakan sprint yang berfokus untuk membuat tampilan aplikasi dari desain user interface yang telah dibuat. Pada sprint kedua ini, dibagi menjadi 6 Sprint ID yang masing-masingnya berfokus pada pembuatan tampilan pada aplikasi. Sprint ID 3 membuat tampilan pada halaman splash screen, *onboarding*, dan menu utama, Sprint ID 4 membuat tampilan pada halaman login, Sprint ID 5 membuat tampilan pada halaman dashboard, Sprint ID 6 membuat tampilan pada halaman menu item, halaman detail item, dan halaman edit item, Sprint ID 7 membuat tampilan pada halaman incoming item dan halaman outcoming item, serta Sprint ID 8 membuat tampilan pada halaman profil. Sprint kedua terlaksana dalam waktu 2 minggu.



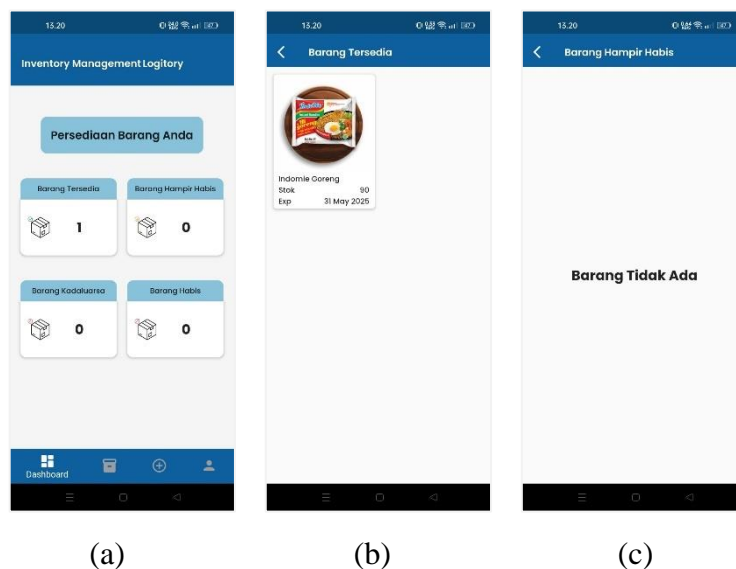
Gambar 16 (a) Halaman Splash Screen, (b) Halaman Onboarding 1, (c) Halaman Onboarding 2, (d) Halaman Onboarding 3

Gambar 16 merupakan hasil dari melakukan sprint dengan Sprint ID 3 yang berisikan pembuatan tampilan halaman splash screen dan halaman onboarding.



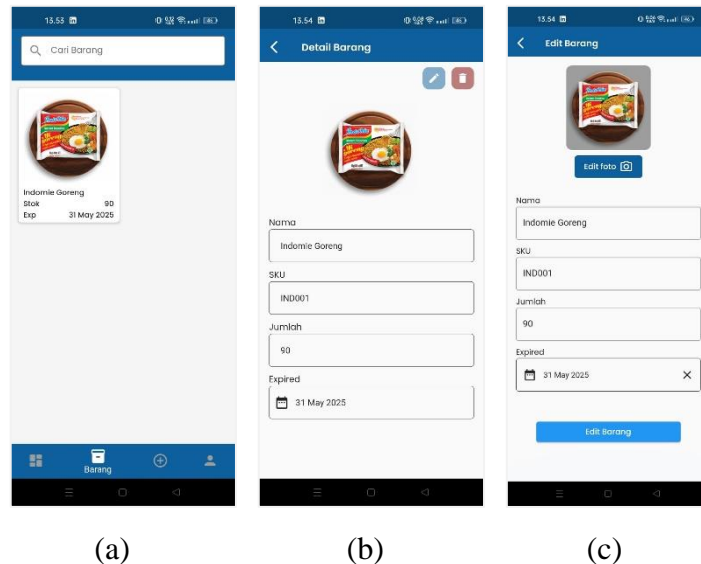
Gambar 17 Halaman Login

Gambar 17 merupakan hasil dari melakukan sprint dengan Sprint ID 4 yang berisikan pembuatan tampilan halaman login.



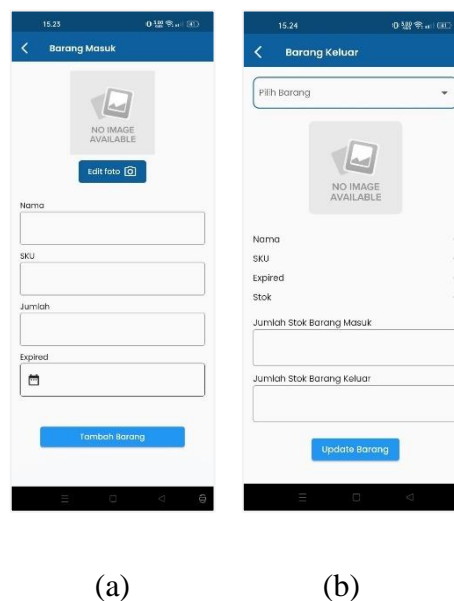
Gambar 18 (a) Halaman Dashboard, (b) Halaman Dashboard Barang Tersedia, (c) Halaman Dashboard Barang Hampir Habis

Gambar 18 merupakan hasil dari melakukan sprint dengan Sprint ID 5 yang berisikan pembuatan tampilan halaman dashboard.



Gambar 19 (a) Halaman Menu Item, (b) Halaman Detail Barang, (c) Halaman Edit Barang

Gambar 19 merupakan hasil dari melakukan sprint dengan Sprint ID 6 yang berisikan pembuatan tampilan menu item, detail item, dan edit item.



Gambar 20 (a) Halaman barang masuk (b) Halaman barang keluar

Gambar 20 merupakan hasil dari melakukan sprint dengan Sprint ID 7 yang berisikan pembuatan tampilan halaman barang masuk dan halaman barang keluar.



Gambar 21 Halaman Profil

Gambar 21 merupakan hasil dari melakukan sprint dengan Sprint ID 8 yang berisikan pembuatan tampilan halaman profil.

#### 4.2.4.3. *Sprint Backlog Ketiga*

##### a. **Product Backlog Item Sprint 3**

Berikut merupakan list dari backlog item yang akan dilakukan pada sprint kedua:

Tabel 4 Sprint Backlog Item Ketiga

<i><b>Sprint ID</b></i>	<i><b>Backlog Item</b></i>
9	Pembuatan dan pemasangan <i>database</i> menggunakan <i>firebase</i> pada aplikasi
10	Implementasi <i>RestAPI</i> untuk <i>login</i> dan <i>logout</i>
11	Implementasi <i>RestAPI</i> untuk pengambilan data barang pada halaman <i>menu item</i>

12	Implementasi <i>RestAPI</i> untuk penambahan barang pada halaman <i>incoming item</i>
13	Implementasi <i>RestAPI</i> untuk pengurangan barang pada halaman <i>outcoming item</i>
14	Implementasi <i>RestAPI</i> untuk pengeditan barang pada halaman <i>edit item</i>
15	Implementasi <i>RestAPI</i> untuk penghapusan barang pada halaman <i>detail item</i>

### b. Hasil Sprint

Sprint ketiga merupakan sprint yang berfokus untuk membuat database untuk aplikasi sekaligus implementasi RestAPI agar aplikasi dapat terhubung dengan database dan aplikasi bisa berjalan sesuai fungsinya. Pada sprint ketiga ini dibagi menjadi 7 bagian sprint, dimana masing masing Sprint ID berfokus untuk implementasi RestAPI. Sprint ID 9 membuat dan memasang database firebase kedalam aplikasi, Sprint ID 10 implementasi ResAPI untuk login dan logout, Sprint ID 11 implementasi RestAPI untuk pengambilan data barang yang akan ditampilkan pada menu item, Sprint ID 12 implementasi RestAPI untuk penambahan data barang pada halaman incoming item, Sprint ID 13 implementasi RestAPI untuk pengurangan data barang pada halaman outcoming item, Sprint ID 14 implementasi RestAPI untuk pengeditan data barang pada halaman edit item, dan Sprint ID 15 implementasi RestAPI untuk penghapusan data barang pada halaman detail item.

```
Future<void> signIn(String email, String password) async {
  try {
    await _firebaseAuth.signInWithEmailAndPassword(
      email: email, password: password);
  } catch (e) {
    throw Exception(e.toString());
  }
}
```

(a)

```

void _onSignInEvent(SignInButtonPressed event, Emitter emit) async {
  emit(AuthLoadingState());
  try {
    await authRepository
      .signIn(event.email, event.password)
      .then((value) => emit(LoginSuccessState()));
  } catch (e) {
    emit(AuthErrorState("email atau password Anda salah!"));
  }
}

```

(b)

```

submitButton() {
  if (emailController.text.isNotEmpty && passwordController.text.isNotEmpty) {
    context.read<AuthBloc>().add(
      SignInButtonPressed(emailController.text, passwordController.text));
  } else {
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text(mustFillField),
      duration: Duration(seconds: 1, milliseconds: 5),
    )); // SnackBar
  }
}

```

(c)

Gambar 22 (a) Kode program login, (b) Kode program event login, (c)  
Kode program login button

Gambar 22 merupakan hasil dari melakukan sprint dengan Sprint ID 10 untuk pengimplementasian RestAPI bagian login. Gambar 22 (a) merupakan kode program yang berfungsi untuk menghubungkan aplikasi dengan database. Function signIn memiliki dua buah parameter yang harus diisi ketika akan dijalankan, yaitu sebuah String email dan String password. Lalu function akan mengirimkan parameter tersebut ke database. Gambar 22 (b) merupakan kode program Event Login yang akan dijalankan oleh sistem ketika pengguna melakukan perubahan event di dalam aplikasi. Event tersebut akan mengubah state saat ini menjadi Loading State saat dijalankan, lalu akan mencoba menjalankan function signIn, jika berhasil maka state akan berubah menjadi Login Success, tetapi jika gagal maka state akan berubah menjadi Authentication Error. Gambar 22 (c) merupakan kode program yang berupa function yang akan berjalan ketika pengguna menekan tombol login pada halaman login. Function tersebut akan berjalan



ketika pengguna telah mengisi form yang telah tersedia, yaitu form email dan form password, setelah itu function akan menjalankan sebuah event bernama `SignInButtonPressed()`, jika form tidak terisi maka aplikasi akan menampilkan pesan peringatan berupa tulisan “Harap bidang diisi terlebih dahulu”.

```
Future logout() async {
  await _firebaseAuth.signOut();
}
```

(a)

```
void _onSignOutEvent(SignOutButtonPressed event, Emitter emit) async {
  emit(AuthLoadingState());
  try {
    await authRepository.logout().then((value) => emit(LogoutSuccessState()));
  } catch (e) {
    emit(AuthErrorState("Gagal untuk logout, coba sekali lagi"));
  }
}
```

(b)

```
child: ElevatedButton(
  onPressed: () {
    BlocProvider.of<AuthBloc>(context)
      .add(SignOutButtonPressed());
  },
  style: ElevatedButton.styleFrom(...), // BorderSide
  child: const Text(logout, style: textHeading3Negative),
), // ElevatedButton
```

(c)

Gambar 23 Kode program logout, (b) Kode program event logout, (c)  
Kode program logout button

Gambar 23 merupakan hasil dari melakukan sprint dengan Sprint ID 10 untuk pengimplementasian RestAPI bagian logout. Gambar 23 (a) merupakan kode program yang berfungsi untuk menghubungkan aplikasi dengan database. Function tersebut merupakan function yang ketika dipanggil maka akan melakukan logout dan menghapus token dari aplikasi.

Gambar 23 (b) merupakan kode program Event Logout yang akan dijalankan oleh sistem ketika pengguna melakukan perubahan event di dalam aplikasi. Event tersebut akan mengubah state saat ini menjadi Loading State saat dijalankan, lalu akan mencoba menjalankan function logout, jika berhasil maka state akan berubah menjadi Logout Success, tetapi jika gagal maka state akan berubah menjadi Authentication Error. Gambar 23 (c) merupakan kode program yang berupa class ElevatedButton yang berisikan tulisan “Logout”, button tersebut memiliki properties onPressed yang ketika ditekan, button tersebut akan menjalankan sebuah event yang bernama SignOutButtonPressed() yang merupakan event ketika melakukan logout.

Setelah menyelesaikan Sprint ID 10, dilanjutkan dengan melakukan sprint dengan Sprint ID 11 yaitu pengimplementasian RestAPI untuk pengambilan sebuah data barang dari database. Berikut hasil dari Sprint ID 11:

```
class ItemModel {
  ItemModel({
    this.id, this.name, this.image,
    this.sku, this.expired, this.stock,});

  ItemModel.fromJson(dynamic json) {
    name = json['name']; image = json['image'];
    sku = json['sku']; id = json['id'];
    expired = json['expired']; stock = json['stock'];
  }
  String? name; String? id;
  String? image; String? sku;
  String? expired; int? stock;

  Map<String, dynamic> toJson() {
    final map = <String, dynamic>{};
    map['name'] = name; map['image'] = image;
    map['sku'] = sku; map['id'] = id;
    map['expired'] = expired; map['stock'] = stock;
    return map;
  }
}
```

(a)

```

getItem() async {
  try {
    final response = await _firebaseFirestore.collection("items").get();
    final result = response.docs.map((e) => ItemModel.fromJson(e)).toList();
    return result;
  } on FirebaseException catch (e) {
    print("Error loaded ${e.toString()}");
  } catch (e) {
    throw Exception(e.toString());
  }
}

```

(b)

```

on<OnGetListItem>((event, emit) async {
  try {
    emit(ItemLoading());
    Future.delayed(const Duration(seconds: 1));
    final itemLoaded = await itemRepository.getItem();
    emit(GetItemListLoaded(itemLoaded));
  } catch (e) {
    emit(ItemError("Error Loaded"));
  }
});

```

(c)

```

@override
void didChangeDependencies() {
  context.read<ItemBloc>().add(OnGetListItem());
  super.didChangeDependencies();
}

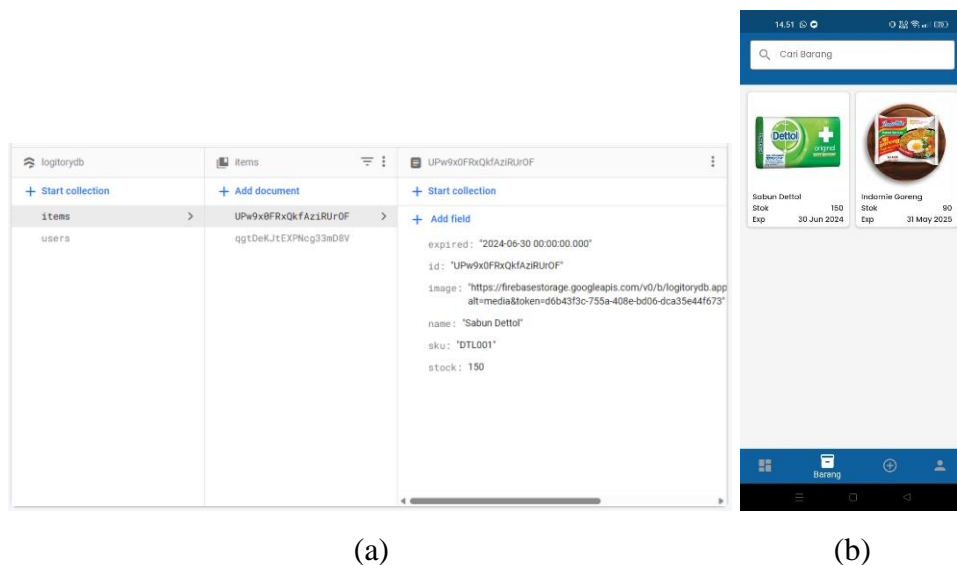
```

(d)

Gambar 24 (a) Kode program item model, (b) Kode program get item, (c) Kode program event get item, (d) Kode program penggunaan get item

Gambar 24 merupakan hasil dari melakukan sprint dengan Sprint ID 11 pengimplementasian RestAPI untuk pengambilan barang. Gambar 24 (a) merupakan sebuah data item model. Item model berfungsi untuk penyimpanan sebuah data yang diambil dari database, data tersebut kemudian akan dikonversikan menjadi sebuah data barang. Gambar 24 (b) merupakan sebuah function untuk pengkoneksian aplikasi dengan database.

Function tersebut berisikan sebuah method get yang digunakan untuk mengambil sebuah data dari collection item firebase, setelah pengambilan data tersebut, function akan mengkonversi data tersebut menjadi sebuah data barang dan menyimpan data tersebut ke dalam variabel result. Gambar 24 (c) merupakan sebuah Event Get Item, event tersebut akan menjalankan dan mengubah state ketika pengguna mengakses Event tersebut, ketika dijalankan akan mengubah state menjadi ItemLoading(), selanjutnya akan mencoba menjalankan function get item seperti pada gambar 24 (b), jika berhasil maka state akan berubah menjadi GetItemListLoaded(), jika gagal maka akan mengubah state menjadi ItemError(). Gambar 24 (d) merupakan function didChangeDependencies() yang akan otomatis berjalan ketika aplikasi dibuka, di dalam function tersebut terdapat sebuah kode yang digunakan untuk memanggil dan mengubah Event yang telah dibuat menjadi OnGetListItem().



Gambar 25 (a) Database item, (b) Halaman Menu Item

Gambar 25 (b) merupakan hasil dari implementasi dari penggunaan RestAPI untuk pengambilan data barang pada halaman menu item. Hal tersebut sesuai dengan data yang berada pada database yaitu pada gambar 25 (a).

Setelah menyelesaikan Sprint ID 11, selanjutnya yaitu menyelesaikan sprint dengan Sprint ID 12 yaitu pengimplementasian RestAPI untuk penambahan barang. Berikut hasil dari menyelesaikan Sprint ID 12:

```
Future<void> addItem(  
    String name,  
    String image,  
    String sku,  
    int stock,  
    String expired,  
) async {  
    final createId = _firebaseFirestore.collection("items").doc().id;  
    await _firebaseFirestore.collection("items").doc(createId).set({  
        "id": createId,  
        "name": name,  
        "image": image,  
        "sku": sku,  
        "stock": stock,  
        "expired": expired,  
    });  
}
```

(a)

```
on<OnSubmittedItem>((event, emit) async {  
    emit(ItemLoading());  
    try {  
        await itemRepository.addItem(  
            event.name, event.image, event.sku, event.stock, event.expired);  
        emit(ItemSubmitted());  
    } catch (e) {  
        emit(ItemError(e.toString()));  
    }  
});
```

(b)

```

Future submitForm() async {
  await ItemRepository().uploadImage(fileImageItem).then((value) {
    imageItemUrl = value;

    if (itemName.text.isNotEmpty && itemStock.text.isNotEmpty &&
        itemSKU.text.isNotEmpty && imageItemUrl!.isNotEmpty) {
      BlocProvider.of<ItemBloc>(context).add(OnSubmittedItem(
        itemName.text, itemSKU.text.toUpperCase(),
        int.parse(itemStock.text), itemExpired,
        imageItemUrl!));
      itemName.clear(); itemSKU.clear();
      itemStock.clear(); itemExpired = "";
      imageItemUrl!;
      setState(() {
        fileImageItem = null;
      });
    } else {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text(mustFillField),
        duration: Duration(seconds: 1),
      )); // SnackBar
    }
  });
}

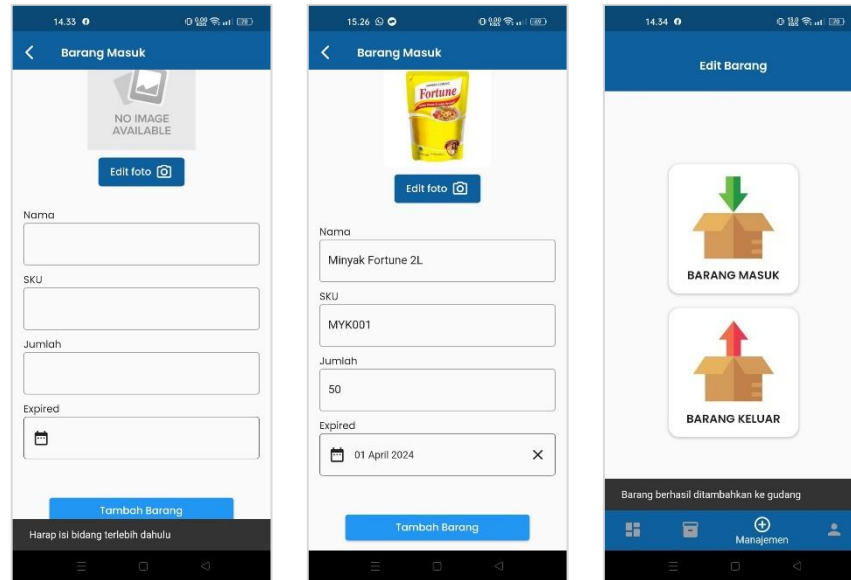
```

(c)

Gambar 26 (a) Kode program add item, (b) Kode program event add item,  
(c) Kode program penggunaan add item

Gambar 26 (a) merupakan sebuah function yang digunakan untuk menambahkan barang dari aplikasi ke database. Function tersebut memiliki beberapa parameter, diantaranya yaitu sebuah nama, gambar, sku, stok, dan tanggal kadaluwarsa. Lalu function tersebut akan mengirimkan data tersebut ke dalam firebase database collection item. Gambar 26 (b) merupakan sebuah Event yang akan merubah state didalamnya ketika pengguna melakukan pemanggilan event tersebut. Event tersebut akan mengubah state awal menjadi ItemLoading(), lalu event akan mencoba menambahkan barang ke dalam database seperti pada gambar 26 (a), jika berhasil maka state akan berubah menjadi ItemSubmitted(), jika gagal maka state akan berubah menjadi ItemError(). Gambar 26 (c) merupakan sebuah function yang di dalamnya terdapat function ItemRepository.uploadImage() yang bergungsi untuk mengunggah sebuah gambar, dan selanjutnya function

akan melakukan pengecekan terhadap inputan-inputan dari pengguna. Jika inputan terisi, maka akan menjalankan sebuah Event yang telah dibuat pada gambar 26 (b), jika inputan tidak terisi maka aplikasi akan menampilkan sebuah pesan pada bagian bawah aplikasi.



(a)

(b)

(c)

logitorydb	items	IB3rQD7M08sfx1EkE6iT
+ Start collection	+ Add document	+ Start collection
items >	IB3rQD7M08sfx1EkE6iT >	+ Add field
users	UPw9x0FRxQkfAz1RUrOF qgtDeKJtEXPNcg33mD8V	expired: "2024-04-01 00:00:00.000" id: "IB3rQD7M08sfx1EkE6iT" image: "https://firebasestorage.googleapis.com/v0/b/logitorydb.app/alt=media&token=e6baa375-34d4-499e-843d-30d4df6883a3" name: "Minyak Fortune 2L" sku: "MYK001" stock: 50

(d)

Gambar 27 (a) Pesan Error ketika bidang kosong, (b) Penambahan barang, (c) Hasil dari penambahan barang, (d) database item

Gambar 27 (a) merupakan sebuah contoh ketika pengguna ingin melakukan penambahan sebuah barang, tetapi pengguna tidak mengisi bidang yang telah disediakan aplikasi. Gambar 27 (b) merupakan contoh ketika

pengguna ingin menambahkan sebuah barang ke dalam database, dan jika berhasil maka pengguna akan diarahkan ke menu utama dan muncul sebuah keterangan bahwa barang telah berhasil ditambahkan, seperti pada gambar 27 (c). Hasil penambahan barang dapat dilihat pada database seperti pada gambar 27 (d).

Setelah menyelesaikan Sprint ID 12, selanjutnya yaitu menyelesaikan sprint dengan Sprint ID 13 dan Sprint ID 14 yaitu pengimplementasian RestAPI untuk pengurangan barang dan pengeditan data barang. Berikut hasil dari menyelesaikan Sprint ID 13 dan Sprint 14:

```
editItem(
  String name,
  String image,
  String sku,
  int stock,
  String expired,
  String docId,
) async {
  DocumentReference docs = _firebaseFirestore.collection("items").doc(docId);

  try {
    docs.update({
      "name": name,
      "image": image,
      "sku": sku,
      "stock": stock,
      "expired": expired,
    });
    print("succes submit repo");
  } on FirebaseException catch (e) {
    print("Error loaded ${e.toString()}");
  } catch (e) {
    print("failed submit repo");
  }
}
```

(a)

```
on<OnEditItem>((event, emit) async {
  emit(ItemLoading());
  try {
    await itemRepository.editItem(
      event.name, event.image, event.sku, event.stock, event.expired, event.docId);
    emit(ItemEdited());
  } catch (e) {
    emit(ItemError(e.toString()));
  }
});
```



(b)

```

Future submitForm() async {
  if (selectedValue.id != null && stockIn.text.isNotEmpty &&
      stockOut.text.isNotEmpty) {
    if(int.parse(stockOut.text) > selectedValue.stock!){
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text("Tidak boleh melebihi stok!"),
        duration: Duration(seconds: 1),
      )); // SnackBar
    }
    BlocProvider.of<ItemBloc>(context).add(
      OnEditItem(
        selectedValue.name!, selectedValue.sku!,
        (selectedValue.stock! - int.parse(stockOut.text) + int.parse(stockIn.text)),
        selectedValue.expired!, selectedValue.image!, selectedValue.id!,
      ),
    );
  } else {
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text(mustFillField),
      duration: Duration(seconds: 1),
    )); // SnackBar
  }
}

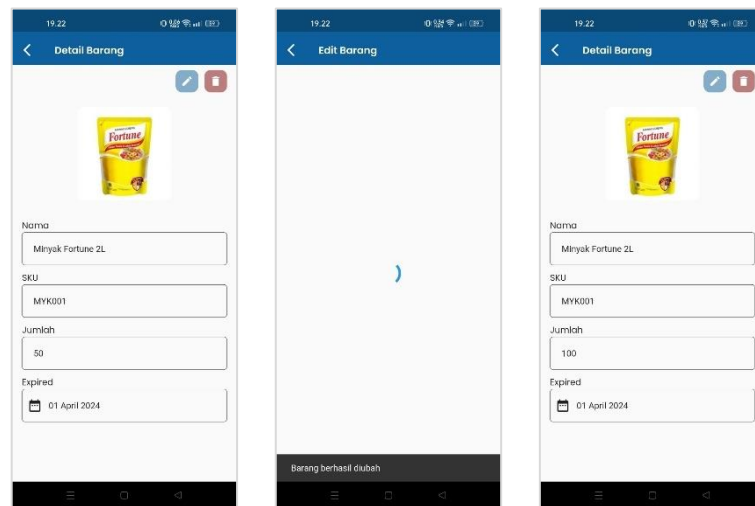
```

(c)

Gambar 28 (a) Kode program edit item, (b) Kode program event add item,  
(c) Kode program penggunaan edit item

Gambar 28 (a) merupakan sebuah function yang digunakan untuk mengupdate barang dari aplikasi ke database. Function tersebut memiliki beberapa parameter, diantaranya yaitu sebuah nama, gambar, sku, stok, , tanggal kadaluwarsa, dan dokumen id. Lalu function tersebut akan mengirimkan data tersebut ke dalam firebase database collection item dan mengupdate barang yang telah diinputkan dari aplikasi. Gambar 26 (b) merupakan sebuah Event yang akan merubah state didalamnya ketika pengguna melakukan pemanggilan event tersebut. Event tersebut akan mengubah state awal mennjadi ItemLoading(), lalu event akan mencoba mengupdate barang ke dalam database seperti pada gambar 28 (a), jika berhasil maka state akan berubah menjadi ItemEdited(), jika gagal maka state akan berubah menjadi ItemError(). Gambar 28 (c) merupakan sebuah function yang berfungsi sebagai trigger dari pengguna, function tersebut akan melakukan pengecekan terhadap inputan-inputan dari pengguna. Jika

inputan terisi, maka akan menjalankan sebuah Event yang telah dibuat pada gambar 28 (b), jika inputan tidak terisi maka aplikasi akan menampilkan sebuah pesan pada bagian bawah aplikasi.



(a)

(b)

(c)

logitordb	items	IB3rQD7M08sfX1EkE6iT
+ Start collection	+ Add document	+ Start collection
items >	IB3rQD7M08sfX1EkE6iT >	+ Add field
users	UPw9x0FRxQkfAz1RUrOF qgtDeKJtEXPncg33mDBV	<p>expired: "2024-04-01 00:00:00.000"</p> <p>id: "IB3rQD7M08sfX1EkE6iT"</p> <p>image: "https://firebasestorage.googleapis.com/v0/b/logitordb.app/alt=media&amp;token=e6baa375-34d4-499e-843d-30d4df6883a3"</p> <p>name: "Minyak Fortune 2L"</p> <p>sku: "MYK001"</p> <p>stock: 100</p>

(d)

Gambar 29 (a) Barang sebelum diedit, (b) Pemberitahuan barang setelah diedit, (c) Barang telah diedit, (d) Database barang yang telah diedit

Gambar 29 (a) merupakan barang yang belum dilakukan perubahan, setelah pengguna melakukan perubahan informasi pada barang, baik nama, gambar, sku, stok, maupun tanggal kadaluwarsa, maka akan muncul pemberitahuan

oleh aplikasi bahwa barang telah teredit, seperti pada gambar 29 (b). Gambar 29 (c) merupakan barang yang telah dilakukan perubahan sesuai dengan apa yang telah dilakukan sebelumnya.

Setelah menyelesaikan Sprint ID 13 dan Sprint ID 14, selanjutnya yaitu menyelesaikan sprint dengan Sprint ID 15 yaitu pengimplementasian RestAPI untuk penghapusan barang pada halaman detail item. Berikut hasil dari menyelesaikan Sprint ID 15:

```
deleteItem(String docId) {
  DocumentReference docRef =
    _firebaseFirestore.collection("items").doc(docId);
  docRef.delete();
}
```

(a)

```
on<OnDeleteItem>((event, emit) async {
  emit(ItemLoading());
  try {
    await itemRepository.deleteItem(
      event.docId);
    emit(ItemDeleted());
  } catch (e) {
    emit(ItemError(e.toString()));
  }
});
}
```

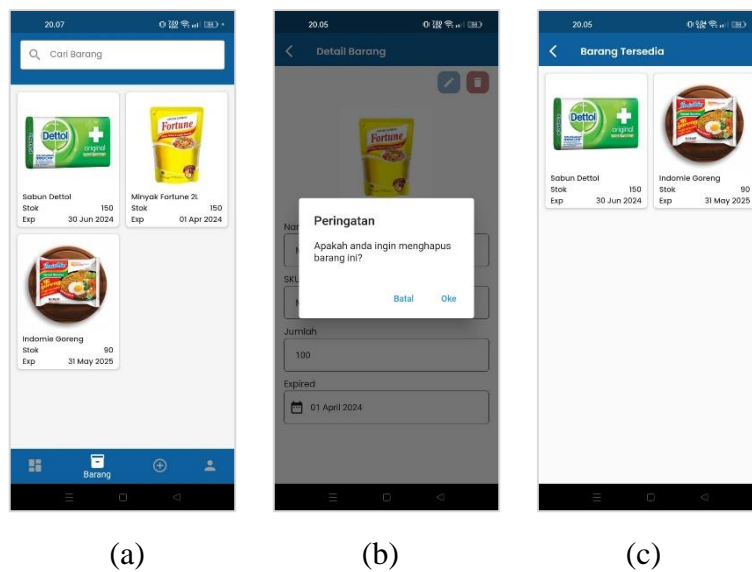
(b)

```
); // TextButton
Widget continueButton = TextButton(
  child: const Text("Ok"),
  onPressed: () {
    BlocProvider.of<ItemBloc>(context).add(
      OnDeleteItem(widget.listItem.id!));
  },
); // TextButton
```

(c)

Gambar 30 (a) Kode program menghapus barang, (b) Kode program Event Delete Item, (c) Kode program penggunaan Event Delete Item

Kode program pada gambar 30 (a) merupakan function yang berfungsi untuk menghapus barang yang ada di database, function tersebut memiliki satu buah parameter yaitu sebuah document id agar bisa berjalan. Kode program pada gambar 30 (b) merupakan sebuah Event Delete Item, yang akan berjalan ketika event dipanggil oleh pengguna, event tersebut akan mengubah state initial menjadi ItemLoading(), kemudian akan menjalankan function deleteItem seperti pada gambar 30 (a), setelah berhasil maka state akan berubah menjadi ItemDeleted(), jika gagal maka state akan berubah menjadi ItemError(). Kode program pada gambar 30 (c) merupakan sebuah Widget TextButton yang ketika ditekan oleh pengguna akan memicu sebuah event bernama OnDeleteItem yang merupakan event yang berfungsi untuk menghapus barang dari database.



logitordb	items	UPw9x0FRxQkfAzIRUrOF
+ Start collection	+ Add document	+ Start collection
items >	UPw9x0FRxQkfAzIRUrOF >	+ Add field
users	qgtDeKJtEXPNcg33mD8V	expired: "2024-06-30 00:00:00.000" id: "UPw9x0FRxQkfAzIRUrOF" image: "https://firebasestorage.googleapis.com/v0/b/logitordb.app/alt=media&token=d6b43f3c-755a-408e-bd06-dca35e44f673" name: "Sabun Dettol" sku: "DTL001" stock: 150

(d)

Gambar 31 (a) Daftar barang sebelum dihapus, (b) Penghapusan barang, (c) Barang setelah dihapus, (d) Database barang

Gambar 31 (a) merupakan daftar barang yang ada di aplikasi ketika barang belum dihapus, setelah pengguna memilih barang yang ingin dihapus, ketika pengguna akan menghapus barang maka akan muncul sebuah dialog seperti gambar 31 (b), ketika pengguna menekan tombol oke, maka barang akan dihapus dari database dan pengguna langsung akan diarahkan ke halaman menu item, seperti pada gambar 31 (c). Gambar 31 (d) merupakan database barang setelah pengguna melakukan penghapusan barang.

### 4.3. Pengujian Aplikasi

Pada tahap penelitian, dilakukan pengujian aplikasi menggunakan metode Black Box Testing untuk menguji apakah aplikasi sudah berjalan sesuai dengan kebutuhan yang dibutuhkan oleh pengguna atau belum. Pengujian dilakukan pada semua fitur yang telah dibuat sebelumnya, seperti fitur login, penambahan barang, pengeditan barang, penghapusan barang, dan logout. Teknik yang digunakan pada pengujian di penelitian ini adalah *decision tables*. Pada *decision tables* dibutuhkan *input* untuk menentukan *output* yang sesuai dengan cara menguji beberapa kombinasi *input*. Adapun fitur-fitur yang telah diuji dengan metode *blacbox* adalah sebagai berikut:

#### 4.3.1. Otentikasi

Pengujian fitur ini dilakukan untuk memastikan bahwa pengguna dapat melakukan manajemen barang ketika sudah masuk ke dalam aplikasi. Pengujian dilakukan dengan skenario yang berbeda. Hasil pengujian sebagai berikut:

Tabel 5 Pengujian Black Box Otentikasi

Skenario	Hasil yang diharapkan	Hasil
Melakukan login dengan memasukkan email dan password yang valid	Berhasil login dan aplikasi beralih menampilkan halaman utama	Sesuai
Melakukan login dengan memasukkan email dan password yang tidak valid	Menampilkan pesan peringatan bahwa email atau password yang dimasukkan salah	Sesuai

Berdasarkan tabel 5, pengujian fitur otentikasi dilakukan dengan dua skenario yang berbeda dan menghasilkan output yang berbeda pula. Pada pengujian fitur ini memiliki hasil yang sesuai, sehingga pengujian fitur ini dikatakan berhasil.

#### 4.3.2. Penambahan Barang

Pengujian fitur ini dilakukan untuk mengetahui apakah fitur penambahan barang bisa digunakan oleh pengguna atau tidak. Pengujian ini memiliki dua skenario yang berbeda. Berikut hasil dari pengujian fitur penambahan barang:

Tabel 6 Pengujian Black Box Penambahan Barang

Skenario	Hasil yang diharapkan	Hasil
Melakukan penambahan barang dan mengisi semua informasi barang yang dibutuhkan.	Barang berhasil ditambahkan ke dalam database, dan aplikasi mengarahkan pengguna ke menu utama	Sesuai
Melakukan melakukan penambahan barang dan tidak mengisi informasi barang yang dibutuhkan	Menampilkan pesan peringatan bahwa harus mengisi informasi barang terlebih dahulu.	Sesuai

Berdasarkan tabel 6, pengujian fitur penambahan barang dilakukan dengan dua skenario yang berbeda. Pada skenario pertama

**4.3.3. Melihat Daftar Barang**

**4.3.4. Penghapusan Barang**

**4.3.5. Pengeditan Barang**

**4.3.6. Logout**

Ini adalah sprint backlog

## **BAB V KESIMPULAN**

Qwe

Qwe

Qwe



## DAFTAR PUSTAKA

- [1] I. Firman Ashari, A. J. Aryani, and A. M. Ardhi, “DESIGN AND BUILD INVENTORY MANAGEMENT INFORMATION SYSTEM USING THE SCRUM METHOD,” *Jurnal Sistem Informasi*, vol. 9, no. 1, pp. 27–35, Mar. 2022.
- [2] A. E. Syaputra, “APLIKASI MANAJEMEN INVENTORY BERBASIS MOBILE,” *Journal of Information and Technology*, vol. 06, no. 01, pp. 23–32, Jun. 2018.
- [3] S. Hadji, M. Taufik, and S. Mulyono, “IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG),” *Prosiding KONFERENSI ILMIAH MAHASISWA UNISSULA (KIMU) 2*, pp. 32–43, Oct. 2019.
- [4] Muslim, R. Puspita Sari, and S. Rahmayuda, “IMPLEMENTASI FRAMEWORK FLUTTER PADA SISTEM INFORMASI PERPUSTAKAAN MASJID,” *Jurnal Komputer dan Aplikasi*, vol. 10, no. 01, pp. 46–59, 2022.
- [5] G. M. Hati, A. Suprayogi, and B. Sasmito, “APLIKASI PENANDA LOKASI PETA DIGITAL BERBASIS MOBILE GIS PADA SMARTPHONE ANDROID,” *Jurnal Geodesi Undip Oktober*, vol. 2, no. 4, pp. 26–40, 2013.
- [6] A. Juansyah, “PEMBANGUNAN APLIKASI CHILD TRACKER BERBASIS ASSISTED-GLOBAL POSITIONING SYSTEM (A-GPS) DENGAN PLATFORM ANDROID,” *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 1, no. 1, pp. 1–8, Aug. 2015.
- [7] M. Hamas and Z. Imaduddin, “PENGEMBANGAN SISTEM JUAL BELI BAHAN POKOK PETANI BERBASIS APLIKASI MOBILE,” *Jurnal Informatika Terpadu*, vol. 5, no. 2, pp. 49–55, 2019.
- [8] F. C. Ningrum, D. Suherman, S. Aryanti, H. A. Prasetya, and A. Saifudin, “Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions,” *Jurnal Informatika*

- Universitas Pamulang*, vol. 4, no. 4, pp. 125–130, 2019, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>
- [9] R. Jappi and D. F. Koan, “PENERAPAN INVENTORY MANAGEMENT DALAM MENINGKATKAN PROFITABILITAS DI TOKO X KUPANG,” *Jurnal Ilmiah Mahasiswa Universitas Surabaya*, vol. 3, no. 1, pp. 1–16, 2014.
- [10] A. A. G. S. Utama, E. V. Restiyantingrum, A. Manasika, E. A. Laila, and I. G. O. M. S. Pramesti, “AUTOMATIC MANAGEMENT INVENTORY SYSTEM (AMIS): Efektifitas Pengawasan dan Pengelolaan Persediaan Terintegrasi,” vol. 3, no. 2, pp. 144–169, 2019.
- [11] M. Yumna Majdina, B. Praptono, and M. Dellarosawati, “PERANCANGAN APLIKASI MANAJEMEN PERSEDIAAN GUDANG BERBASIS WEBSITE PADA UMKM BATIK SINUWUN DENGAN AGILE SCRUM DEVELOPMENT METHOD,” *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 5630–5637, Aug. 2020.
- [12] O. Y. Iliashenko and S. V. Shirokova, “Application of Database Technology to Improve the Efficiency of Inventory Management for Small Businesses,” *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, vol. 11, pp. 810–818, 2014, [Online]. Available: <http://www.isem-fem.spb.ru>