

BAB V

HASIL DAN PEMBAHASAN

5.1 Pengumpulan Data

Pada tahap pengumpulan data ini merupakan tahap awal untuk mendapatkan informasi mengenai permasalahan pengelolaan data inventori gudang di Toko Kosmetik XYZ Kabupaten Ciamis. Data-data yang didapatkan akan menjadi dasar untuk mengembangkan sistem yang di butuhkan oleh pengguna agar membantu proses pengelolaan data barang. Pengumpulan data ini dilakukan dengan cara observasi ke Toko Kosmetik XYZ, melakukan wawancara pada pengguna (Admin dan Karyawan) dan dokumentasi. Hasil dari pengumpulan data tersebut akan menjadi acuan untuk menyusun *Product Backlog* pada metode *Scrum*.

5.1.1 Hasil Observasi

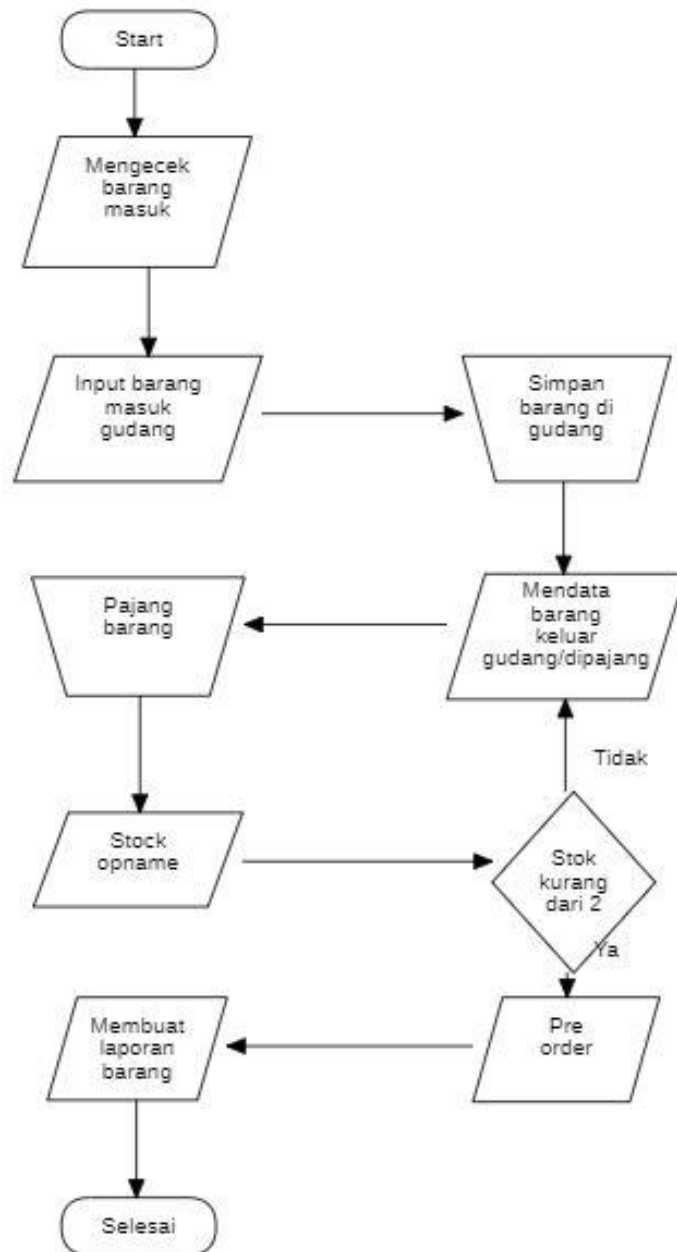
Dari hasil observasi di ketahui sistem kerjanya masih menggunakan kertas dan excel yang membuat pengerjaannya tidak efisien dan waktu kerjanya yang jadi lama. Selain itu, stok barang di gudang juga tidak terpantau karena tidak adanya data stok barang yang *real time* membuat persediaan barang jadi kosong dan kehilangan penjualan.

5.1.2 Sistem yang sedang berjalan

Berdasarkan data hasil observasi, diketahui bahwa pengelolaan inventori gudang di Toko Kosmetik XYZ masih menggunakan sistem manual dan menimbulkan beberapa permasalahan seperti yang telah dijelaskan di hasil observasi. Sistem manual tersebut sudah digunakan selama ini, tapi datanya masih tersimpan pada dokumen fisik yaitu di kertas dan sering kali kertas tersebut hilang atau rusak.

Berikut ini merupakan sistem manual yang digambarkan dengan flowchart :

a. *Flowchart* sistem yang sedang berjalan



Gambar 5. 1. Flowchart sistem yang sedang berjalan

Penjelasan :

Pada tahap awal barang yang baru datang di cek oleh admin dan karyawan, setelah barang terkonfirmasi sesuai dengan pesanan maka barang di input masuk gudang

dengan menggunakan kertas. Kemudian barang di masukkan ke Gudang oleh karyawan, ketika barang pajangan di area jualan sudah menipis maka barang gudang akan di keluarkan atau di pajang.

Sebelum di pajang barang akan di data dulu dengan kertas yang telah di sediakan admin lalu di pajang di area jualan, dikarenakan barang Gudang sudah keluar maka dilakukan *stock opname* untuk mengecek fisik barang yang masih ada. Jika stok barang kurang dari 2 pcs maka dilakukan pemesanan *pre-order*. Tahap akhirnya membuat laporan barang yang di *pre-order* dan barang.

5.1.3 Hasil wawancara

Dari hasil wawancara, sistem manual menggunakan kertas memang membuat efisiensi dan waktu kerja jadi efektif. Kepala toko merasa laporan barang bulanan dari admin selalu terhambat dan mendapat teguran dari atasan karena laporan yang tidak selesai sesuai tanggal *deadline*. Admin dan karyawan kerepotan jika harus mengecek barang di gudang jika di area jual (*display*) kosong maka harus di isi lagi dari stok barang di gudang.

Admin terhambat untuk melakukan *pre-order* karena tidak tau barang yang tinggal sedikit apa saja dan harus mengeceknya dari kertas atau mengeceknya langsung ke gudang. Karayawan kesulitan saat datang barang karena harus mengecek barang dari faktur kertas mencari *barcode* barangnya dan proses tersebut memakan waktu yang lama. Selain itu, saat proses *stock opname* harus menggunakan kertas atau excel yang agak ribet harus mencari data *barcode* barangnya dulu. Oleh karena itu, admin dan karyawan mengharapkan adanya sistem yang membantu proses pengelolaan barang inventori gudang agar bisa lebih cepat, akurat dan efesien.

5.1.4 Hasil dokumentasi

Setelah observasi dan wawancara, penulis juga melakukan kegiatan dokumentasi di Toko Kosmetik XYZ. Hal tersebut dilakukan untuk mendapatkan bukti fisik yang berkaitan dengan proses pengelolaan inventori gudang yang sedang berjalan saat ini. Dokumen ini dikumpulkan digunakan sebagai data informasi serta patokan

untuk merancang sistem informasi inventori gudang berbasis web yang dibutuhkan oleh pengguna.

5.2 Produk Backlog

Pada penelitian ini, penulis memakai metode *scrum* karena dinilai pengembangan sistemnya bisa fleksibel dan tidak kaku seperti metode lainnya. Tahap awal yang dilakukan yaitu membuat daftar kebutuhan sistem atau *Product Backlog* dari hasil dokumentasi, observasi dan wawancara yang telah dilakukan dengan kepala toko, admin dan karyawan Toko Kosmetik XYZ. Informasi dari tiga sumber tersebut dianalisis untuk mendapatkan sistem atau fitur apa saja yang di butuhkan oleh pengguna agar membantu memudahkan pengelolaan inventori gudang.

5.2.1 Identifikasi kebutuhan sistem

Identifikasi kebutuhan sistem ini dilakukan agar kebutuhan fungsional yang di butuhkan oleh pengguna tersedia. Dari hasil analisis dokumentasi, wawancara dan observasi tersebut penulis akan merancang beberapa kebutuhan fitur akan yang membuat sistem kerjanya jadi lebih efisien. Beberapa kebutuhan fitur (*Product Backlog*) tersebut ada pada Tabel 5.1 :

Tabel 5. 1 *Product Backlog*

No	Nama Menu	User story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
1	Login	Sebagai pengguna (Admin), saya ingin login dengan menggunakan nama, password dan role pengguna. Pengguna (karyawan)	Semua Pengguna	Tinggi	1 Hari

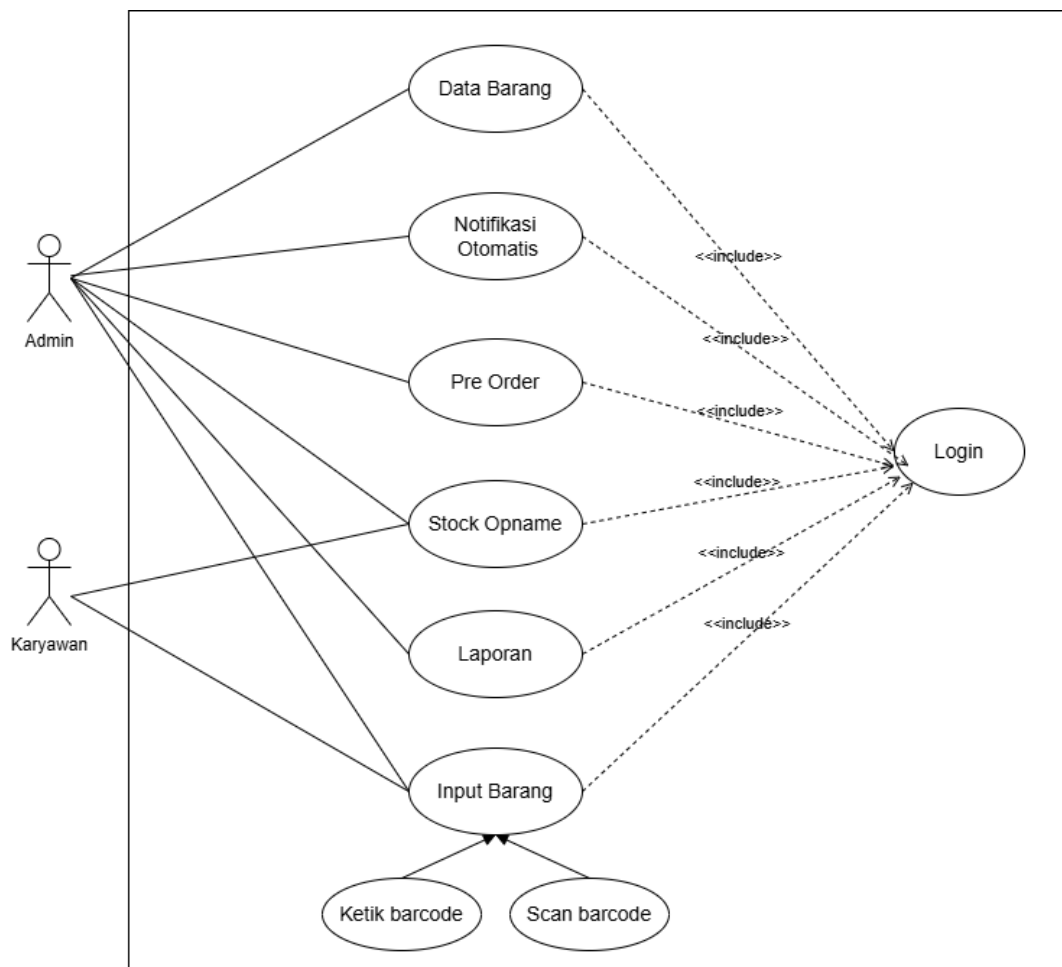
No	Nama Menu	User story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
		menggunakan nama dan role pengguna			
2	Pengguna	Sebagai Admin, saya ingin mengelola data barang Gudang, membuat laporan, membuat <i>pre-order</i> , <i>stock opname</i> dan input barang masuk atau keluar.	Admin	Sedang	1 Hari
3	Pengguna	Sebagai Karyawan, saya ingin melakukan <i>stock opname</i> dan input barang masuk atau keluar.	Karyawan	Sedang	1 Hari
4	<i>Dashboard</i>	Sebagai pengguna saya ingin memakai fitur dan melihat ringkasan data barang atau aksi cepat.	Semua Pengguna	Sedang	2 Hari
5	Data Barang	Sebagai pengguna, saya ingin melihat data barang gudang yang masih tersedia.	Semua Pengguna	Sedang	2 Hari
6	Pengelolaan barang masuk	Sebagai pengguna, saya ingin input barang yang masuk ke gudang dengan mudah	Semua Pengguna	Sedang	2 Hari

No	Nama Menu	User story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
7	Pengelolaan barang keluar	Sebagai pengguna, saya ingin input barang yang keluar dari gudang dengan mudah	Semua Pengguna	Sedang	2 Hari
8	Perhitungan stok otomatis	Sebagai pengguna, saya ingin stok otomatis, jika barang masuk atau keluar gudang.	Semua Pengguna	Sedang	1 Hari
9	<i>Stock opname</i>	Sebagai pengguna, saya ingin melakukan pengecekan fisik barang gudang dengan mudah dan efisien.	Semua Pengguna	Tinggi	3 Hari
10	Laporan Barang	Sebagai pengguna, saya ingin melakukan laporan (laporan inventori, laporan <i>pre-order</i> , laporan stok sedikit).	Admin	Tinggi	2 Hari
11	<i>Barcode Scanner</i>	Sebagai pengguna, saya ingin <i>scan barcode</i> yang responsive dan cepat.	Semua Pengguna	Tinggi	4 Hari
12	Notifikasi stok sedikit	Sebagai pengguna, saya ingin mengetahui barang yang tinggal sedikit secara otomatis.	Admin	Sedang	1 Hari

No	Nama Menu	User story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
13	<i>Pre-order</i>	Sebagai pengguna, saya ingin melakukan pemesanan barang untuk persediaan barang gudang.	Admin	Tinggi	2 Hari

5.2.2 Use case diagram Sistem usulan

Daftar kebutuhan fitur tersebut kemudian menjadi acuan dalam proses pengembangan sistem agar lebih terstruktur dan efisien. Rancangan sistem usulannya menggunakan *Use Case Diagram*. Penulis memakai Diagram ini karena menggambarkan hubungan antara pengguna (aktor) dengan fitur-fitur yang akan dirancang di Sistem Informasi Inventori Gudang Berbasis Web di Toko Kosmetik XYZ di Kabupaten Ciamis. Dengan adanya *Use Case Diagram*, akses setiap aktor dibedakan karena hak aksesnya berbeda seperti aktor admin yang dapat mengakses pengelolaan data barang. Sedangkan untuk aktor karyawan tidak dapat akses mengelola data barang, hanya input barang masuk dan barang keluar. Berikut ini *use case diagram* yang menggambarkan kebutuhan sistemnya ada pada Gambar 5.2:



Gambar 5. 2. Use case sistem usulan

Sistem ini memiliki dua aktor yang dapat mengakses dan mengelolanya yaitu admin dan karyawan. Pengguna wajib melakukan login untuk menggunakan sistem web ini, Kelola data barang dilakukan oleh dua aktor tersebut yaitu karyawan hanya dapat melihat produk, melakukan *stock opname* dan input barang masuk atau barang keluar. Sementara itu, untuk admin bisa mengelola data produk, melakukan input barang masuk atau barang keluar, melakukan *stock opname*, melakukan *pre-order* dan melakukan pelaporan.

Notifikasi otomatis hanya akan muncul ke *dashboard* admin karena admin bertanggung jawab atas pengelolaan data barang. Saat notifikasi stok sedikit muncul yaitu jika stok kurang dari dua pcs, maka admin harus melakukan *pre-order*

agar barang yang kosong dapat terpenuhi dan penjualan tidak *lost sale*. Namun sebelum admin melakukan *pre-order*, admin atau karyawan harus melakukan *stock opname* dulu mengecek fisik barang yang ada di gudang. Data *stock opname* yang harus di isi terdiri dari *barcode* barang yang bisa di *scan barcode* atau ketik manual, fisik barang, order barangnya mau berapa, penjualan sebelumnya, harga barang dan total beli yang akan terisi otomatis jika semua datanya sudah terisi. Jika sudah sesuai maka admin mengklik “tambahkan ke *opname*” maka data *stock opname* sudah bisa di masukkan ke *pre-order*.

Setelah melakukan *stock opname*, admin akan masuk ke halaman *pre-order* dan mengisi data-datanya seperti nama supplier, tanggal order, tanggal diharapkan, sumber data *stock opname* dan catatan. Untuk data *stock opname* itu di isi dari data *stock opname* yang telah dilakukan, admin mengklik keranjang pada data *stock opname* mana saja yang akan *pre-order* pada list *stock opname*.

Kemudian saat karyawan atau admin akan memasukan barang atau mengeluarkan barang dari gudang, maka masuk ke halaman input barang yang bisa melakukan *scan barcode* barang dengan menggunakan kamera *handphone* atau dengan mengetik *barcodenya*. Jika *barcode* sudah terinput maka akan tampil data barangnya, opsi barang masuk atau keluar dan jumlah barangnya. Jika sudah sesuai maka akan data barang akan terupdate otomatis.

Tahap terakhirnya yaitu admin akan melakukan pelaporan berupa laporan inventory, barang stok rendah, laporan *pre-order*, laporan perkategori. Admin mengisi data pelaporan yaitu jenis laporan, tanggal mulai dan tanggal akhir. Setelah itu *generate* laporan, maka data laporan akan ditampilkan dan bisa untuk di cetak pdf .

5.2.2.1 Deskripsi aktor

Untuk memperjelas mengenai deskripsi masing-masing *use case diagram* sistem usulannya, di jelaskan pada Tabel 5.2 :

Tabel 5. 2 Deskripsi aktor

No.	Aktor	Deskripsi
1	Admin	Admin adalah aktor (pengguna) yang dapat mengakses semua fitur dalam sistem seperti pengelolaan data barang, input barang masuk atau keluar, <i>stock opname</i> , <i>pre-order</i> dan laporan.
2	Karyawan	Karyawan adalah aktor yang menggunakan sistem untuk input barang masuk atau keluar, melihat data barang tanpa bisa mengelola datanya dan <i>stock opname</i>

5.2.2.2 Deskripsi use case

Untuk memperjelas mengenai deskripsi masing-masing *use case diagram* sistem usulannya, di jelaskan pada Tabel 5.3 :

Tabel 5. 3. Definisi use case diagram

No.	Use case	Deskripsi	Aktor
1	Login	Proses pertama dan wajib untuk bisa mengakses pengelolaan data perusahaan dengan menggunakan username dan password	Admin dan karyawan
2	Data barang	Admin dapat mengelola data barang Gudang berupa menambah, mengedit dan menghapus data barang.	Admin
3	<i>Pre-order</i>	Use case ini digunakan untuk memesan barang yang stoknya tinggal sedikit menghindari stok kosong.	Admin
4	<i>Stock opname</i>	Sebelum <i>pre-order</i> , admin dan karyawan mengecek stok fisik barang yang ada di Gudang apakah sesuai dengan data sekaligus mengecek barang yang <i>slow moving</i> atau barang yang sudah lama tertimbun.	Admin dan karyawan

No.	Use case	Deskripsi	Aktor
5	Laporan	Proses ini penting untuk mengetahui stok barang yang cepat habis dan di serahkan ke kepala toko.	Admin
6	Input Barang	Admin dan karyawan menginput data barang yang masuk maupun keluar	Admin dan karyawan
7	Scan barcode	Admin dan karyawan dapat <i>scan barcode</i> untuk menginput data barang masuk ataupun keluar	Admin dan karyawan
8	Ketik barcode	Admin dan karyawan dapat mengetik juga <i>barcode</i> barang apabila barang tidak dapat di <i>scan</i> .	Admin dan karyawan
9	Notifikasi otomatis	Fitur ini memudahkan admin untuk mengetahui stok barang yang tinggal sedikit ataupun habis.	Admin

5.3 Sprint 1

Sprint satu adalah tahap pertama yang terdiri dari fitur dasar sebagai fondasi karena fitur dasar sistem sangat penting untuk operasional sistem selanjutnya. Pada *sprint* ini seluruh pelaksanaannya bersumber dari *product backlog* yang telah dibuat berdasarkan hasil analisis kebutuhan sistem di Toko Kosmetik XYZ Kabupaten Ciamis.

Fokus utama dari *sprint* ini yaitu perancangan fitur fondasi seperti pembuatan halaman login, pembuatan *dashboard*, pembuatan data barang, pengguna (Admin dan Karyawan). Dengan adanya perancangan tersebut, pengguna dapat login dengan rolenya masing-masing, pengguna dapat masuk ke *dashboard* sesuai dengan rolenya dan admin dapat mengelola data barang (sedangkan karyawan hanya bisa melihat data barang saja).

5.3.1 *Sprint planning*

Sprint planning adalah tahapan perencanaan fitur yang akan di rancang untuk menentukan batasan pekerjaan yang akan dikerjakan pada setiap *sprint*. Pada *sprint* satu, berfokus pada fitur-fitur fondasi yang sebagai acuan untuk *sprint* berikutnya.

Tabel 5. 4 *Sprint planning (sprint 1)*

No	Target Sprint	Product Backlog	Estimasi Waktu
1	Pembuatan dan mengembangkan fitur login	Halaman login	2 Hari
2	Mengembangkan role pengguna (Admin memakai kata sandi dan karyawan tidak pakai kata sandi)	Role pengguna	2 Hari
3	Pembuatan <i>Dashboard</i> menampilkan menu-menu yang tersedia setiap rolenya	<i>Dashboard</i>	3 Hari
4	Pembuatan dan mengembangkan Fitur data barang yang menampilkan rincian barang gudang	Data barang	3 Hari

Berdasarkan Tabel 5.4, menghasilkan fokus utama pengembangannya yaitu fitur login dengan role masing-masing pengguna. Kemudian menampilkan *Dashboard* dengan menu atau fitur yang tersedia di rolenya.

5.3.2 *Sprint development*

Sprint development pada *sprint* satu harus benar-benar selesai karena sebagai dasar fondasi sistem dan agar bisa melanjutkan ke fitur selanjutnya. Jika salah sedikit saja akan menyebabkan pengguna tidak dapat mengakses fitur selanjutnya.

5.3.2.1 Login

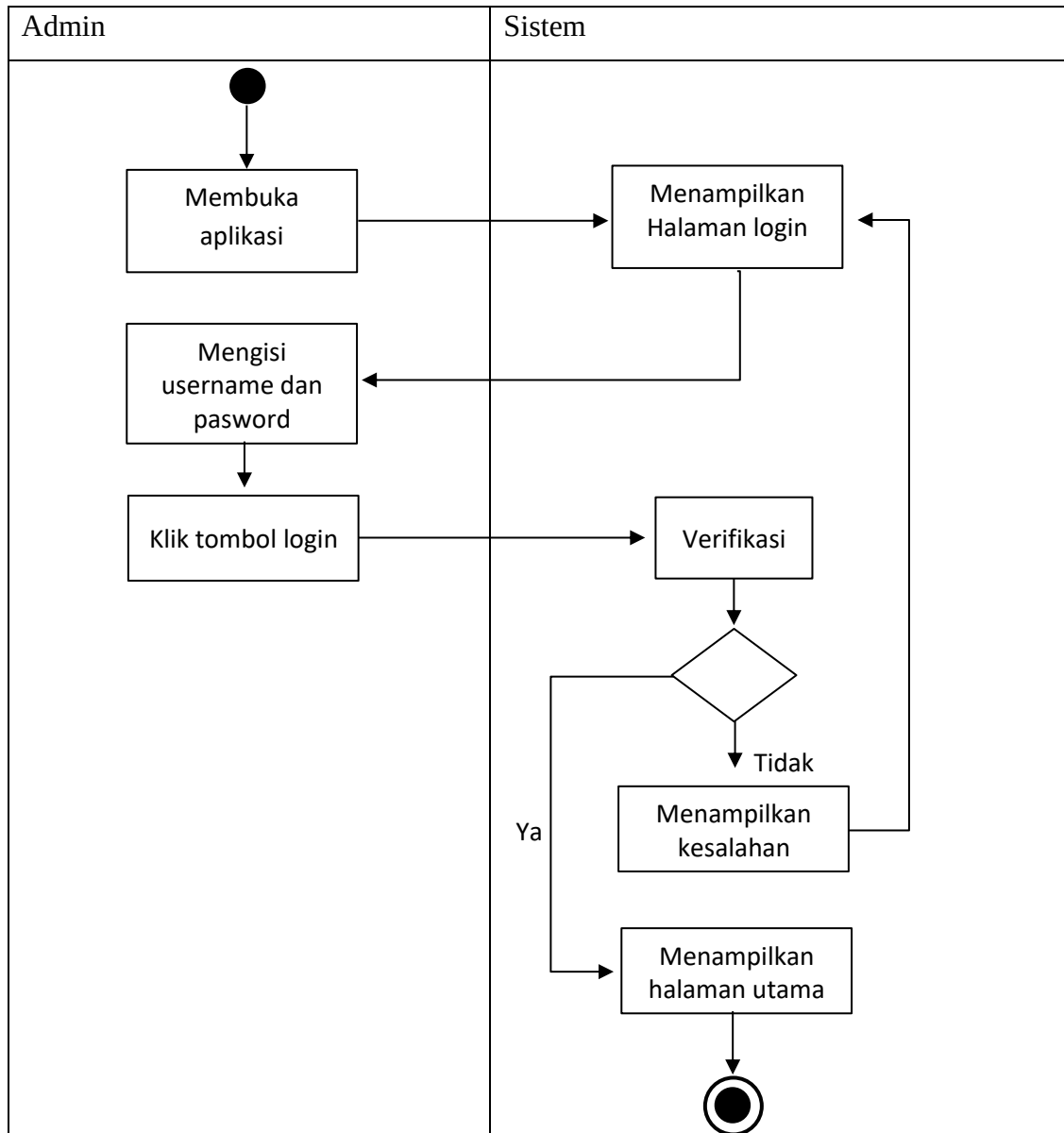
Fitur *login* merupakan pintu masuk untuk pengguna kedalam sistem yang menentukan akses ke fitur selanjutnya. Dengan adanya fitur ini, pengguna dapat

melanjutkan ke akses selanjutnya dengan input nama pengguna, password dan rolenya masing-masing.

Rancangan fitur *login* menggunakan UML yaitu *activity diagram* dan *class diagram*, sebagai acuan untuk implementasi fitur yang akan dibuat dengan mengikuti alur prosesnya. Dengan adanya diagram tersebut, membantu sebagai gambaran perancangan yang akan di buat agar implementasinya sesuai dengan kebutuhan pengguna.

1. Activity diagram login

Tabel 5. 5. Activity diagram login

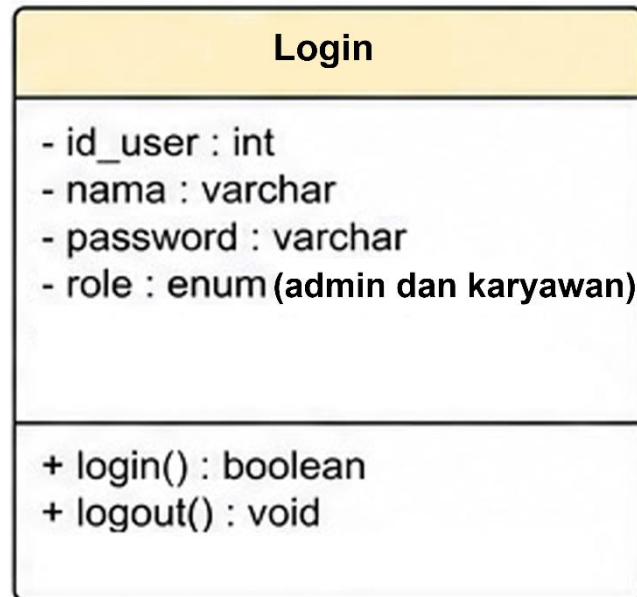


Penjelasan :

Proses yang pertama admin membuka web untuk melakukan login, sistem merespon dengan menampilkan halaman login. Admin mengisi data username dan password untuk login, admin kemudian mengklik tombol login. Kemudian sistem akan mengvalidasi username dan password, Jika salah maka akan ada pesan “Login

gagal” dan harus memasukan username dan password yang benar. Kemudian sistem akan menampilkan halaman utama untuk mengelola data barang gudang.

2. *Class Diagram login*



Gambar 5. 3 *Class Diagram login*

Pada Gambar 5.3 *Class Diagram login* merupakan gambaran struktur kelas yang berfungsi sebagai proses validasi data pengguna. Diagram ini juga menunjukkan hubungan antar atribut dan operasi yang berfungsi saat proses *login*.

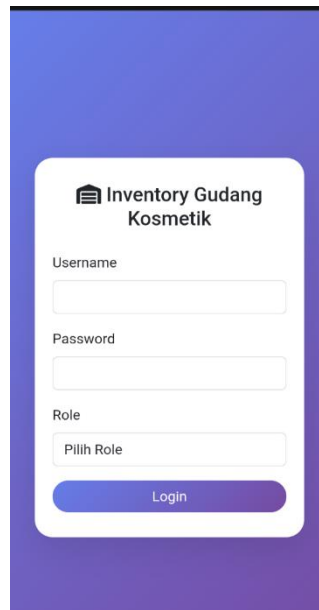
3. Implementasi *login*

```
2    <html lang="id">
49   <body>
80   <script>
85       function updatePasswordVisibility() {
86           if (roleSelect.value === 'karyawan') {
87               passwordGroup.style.display = 'none';
88               passwordInput.removeAttribute('required');
89           } else {
90               passwordGroup.style.display = 'block';
91               passwordInput.setAttribute('required', '');
92           }
93       }
94
95       roleSelect.addEventListener('change', updatePasswordVisibility);
96       updatePasswordVisibility();
97
98       document.getElementById('loginForm').addEventListener('submit', function(e) {
99           e.preventDefault();
100          const username = document.getElementById('username').value;
101          const password = document.getElementById('password').value;
102          const role = roleSelect.value;
103
104          if (role === 'admin' && username === 'admin' && password === 'admin') {
105              localStorage.setItem('userRole', 'admin');
106              window.location.href = 'dashboard-admin.html';
107          } else if (role === 'karyawan') {
108              localStorage.setItem('userRole', 'karyawan');
109              window.location.href = 'dashboard-karyawan.html';
110          } else {
111              alert('Role atau kredensial tidak valid!');
112          }
113      });
114   </script>
115 </body>
116 </html>
```

Gambar 5. 4 Implementasi *login*

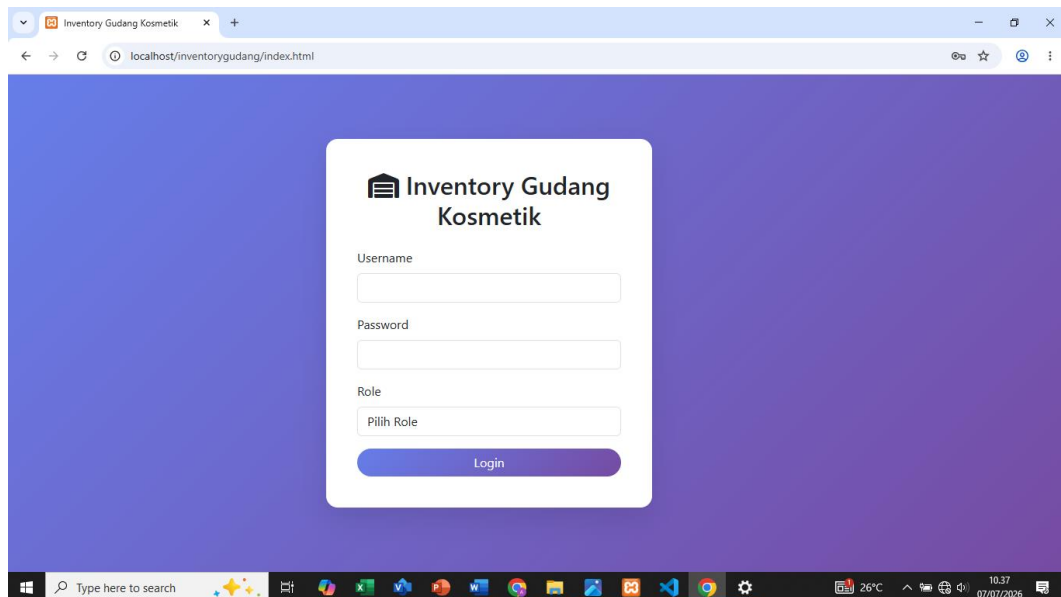
Berdasarkan Gambar 5.4 implementasi *login* menunjukkan potongan kodingan untuk logika saat *login* admin dan karyawan. Nama bisa di input dengan nama masing-masing pengguna untuk role karyawan. Sedangkan untuk role admin nama tidak bisa di input dengan nama pengguna tapi hanya bisa di input dengan “admin” jika salah maka sistem akan melakukan validasi dan menampilkan notifikasi “role atau kredensial tidak valid”.

4. Tampilan *login*



The image shows a vertical smartphone screen with a purple gradient background. In the center is a white rounded rectangle containing the login form. At the top of the form is a warehouse icon followed by the text 'Inventory Gudang Kosmetik'. Below this are three input fields: 'Username', 'Password', and 'Role'. The 'Role' field has a dropdown menu with 'Pilih Role' selected. At the bottom of the form is a purple rounded button with the text 'Login'.

Gambar 5. 5 Tampilan *login* di *handphone*



The image shows a desktop browser window with a purple gradient background. The login form is centered and identical to the one in the previous image. The browser's address bar shows 'localhost/inventorygudang/index.html'. The Windows taskbar is visible at the bottom with various application icons and a system clock showing 10:37 on 07/07/2026.

Gambar 5. 6 Tampilan *login*

Kedua Gambar diatas menampilkan halaman *login*, terdapat *field* yaitu *username* dan *password* yang harus di isi oleh pengguna agar bisa mengakses web. Setelah di isi klik tombol login agar di verifikasi oleh sistem, jika benar maka akan masuk ke halaman *dashboard* pengguna masing-masing.

5.3.2.2 Dashboard

Dashboard merupakan tampilan antarmuka dari masing-masing pengguna dengan akses yang berbeda sesuai kebutuhan. Pada penelitian ini terdapat dua aktor yaitu admin dan karyawan yang memiliki peran berbeda. *Dashboard* admin berperan untuk mengelola data inventori, sedangkan *dashboard* berperan ke operasional masuk dan keluar barang gudang.

1. Implementasi *dashboard* admin

```
2 id">
52
53 class="container-fluid">
54 class="row">
55
56 <div class="col-12 col-md-2 sidebar p-3">
57   <h4 class="mb-4"><i class="fas fa-warehouse"></i> Admin</h4>
58   <nav class="nav flex-column">
59     <a class="nav-link active" href="#dashboard"><i class="fas fa-tachometer-alt"></i> Dashboard<
60     <a class="nav-link" href="crud.html"><i class="fas fa-boxes"></i> Data Barang</a>
61     <a class="nav-link" href="scanner.html"><i class="fas fa-qrcode"></i> Input Barang</a>
62     <a class="nav-link" href="stock-opname.html"><i class="fas fa-clipboard-list"></i> Stock Opna
63     <a class="nav-link" href="pre-order.html"><i class="fas fa-shopping-cart"></i> Pre Order</a>
64     <a class="nav-link" href="reports.html"><i class="fas fa-chart-bar"></i> Laporan</a>
65     <a class="nav-link" href="#" onclick="logout()"><i class="fas fa-sign-out-alt"></i> Logout</a>
66   </nav>
67 </div>
68
69 <!-- Main Content -->
70 <div class="col-12 col-md-10 p-4">
71   <h2 class="mb-4">Dashboard Admin</h2>
72
73   <!-- Stats Cards -->
74   <div class="row mb-4">
75     <div class="col-md-3">
76       <div class="card stat-card">
77         <div class="card-body text-center">
78           <i class="fas fa-box fa-2x mb-2"></i>
79           <h4 id="totalItems">0</h4>
80           <p>Total Barang</p>
81         </div>
82       </div>
```

Gambar 5. 7 Implementasi *Dashboard* admin

Gambar 5.7 *Dashboard* admin menunjukkan potongan *code* HTML untuk membuat antarmuka halaman *dashboard* admin dengan menggunakan framework bootstrap pada desainnya yang memiliki menu pengelolaan data barang, input barang, *stock opname*, *pre-order* dan laporan barang. Selain fitur tersebut, *code* ini juga menunjukkan tampilan keterangan jumlah item barang, stok barang yang sedikit, notifikasi dan jumlah harga barangnya.

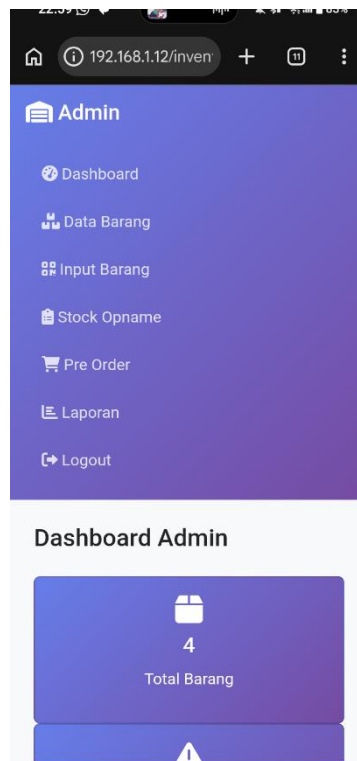
2. Implementasi *dashboard* karyawan

```
2   id">
48
49   ss="container-fluid">
50   class="row">
52   <div class="col-12 col-md-2 sidebar p-3">
53     <h4 class="mb-4"><i class="fas fa-user"></i> Karyawan</h4>
54     <nav class="nav flex-column">
55       <a class="nav-link active" href="#dashboard"><i class="fas fa-tachometer-alt"></i> Dashboard</a>
56       <a class="nav-link" href="scanner.html"><i class="fas fa-qrcode"></i> Input Barang</a>
57       <a class="nav-link" href="data-barang-karyawan.html"><i class="fas fa-boxes"></i> Data Barang
58       <a class="nav-link" href="stock-opname.html"><i class="fas fa-clipboard-list"></i> Stock Opna
59       <a class="nav-link" href="#" onclick="logout()"><i class="fas fa-sign-out-alt"></i> Logout</a>
60     </nav>
61   </div>
62
63   <!-- Main Content -->
64   <div class="col-12 col-md-10 p-4">
65     <h2 class="mb-4">Dashboard Karyawan</h2>
66
67     <!-- Quick Actions -->
68     <div class="row">
69       <div class="col-md-6">
70         <div class="card card-custom">
71           <div class="card-header">
72             <h5><i class="fas fa-plus-circle"></i> Aksi Cepat</h5>
73           </div>
74           <div class="card-body">
75             <div class="row row-cols-1 row-cols-sm-2 g-2 mb-3">
76               <div class="col"><a href="scanner.html?action=in" class="btn btn-success w-100">Input Barang</a>
77               <div class="col"><a href="scanner.html?action=out" class="btn btn-danger w-100">Output Barang</a>
78             </div>
79             <div class="row row-cols-1 g-2">
```

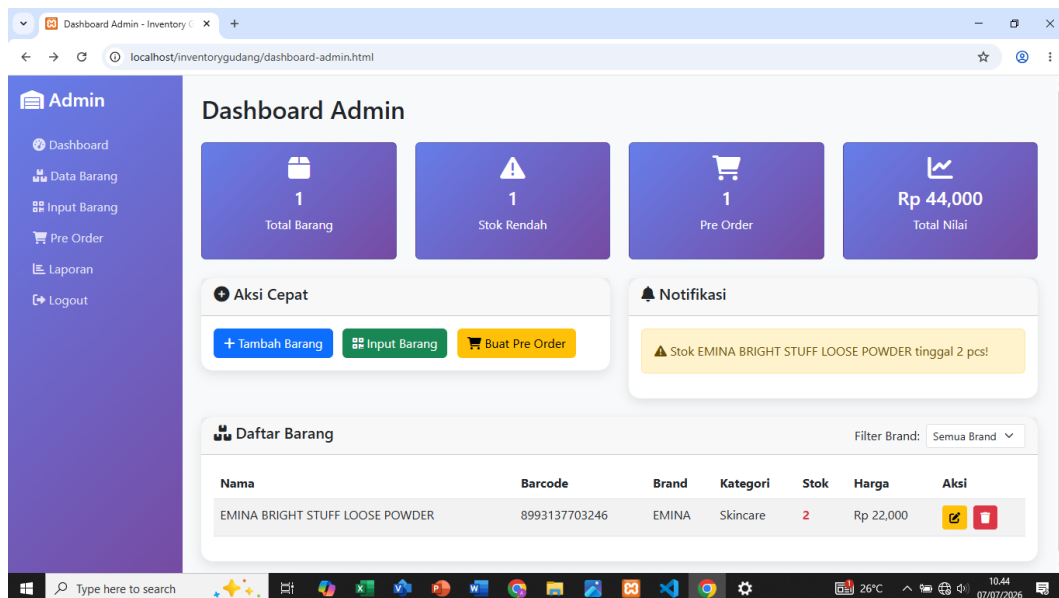
Gambar 5. 8 Implementasi *dashboard* karyawan

Gambar 5.8 *dashboard* karyawan menunjukkan potongan *code* HTML untuk membuat antarmuka halaman *dashboard* karyawan dengan menggunakan framework bootstrap pada desainnya yang memiliki lebih sedikit menu yaitu pengelolaan data barang, input barang dan *stock opname*. Selain fitur tersebut, ada juga pilihan aksi cepat yang bisa langsung akses *scan barcode* untuk input barang.

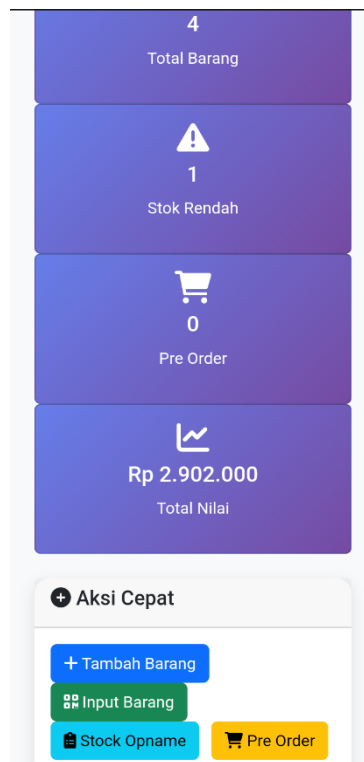
3. Tampilan *Dashboard* admin



Gambar 5. 9 Tampilan *dashboard* admin (*handphobe*)



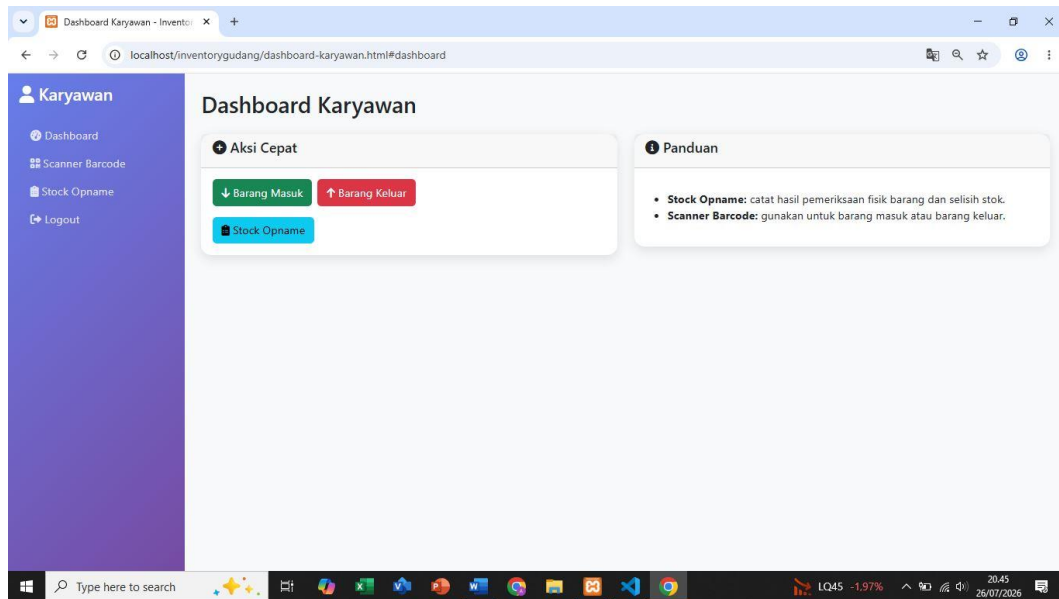
Gambar 5. 10 Tampilan *dashboard* admin



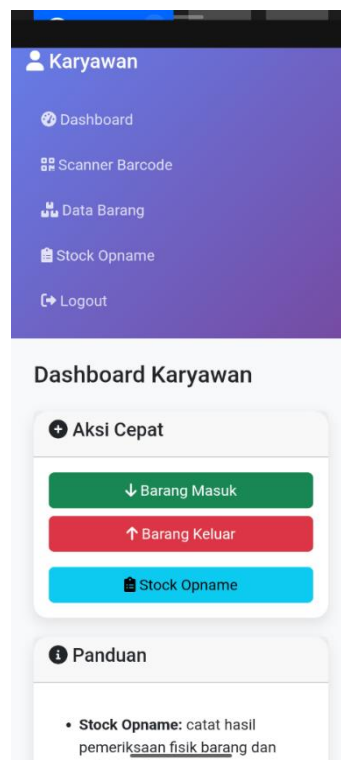
Gambar 5. 11 Tampilan dashboard admin (handphone 2)

Setelah admin berhasil login maka menampilkan *dashboard* admin, yang terdapat beberapa fitur yaitu data barang, input *barcode*, *stock opname*, *pre-order*, laporan dan logout. Ada pula aksi cepat, display data barang, notifikasi dan tabel data barang.

4. Tampilan *dashboard* karyawan



Gambar 5. 13. *Dashboard* karyawan



Gambar 5. 12 Tampilan *dashboard* karyawan (*Handphone*)

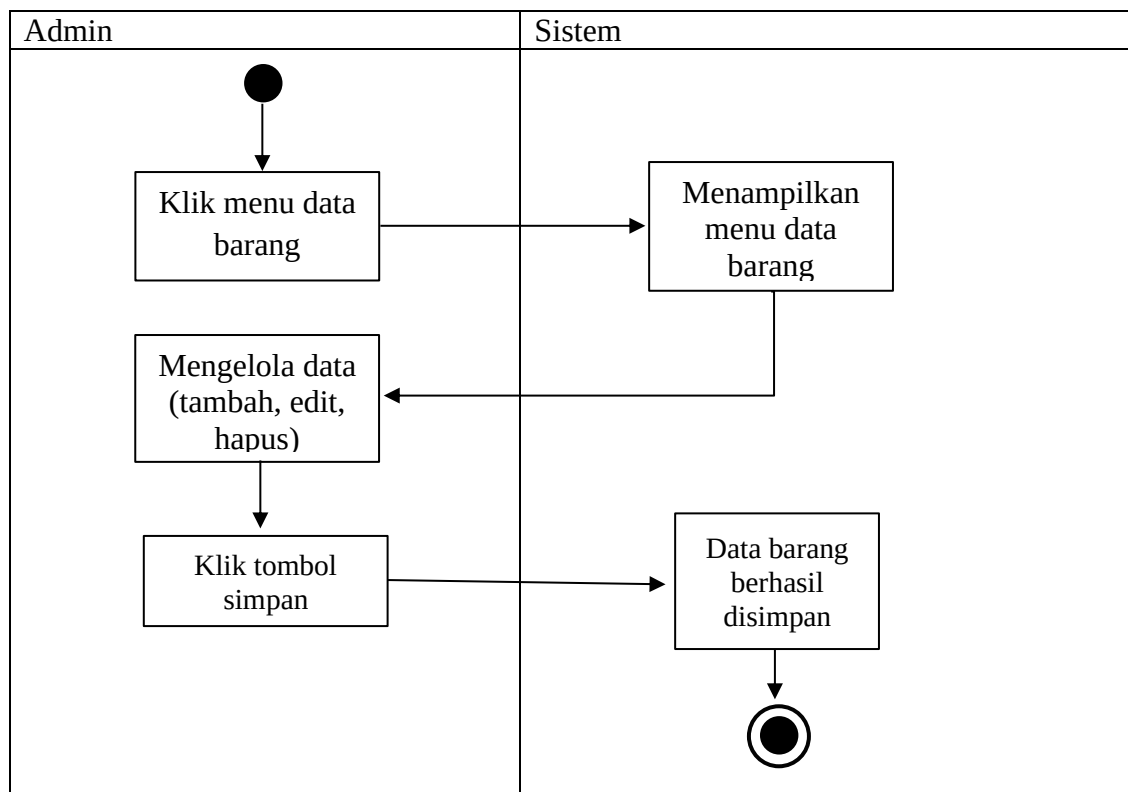
Setelah karyawan berhasil login maka menampilkan *dashboard* karyawan, terdapat beberapa fitur yaitu *scanner barcode*, *stock opname* dan *logout*. Ada pula aksi cepat untuk barang masuk dan keluar, terdapat panduan juga agar pengguna baru atau karyawan baru bisa cepat paham dan bisa menggunakannya dengan mudah.

5.3.2.3 Data barang

Data barang merupakan halaman yang ada data barang inventori gudang terdiri dari nama barang, barcode, stok, brand, kategori dan harga. Dalam data barang memiliki fitur pengelolaan data yaitu *create*, *read*, *update* dan *delete* (CRUD).

1. Activity diagram data barang

Tabel 5. 6 Activity diagram data barang

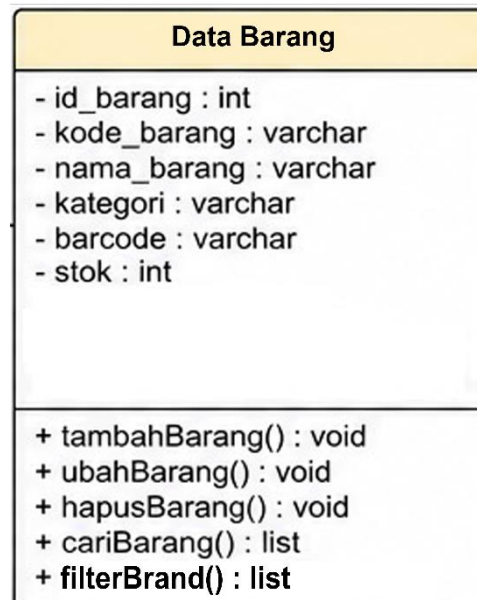


Penjelasan :

Admin mengklik data barang, sistem merespon dengan menampilkan menu data barang. Disini admin bisa mengelola data seperti menambahkan data barang, mengedit data barang dan menghapus data barang. Kemudian klin tombol simpan

jika sudah selesai mengelola datanya, Sistem merespon dengan berhasil menyimpan data barang.

2. *Class Diagram* data barang



Gambar 5. 14 *Class diagram* data barang

Gambar 5. 14 *Class Diagram* data barang menunjukkan informasi data barang dan operasi yang bisa dilakukan. Ada beberapa atribut didalamnya yaitu `id_barang` untuk identifikasi barang berupa angka, `kode_barang` untuk kode barang berbentuk teks, `nama_barang` untuk nama barangnya, `kategori` untuk kategori barang (lipstik, skincare, dan lainnya), `barcode` untuk *barcode* setiap barang, `stok` untuk jumlah ketersediaan stok barang. Selain atribut ada juga *methods* atau operasinya yaitu `tambahbarang()` untuk menambahkan data barang baru ke dalam sistem, `ubahbarang()` untuk mengubah data barang yang sudah ada, `hapusbarang()` untuk menghapus data barang yang sudah ada, `caribarang()` untuk mencari data barang serta menampilkannya dalam bentuk daftar, `filterbarang()` untuk memfilter data barang serta menampilkannya dalam bentuk daftar.

3. Implementasi data barang admin

```
2   <html lang="id">
34  <body>
35    <div class="container-fluid p-4">
36
37      <div class="d-flex justify-content-between align-items-center mb-4">
38        <h2><i class="fas fa-boxes"></i> CRUD Barang</h2>
39        <div>
40          <button class="btn btn-custom text-white" onclick="openAddModal()">
41            <i class="fas fa-plus"></i> Tambah Barang
42          </button>
43          <a href="dashboard-admin.html" class="btn btn-secondary ms-2"><i class="fas fa-arrow-
44            </div>
45        </div>
46
47      <div class="row mt-4">
48        <div class="col-12">
49          <div class="card card-custom">
50            <div class="card-body d-flex flex-column flex-md-row flex-wrap justify-content-be
51              <div class="d-flex flex-column flex-sm-row gap-2 align-items-start align-item
52                <label class="mb-0">Filter Brand:</label>
53                <select id="brandFilter" class="form-select form-select-sm flex-grow-1 fl
54                  <option value="all">Semua Brand</option>
55                </select>
56              </div>
57              <div class="d-flex flex-column flex-sm-row gap-2 align-items-start align-item
58                <label class="mb-0">Cari:</label>
59                <input id="searchInput" class="form-control form-control-sm" type="search"
60              </div>
61            </div>
62          </div>
63        </div>
64      </div>
</div>
```

Gambar 5. 15 Implementasi data barang admin

Gambara 5. 15 Implementasi data barang admin menunjukkan potongan *code* untuk fitur CRUD seperti menggunakan javascript `openAddModal()` berfungsi memunculkan form input data barang yang baru saat di klik, menggunakan elemen `<select id="brandFilter">` untuk memfilter data barang berdasarkan brandnya, pencarian data dengan menggunakan `<input id="searchInput">` untuk mencari data barang berdasarkan kata kunci.

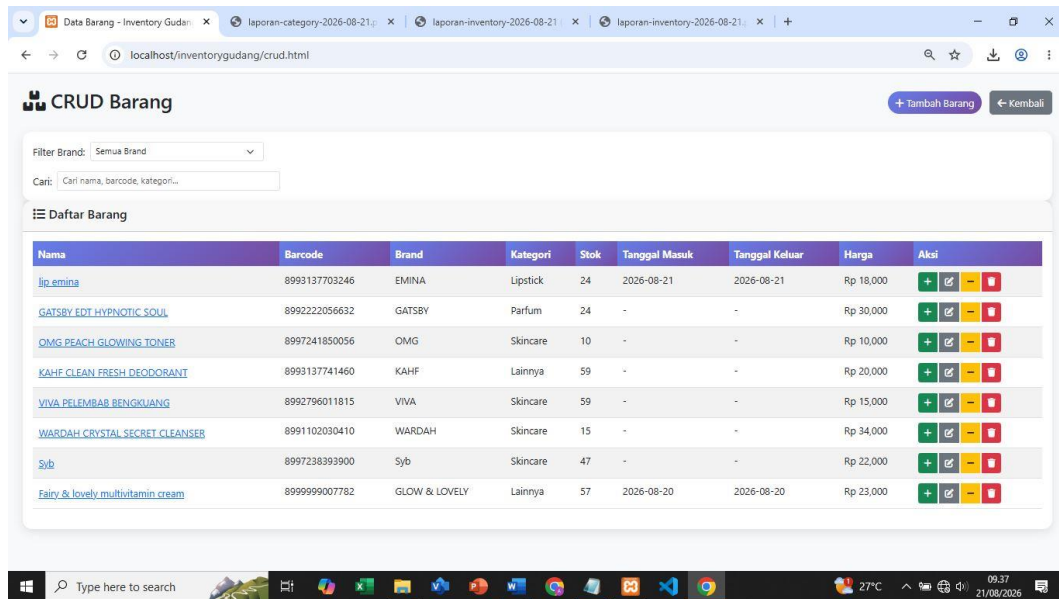
4. Implementasi data barang karyawan

```
2   <html lang="id">
16  <body>
61    <script>
81      function populateBrandKaryawan() {
82        const sel = document.getElementById('brandFilterKaryawan');
83        if (!sel) return;
84        const brands = Array.from(new Set(kItems.map(i => (i.brand || '').trim()).filter(b => b)));
85        sel.innerHTML = '<option value="all">Semua Brand</option>';
86        brands.sort().forEach(b => {
87          const o = document.createElement('option'); o.value = b; o.textContent = b; sel.append
88        });
89      }
90
91      function renderKaryawanItems() {
92        const tbody = document.getElementById('karyawanItemsBody');
93        tbody.innerHTML = '';
94        const brand = (document.getElementById('brandFilterKaryawan')?.value || 'all').toLowerCase();
95        const q = (document.getElementById('searchInputKaryawan')?.value || '').trim().toLowerCase();
96
97        const filtered = kItems.filter(item => {
98          if (brand !== 'all' && (item.brand || '').toLowerCase() !== brand) return false;
99          if (q) {
100            const hay = (item.name + ' ' + item.barcode + ' ' + (item.category || '')) + ' ' +
101              return hay.indexOf(q) !== -1;
102          }
103          return true;
104        });
105
106        if (filtered.length === 0) {
107          tbody.innerHTML = '<tr><td colspan="6" class="text-muted">Tidak ada data</td></tr>';
108          return;
109        }
110      }
```

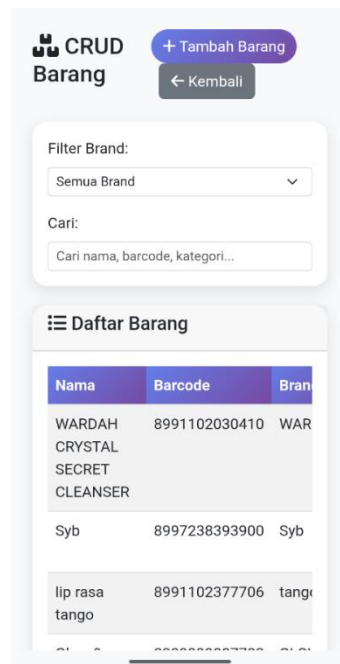
Gambar 5. 16 Implementasi data barang karyawan

Gambara 5. 16 Implementasi data barang karyawan menunjukkan potongan *code* yang berfungsi hampir sama dengan Implementasi data barang admin tapi tidak dengan fitur pengelolaan data CRUD.

5. Tampilan data barang admin



Gambar 5. 17 Tampilan data barang admin

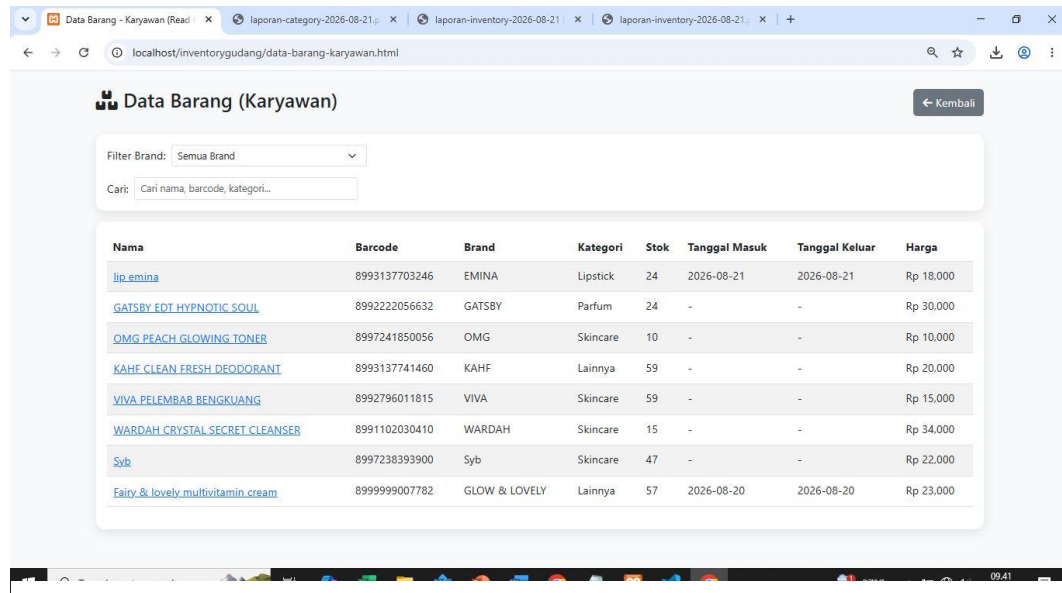


Gambar 5. 18 Tampilan data barang admin di *handphone*

Gambar 5. 17 dan Gambar 5. 18 menampilkan halaman data barang admin yang memiliki fitur tambah barang, edit, hapus, cari barang dan filter brandnya. Admin

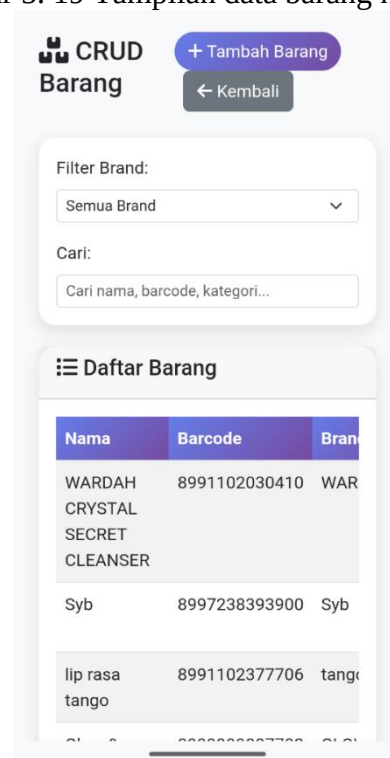
dapat mengelola data dan mengecek stok barangnya di sini dan menjadi lebih mudah.

6. Tampilan data barang karyawan



Nama	Barcode	Brand	Kategori	Stok	Tanggal Masuk	Tanggal Keluar	Harga
lip emina	8993137703246	EMINA	Lipstick	24	2026-08-21	2026-08-21	Rp 18,000
GATSBY EDT HYPNOTIC SOUL	8992222056632	GATSBY	Parfum	24	-	-	Rp 30,000
OMG PEACH GLOWING TONER	8997241850056	OMG	Skincare	10	-	-	Rp 10,000
KAHE CLEAN FRESH DEODORANT	8993137741460	KAHF	Lainnya	59	-	-	Rp 20,000
VIVA PELEMBAB BENGIKUANG	8992796011815	VIVA	Skincare	59	-	-	Rp 15,000
WARDAH CRYSTAL SECRET CLEANSER	8991102030410	WARDAH	Skincare	15	-	-	Rp 34,000
Syb	8997238393900	Syb	Skincare	47	-	-	Rp 22,000
Fairy & lovely multivitamin cream	8999999007782	GLOW & LOVELY	Lainnya	57	2026-08-20	2026-08-20	Rp 23,000

Gambar 5. 19 Tampilan data barang karyawan



Nama	Barcode	Brand
WARDAH CRYSTAL SECRET CLEANSER	8991102030410	WAR
Syb	8997238393900	Syb
lip rasa tango	8991102377706	tango

Gambar 5. 20 Tampilan data barang karyawan di *handphone*

Gambar ini menampilkan halaman data barang karyawan, di sini hanya ada fitur cari barang dan filter brand karena karyawan tidak memiliki izin mengelola data barang. Jadi hanya dapat melihat stok barangnya saja di gudang.

5.3.3 Black box testing

Untuk memastikan rancangan berjalan dengan baik dan benar maka dilakukan *black box testing* pada setiap fitur, menu dan halaman agar berjalan dengan lancar. Berikut ini adalah *black box testing* yang ada pada Tabel 5.7 :

Tabel 5. 7 Blackbox testing

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
1	Login pengguna	Input nama dan kata sandi admin yang benar	Sistem berhasil melakukan verifikasi dan menampilkan <i>dashboard</i> admin	Sistem berhasil menampilkan <i>dashboard</i> admin	Berhasill
		Input nama dan password admin salah atau kosong	Sistem menolak akses dan akan menampilkan “Role atau kredensial tidak valid!”	Sistem tetap di halaman login dan menampilkan peringatan.	
		Input nama dan role yang benar(tanpa password)	Sistem memverifikasi role dan menampilkan <i>dashboard</i> karyawan	Sistem berhasil masuk menampilkan <i>dashboard</i> karyawan	

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
2	Pengelolaan data barang	Mengisi form Tambah Barang lengkap (Kode, Nama, Kategori, Stok, <i>Barcode</i>).	Data barang baru berhasil tersimpan ke database dan tampil pada tabel data barang.	Data tersimpan secara real-time pada tabel master barang.	Berhasill
		Mengubah (edit) informasi detail data barang.	Sistem memperbarui informasi barang sesuai perubahan yang diinputkan.	Informasi data barang berhasil terupdate secara real-time.	
		Menghapus data barang pilihan.	Sistem menampilkan konfirmasi hapus dan menghapus data dari database.		

5.3.4 Sprint review (sprint 1)

Sesudah setiap sprint selesai di uji coba dengan menggunakan *blackbox testing*, kemudian dilakukan Sprint Review bersama pengguna sistem untuk penilaian sistem.

Saat sistem usulan sedang di jelaskan oleh peneliti kepada pengguna (Admin), pengguna juga mencoba sistem usulan dan memberikan *feedback* yang baik dan merasa terbantu oleh sistem usulan ini. Dari sprint review diperoleh beberapa *feedback* positif,dari pengguna (admin) di antaranya:

Tabel 5. 8 *Sprint review (sprint 1)*

Pengguna	Sprint	Review
Admin	Sprint 1	Login berhasil dengan baik, autentikasi berhasil dengan kata sandi dan role pengguna yang sesuai lalu menuju <i>dashboard</i> Admin dan data barang sudah benar sesuai datanya serta bisa mengelola data barang.
Karyawan		Login berhasil dengan baik, menggunakan nama dan role saja bisa masuk ke <i>dashboard</i> karyawan dan dapat melihat data barang tanpa bisa mengelola data barang.

Hasil review ini menjadi bahan evaluasi untuk penyempurnaan sistem pada sprint berikutnya.

5.3.5 *Sprint retrospective (sprint 1)*

Setelah selesai dengan *sprint review*, maka masuk ke tahap selanjutnya yaitu *sprint restrospective*. Dalam tahap ini, peneliti melakukan evaluasi dari hasil *sprint review* untuk pengembangan sistem jadi lebih baik.

Tabel 5. 9 *Sprint retrospective sprint 1*

Pengguna	Aspek evaluasi	<i>Sprint retrospective</i>
Peneliti	<i>Sprint 1</i>	Seluruh sprint sudah di selesaikan sesuai dengan hasil <i>sprint review</i> . Terdapat masukan dari pengguna mengenai tampilan yang terlalu ramai. Masukan tersebut sudah diperbaiki dengan menyederhanakan tampilan supaya tidak ramai dan gampang dipahami oleh pengguna.

5.4 Sprint 2

Pada *sprint* ini berfokus pada sistem input barang masuk atau barang keluar dari gudang dan logika perhitungan stok barang. Dalam menu input barang, menggunakan teknologi *scan barcode* untuk menginput barang jadi lebih mudah dan cepat. Teknologi tersebut terintegrasi dengan kamera *handphone smartphone* pengguna dan stok barang yang otomatis terhitung jika masuk atau keluar barang.

5.4.1 *Sprint planning*

Tabel 5. 10 *Sprint planning (sprint 2)*

No	Target Sprint	Product Backlog	Estimasi Waktu
1	Pembuatan dan pengembangan fitur input barang	Input barang	5 hari
2	Pengembangan logika perhitungan stok otomatis	perhitungan stok otomatis	5 hari

Berdasarkan Tabel 5.10, menghasilkan fokus utama pengembangannya yaitu fitur pembuatan dan pengembangan fitur input barang. Selain itu, pada *sprint* ini mengembangkan proses logika perhitungan stok barang otomatis agar datanya benar.

5.4.2 *Sprint development*

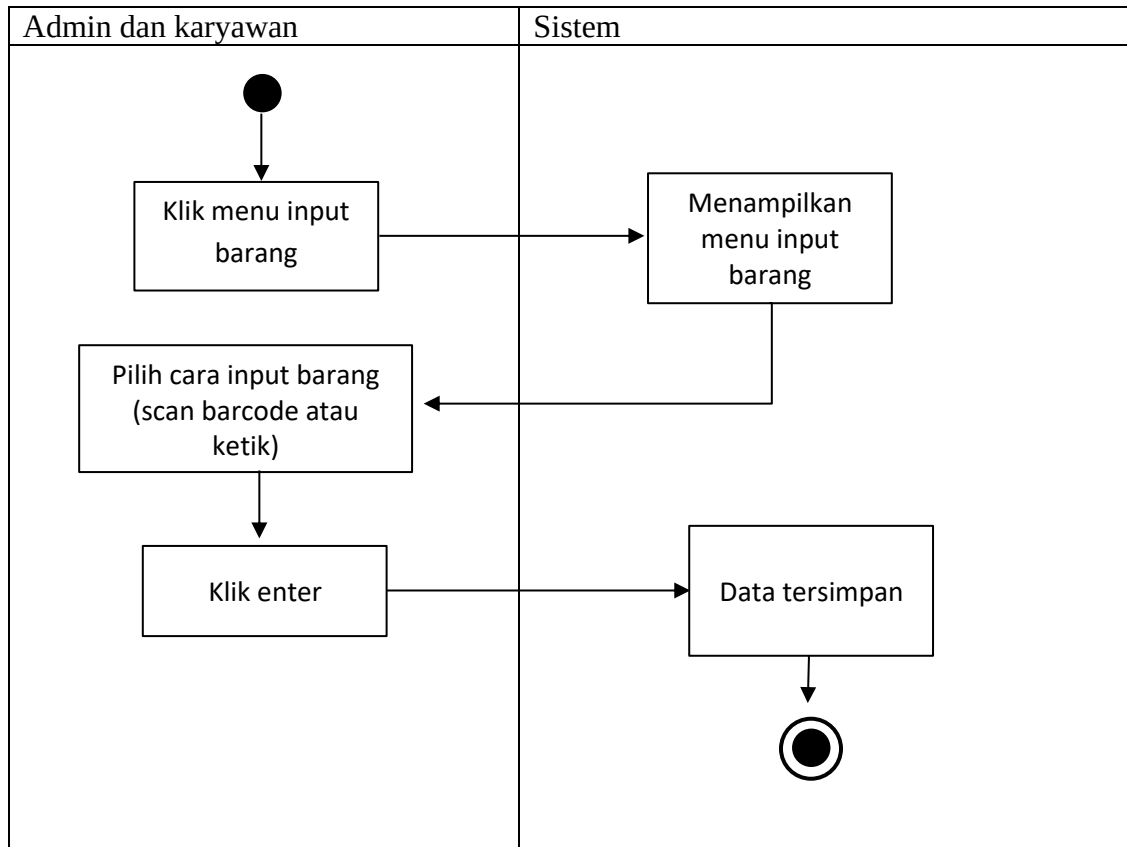
Pada *sprint* ini, peneliti merancang dulu kerangka kerangka kerjanya dengan menggunakan UML terdiri dari *activity diagram*, *class diagram*. Selain kerangka kerja, ada juga *coding* untuk membuat fitur di sistem dan tampilan hasil *coding* yang bisa di jalankan.

5.4.2.1 Input barang

Input barang merupakan halaman yang dapat menginput data barang dengan cara menggunakan *scan barcode* atau jika ada *barcode* yang rusak maka bisa diketik barcodenya.

1. *Activity diagram* input barang

Tabel 5. 11 *Activity diagram* input barang

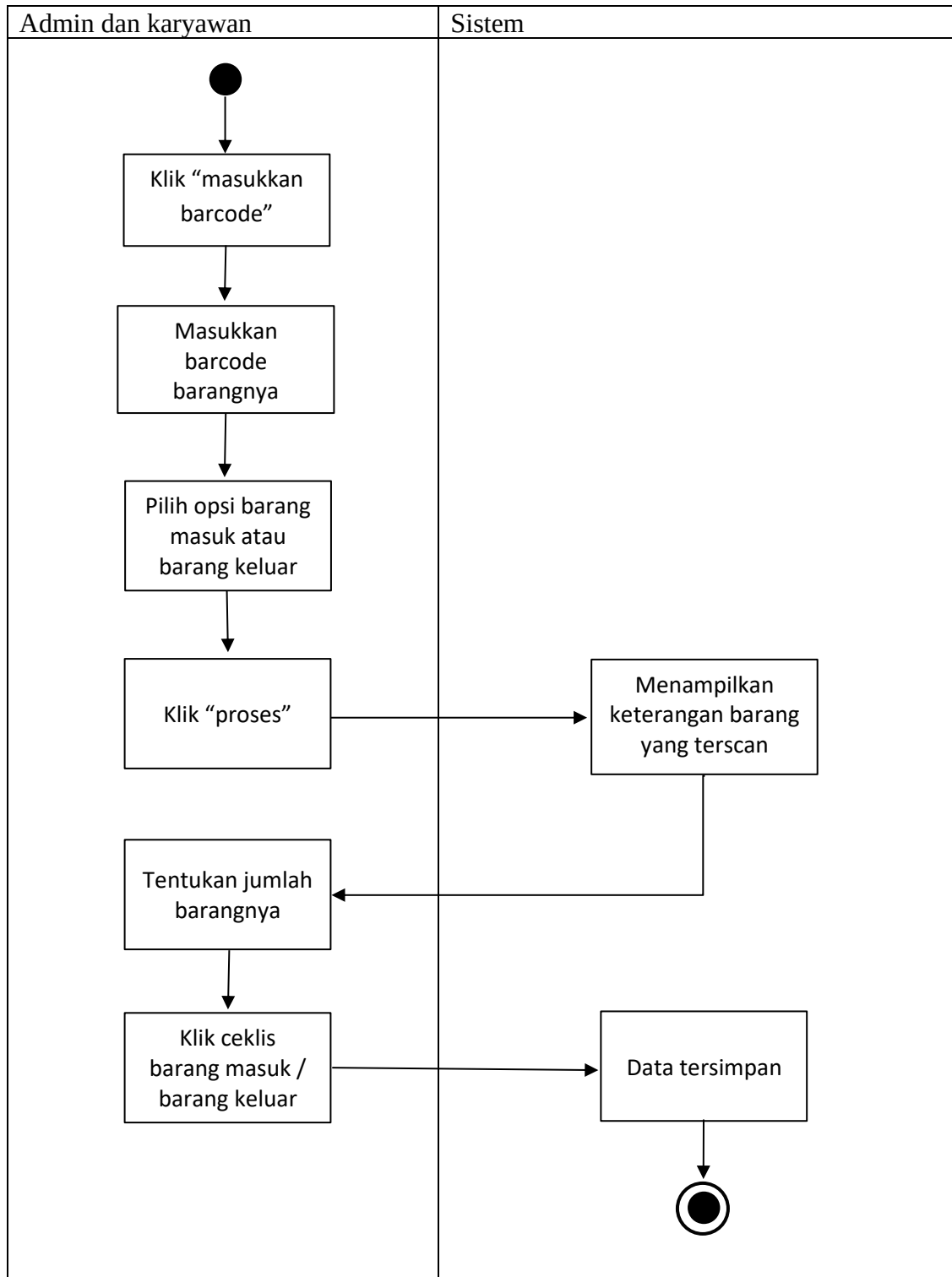


Penjelasan :

Admin dan karyawan mengklik menu input barang, sistem menampilkan menu input barang. Pilih cara input barang ketik atau *scan barcode*, setelah benar klik enter lalu sistem merespon dengan data berhasil tersimpan.

2. Activity diagram ketik barcode

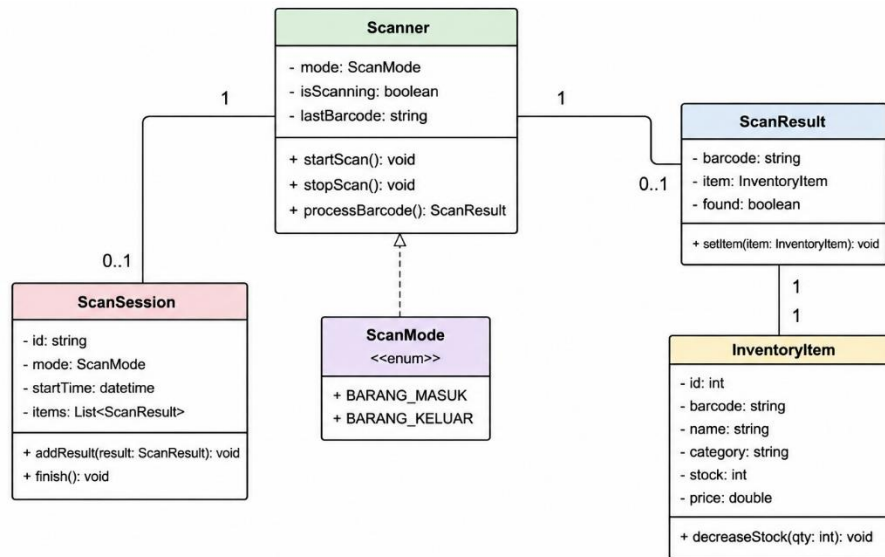
Tabel 5. 12. Activity diagram ketik barcode



Penjelasan :

Admin dan karyawan mengklik masukkan *barcode*, mengisi *barcode* barang lalu pilih opsi barang masuk atau barang keluar. Jika *barcode* sudah benar maka klik proses, sistem merespon dengan menampilkan keterangan barang yang terscan. Setelah itu tentukan jumlah barangnya, klik ceklis barang masuk atau barang keluar yang sudah sesuai dan sistem merespon dengan data berhasil disimpan.

3. Class diagram



Gambar 5. 21 Class Diagram input masuk

Gambar 5. 21 Class Diagram input masuk menunjukkan *class scanner* fitur utama dari proses sistem *scan barcode*. Pada *class scanner* terdapat atribut *mode* untuk pilihan mode barang masuk atau barang keluar, *isScanning* untuk mengecek proses *scan* sedang berjalan, *lastBarcode* untuk menyimpan *barcode* yang berhasil dibaca.

Class scanmode adalah proses menentukan mode *scanning* yang akan dilakukan, di dalamnya terdapat mode *barang_masuk* untuk barang yang masuk ke gudang dan *barang_keluar* untuk barang yang keluar dari gudang.

Class scanresult adalah proses menyimpan *barcode* yang telah berhasil di pindai, memiliki beberapa atribut yaitu dengan *barcode* untuk kode barang, *item* untuk

informasi barang yang ditemukan, found untuk memvalidasi *barcode* ditemukan atau tidak ditemukan.

Class inventoryitem adalah data-data barang yang ada di dalam gudang. *Class* ini memiliki beberapa atribut yaitu *id* untuk ID barang, *barcode* untuk kode barangnya, *name* untuk nama barang, *category* untuk kategori barang, *stock* jumlah stok dan *price* untuk harga barang.

Class scansession adalah penyimpanan aktivitas *scan* yang sedang berjalan atau *scanning* sesi. *Class* ini memiliki beberapa atribut yaitu *id* sebagai identitas *scanning session*, *mode* sebagai opsi jenis *scan barcode* barang masuk atau barang keluar, *starttime* sebagai mulainya proses *scan barcode*.

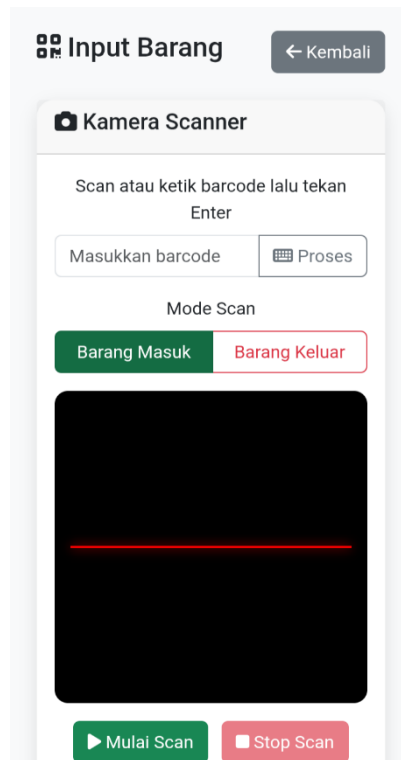
4. Implentasi input barang

```
2  <html lang="id">
3  <head>
9    <style>
13      .card-custom {
14        border: none;
15        border-radius: 15px;
16        box-shadow: 0 5px 15px rgba(0,0,0,0.1);
17      }
18      #interactive {
19        width: 100%;
20        max-width: 100%;
21        margin: 0 auto;
22        min-height: 280px;
23        background: #000;
24        border-radius: 12px;
25        overflow: hidden;
26        display: flex;
27        align-items: center;
28        justify-content: center;
29      }
30      #interactive video,
31      #interactive canvas {
32        width: 100% !important;
33        height: auto !important;
34        max-height: 420px;
35        object-fit: cover;
36        border-radius: 12px;
37      }
38      #interactive video {
39        background: #000;
40      }
41      scanner-container {
```

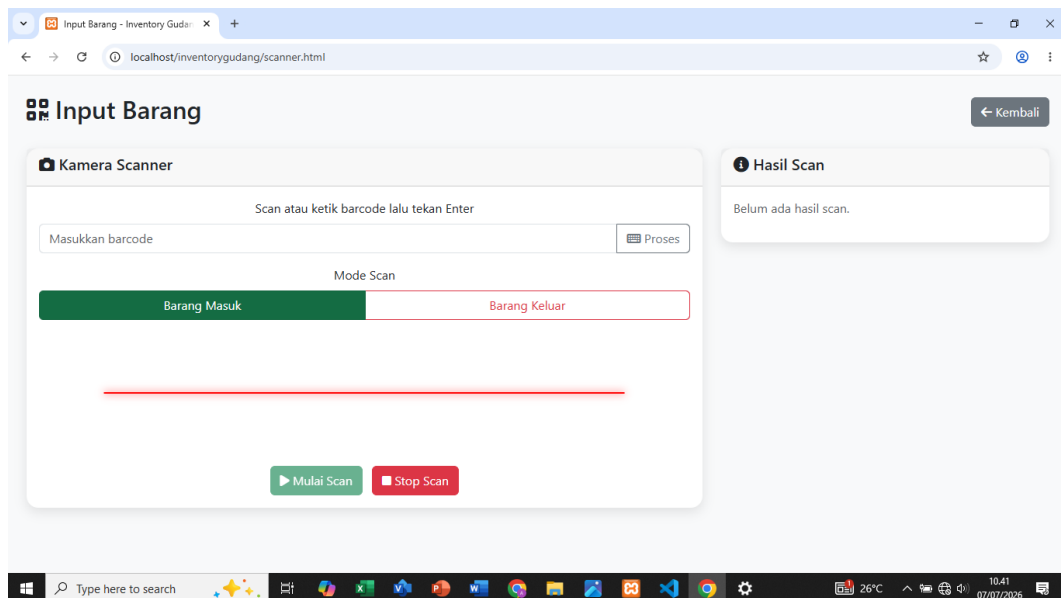
Gambar 5. 22 Implementasi input barang

Gambar 5. 22 implementasi input masuk menunjukkan potongan kode CSS untuk mengatur posisi wadah pemindai dan tampilan visual (seperti tampilan kamera *scan barcode*) agar terlihat rapi, responsive dan modern.

5. Tampilan input barang



Gambar 5. 23 Tampilan input barang (*handphone*)



Gambar 5. 24 Tampilan input barang

Gambar ini menampilkan halaman input barang yang memiliki beberapa fitur yaitu opsi barang masuk atau barang keluar, tombol mulai *scan* untuk memulai *scan barcode*, tombol stop *scan* untuk menghentikan *scanner barcode*, ada juga keterangan hasil *scan* yang nantinya jika sudah terscan maka akan muncul data barangnya.

5.4.3 Blackbox testing

Tabel 5. 13 Blackbox testing sprint 1

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
1	Input Barang Masuk	<i>Scan barcode</i> / ketik <i>barcode</i> barang masuk dan menentukan jumlah (QTY)	Sistem mengenali barang, mencatat riwayat masuk, dan stok fisik bertambah otomatis.	Keterangan barang tampil, jumlah barang tersimpan, dan stok bertambah.	Berhasill
		Input jumlah barang masuk dengan angka minus (≤ 0) atau non numerik.	Sistem menolak input data dan menampilkan peringatan validasi input.	Sistem menampilkan pesan kesalahan input jumlah barang.	
2	Input Barang Keluar	<i>Scan/ketik barcode</i> barang keluar ke	Sistem mengurangi stok barang di gudang secara otomatis	Stok ter-update berkurang secara	Berhasill

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
		area display dan tentukan jumlah.	sesuai QTY keluar.	otomatis dan akurat.	
		Input QTY keluar melebihi sisa stok gudang yang tersedia.	Sistem menolak transaksi dan memberikan peringatan stok tidak mencukupi.	Sistem menampilkan notifikasi error persediaan tidak cukup.	
3	<i>Barcode Scanner</i> (Kamera HP / Ketik)	Arahkan kamera smartphone ke <i>barcode</i> pada fisik produk kosmetik.	Kamera merespons dengan cepat (responsive) dan mengidentifikasi kode <i>barcode</i> barang.	Kode <i>barcode</i> dan nama barang langsung terdeteksi otomatis.	Berhasill
		Ketik kode <i>barcode</i> secara manual saat kondisi <i>barcode</i> fisik rusak.	Sistem mencari data sesuai inputan teks <i>barcode</i> dan menampilkan detail barang.	Data barang berhasil dipanggil dan ditampilkan oleh sistem	

5.4.4 Sprint review

Pada *sprint* ini, pengguna mencoba fitur *scan barcode* pada *handphone smartphone* memilih opsi barang masuk atau barang (*barcode* terbaca oleh kamera *handphone smartphone* dengan responsive).

Tabel 5. 14 *Sprint review (sprint 2)*

Pengguna	Sprint	Review
Admin dan Karyawan	Sprint 2	Menginput barang masuk atau keluar sudah bisa pakai <i>Scan Barcode</i> menggunakan kamera <i>handphone smartphone (responsive</i> dan data benar sesuai data barangnya) dan stoknya sudah otomatis menyesuaikan dengan barang masuk atau kelua.

5.4.5 *Sprint retrospective*

Pada *sprint* ini merupakan tahap terakhir yang dilakukan evaluasi terhadap seluruh proses pengembangan *sprint* dua. Ada aspek yang telah di evaluasi yang telah di perbaiki disajikan di Tabel 5. 16 :

Tabel 5. 15 Sprint retrospective (sprint 2)

Pengguna	Aspek evaluasi	<i>Sprint retrospective</i>
Peneliti	<i>Sprint 2</i>	Seluruh sprint sudah di selesaikan sesuai dengan hasil <i>sprint review</i> . Terdapat masukan dari pengguna mengenai tampilan <i>scan barcode</i> yang tidak pas di layar <i>handphone</i> . Masukan tersebut sudah diperbaiki dengan menata ulang tampilan <i>scan barcode</i> supaya pas dan sesuai dengan layar <i>handphone</i> pengguna.

5.5 Sprint 3

5.5.1 *Sprint planning*

Tabel 5. 16 *Sprint planning (sprint 3)*

No	Target Sprint	Product Backlog	Estimasi Waktu
1	Pembuatan dan pengembangan fitur <i>stock opname</i>	<i>Stock opname</i>	5 hari
2	Pembuatan dan pengembangan laporan barang	Laporan barang	5 hari

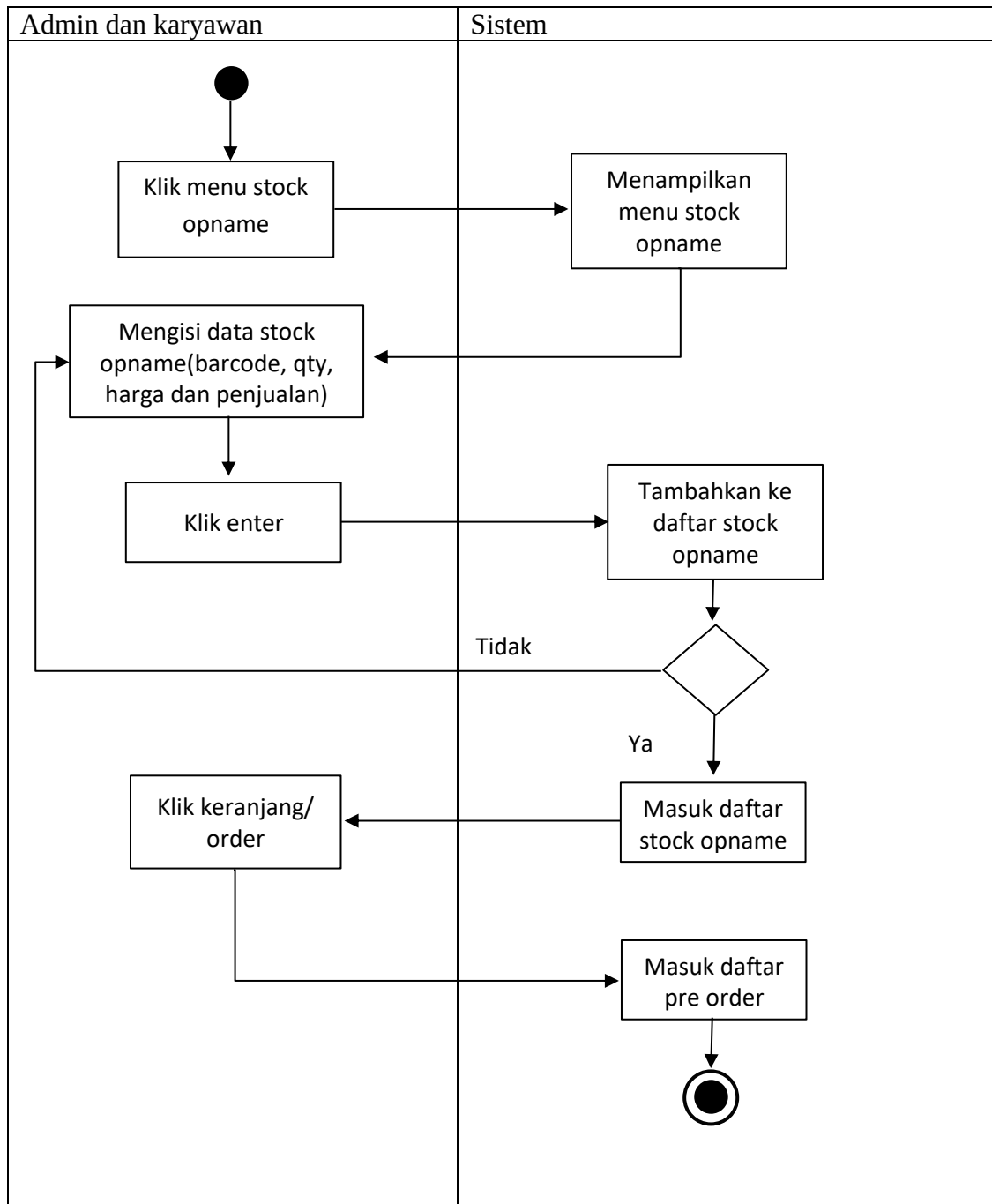
Berdasarkan Tabel 5.17, menghasilkan fokus utama pengembangannya yaitu fitur pembuatan dan pengembangan fitur *stock opname* dan laporan barang. Dalam fitur *stock opname* juga memakai teknologi *scan barcode* untuk mengecek stok fisik barang.

5.5.2 Sprint development

5.5.2.1 Stock opname

1. Activity diagram stock opname

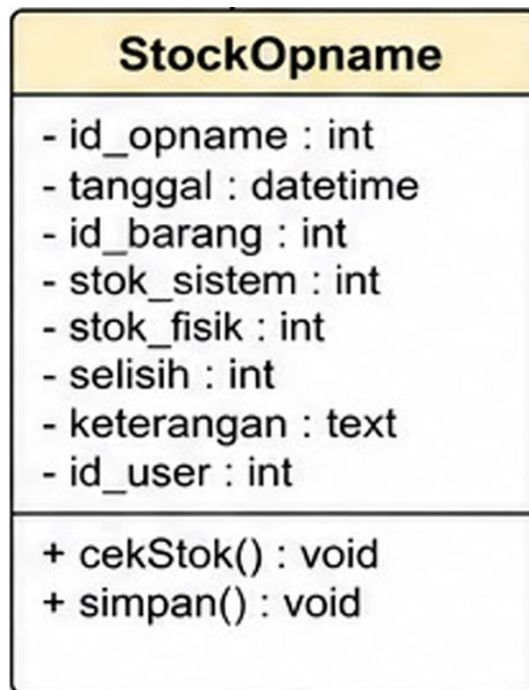
Tabel 5. 17 Activity diagram stock opname



Penjelasan :

Admin dan karyawan mengklik *stock opname*, sistem merespon dengan menampilkan menu *stock opname*. Admin dan karyawan mengisi data stok seperti *barcode* barang, QTY order, harga dan penjualan. Klik enter, kemudian jika belum benar atau salah maka bisa mengisi data Kembali dan jika sudah benar maka tambahkan ke daftar stok opname. Kemudian klik order atau keranjang, sistem merespon dengan orderan masuk ke daftar *pre-order*.

2. Class Diagram stock opname



Gambar 5. 25 Class Diagram stock opname

Class Diagram stock opname adalah class yang digunakan untuk menyimpan hasil data pengecekan stok barang fisik. Ada beberapa atributnya yaitu *id_opname* sebagai nomor identitas setiap melakukan pengecekan barang, tanggal sebagai penanggalan saat pengecekan barang, *id_barang* sebagai nomor identitas barang, *stok_sistem* sebagai data stok barang yang ada di sistem, *stok_fisik* sebagai data stok yang akan di cek, *selisih* sebagai data barang yang terdapat perbedaan antara barang fisik dan data stok di sistem.

3. Implementasi *stock opname*

```
2
133
134   uid p-4">
140   d-custom mb-4">
144   |body">
157     <div class="col-12 col-sm-6 col-lg-3">
158       <label class="form-label">Fisik</label>
159       <input type="number" class="form-control" id="countedQty" min="0" value="1">
160     </div>
161     <div class="col-12 col-sm-6 col-lg-3">
162       <label class="form-label">Order</label>
163       <input type="number" class="form-control" id="orderQty" min="0" value="0">
164     </div>
165     <div class="col-12 col-sm-6 col-lg-3">
166       <label class="form-label">Sales</label>
167       <input type="number" class="form-control" id="salesQty" min="0" value="0">
168     </div>
169     <div class="col-12 col-sm-6 col-lg-3">
170       <label class="form-label">Harga Beli</label>
171       <input type="number" class="form-control" id="buyPrice" min="0" value="0">
172     </div>
173   </div>
174   <div class="row g-3 mt-1 align-items-end">
175     <div class="col-12 col-md-6 col-lg-4">
176       <label class="form-label">Total Beli</label>
177       <input type="text" class="form-control" id="totalBuy" value="0" readonly>
178     </div>
179     <div class="col-12 col-md-6 col-lg-8">
180       <div class="opname-action-group justify-content-md-end">
181         <button id="addOpnameBtn" class="btn btn-primary"><i class="fas fa-plus"></i> Tambah ke O
182         <button id="clearOpnameBtn" class="btn btn-secondary"><i class="fas fa-times"></i> Bersih
183       </div>
```

Gambar 5. 26 Implementasi *stock opname*

Berdasarkan gambar 5. 26 menunjukkan potongan kode untuk form input untuk melakukan *stock opname* yang memiliki beberapa data inputnya yaitu fisik untuk input data barang yang ada di gudang, order untuk pemesanan barang, sales untuk barang yang sudah terjual, harga beli untuk harga modal pembelian barang dan total beli untuk jumlah harga barang yang akan di pesan yang tidak bisa di edit (*read only*). Selain itu, ada juga *addOpname* sebagai tombol *controller* untuk menambahkan data *stock opname* dan *clearOpname* sebagai tombol *controller* untuk membersihkan data *stock opname* yang salah atau tidak jadi.

4. Tampilan *stock opname*

The screenshot shows a web browser window with the address bar displaying 'localhost/inventorygudang/stock-opname.html'. The page title is 'Stock Opname'. Below the title, there is a '+ Catat Hasil Stock Opname' section. This section contains input fields for 'Barcode', 'Fisik' (Physical), 'Order', 'Sales', and 'Harga Beli' (Purchase Price). The 'Barcode' field has a 'Scan' button next to it. Below these fields, there is a 'Total Beli' (Total Purchase) field. A 'Cetak PDF' (Print PDF) button is located below the 'Total Beli' field. At the bottom of the section, there is a table with the following data:

Barcode	Nama Barang	Brand	Stok	Fisik	Order	Sales	Harga Beli	Total Beli	Tanggal
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1	0	0	Rp 0	Rp 0	4/7/2026
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1	4	1	Rp 22.000	Rp 88.000	4/7/2026
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1	0	0	Rp 0	Rp 0	4/7/2026

Gambar 5. 27 Tampilan *stock opname*

The screenshot shows the same 'Stock Opname' web application on a smartphone. The layout is adapted for a smaller screen. The '+ Catat Hasil Stock Opname' section is at the top. Below it, there is a 'Scan / Masukkan Barcode' section with a 'Barcode' input field and a 'Scan' button. Below this, there are input fields for 'Fisik', 'Order', 'Sales', 'Harga Beli', and 'Total Beli'. A 'Cetak PDF' button is located below the 'Total Beli' field. At the bottom, there is a table with the following data:

Barcode	Nama Barang	Brand	Stok	Fisik
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1
8993137703246	EMINA BRIGHT STUFF LOOSE POWDER	EMINA	1	1

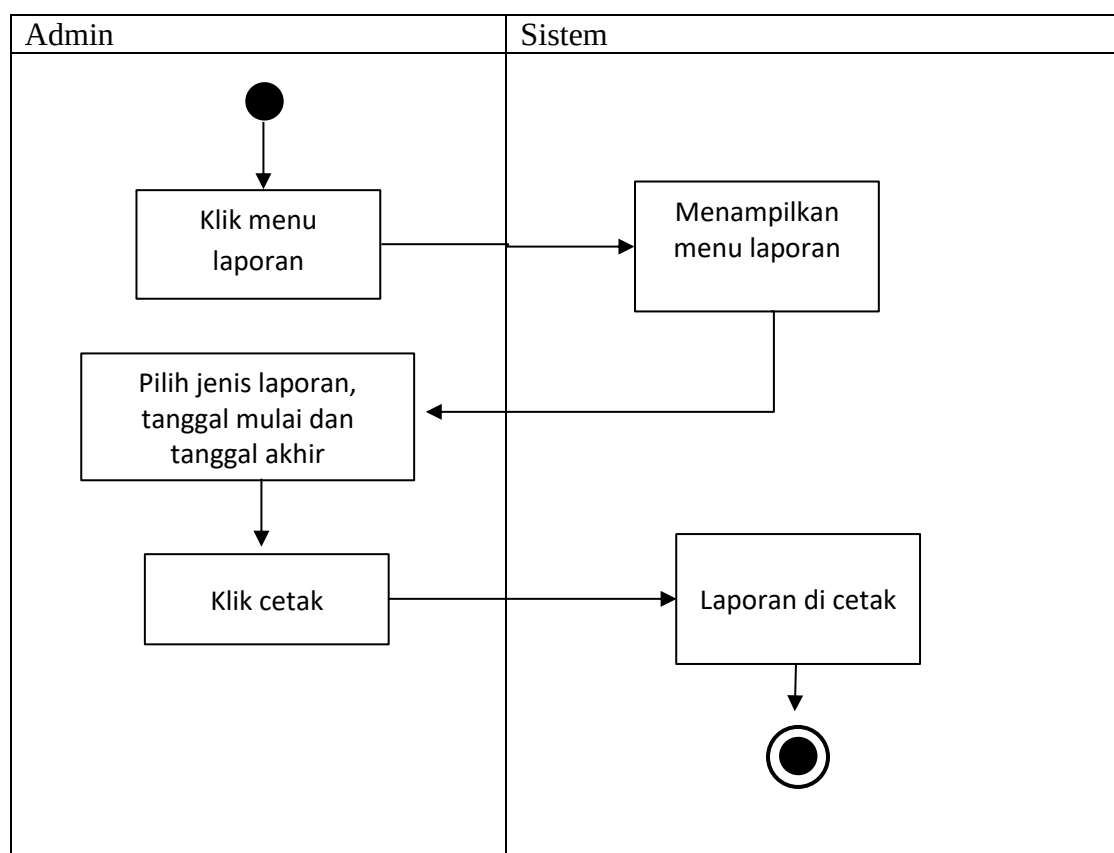
Gambar 5. 28 Tampilan *stock opname* di *handphone*

Gambar 5. 27 dan gambar 5.28 menampilkan halaman *stock opname* yang bisa memakai *scan barcode* atau pun ketik *barcode* dengan tampilan yang mudah di mengerti. Di halaman ini juga data *stock opname* bisa di *export* jadi format pdf.

5.5.2.2 Laporan barang

1. Activity diagram laporan barang

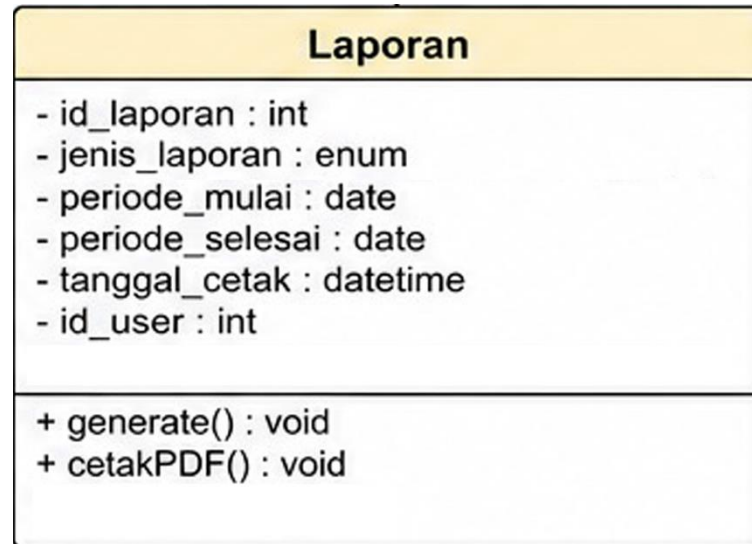
Tabel 5. 18 Activity diagram laporan barang



Penjelasan :

Admin klik menu laporan, sistem merespon dengan menampilkan menu laporan. Kemudian pilih jenis laporan, tanggal mulai dan laporan tanggal akhir. Klik *export* jika sudah benar, laporan pun berhasil di *export*.

2. *Class Diagram* laporan barang



Gambar 5. 29 *Class Diagram* laporan barang

Class Diagram laporan barang adalah *class* yang digunakan untuk membuat laporan barang. Ada beberapa atributnya yaitu *id_laporan* sebagai nomor identitas setiap melakukan laporan, *jenis_laporan* sebagai pemilihan jenis laporan (laporan inventori, laporan stok sedikit, laporan *pre-order* dan laporan per kategori), *periode_mulai* sebagai tanggal waktu membuat laporan, *periode_selesai* sebagai tanggal diharapkan datangnya barang pesanan, *tanggal_cetak* sebagai tanggal saat laporan di cetak. Selain itu, ada juga *method generate* sebagai prose membuat laporan berdasarkan jenis laporan dan *cetakPDF* sebagai pencetakan laporan jadi PDF.

3. Implementasi laporan barang

```
2   <html lang="id">
35  <body>
85  <script>

157  function generateInventoryReport() {
158      const items = reportItems;
159      const totalValue = items.reduce((sum, item) => sum + (Number(item.price) * Number(item.sto
160      const totalItems = items.length;
161      const lowStockItems = items.filter(item => item.stock <= 2).length;
162
163      return `
164          <div class="report-section">
165              <div class="card card-custom">
166                  <div class="card-header">
167                      <h5><i class="fas fa-boxes"></i> Laporan Inventory</h5>
168                  </div>
169                  <div class="card-body">
170                      <div class="row mb-4">
171                          <div class="col-md-4">
172                              <div class="text-center">
173                                  <h3>${totalItems}</h3>
174                                  <p>Total Barang</p>
175                              </div>
176                          </div>
177                          <div class="col-md-4">
178                              <div class="text-center">
179                                  <h3>${lowStockItems}</h3>
180                                  <p>Barang Stok Rendah</p>
181                              </div>
182                          </div>
183                          <div class="col-md-4">
184                              <div class="text-center">
185                                  <h3>Rp. ${totalValue.toLocaleString()}</h3>
```

Gambar 5. 30 Implementasi laporan barang

Berdasarkan gambar 5.30 menunjukkan potongan kode untuk pembuatan laporan inventory (*generateinventoryreport*) dari data barang yang tersedia dia gudang. Program ini mengambil data barang, kemudian menghitung semua jumlah barang, menghitung total persediaan barang dan filter data laporan (laporan inventori, laporan stok sedikit, laporan *pre-order* dan laporan per kategori).

4. Tampilan laporan barang

Laporan ← Kembali

Filter Laporan

Jenis Laporan

Tanggal Mulai

Tanggal Akhir

Generate Laporan Cetak PDF

Laporan Inventory

4
Total Barang

1
Barang Stok Rendah

Rp 2.732.000
Total Nilai Inventory

Gambar 5. 31 Tampilan laporan barang di *handphone*

Laporan ← Kembali

Filter Laporan

Jenis Laporan

Tanggal Mulai

Tanggal Akhir

Generate Laporan Cetak PDF

Laporan Inventory

8
Total Barang

0
Barang Stok Rendah

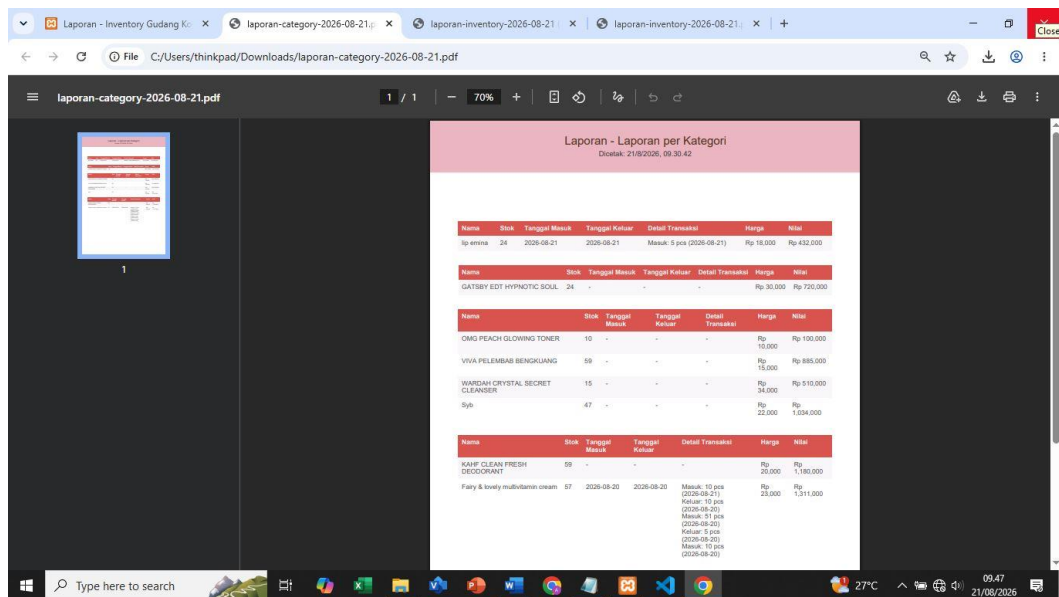
Rp 6,172,000
Total Nilai Inventory

Nama Barang	Kategori	Stok	Tanggal Masuk	Tanggal Keluar	Detail Transaksi	Harga	Total Nilai
lip emina	Lipstick	24	2026-08-21	2026-08-21	Masuk 5 pcs (2026-08-21)	Rp 18,000	Rp 432,000
GATSBY EDT HYPNOTIC SOUL	Parfum	24	-	-	-	Rp 30,000	Rp 720,000
OMG PEACH GLOWING TONER	Skincare	10	-	-	-	Rp 10,000	Rp 100,000
KAHIF CLEAN FRESH DEODORANT	Lainnya	59	-	-	-	Rp 20,000	Rp 1,180,000
VIVA PELEMBAB BENGKUANG	Skincare	59	-	-	-	Rp 15,000	Rp 885,000
WARDAH CRYSTAL SECRET CLEANSER	Skincare	15	-	-	-	Rp 34,000	Rp 510,000
Syo	Skincare	47	-	-	-	Rp 22,000	Rp 1,034,000
Fairy & lovely multivitamin cream	Lainnya	57	2026-08-20	2026-08-20	Masuk 10 pcs (2026-08-21) Keluar 10 pcs (2026-08-20) Masuk 51 pcs (2026-08-20) Keluar 5 pcs (2026-08-20) Masuk 10 pcs (2026-08-20)	Rp 23,000	Rp 1,311,000

Gambar 5. 32 Tampilan laporan barang

Gambar 5. 31 dan gambar 5. 32 menampilkan halaman laporan, ada beberapa jenis laporan yang bisa di print yaitu laporan inventory, laporan barang stok rendah, laporan *pre-order* dan laporan per kategori. Data-data tersebut juga bisa di *export* jadi format pdf yang mudah dipahami admin.

5. Tampilan *export* PDF



Gambar 5. 33 Tampilan *export* PDF

Gambar ini menampilkan data yang sudah di *export*, tertata rapih dan sesuai dengan keperluan admin.

5.5.3 *Blackbox testing*

Tabel 5. 19 *Blackbox testing*

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
1	Stock opname	Mengisi data fisik barang (Barcode, QTY	Sistem dapat menginput data, menghitung total	Bisa input data, total selisih muncul sesuai hitungan dan data berhasil	Berhasil

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
		fisik, Harga, sales penjualan).	selisih dan memasukkan data ke list <i>opname</i> .	masuk ke <i>list</i> keranjang <i>opname</i> .	
		Klik tombol "Keranjang" pada item hasil <i>stock opname</i> .	Data item <i>opname</i> masuk secara otomatis menjadi daftar item pesanan <i>Pre-order</i> .	Item terhubung langsung ke daftar acuan <i>Pre-order</i> .	Berhasil
2	Laporan barang dan Export PDF	Pilih Jenis Laporan (Laporan Inventory, Laporan Stok Rendah, PO, Per Kategori) dan Tanggal.	Sistem menampilkan data laporan yang sesuai dengan opsi laporan yang dipilih.	Tabel laporan menampilkan data yang akurat dan filter laporan berfungsi lancar.	Berhasil
		Klik tombol "Cetak / Export PDF".	Sistem membuat dokumen laporan yang berformat PDF secara rapi dan siap cetak.	File format PDF berhasil diunduh dengan tata letak data yang rapi dan akurat.	Berhasil

5.5.4 *Sprint review*

Pada *sprint* ini pengguna mencoba fitur *stock opname* untuk pengecekan fisik barang yang ada di gudang menggunakan *scan barcode* yang terhubung dengan kamera *handphone smartphone*. Selain itu, pengguna juga mencoba fitur laporan barang dengan beberapa jenis laporan seperti laporan inventori, laporan stok rendah, laporan perkategori dan laporan *pre-order*.

Tabel 5. 20 *Sprint review (sprint 3)*

Pengguna	<i>Sprint</i>	<i>Review</i>
Admin	<i>Sprint</i> 3	Pengecekan barang juga dapat digunakan dengan baik, admin dapat membuat laporan jadi lebih cepat dan efisien.
Karyawan		Pengecekan barang juga dapat digunakan dengan baik dan responsif

5.5.5 *Sprint retrospective*

Pada *sprint* ini merupakan tahap terakhir yang dilakukannya evaluasi terhadap seluruh proses pengembangan *sprint* tiga. Ada aspek yang telah di evaluasi yang telah di perbaiki disajikan di Tabel 5. 22 :

Tabel 5. 21 *Sprint retrospective (sprint 3)*

Pengguna	Aspek evaluasi	<i>Sprint retrospective</i>
Peneliti	<i>Sprint 2</i>	Seluruh sprint sudah di selesaikan sesuai dengan hasil <i>sprint review</i> . Terdapat masukan dari pengguna mengenai tampilan <i>scan barcode</i> yang tidak pas di layar <i>handphone</i> . Masukan tersebut sudah diperbaiki dengan menata ulang tampilan <i>scan barcode</i> supaya pas dan sesuai dengan layar <i>handphone</i> pengguna.

5.6 Sprint 4

5.6.1 Sprint planning

Tabel 5. 22 *Sprint planning (sprint 4)*

No	Target Sprint	Product Backlog	Estimasi Waktu
1	Pengembangan fitur <i>scan barcode</i>	<i>scan barcode</i>	4 hari
2	Pengembangan notifikasi stok sedikit otomatis	notifikasi stok sedikit otomatis	3 hari
3	Pembuatan dan pengembangan fitur <i>pre_order</i>	<i>pre_order</i>	3 hari

Berdasarkan Tabel 5.17, menghasilkan fokus utama pengembangannya yaitu pembuatan fitur *pre_order*. Selain pembuatan fitur, ada juga pengembangan fitur yaitu pengembangan fitur *scan barcode*, notifikasi stok sedikit otomatis dan *pre_order* dengan waktu yang telah di tentukan.

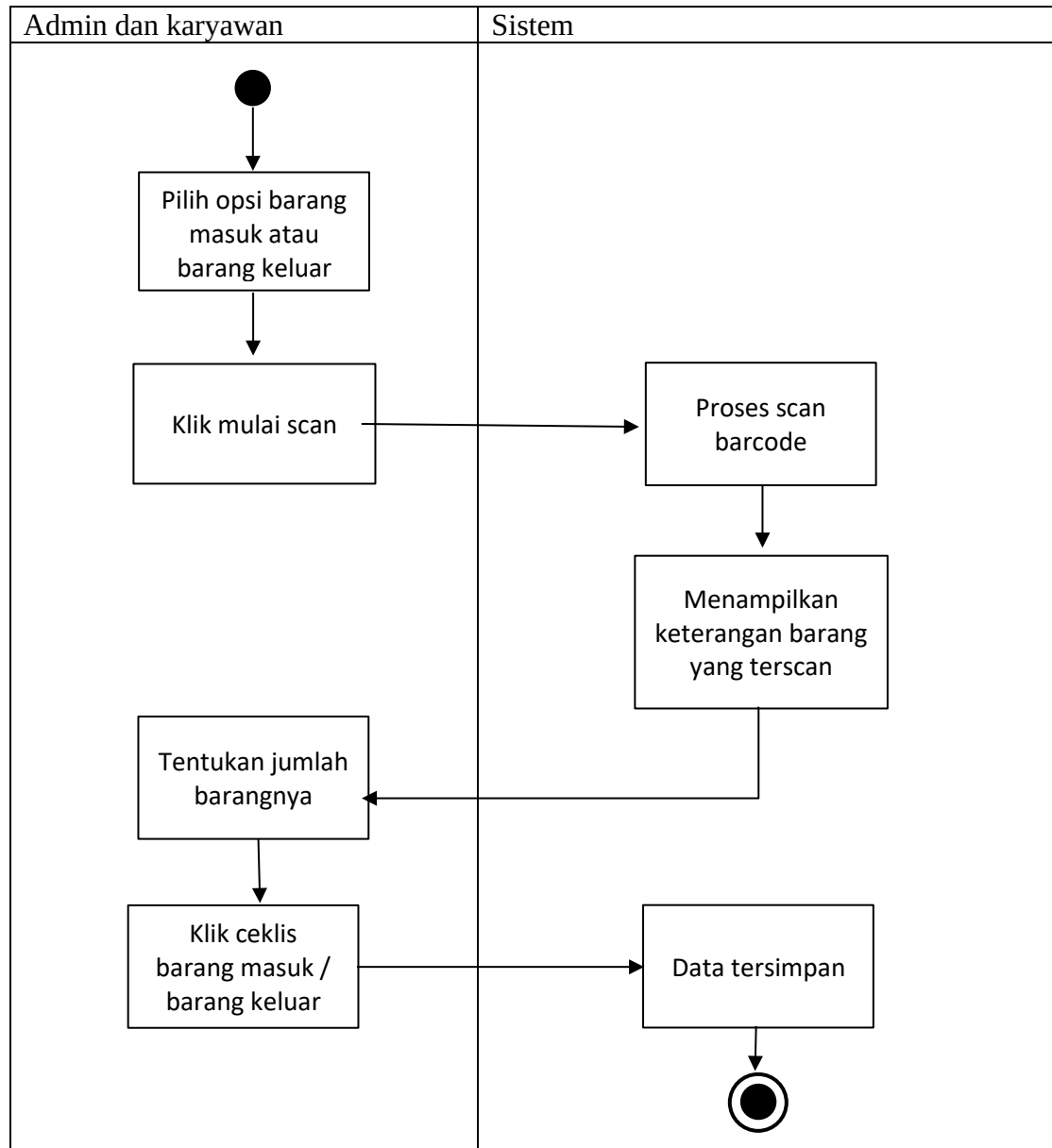
5.6.2 Sprint development

5.6.2.1 Scan barcode

Scan barcode merupakan teknologi yang berguna untuk memindai *barcode* pada barang dengan menggunakan kamera. Pada penelitian terdahulu ada juga yang memakai teknologi ini, namun memakai alat ketiga berupa alat pemindai yang masih terhubung dengan kabel. Pada penelitian ini, *scan barcode* menggunakan *handphone smartphone* yang lebih efisien, fleksibel dan jadi cepat mengelola data. Fitur ini terdapat pada dua menu yaitu di menu input barang untuk memindai *barcode* barang yang masuk atau keluar dan di menu *stock opname* untuk mengecek barang yang tersedia di gudang.

1. Activity diagram scan barcode

Tabel 5. 23 Activity diagram scan barcode



Penjelasan :

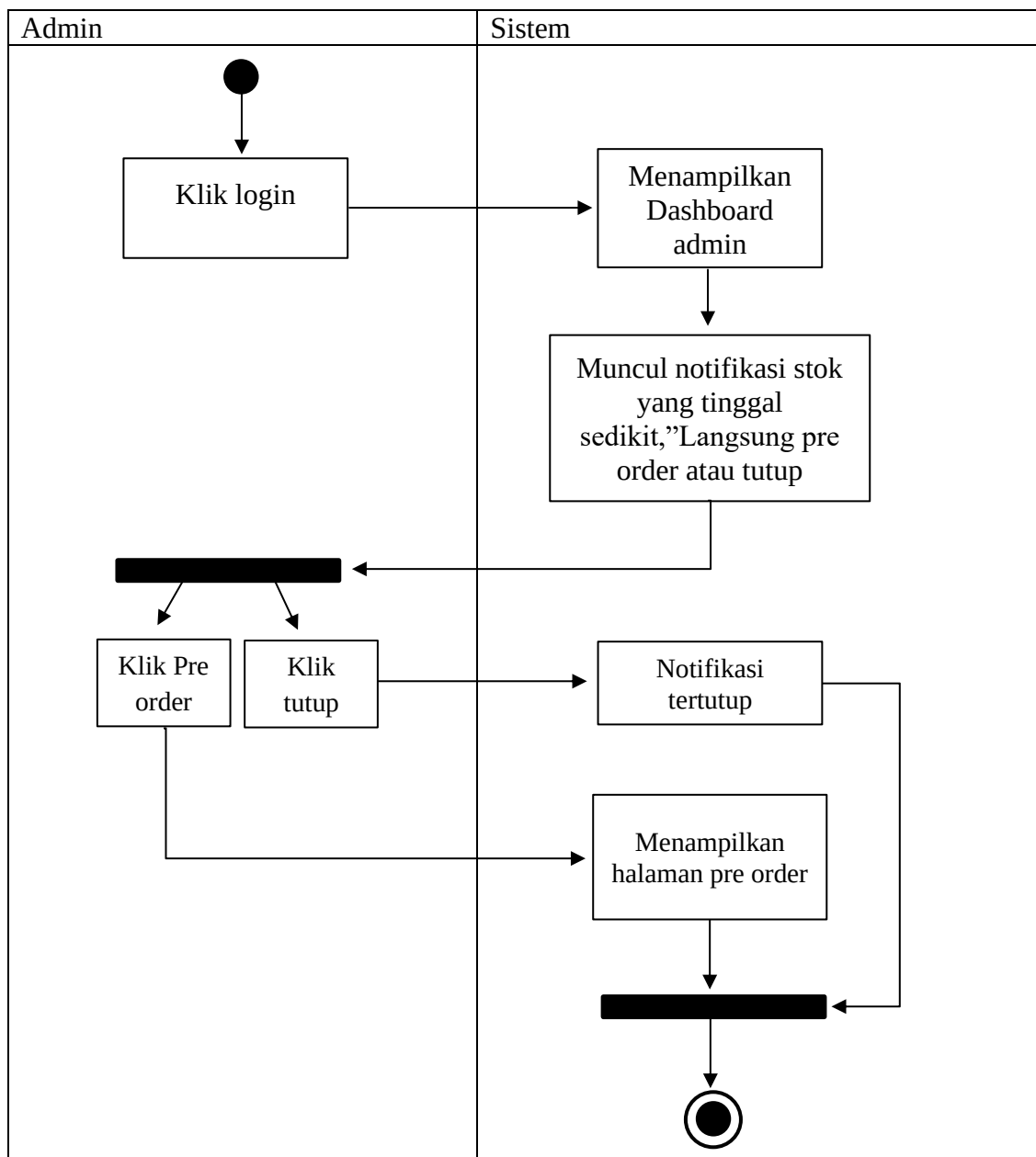
Admin dan karyawan pilih opsi barang masuk atau barang keluar lalu klik mulai scan, sistem merespon dengan memproses scan barcode kemudian sistem menampilkan keterangan barang yang terscan. Admin atau karyawan menentukan jumlah barangnya, setelah jumlah sesuai maka klik ceklis barang masuk atau barang keluar yang di inginkan. Sistem merespon dengan data berhasil di simpan.

5.6.2.2 Notifikasi stok sedikit otomatis

Notifikasi stok sedikit otomatis adalah fitur yang akan muncul jika stok barang gudang kurang dari dua pcs dan notifikasi ini muncul dengan format *pop-up*.

1. Activity diagram notifikasi stok sedikit otomatis

Tabel 5. 24 Activity diagram notifikasi stok sedikit otomatis



Penjelasan :

Pada proses ini admin sudah login dan menampilkan *dashboard* admin, lalu akan muncul *pop up* notifikasi stok tinggal sedikit (kurang dari 2 pcs). Akan ada 2 opsi yaitu klik tutup jika tidak ingin melakukan *pre-order* dan klik *pre-order* jika ingin melakukan *pre-order* untuk memesan barang. Jika memilih opsi *pre-order* maka akan membuka halaman *pre-order*.

2. *Class Diagram* notifikasi stok sedikit otomatis



Gambar 5. 34 *Class Diagram* notifikasi stok sedikit otomatis

Gambar 5. 34 merupakan gambaran dari atribut pada notifikasi otomatis yang terdiri dari `id_notifikasi` sebagai penomoran stok, `id_barang` sebagai penomoran barang, `batas_minimum` sebagai pembatasan munculnya notif dan `pesan` sebagai notifikasi itu sendiri. Selain itu, ada juga *method* yaitu `cekstok()` untuk mengecek stok di sistem dan `kirimnotifikasi()` kemunculan notif dari sistem.

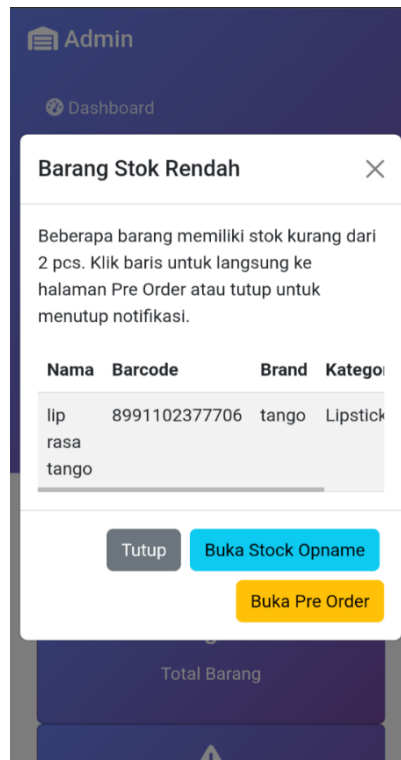
3. Implementasi notifikasi stok sedikit otomatis

```
2    <html lang="id">
52   <body>
181   <script>
333       function checkLowStockNotifications() {
334           if ('Notification' in window) {
335               if (Notification.permission === 'granted') {
336                   const lowStockItems = dashboardItems.filter(item => item.stock <= 2);
337                   lowStockItems.forEach(item => {
338                       const notificationKey = `notified_${item.id}`;
339                       if (!localStorage.getItem(notificationKey)) {
340                           new Notification('Stok Rendah!', {
341                               body: `Stok ${item.name} tinggal ${item.stock} pcs`,
342                               icon: 'https://via.placeholder.com/64x64?text=▲'
343                           });
344                           localStorage.setItem(notificationKey, 'true');
345                       }
346                   });
347               } else if (Notification.permission !== 'denied') {
348                   Notification.requestPermission();
349               }
350           }
351       }
352
353       setInterval(() => {
354           loadDashboard().catch(() => {});
355       }, 30000);
356
357       loadDashboard();
358   </script>
359 </body>
360 </html>
```

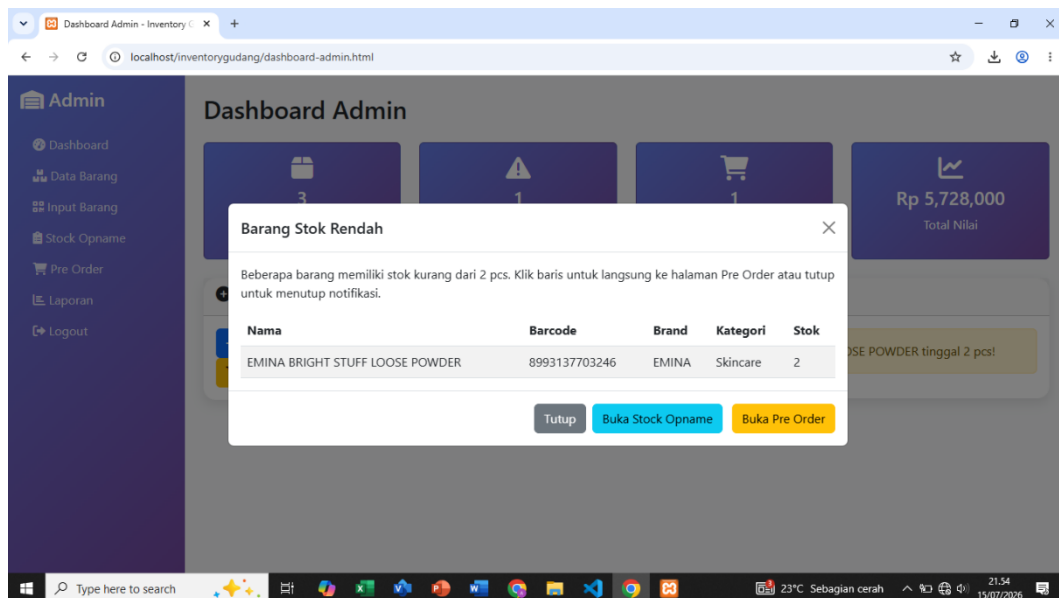
Gambar 5. 35 Implementasi Notifikasi stok sedikit otomatis

Berdasarkan gambar 5.35 menunjukkan potongan kode notifikasi stok sedikit otomatis. *Checklowstocknotifications()* sebagai pengecekan stok data barang yang sedikit di sistem, *'notification' in window* untuk menampilkan notif dan logika sistem untuk memunculkan notif.

4. Tampilan notifikasi stok sedikit otomatis



Gambar 5. 36 Tampilan notifikasi stok sedikit otomatis di *handphone*



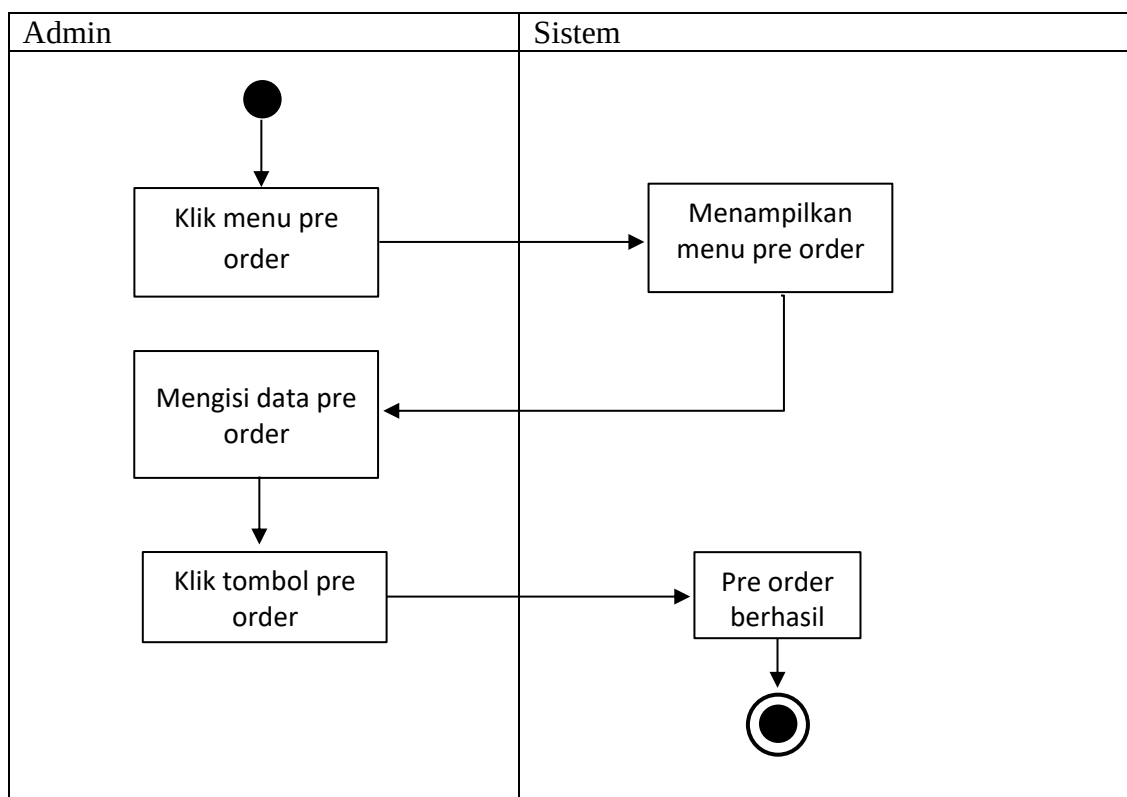
Gambar 5. 37 Tampilan notifikasi stok sedikit otomatis

Gambar 5. 36 dan gambar 5. 37 menampilkan notifikasi otomatis jika stok barang di Gudang kurang dari dua pcs yang di sebabkan oleh karyawan atau admin yang mengeluarkan stok barang dari gudang. Notifikasi ini akan otomatis muncul saat admin membuka web yang memungkinkan untuk admin bisa langsung *pre-order*.

5.6.2.3 Pre_order

1. Activity diagram pre_order

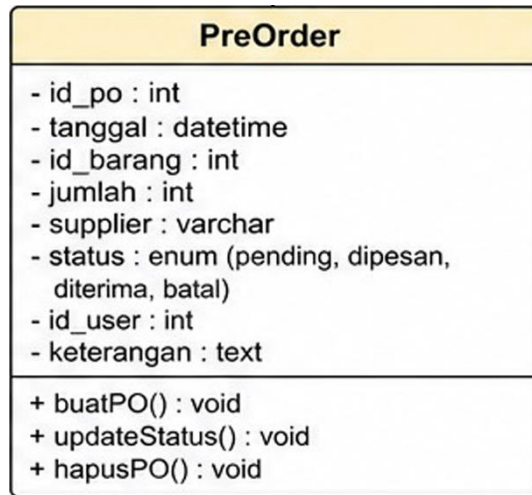
Tabel 5. 25 Activity diagram pre-order



Penjelasan :

Admin mengklik menu *pre-order*, sistem merespon dengan menampilkan menu *pre-order*. Admin mengisi data *pre-order* seperti supplier, tanggal *pre-order*, tanggal diharapkan datang barang, catatan jika perlu dan pilih data stok opname.

2. Class Diagram pre-order



Gambar 5. 38 Class Diagram pre-order

Berdasarkan gambar 5. 38 *class pre-order* merupakan menu utama untuk pemesanan barang yang memiliki atribut, diantaranya `id_po` untuk penomoran, `tanggal` untuk penanggalan pemesanan, `supplier` untuk nama perusahaan suppliernya. Selain itu, ada juga *method* yaitu pembuatan PO, update statusnya (*pending dan sukses*).

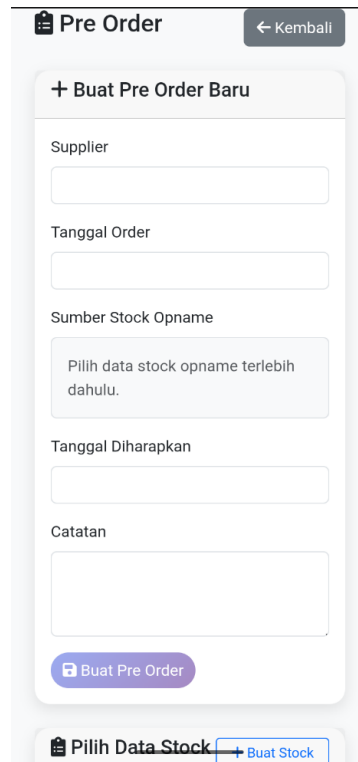
3. Implementasi *pre_order*

```
295 function setPreOrderSource(record) {
296     document.getElementById('selectedOpnameId').value = record.id;
297     document.getElementById('preOrderSourceForm').classList.remove('text-muted');
298     document.getElementById('preOrderSourceForm').innerHTML = `
299         <strong>${record.itemName}</strong> (${record.brand || '-'})<br>
300         Barcode: ${record.barcode}<br>
301         Stok: ${record.currentStock}<br>
302         Fisik: ${record.countedQty}<br>
303         Order: ${record.orderQty}<br>
304         Sales: ${record.salesQty}<br>
305         Harga Beli: Rp ${Number(record.buyPrice).toLocaleString('id-ID')}<br>
306         Total Beli: Rp ${Number(record.totalBuy).toLocaleString('id-ID')}`;
307     document.getElementById('preOrderSourceList').classList.remove('text-muted');
308     document.getElementById('preOrderSourceList').innerHTML = `
309         <strong>${record.itemName}</strong> (${record.brand || '-'})<br>
310         Barcode: ${record.barcode}<br>
311         Stok: ${record.currentStock}<br>
312         Fisik: ${record.countedQty}<br>
313         Order: ${record.orderQty}<br>
314         Sales: ${record.salesQty}`;
315     document.getElementById('preOrderSubmitBtn').disabled = false;
316 }
317
318 function clearPreOrderSelection() {
319     document.getElementById('selectedOpnameId').value = '';
320     document.getElementById('preOrderSourceForm').classList.add('text-muted');
321     document.getElementById('preOrderSourceForm').innerHTML = 'Pilih data stock opname terlebih dahulu';
322     document.getElementById('preOrderSourceList').classList.add('text-muted');
323     document.getElementById('preOrderSourceList').innerHTML = 'Pilih data stock opname terlebih dahulu';
324 }
```

Gambar 5. 39 Implementasi *pre-order*

Berdasarkan gambar 5. 39 menunjukkan potongan kode fitur *pre-order* seperti *setPreOrderSource(record)* untuk memilih data stok yang akan di pesankan, *selectedOpnameid* untuk id stock opname yang akan dipilih dan untuk tampilan data barang yang dilakukan *pre-order*.

4. Tampilan *pre_order*



Pre Order [← Kembali](#)

+ Buat Pre Order Baru

Supplier

Tanggal Order

Sumber Stock Opname

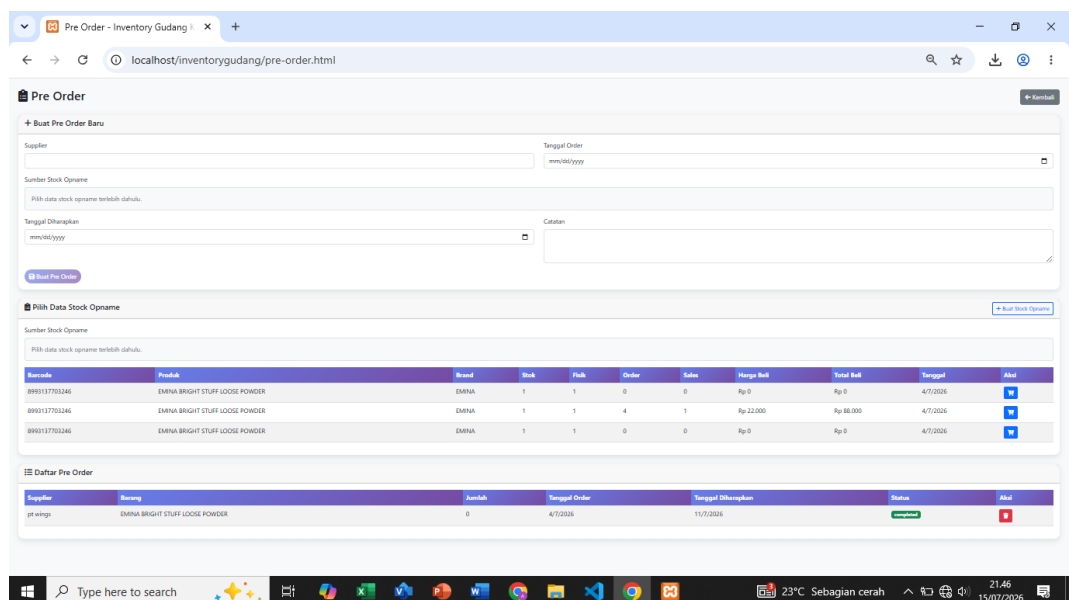
Tanggal Diharapkan

Catatan

[Buat Pre Order](#)

[Pilih Data Stock](#) [+ Buat Stock](#)

Gambar 5. 40 Tampilan *pre-order* di handphone



Pre Order [← Kembali](#)

+ Buat Pre Order Baru

Supplier Tanggal Order

Sumber Stock Opname

Tanggal Diharapkan Catatan

[Buat Pre Order](#)

Pilih Data Stock Opname [+ Buat Stock Opname](#)

Sumber Stock Opname

Barcode	Produk	Brand	Stok	Piik	Order	Saldo	Harga Beli	Total Beli	Tanggal	Aksi
8993137703246	EMANA BRIGHT STUFF LOOSE POWDER	EMANA	1	1	0	0	Rp 0	Rp 0	4/7/2026	+
8993137703246	EMANA BRIGHT STUFF LOOSE POWDER	EMANA	1	1	4	1	Rp 22.000	Rp 88.000	4/7/2026	+
8993137703246	EMANA BRIGHT STUFF LOOSE POWDER	EMANA	1	1	0	0	Rp 0	Rp 0	4/7/2026	+

Daftar Pre Order

Supplier	Barang	Jumlah	Tanggal Order	Tanggal Diharapkan	Status	Aksi
pt wings	EMANA BRIGHT STUFF LOOSE POWDER	0	4/7/2026	11/7/2026	Completed	-

Gambar 5. 41 Tampilan *pre-order*

Gambar ini menampilkan halaman *pre-order*, halaman ini berfungsi sebagai untuk memesan barang yang stoknya sedikit lagi atau habis. Ada beberapa data yang harus di isi yaitu supplier, tanggal order, tanggal di harapkan datang dan pilih data *stock opname*. Ada juga daftar *stock opname* yang harus di pilih agar bisa di *pre-order*kan dan ada daftar *pre-order* yang sudah di lakukan.

5.6.3 Blackbox testing

Tabel 5. 26 Blackbox testing (sprint 4)

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
1	Notifikasi Otomatis Stok Minimum	Stok barang gudang tersisa < 2 pcs akibat transaksi barang keluar.	Sistem secara otomatis memunculkan <i>pop-up</i> notifikasi stok menipis pada <i>Dashboard Admin</i> .	<i>Pop-up</i> peringatan stok rendah muncul saat Admin membuka <i>dashboard</i>	Berhasill
		Admin mengklik tombol "Langsung <i>Pre Order</i> " pada notifikasi <i>pop-up</i> .	Sistem mengarahkan Admin secara otomatis menuju halaman pembuatan <i>Pre Order</i> .	Sistem langsung berpindah ke menu <i>form Pre-Order</i> .	Berhasill
2	Pengelolaan <i>Pre-order</i>	Mengisi form PO (Supplier, Tgl Order,	Transaksi <i>Pre-Order</i> berhasil tercatat dan tersimpan ke	Data <i>Pre-Order</i> berhasil	Berhasill

No	Scenario pengujian	Input Data	Hasil yang diharapkan	Hasil pengujian	Status
		Tgl Diharapkan, Sumber Stock Opname).	dalam daftar riwayat PO.	disimpan dan dapat dipantau statusnya.	

Berdasarkan tabel 5. 26 peneliti mengecek sistem perfiturnya, dengan pengetesan sistem bisa di pakai pengguna dan *review* untuk mendapatkan *feedback* dari pengguna.

5.6.4 *Sprint review*

Pada *sprint* ini pengguna mencoba fitur *stock opname* untuk pengecekan fisik barang yang ada di gudang menggunakan *scan barcode* yang terhubung dengan kamera *handphone smartphone*. Selain itu, pengguna juga mencoba fitur laporan barang dengan beberapa jenis laporan seperti laporan inventori, laporan stok rendah, laporan perkategori dan laporan *pre-order*.

Tabel 5. 27 *Sprint review (sprint review)*

Pengguna	<i>Sprint</i>	<i>Review</i>
Admin	Sprint 4	<i>Scan Barcode</i> berjalan dengan responsive dan benar, notifikasi otomatis juga muncul saat ada stok yang tinggal sedikit. Admin dapat melakukan <i>pre-order</i> lebih cepat.
Karyawan		<i>Scan Barcode</i> berjalan dengan responsive dan benar,

5.6.5 *Sprint retrospective*

Pada sprint ke empat ini, melakukan evaluasi demi kenyamanan pengguna dan fleksibilitas biar cepat melakukan pengecekan stok barang di gudang dan pemesanan barang. Pernyataan itu disajikan pada Tabel 5. 28 :

Tabel 5. 28 *Sprint retrospective (sprint 4)*

Pengguna	Aspek evaluasi	<i>Sprint retrospective</i>
Peneliti	<i>Sprint 4</i>	Seluruh sprint sudah di selesaikan sesuai dengan hasil <i>sprint review</i> . Terdapat masukan dari pengguna mengenai notifikasi otomatis yang mau bisa langsung buka <i>stock opname</i> dan <i>pre-order</i> . Masukan tersebut sudah diperbaiki dengan menambahkan tombol <i>controller</i> agar bisa masuk ke <i>stock opname</i> dan <i>pre-order</i> dengan cepat.

5.7 Penilaian *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah tahapan pengujian sistem oleh pengguna untuk mengetahui nilai penggunaan terhadap Sistem *Inventory* gudang berbasis web. Pengujian ini dilakukan setelah semua sistem sudah selesai dikerjakan dengan penilaian kepada penggunanya yaitu admin dan karyawan. Penilaian ini dilakukan dengan cara mengisi kuesoner oleh pengguna dengan lima tingkat skala likert. Pada setiap tingkat memiliki bobot nilai yang berbeda, bobot tersebut disajikan pada Tabel 5. 29 :

Tabel 5. 29 Penilaian *user acceptance testing*

No	Jawaban	Bobot
1	Sangat setuju (SS)	5

No	Jawaban	Bobot
2	Setuju (S)	4
3	Kurang setuju (KS)	3
4	Tidak setuju (TS)	2
5	Sangat tidak setuju (STS)	1

Kuesioner yang dilakukan memiliki beberapa pernyataan mengenai penggunaan Sistem *inventory* gudang berbasis web yang digunakan oleh admin dan karyawan Toko Kosmetik XYZ (karyawan internal dan karyawan eksternal atau SPG), pertanyaan tersebut di sajikan pada Tabel 5. 30 :

Tabel 5. 30 *Kuesioner User Acceptance Testing*

No	Pernyataan	SS	S	KS	TS	STS
1	Sistem informasi <i>inventory</i> gudang berbasis web mudah dipahami	1	1	1	0	0
2	Sistem informasi <i>inventory</i> gudang berbasis web mudah digunakan	2	1	0	0	0
3	Proses <i>login</i> bisa di lakukan dengan baik dan mudah	2	1	0	0	0
4	Sistem informasi <i>inventory</i> gudang berbasis web berguna membantu operasional Toko Kosmetik XYZ	1	2	0	0	0

No	Pernyataan	SS	S	KS	TS	STS
5	Fitur <i>scan barcode</i> berjalan dengan reponsif	2	1	0	0	0
6	Fitur <i>stock opname</i> membuat pengecekan barang jadi cepat	1	2	0	0	0
7	Pemantauan stok dapat di lihat secara <i>real-time</i>	3	0	0	0	0
8	Notifikasi stok sedikit otomatis berjalan dengan baik	1	2	0	0	0
9	Laporan barang gudang jadi lebih cepat dibuat	1	2	0	0	0
10	Pemesanan barang <i>pre-order</i> jadi cepat	1	2	0	0	0

(Sumber: Penelitian 2026)

Berdasarkan Tabel 5. 30, dapat diperoleh hasil data responden kuesioner terhadap sepuluh pernyataan mengenai Sistem Informasi *Inventory Gudang Berbasis Web*. Kuesioner tersebut di berikan kepada tiga responden yang mencoba dan menggunakan sistem ini. Hasil kuesioner ini menghasilkan lima tingkat skala yaitu Sangat Setuju (SS), Setuju (S), Kurang Setuju (KS), Tidak Setuju (TS) dan Sangat Tidak Setuju(STS). Data-data tersebut kemudian akan diolah jadi perhitungan bobot nilai, persentase dan rata-rata nilai.

5.7.1 Perhitungan User Acceptance Testing (UAT)

Perhitungan *User Acceptance Testing (UAT)* dilakukan untuk menghasilkan data kuesioner yang menentukan efesiensi sistem yang dirancang. Perhitungan tersebut menggunakan rumus diantaranya yaitu :

1. Menghitung jumlah nilai

$$(SS \times 5) + (S \times 4) + (KS \times 3) + (TS \times 2) + (STS \times 1) = \text{jumlah nilai}$$

Keterangan:

SS: Sangat Setuju

S: Setuju

TS: Tidak Setuju

KS: Kurang Setuju

STS: Sangat Tidak Setuju

2. Menghitung nilai rata-rata

$$\frac{\text{Jumlah nilai}}{\text{Jumlah responden}}$$

Keterangan : Jumlah nilai dibagi Jumlah responden

3. Menghitung persentase

$$\frac{\text{rata-rata}}{\text{Total}} \times 100\%$$

Keterangan: rata-rata dibagi 5, dikali 100%

Dengan adanya rumus perhitungan ini maka kuesioner diolah jadi data hasil sebagai berikut ini :

Tabel 5. 31 Data hasil bobot nilai UAT

No	Pernyataan	Nilai					Jumlah	Rata-rata	Persentase
		SS x 5	S x 4	KS x 3	TS x 2	STS x 1			
1	Sistem informasi inventory gudang	5	4	3	0	0	12	2,4	80%

No	Pernyataan	Nilai					Jumlah	Rata-rata	Persentase
		SS x 5	S x 4	KS x 3	TS x 2	STS x 1			
	berbasis web mudah dipahami								
2	Sistem informasi <i>inventory</i> gudang berbasis web mudah digunakan	10	4	0	0	0	14	2,8	93%
3	Proses <i>login</i> bisa dilakukan dengan baik dan mudah	10	4	0	0	0	14	2,8	93%
4	Sistem informasi <i>inventory</i> gudang berbasis web berguna membantu operasional Toko	5	8	0	0	0	13	2,6	87%

No	Pernyataan	Nilai					Jumlah	Rata-rata	Persentase
		SS x 5	S x 4	KS x 3	TS x 2	STS x 1			
	Kosmetik XYZ								
5	Fitur <i>scan barcode</i> berjalan dengan reponsif	10	8	0	0	0	18	3,6	120%
6	Fitur <i>stock opname</i> membuat pengecekan barang jadi cepat	5	8	0	0	0	13	2,6	87%
7	Pemantauan stok dapat di lihat secara <i>real-time</i>	15	0	0	0	0	15	3	100%
8	Notifikasi stok sedikit otomatis berjalan dengan baik	5	8	0	0	0	13	2,6	87%
9	Laporan barang	5	8	0	0	0	13	2,6	87%

No	Pernyataan	Nilai					Jumlah	Rata-rata	Persentase
		SS x 5	S x 4	KS x 3	TS x 2	STS x 1			
	gudang jadi lebih cepat dibuat								
10	Pemesanan barang <i>pre-order</i> jadi cepat	5	8	0	0	0	13	2,6	87%

(Sumber: Penelitian 2026)

Berdasarkan Tabel 5.31 didapatkan hasil *User Acceptance Testing (UAT)* terhadap 10 pernyataan kepada tiga orang responden. Hasilnya menunjukkan bahwa Sistem Informasi *Inventory* gudang berbasis Web mendapatkan penilaian positif dengan total nilai skor 138, nilai rata-ratanya 2,76 dan persentase 87%. Dengan penilaian tersebut dapat disimpulkan bahwa sistem ini dapat memenuhi kebutuhan pengguna dan layak dipakai sebagai sistem untuk pengelolaan data barang gudang di Toko Kosmetik XYZ.

5.8 Pembahasan

5.8.1 Pembahasan rumusan masalah pertama

Rumusan masalah pertama, berfokus pada pembahasan mengenai bagaimana cara merancang sistem informasi untuk inventori gudang yang tadinya manual memakai pencatatan dengan kertas menjadi digital.

Penelitian ini menghasilkan rancangan sistem informasi inventori gudang berbasis web yang bisa menggantikan pencatatan manual dengan kertas menjadi digital, sistem ini bisa dirancang karena penulis melakukan observasi merasakan langsung sistem yang masih manual, wawancara (kepada admin, karyawan dan kepala toko). Dengan adanya sistem ini membuat data barang menjadi lebih rapih, aman dan

mudah dicari kembali karena tersimpan di dalam database. Admin jadi terbantu saat mengelola persediaan stok barang yang lebih akurat dan efektif karena stok barangnya dapat dipantau secara langsung dengan media *handphone* atau *Komputer*.

5.8.2 Pembahasan rumusan masalah kedua

Pada rumusan masalah ini, yang di bahas adalah berfokus pada penerapan metode *scrum* saat pengembangan sistemnya.

Penelitian ini memperlihatkan pengembangan sistem yang menggunakan metode *scrum* yang fleksibel dan terarah. Pengembangannya dilakukan secara bertahap dengan memakai *sprint* yang sudah dirancang sehingga kebutuhan pengguna dapat di penuhi.

Peneliti ini juga dapat memperbaiki sistem melalui masukan atau saran dari pengguna dengan proses *sprint review* dan *sprint retrospective* yang menghasilkan sistem sesuai dengan kebutuhan Toko Kosmetik XYZ.

5.8.3 Pembahasan rumusan masalah ketiga

Pada rumusan masalah ini berfokus mengenai bagaimana rancangan sistem bisa mengurangi kesalahan pencatatan data barang gudang dan mempercepat laporan barang.

Penerapan rancangan sistem informasi inventori gudang mampu mengurangi kesalahan pencatatan data barang karena semua data disimpan dalam database dan adanya validasi data. Selain itu, sistem ini memakai teknologi *barcode scanner* yang terhubung dengan kamera *handphone* untuk membantu mempercepat proses input data barang.

Laporan barang yang tadinya dilakukan secara manual sekarang bisa dilakukan secara otomatis sehingga lebih cepat dan efisien.

5.8.4 Pembahasan penelitian

Berdasarkan hasil penerapan rancangan sistem informasi inventori gudang, penelitian ini memiliki *novelty* atau kebaruan. Karena itu, membuat penelitian ini mempunyai keunggulan dari penelitian sebelumnya yang serupa.

Keunggulan yang pertama yaitu penggunaan teknologi *barcode scanner* yang terhubung dengan kamera *handphone*, sehingga tidak membutuhkan alat tambahan *scanner barcode* yang mahal dan ribet.

Keunggulan yang kedua yaitu, penggunaan fitur notifikasi stok sedikit otomatis yang bisa membantu memperingatkan admin adanya barang yang hampir habis.

Keunggulan yang ketiga, penggunaan fitur *pre-order* yang membantu proses pemesanan barang kepada *supplier* lebih cepat sehingga mengantisipasi kehabisan stok.

Berdasarkan hasil penelitian ini, terbukti bahwa “ Perancangan Sistem Informasi *Inventory Gudang Berbasis Web Di Toko Kosmetik Xyz Kabupaten Ciamis Dengan Metode Scrum* ” dapat mengelola data inventori barang gudang dengan lebih cepat, efisien dan *real-time* dibandingkan dengan sistem pencatatan inventori manual yang sebelumnya digunakan.