

BAB IV

PERANCANGAN, PEMBUATAN DAN UJI COBA

4.1 Permasalahan Bisnis

Toko retail ini menggunakan sistem manajemen secara manual tanpa menggunakan aplikasi berbasis teknologi informasi apapun sebelumnya. Pembagian tugas dan manajemen toko bergantung pada perintah dari pemilik. Karena kurangnya kepastian pembagian tugas dan wewenang di antara karyawan sehingga setiap karyawan memiliki wewenang di setiap sektor, hal ini menyebabkan munculnya potensi terjadinya permasalahan *fraud* atau ke dalam proses bisnis.[13]

Karena pemantauan stok dilakukan secara manual tanpa adanya pencatatan data secara daring, pemilik toko mengeluhkan bahwa seringkali terjadi salah *order* untuk menambah stok, yang dimana menyebabkan stok barang menjadi menumpuk atau kurangnya restok untuk barang yang sudah habis.

Seringnya terjadi kesalahan dalam *restock* inilah yang menyebabkan terjadinya kerugian kepada bisnis yang sedang berjalan. Untuk menghindari kerugian, bisnis yang sedang berjalan memerlukan sebuah sistem yang dapat melayani pencatatan dan pengecekan arus keluar masuk stok barang yang ada dalam gudang.[14]

4.2 Perancangan Aplikasi

Dalam perancangan aplikasi, pendekatan untuk melakukan pembangunan aplikasi, menggunakan teknik yang biasa disebut sebagai *Rapid Application Development (RAD)*. [15] Dalam pembangunan aplikasi menggunakan teknik ini, analisa mendalam akan kebutuhan dari bisnis haruslah matang dari awal pembuatan aplikasi. Sebelum aplikasi ini dibuat, wawancara kepada pemilik bisnis telah

dilakukan. Dan dari sanalah didapatkan apa saja yang dibutuhkan oleh perusahaan untuk dibangun dalam sistem.

Dari hasil wawancara yang dilakukan oleh pemilik toko retail, kebutuhan mendasar yang dibutuhkan bisnis ini adalah sebuah aplikasi yang dapat melayani pencatatan produk-produk beserta dengan stok yang ada dalam gudang. Kebutuhan ini muncul karena pemilik menyadari bahwa sering terjadi salah order ulang terhadap beberapa macam barang yang menyebabkan stok menumpuk dan juga terjadi kelalaian dalam stok ulang barang yang sudah hampir habis stoknya. Aplikasi ini dibuat untuk memantau alur keluar masuk penjualan barang yang terjadi dalam bisnis retail.

Karena zaman semakin maju, penulis menambahkan fitur bonus yang diberikan kepada pemilik toko retail berupa formulir order barang secara online tanpa harus menggunakan *platform marketplace* yang ada. Formulir daring ini mempermudah penjualan secara daring bagi beberapa pengguna yang ingin melihat-lihat dulu apa saja barang yang tersedia dalam daftar barang.

Karena banyaknya barang yang ada, aplikasi ini akan memiliki fitur berupa *QR-Code Generator* yang memberikan kemudahan dalam mengecek kondisi secara deskriptif dari sebuah barang. Hasil dari *generator* ini menghasilkan kode yang jika dilakukan *scan* akan mempermudah pengecekan barang beserta dengan stoknya. Pengecekan ini juga berguna untuk semua nota baik pembelian maupun penjualan. Sehingga pengecekan nota tidak perlu lagi membuka-buka nota kertas yang sangat sulit dicari. Pengurutannya berdasarkan tanggal dimana transaksi dilakukan, sehingga mempermudah untuk melihat kapan transaksi terjadi.

Untuk menghindari *fraud* atau kecurangan dari pekerja dalam toko retail ini. Sistem juga akan dilengkapi dengan *action executor recorder* yang akan mencatat siapakah yang melakukan eksekusi dari sebuah penambahan, perubahan maupun penghapusan. Hal ini dapat mempermudah pemilik untuk memantau aktivitas pekerja yang bekerja di dalam sistem sehingga kemungkinan kecurangan menjadi lebih kecil.

4.2.1 Persiapan Awal Pembangunan *Back-End*

Untuk membangun aplikasi ini, kerangka kerja utama yang digunakan adalah Laravel 7.0. Laravel memiliki struktur kerangka kerja yang menerapkan *MVC* untuk mempermudah manajemen kode menjadi lebih terstruktur dan rapi karena memisahkan antara transaksi data dengan basis data, logika dalam aplikasi dan struktur tampilan menjadi 3 bagian yang berbeda.

Untuk melakukan instalasi Laravel, pertama-tama langkah yang harus dilakukan adalah mempersiapkan Composer sebagai *package installer* dari Laravel. Composer akan melakukan instalasi paket-paket yang telah tertera di dalam *file composer.json* yang telah tersedia ketika folder Laravel telah dibuat.

Setelah itu, hal yang harus dilakukan adalah membuat semua *database migration* berdasarkan struktur yang telah dibuat ketika melakukan analisa awal dan perancangan awal. Pada tahap ini basis data telah siap dipakai untuk melakukan transaksi-transaksi data berdasarkan logika bisnis yang nanti akan ditulis pada *controller*. Setelah kedua hal ini selesai dipersiapkan, selanjutnya hal yang harus disiapkan adalah bagian *front-end*.

4.2.2 Persiapan Awal Pembangunan *Front-End*

Untuk membangun sistem ini, hal yang harus dipersiapkan adalah *dashboard*. Pembangunan bagian ini akan memerlukan *template* yang tersedia secara gratis dan bebas hak cipta. Template yang digunakan adalah *Stisla Dashboard Admin*. [16]

Untuk mempermudah interaksi antara pengguna dan aplikasi, dibutuhkan sifat-sifat atau *behaviour* dari tampilan aplikasi agar mempermudah aksesibilitasnya, untuk memenuhi kebutuhan ini, aplikasi ini menggunakan React JS. Instalasi React JS sudah tersedia dalam instalasi

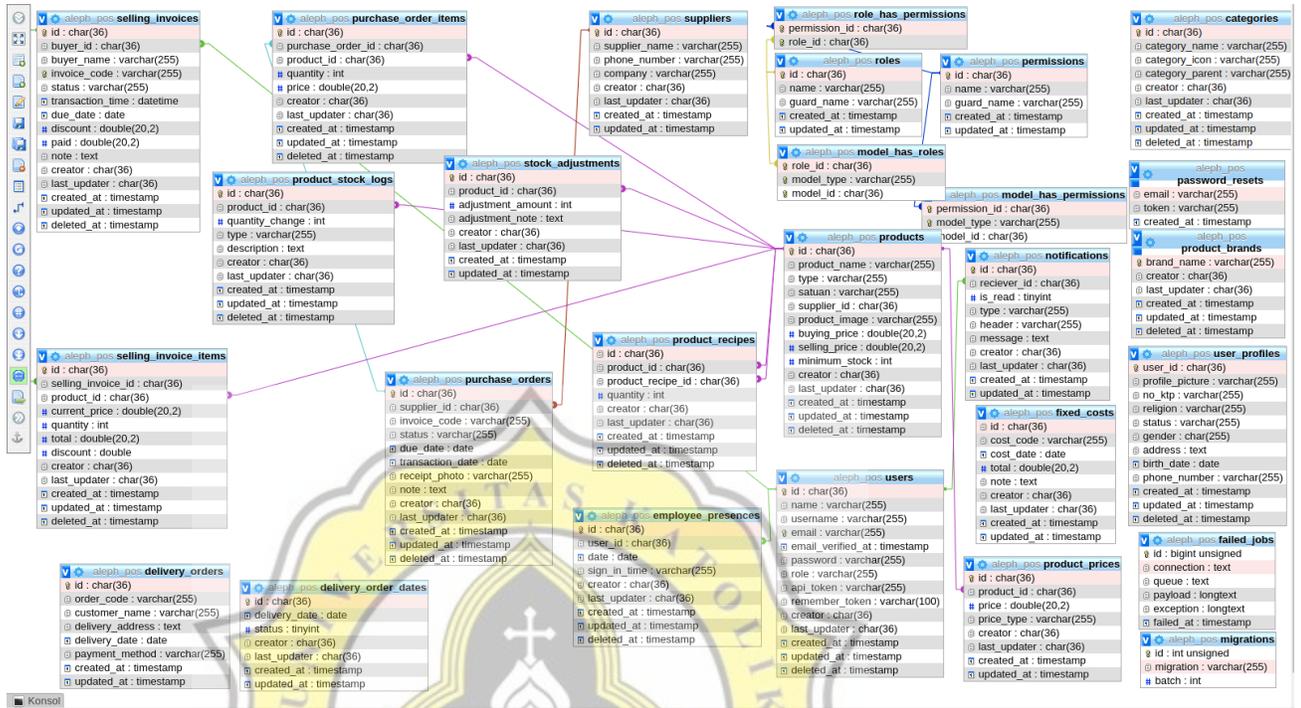
Laravel sehingga tidak diperlukan penambahan paket tertentu untuk menggunakan layanan ini.

Karena secara *default* Laravel menggunakan *blade* templating, maka penggunaan *blade templating* tidak akan ditinggalkan, hanya saja terdapat tambahan komponen dalam folder *js* dalam *resources* folder dalam Laravel yang akan dituliskan kode yang menggunakan basis React JS.

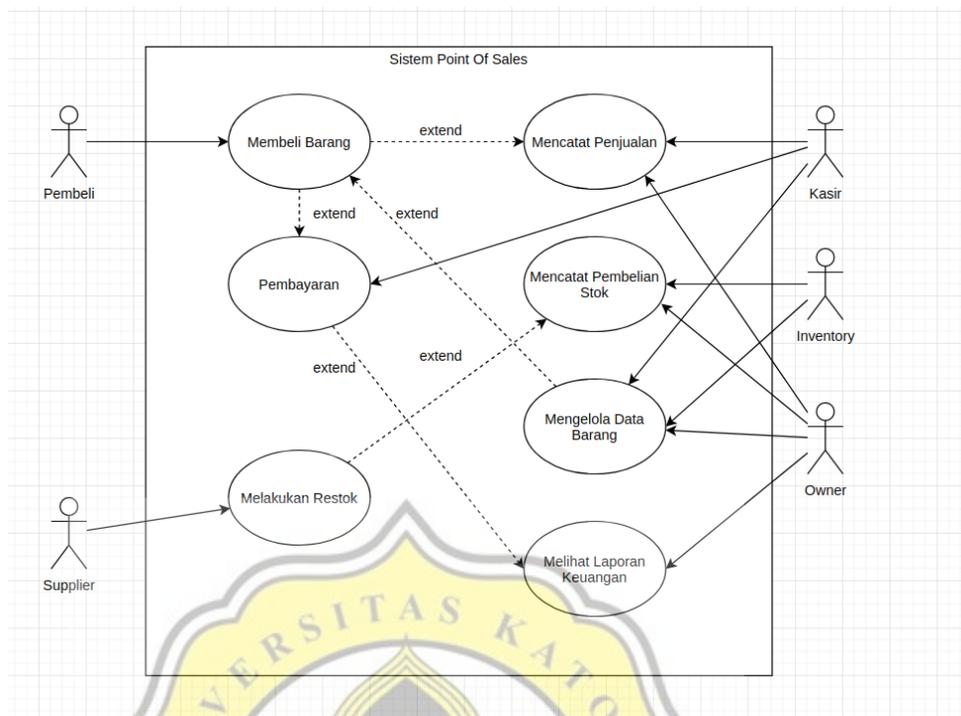
Komponen seperti tombol, form, tabel dan juga *sidebar* sudah tersedia ketika menggunakan Stisla Dashboard Admin sehingga tidak diperlukan lagi menulis ulang untuk membangun aplikasi dari awal. Yang diperlukan adalah memecah komponen-komponen menjadi kecil sehingga lebih mudah digunakan. Prinsip yang diaplikasikan adalah *DRY (Don't Repeat Yourself)* yang dimana mempermudah programmer supaya tidak perlu menuliskan kode yang sama berulang-ulang[17]. Implementasi prinsip ini sangat membantu dalam pembangunan aplikasi yang menggunakan teknik *RAD*.

4.2.3 Perancangan Struktur Basis Data

Struktur Basis Data berikut disusun untuk menyimpan data yang diperoleh dari sistem yang telah dibuat. Struktur ini memanfaatkan relasi antar tabel dan mampu menerima *query* mentah dari aplikasi. Proses optimalisasi data terjadi dalam aplikasi, optimalisasi yang terjadi memanfaatkan metode sekali *query* dan melakukan pengolahan data hasil *query* sebaik mungkin, sehingga mengurangi kebutuhan untuk *query* ke dalam basis data.



Gambar 4.1 Visual Basis Data Aplikasi



Gambar 4.3 Use Case Aplikasi

4.3 Tampilan Aplikasi

Sistem yang diimplementasikan pada toko retail ini menggunakan sistem *dashboard* dan *multi level auth*. Sistem ini memungkinkan perbedaan wewenang dalam melakukan input dan pembacaan data dengan lebih mudah. Sistem semacam ini memberikan kemudahan dalam pengawasan karena, setiap action yang penting seperti menginput penjualan akan tercatat dalam sistem siapakah yang melakukan input maupun edit terakhir.

4.3.1 Tampilan Halaman Order

Menu untuk melakukan pindah antar halaman ditampilkan pada *sidebar menu* yang mempermudah pemilihan halaman-halaman yang dikelompokkan berdasarkan kategorinya. Tampilan yang ada pada aplikasi ini dibuat *responsive* sehingga memberikan kemudahan akses ketika pengguna menggunakan aplikasi ini pada perangkat yang kecil seperti ponsel pintar.

Tampilan yang akan muncul ketika website diakses, akan mengarahkan pengguna ke dalam halaman pembelian order online. Tampilan order online adalah sebagai berikut:

#	NAMA PRODUK	TIPE	SATUAN	HARGA JUAL	ACTION
1	SPOOL KIPAS 18" COSMOS BULAT	SPOOL KIPAS	PCS	Rp. 130.000	+ Tambahkan
2	FLYBACK BSC 25-0231	FLYBACK	PCS	Rp. 50.000	+ Tambahkan
3	FLYBACK JF 0501-19587	FLYBACK	PCS	Rp. 57.500	+ Tambahkan
4	SPK TV SPK 613	SPK TV	PCS	Rp. 22.500	+ Tambahkan
5	FLYBACK FA 060 WISA	FLYBACK	PCS	Rp. 60.000	+ Tambahkan
6	FLYBACK F 0147	FLYBACK	PCS	Rp. 57.800	+ Tambahkan
7	FLYBACK 154-064D	FLYBACK	PCS	Rp. 35.000	+ Tambahkan
8	REMOTE TV WAYANG WTV-3	REMOTE TV	PCS	Rp. 22.500	+ Tambahkan
9	KABEL MIC + JEK 15 M HARKITA	KABEL MIC + JEK	PCS	Rp. 70.000	+ Tambahkan
10	REMOTE TV CHINA SUCKER 808	REMOTE TV	PCS	Rp. 22.500	+ Tambahkan

Data Pemesanan

Nama Pemesan:

Alamat Pengiriman:

Tanggal Pengiriman: 05 Oct 2020

Metode Pembayaran: CASH ON DELIVERY

Nama Produk	Harga	Qty.	Sub Total	Action
Grand Total				
				Rp. 0

PESAN SEKARANG

Gambar 4.4 Halaman Form Order Menu

Pada halaman ini, seorang pembeli dapat melakukan pembelian dengan menambahkan barang-barang yang akan dibeli dan mengisi data diri berupa nama pembeli, alamat pengantaran, tanggal pengantaran dan metode pembayaran.

Penampilan produk dalam menu ditampilkan berdasarkan tipe produk yang dapat dipilih oleh pengguna. Pengelompokan produk berdasarkan tipe produk ini mempermudah pencarian barang dalam satu kelompok tipe produk. Untuk mempermudah pengguna jika ingin melakukan pencarian berdasarkan nama, disediakan fitur pencarian dengan basis teks. Pencarian ini akan mencocokkan semua produk yang mengandung kata yang ditulis dalam kolom pencarian.

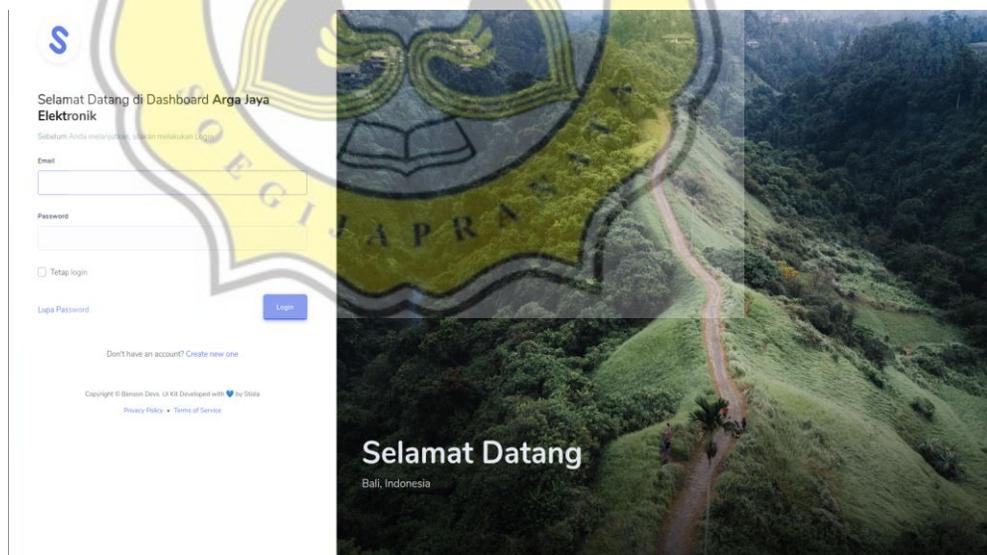
Eksekusi penambahan produk yang dilakukan tidak memerlukan muat ulang halaman, karena semua aksi dalam halaman ini mengimplementasikan

konsep AJAX dalam React JS. Untuk meringankan kerja server, penyimpanan sementara produk yang ditambahkan akan disimpan dalam *session*. Ketika input pemesanan sudah selesai dan berhasil, list tersebut akan dihancurkan dan datanya akan dihilangkan dari data *session*.

Dalam praktek aslinya, pengiriman berdasarkan tanggal masih belum terlalu membantu pengiriman barang, karena pada beberapa kasus pemesanan, pelanggan meminta untuk pesanan diantarkan pada pukul tertentu. Sehingga, penambahan fitur dalam *form order* dibutuhkan pada kemudian hari.

4.3.2 Halaman Login

Untuk melakukan login, pengguna harus tersambung dengan internet dan intranet untuk dapat mengakses domain <http://argajaya.alephpos.com/login>. Ketika halaman tersebut diakses, akan menampilkan halaman *login* untuk para karyawan maupun admin. Tampilan halaman login tersebut adalah sebagai berikut.

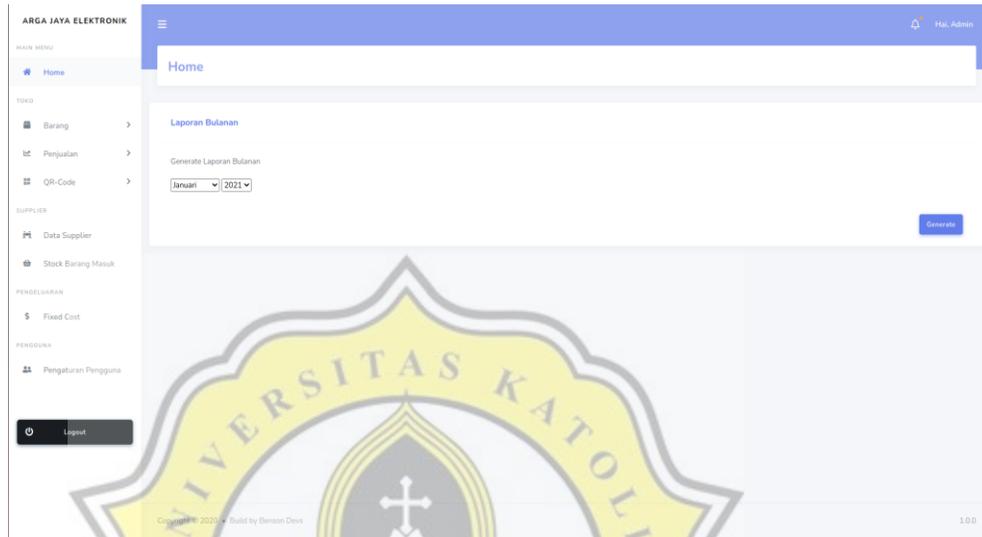


Gambar 4.5 Halaman Login

Halaman login berfungsi sebagai portal *autentikasi* pengguna yang memiliki akun/hak untuk mengakses *dashboard*.

4.3.3 Halaman Awal Dashboard

Halaman awal dashboard adalah yang akan dikunjungi seketika setelah aksi *login* berhasil dilakukan oleh pengguna. Tampilannya adalah sebagai berikut.



Gambar 4.6 Halaman Awal Dashboard

Pada halaman ini kita bisa langsung melakukan pembuatan laporan penjualan, pengeluaran dan *fixed cost*. Pada *sidebar*, kita dapat melihat berbagai menu dan pilihan halaman lain berdasarkan fiturnya yang dikelompokkan berdasarkan kategori bagian dalam toko.

4.3.4 Halaman Daftar Barang

Ketika dilakukan klik pada kategori barang, lalu memilih Daftar Barang, tampilan aplikasi yang muncul adalah berbagai macam daftar barang yang ada dalam aplikasi. Pengurutan barang adalah berdasarkan nama barang. Sehingga memudahkan untuk dilakukan pencarian.

ARGA JAYA ELEKTRONIK

Semua Produk

+ Tambahkan Produk

Cari

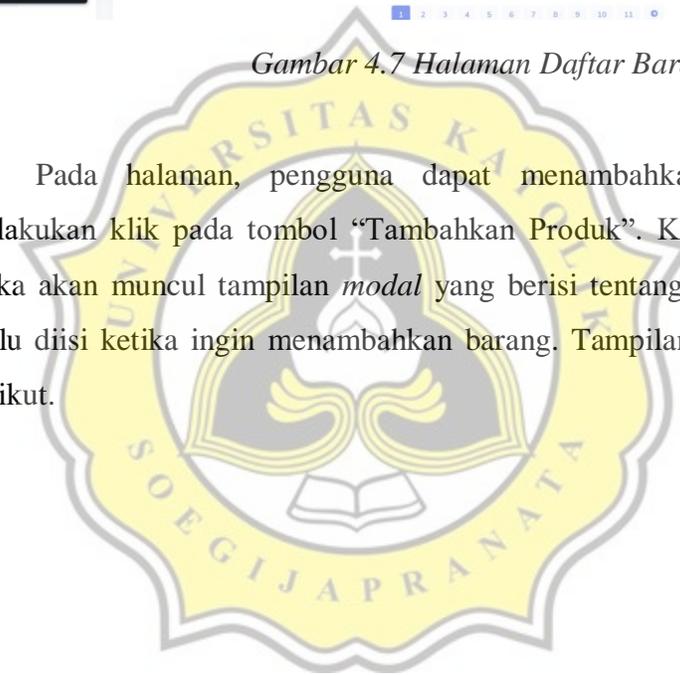
Tabul Produk

Nama Produk	Tipe	Satuan	Stock	Supplier	Belu	Juat	Action	Info
ADAPTOR 1 A/12V	ADAPTOR	PCS	0	ALJONG	Rp. 13.000	Rp. 25.000	Ubah X Hapus	Info
ADAPTOR 1 A/12V	ADAPTOR	PCS	-3	INDOSARI	Rp. 35.500	Rp. 55.000	Ubah X Hapus	Info
ADAPTOR 2 A/12V	ADAPTOR	PCS	0	ALJONG	Rp. 15.500	Rp. 30.000	Ubah X Hapus	Info
ADAPTOR 2 A/12V IN	ADAPTOR	PCS	0	INDOSARI	Rp. 15.000	Rp. 35.000	Ubah X Hapus	Info
ADAPTOR 3 A/12V	ADAPTOR	PCS	0	ALJONG	Rp. 26.000	Rp. 50.000	Ubah X Hapus	Info
ADAPTOR 4 A/9V	ADAPTOR	PCS	5	BANDUNG	Rp. 35.000	Rp. 60.000	Ubah X Hapus	Info
ADAPTOR 5 A/12V	ADAPTOR	PCS	0	ALJONG	Rp. 37.000	Rp. 65.000	Ubah X Hapus	Info
ADAPTOR 500 A/12V	ADAPTOR	PCS	0	INDOSARI	Rp. 30.500	Rp. 50.000	Ubah X Hapus	Info
AMPLI	AMPLI	PCS	-1	HARCO	Rp. 1.000.000	Rp. 1.500.000	Ubah X Hapus	Info

1 2 3 4 5 6 7 8 9 10 11

Gambar 4.7 Halaman Daftar Barang

Pada halaman, pengguna dapat menambahkan barang dengan melakukan klik pada tombol “Tambahkan Produk”. Ketika tombol diklik, maka akan muncul tampilan *modal* yang berisi tentang data apa saja yang perlu diisi ketika ingin menambahkan barang. Tampilannya adalah sebagai berikut.



Tambahkan Produk

×

Foto Produk

Choose File No file chosen

Nama Produk:

Tipe:

Satuan:

Supplier:

ARGA JAYA - ARGA JAYA

Jumlah Minimum: (ISI 0 JIKA PRODUK INI AKAN MEMILIKI RESEPI)

0

Harga Beli:

0

Harga Jual:

0

Close

+ Tambahkan

Gambar 4.8 Modal Tambah Barang

Untuk dapat memunculkan modal seperti di atas, source code yang dipakai adalah sebagai berikut.

```

• <div class="modal fade" id="addProductModal" tabindex="-
1" role="dialog" aria-labelledby="exampleModalLabel" aria-
hidden="true">
• <div class="modal-dialog modal-lg" role="document">
• <div class="modal-content">
• <div class="modal-header">
• <h5 class="modal-title">Tambahkan
Produk</h5>
• <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
• <span aria-hidden="true">x</span>
• </button>
• </div>
• <div class="modal-body">
• <form id="addProductForm"
action="{{ route('dashboard.products.add') }}" method="POST"
enctype="multipart/form-data">
• @csrf
•
•
• @include('components.modals.products.form_content')
• </form>
• </div>
• <div class="modal-footer">
• <button type="button" class="btn btn-
secondary" data-dismiss="modal">Close</button>
• <button form="addProductForm"
type="submit" class="btn btn-dark"><i class="fas fa-plus mr-
1"></i>Tambahkan</button>
• </div>
• </div>
• </div>
• </div>
• <script type="text/javascript">
• $(document).ready(function () {
• $('#addProductModal').on('hidden.bs.modal',
function () {
• //remove the backdrop
• $('.modal-backdrop').remove();
•
• })
•
• $('#addProductForm
input[name=product_image]').change((event) => {
• var imageUrl =
URL.createObjectURL(event.target.files[0])
• $('#addProductForm
img[name=image_preview]').attr('src', imageUrl)
• })
• })
• </script>

```

Gambar 4.9 Kode Modal Tambah Barang

Ketika tombol edit pada produk diklik, maka modal yang sama akan muncul dan diisi dengan data dari *JSON* yang didapatkan dari produk tersebut lalu dengan *jQuery* akan melakukan *fetching* ke dalam kolom-kolom pada modal.

```
• <div class="form-group">
• <label class="col-form-label">
• Foto Produk
• </label>
• <input type="file" class="form-control"
name="product_image">
• </div>
•
• <div class="form-group col-4">
• <img name="image_preview" class="img-square img-
fluid">
• </div>
•
• <div class="form-group">
• <label class="col-form-label">
• Nama Produk:
• </label>
• <input
• type="text"
• class="form-control"
• name="product_name"
• required>
• </div>
•
• <div class="form-group">
• <label class="col-form-label">
• Tipe:
• </label>
• <input
• id="type_input"
• type="text"
• class="form-control"
• name="type"
• autocomplete="off"
• required>
• </div>
•
• <div class="form-group">
• <label class="col-form-label">
• Satuan:
```

```

•         </label>
•         <input
•             type="text"
•             class="form-control"
•             name="satuan"
•             required>
•     </div>
•
•     <div class="form-group">
•         <label class="col-form-label">
•             Supplier:
•         </label>
•         <select class="form-control" name="supplier_id"
•             required>
•             <option disabled="disabled">Silakan Pilih
•             Supplier</option>
•             @foreach (App\Models\Supplier::all() as $key =>
•             $supplier)
•                 <option value="{{ $supplier->id }}">
•                     {{ $supplier->supplier_name}} -
•                     {{ $supplier->company }}
•                 </option>
•             @endforeach
•         </select>
•     </div>
•
•     <div class="form-group">
•         <label class="col-form-label">
•             Jumlah Minimum: (ISI 0 JIKA PRODUK INI AKAN
•             MEMILIKI RESEP!)
•         </label>
•         <input
•             type="number"
•             class="form-control"
•             name="minimum_stock"
•             value="0"
•             required>
•     </div>
•
•     <div class="form-group">
•         <label class="col-form-label">
•             Harga Beli:

```

```

O     </label>
O     <input
O         type="number"
O         class="form-control"
O         name="buying_price"
O         value="0"
O         required>
O </div>
O
O <div class="form-group">
O     <label class="col-form-label">
O         Harga Jual:
O     </label>
O     <input
O         type="number"
O         class="form-control"
O         name="selling_price"
O         value="0"
O         required>
O </div>
O
O

```

Gambar 4.10 Kode Konten Form

4.3.5 Halaman Riwayat Stok Barang

Pada halaman ini, pengguna dapat melihat dan memastikan jumlah stok barang. Ketika stok barang berada di bawah minimum, halaman ini akan menyajikan daftar barang-barang yang perlu untuk mendapatkan stok ulang.

Stok di Bawah Minimum

#	Nama Produk	Minimum Qty	Current Qty
1	FLYBACK FA 100 WISA	10	0
2	SPOOL KIPAS DISK FAN BESAR	10	0
3	REMOTE TV LED SAMSUNG	10	0
4	REMOTE TV CHINA VISERO 808	10	0
5	FLYBACK JF 0501-19959	10	0
6	KABEL MIC + JEK TOA - TOA	10	0
7	FLYBACK TFB 4132 ED	10	0
8	TRAFO 500 MA CT 18V BIG	10	0
9	KABEL MIC + JEK 5 M MAKITA	10	0
10	BOSTER ANTENA REMOT TD	10	0

Gambar 4.11 Tabel Barang Butuh Restock

Halaman ini mengimplementasikan fitur *AJAX* dari *React JS*. Kode untuk menampilkan halaman tersebut adalah sebagai berikut

```
• import React, { Component, Tooltip } from 'react';
• import Axios from 'axios';
• import Pagination from '../Pagination'
• import LoadingOverlay from 'react-loading-overlay';
•
O class MinimumStock extends Component {
O   constructor(props) {
O     super(props)
O
O     this.state = {
O       minimumStocks: [],
O       shownMinimumStocks: [],
O       startIndex: 0,
O
O       isLoading: false,
O     }
O     this.onChangePage = this.onChangePage.bind(this)
O   }
O   async componentDidMount() {
O     await this.populateMinimumStocks()
O   }
O   async populateMinimumStocks() {
O     await this.setState({
O       isLoading: true
O     })
O     await
Axios.get('/dashboard/products/stocks/low_stocks')
O       .then((response) => {
O         var minimumStocks =
Object.values(response.data)
O
O         console.log('Response Minimum')
O         console.log(minimumStocks)
O
O         this.setState({
O           minimumStocks: minimumStocks,
```

```

O         isLoading: false,
O         },
O console.log(this.state.minimumStocks)
O         })
O
O         this.onChangePage = this.onChangePage.bind(this)
O     }
O
O     async onChangePage(shownMinimumStocks, startIndex) {
O         await this.setState({
O             shownMinimumStocks: shownMinimumStocks,
O             startIndex: startIndex
O         })
O     }
O
O     render() {
O         return (
O             <div>
O                 <div className="card">
O                     <div className="card-header">
O                         <h2 className="font-weight-
bold">Stok di Bawah Minimum</h2>
O                     </div>
O
O                     <div className="card-body">
O                         <LoadingOverlay
O                         active={this.state.isLoading}
O                         spinner
O                         text='Memuat...'
O                     >
O                         <div className="table-
responsive">
O                             <table className="table
table-bordered table-md thead-dark">
O                                 <thead
O                                 className="thead-dark">
O                                     <tr>
O                                         <th
O                                         scope="col">#</th>
O                                     </tr>

```



```

O         </div>
O     </div>
O </div>
O     );
O }
O }
O
O
O     export default MinimumStock;

```

Gambar 4.12 Tabel Barang Butuh Restock

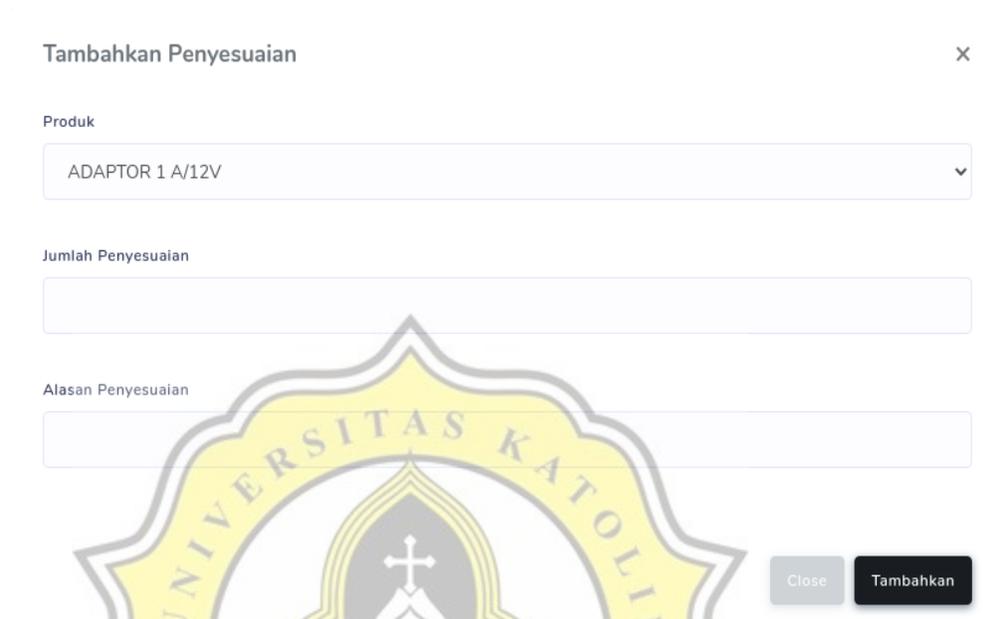
4.3.6 Halaman Penyesuaian Stok Barang

Pada halaman ini, pengguna diberikan wewenang untuk melakukan penyesuaian stok barang. Jika barang yang tertera di dalam aplikasi tidak sama dengan yang ada pada dunia nyata, halaman inilah yang akan dituju oleh pengguna. Dengan fitur ini, aplikasi dapat terus menyesuaikan dengan keadaan di dunia nyata. Sehingga mempermudah pemantauan stok barang secara langsung.

#	Tanggal Penyesuaian	Nama Barang	Jumlah Penyesuaian	Catatan	Action	Info
1	[14:02:47] 30 Oct 2020	KABEL MIC + JER 10 M ROKTANA	10	Stok lama	Edit Hapus	Info
2	[14:02:59] 30 Oct 2020	SPK TV SPK 5090	11	Stok lama	Edit Hapus	Info

Gambar 4.13 Tabel Penyesuaian Barang

Untuk melakukan penambahan penyesuaian barang, dapat dilakukan melalui *modal* untuk menambahkan penyesuaian yang memiliki tampilan sebagai berikut ini.



Tambahkan Penyesuaian ×

Produk

ADAPTOR 1 A/12V ▾

Jumlah Penyesuaian

Alasan Penyesuaian

Close Tambahkan

Gambar 4.14 Modal Penyesuaian Barang

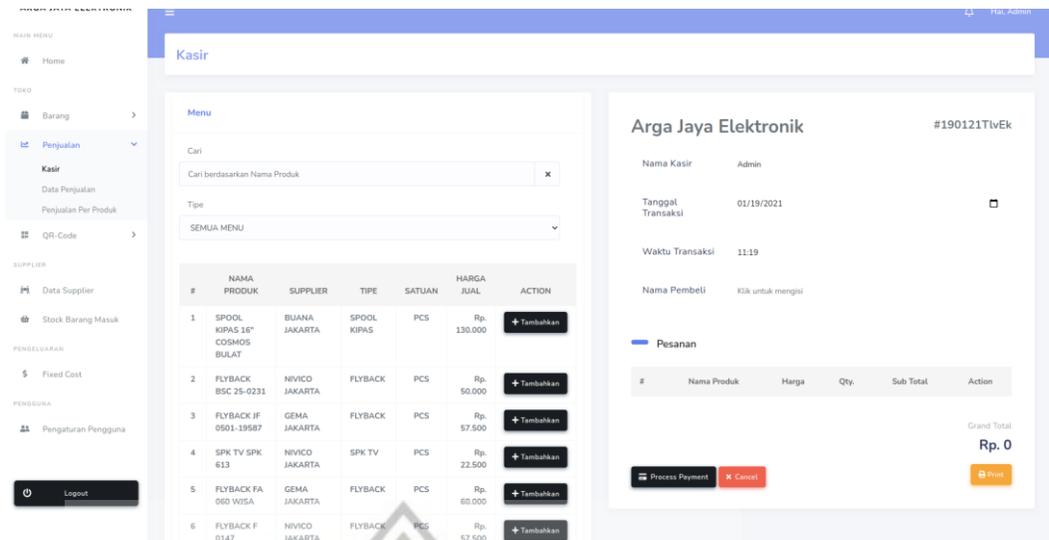
Pada modal ini pengguna dapat menambahkan produk yang memerlukan penyesuaian. Setelah itu, data penyesuaian akan tetap dicatat di dalam sistem. Pengguna yang melakukan pencatatan dan perubahan data akan tercatat nama penggunanya. Untuk melihat hal itu, pengguna dapat melakukan klik pada *icon* info seperti di bawah ini.

	Info	Info
ia	Dibuat Oleh: System Dibuat Pada: 13 detik yang lalu	
ia	Pengubah Terakhir: Perubahan Terakhir: 13 detik yang lalu	

Gambar 4.15 Info Penyesuaian Barang

4.3.7 Halaman Kasir

Pada halaman kasir, seorang kasir dapat melayani pembeli dengan menambahkan produk ke dalam list belanjaan lalu mengisikan nama pembeli tersebut. Nilai dari pembelian akan langsung ditampilkan dan dicatatkan sehingga pembayaran tidak perlu lagi disulitkan dengan menghitung total belanjaan. Terutama karena toko ini memiliki banyak sekali *reseller* yang sering membeli barang secara masif dan banyak



Gambar 4.16 Halaman Kasir

Karena dituntut untuk melayani pembeli secara cepat, pencarian produk harus cepat pula. Pencarian produk dalam halaman ini mengimplementasikan *AJAX* sehingga kasir tidak lagi perlu direpotkan dengan refresh halaman ketika mencari produk yang dibutuhkan. Karena halaman ini penuh dengan *AJAX*, sehingga tidak memerlukan refresh kecuali terdapat penambahan produk di dalam *database*.

4.3.8 Halaman Pre-Order

Halaman ini berisi pesanan-pesanan yang masuk ketika dilakukan *pre-order*. Dalam halaman ini berisi data mendasar yang dibutuhkan untuk pengiriman produk.

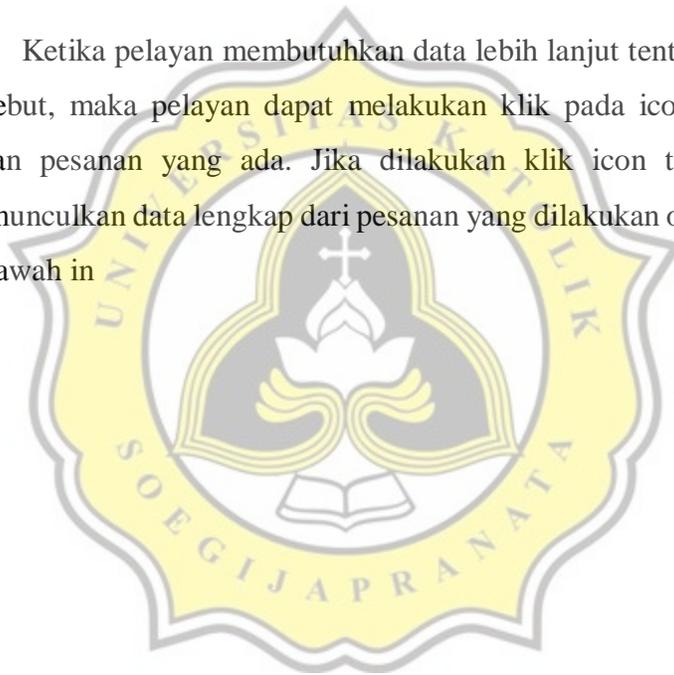
Daftar Pre-Order

Kode Order: mm/dd/yyyy

#	Kode PO	Nama Pemesan	Alamat Pengantaran	Tanggal Pengiriman	Metode Pembayaran	Total	Action
1	PO-06102020001	Tommy Hiananto	Jl	05 Oct 2020	COD	Rp. 1.025.000	👁️ 📄 🗑️
2	PO-13102020001	Benson	Jalan ampera	05 Oct 2020	COD	Rp. 575.000	👁️ 📄 🗑️

Gambar 4.17 Halaman Pre-Order

Ketika pelayan membutuhkan data lebih lanjut tentang isi dari pesanan tersebut, maka pelayan dapat melakukan klik pada icon mata pada pojok kanan pesanan yang ada. Jika dilakukan klik icon tersebut maka akan memunculkan data lengkap dari pesanan yang dilakukan oleh pembeli, seperti di bawah in



Detail Invoice ×

Kode PO:
PO-13102020001

Nama Pemesan:
Benson

Alamat Pengiriman:
Jalan ampera

Metode Pembayaran:
Cash On Delivery

#	Nama Produk	Harga	Qty.	Sub Total	Action
1	FLYBACK JF 0501-19587	Rp. 57.500	10	Rp. 575.000	

+ Tambahkan

Grand Total
Rp. 575.000

Close Simpan

Gambar 4.18 Modal Detil Pre-Order

4.3.9 Halaman Pengiriman Pre-Order

Dalam halaman ini, pengguna dapat mengatur jadwal pengiriman pemesanan yang akan dilakukan oleh pembeli.

#	Tanggal Pengiriman	Status	Action
1	05 Oct 2020	AKTIF	Lihat Pesanan Deactive Hapus

Gambar 4.19 Halaman Pengiriman Pre-Order

Ketika karyawan ingin menyiapkan barang yang perlu dikirim pada hari tersebut, pelayan dapat melakukan lihat pesanan. Maka akan memunculkan list barang apa saja yang perlu dipersiapkan terlebih dahulu. Tampilan list tersebut akan muncul sebagai berikut.

List Produk Tanggal 05 Oct 2020 ×

#	Nama Produk	Harga	Qty.	Sub Total
1	SWITCHING 30 A/12V	Rp. 205.000	5	Rp. 1.025.000
2	FLYBACK JF 0501-19587	Rp. 57.500	10	Rp. 575.000

Grand Total
Rp. 1.600.000

Gambar 4.20 Data Barang Pengiriman Pre-Order

4.3.10 Halaman Data Penjualan

Pada halaman data penjualan, pengguna dapat melihat semua rekam jejak penjualan yang terjadi dalam aplikasi, sehingga dapat memantau berapa banyak barang yang terjual dan omset yang sudah masuk ke dalam aplikasi.

Data Penjualan

Penjualan 2 Omzet Rp. 575.000 Item Terjual 2

Kode Nota 🔍 🗑️

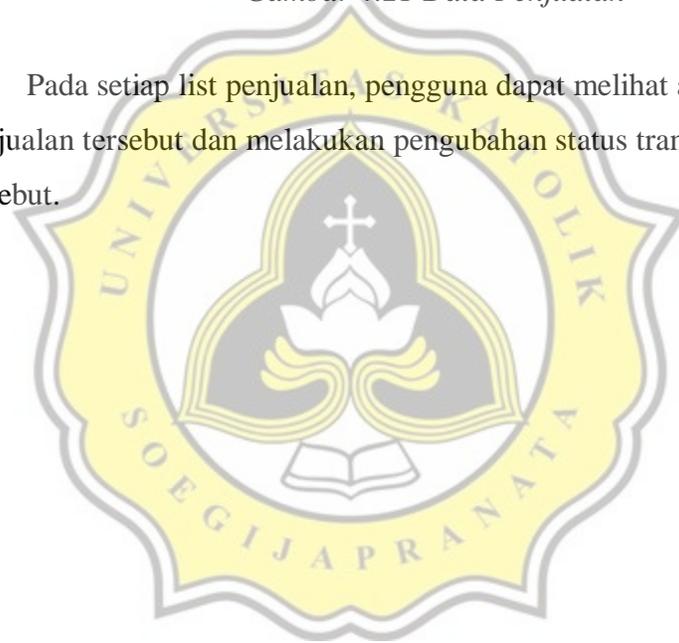
Daftar Penjualan

#	Kode Nota	Nama Pembeli	Grand Total	Status	Waktu Transaksi	Action
1	PO-13102020001	[Pre-Order] Benson	Rp. 575.000	LUNAS	30 Oct 2020 [07:32]	👁️ 🗑️
2	PO-06102020001	[Pre-Order] Tommy Hiananto	Rp. 1.025.000	BELUM LUNAS	TIDAK DIKETAHUI	👁️ 🗑️

Copyright © 2020 • Build by Benson Davs 1.0.0

Gambar 4.21 Data Penjualan

Pada setiap list penjualan, pengguna dapat melihat apa saja isi penjualan tersebut dan melakukan perubahan status transaksi dalam nota tersebut.



Invoice X

Invoice code:

Nama Pembeli:

Kasir:

Status:

Daftar Pesanan

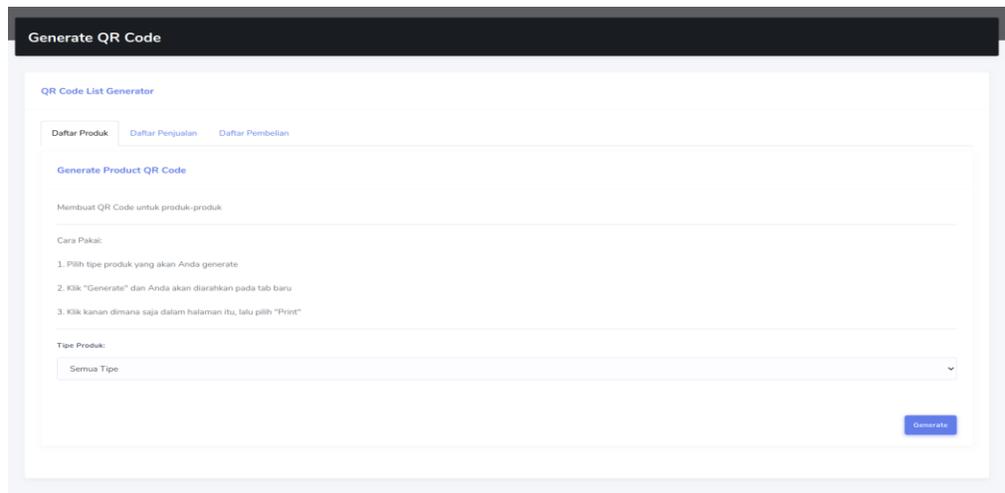
#	Nama Produk	Harga	Jumlah	Total
1	SWITCHING 30 A/12V	Rp. 205.000	5	Rp. 1.025.000

Gambar 4.22 Data Nota

Pada modal ini, pengguna yang melakukan *checking* dapat melakukan *cross-check* dengan pembelian yang sebenarnya terjadi, sehingga mengurangi kemungkinan salah beli ataupun *fraud*.

4.3.11 Halaman QR-Code

Halaman ini memberikan pengguna kemudahan untuk menghasilkan *QR-Code* untuk melakukan *checking* data dari sebuah produk tersebut. Ketika implementasi sistem di dunia nyata, fitur ini mempermudah pengecekan pada dunia nyata dan pada dunia maya.



Gambar 4.23 Halaman Generate QR-Code

Ketika kita melakukan *generate* QRCode maka aplikasi akan menampilkan halaman baru yang berisi sekumpulan kode QR yang dapat discan dan menghasilkan data yang dibutuhkan.



Gambar 4.24 Halaman Hasil QR-Code

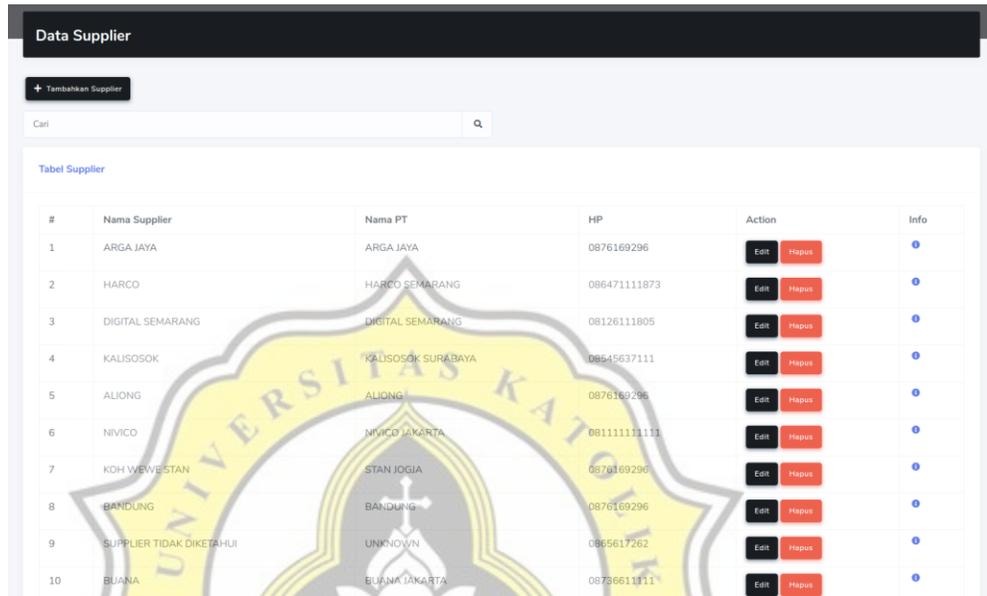
Jika dilakukan scan dengan ponsel pintar maka akan menghasilkan data terkait produk tersebut seperti berikut.

Deskripsi	Keterangan
Nama Produk:	SPK TV SPK 613
Tipe Produk:	SPK TV
Harga Produk:	Rp. 22.500
Stok Tersedia:	0
Gambar:	

Gambar 4.25 Halaman Hasil Scan QR-Code

4.3.12 Halaman Daftar Supplier

Pada halaman Daftar Supplier, pengguna dapat menambahkan siapa saja supplier dari setiap produk yang tersedia di dalam sistem. Pengelompokan produk juga tergantung pada supplier yang tersedia di dalam list di bawah ini.



#	Nama Supplier	Nama PT	HP	Action	Info
1	ARGA JAYA	ARGA JAYA	0876169296	Edit Hapus	•
2	HARCO	HARCO SEMARANG	086471111873	Edit Hapus	•
3	DIGITAL SEMARANG	DIGITAL SEMARANG	08126111805	Edit Hapus	•
4	KALISOSOK	KALISOSOK SURABAYA	08545637111	Edit Hapus	•
5	ALIONG	ALIONG	0876169296	Edit Hapus	•
6	NIVICO	NIVICO JAKARTA	081111111111	Edit Hapus	•
7	KOH WEWESTAN	STAN JOGJA	0876169296	Edit Hapus	•
8	BANDUNG	BANDUNG	0876169296	Edit Hapus	•
9	SUPPLIER TIDAK DIKETAHUI	UNKNOWN	0865617262	Edit Hapus	•
10	BUANA	BUANA JAKARTA	08736611111	Edit Hapus	•

Gambar 4.26 Halaman Data Supplier

Pada halaman data supplier, pengguna dapat menambahkan supplier melalui modal penambah supplier yang ditampilkan sebagai berikut.



Tambahkan Supplier X

Nama Supplier:

Nama PT:

Nomor HP:

Close Tambahkan

Gambar 4.27 Modal Tambah Supplier

4.3.13 Halaman Stok Barang Masuk

Pada halaman ini, barang yang masuk dapat disimpan dalam fitur yang tersedia. Fitur ini memperbolehkan penggunanya untuk mencatat nota fisik menjadi nota digital dan mengelompokkan berdasarkan hari dimana nota itu didapatkan.

Kode Invoice	Nama Supplier	Total Harga	Action	Info

#	Nama Produk	Jenis Barang	Jumlah Barang	Harga Beli	Sub Total	Aksi

Gambar 4.28 Halaman Stok Barang Masuk

4.3.14 Halaman Fixed Cost

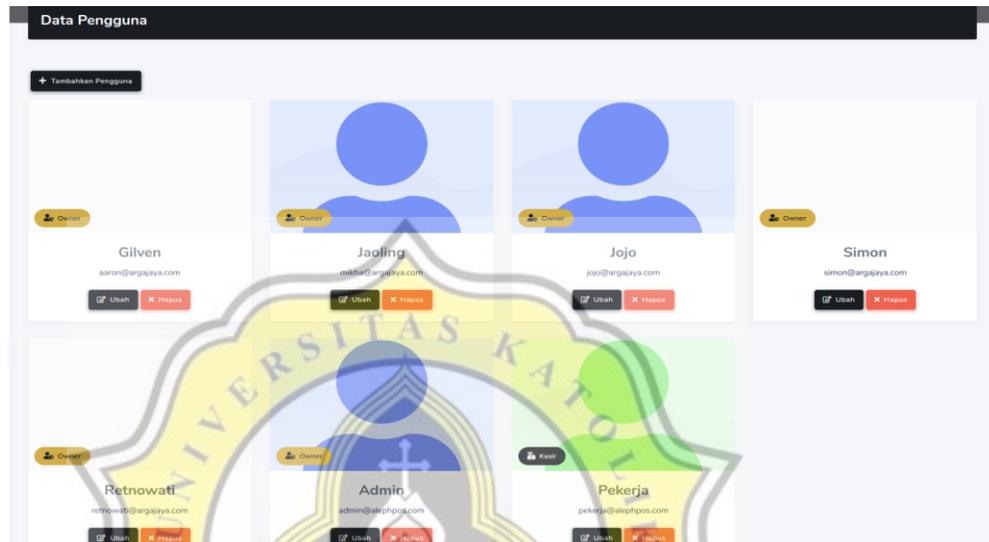
Halaman ini berfungsi untuk menampilkan pengeluaran yang terjadi dan termasuk dalam fixed cost dari perusahaan yang memakai aplikasi ini.

Kode Fixed Cost	Tanggal Pengeluaran	Total	Keterangan	Action	Info
11030020	30 Oct 2020	Rp. 500.000	Internet		

Gambar 4.29 Halaman Fixed Cost

4.3.15 Halaman Data Pengguna

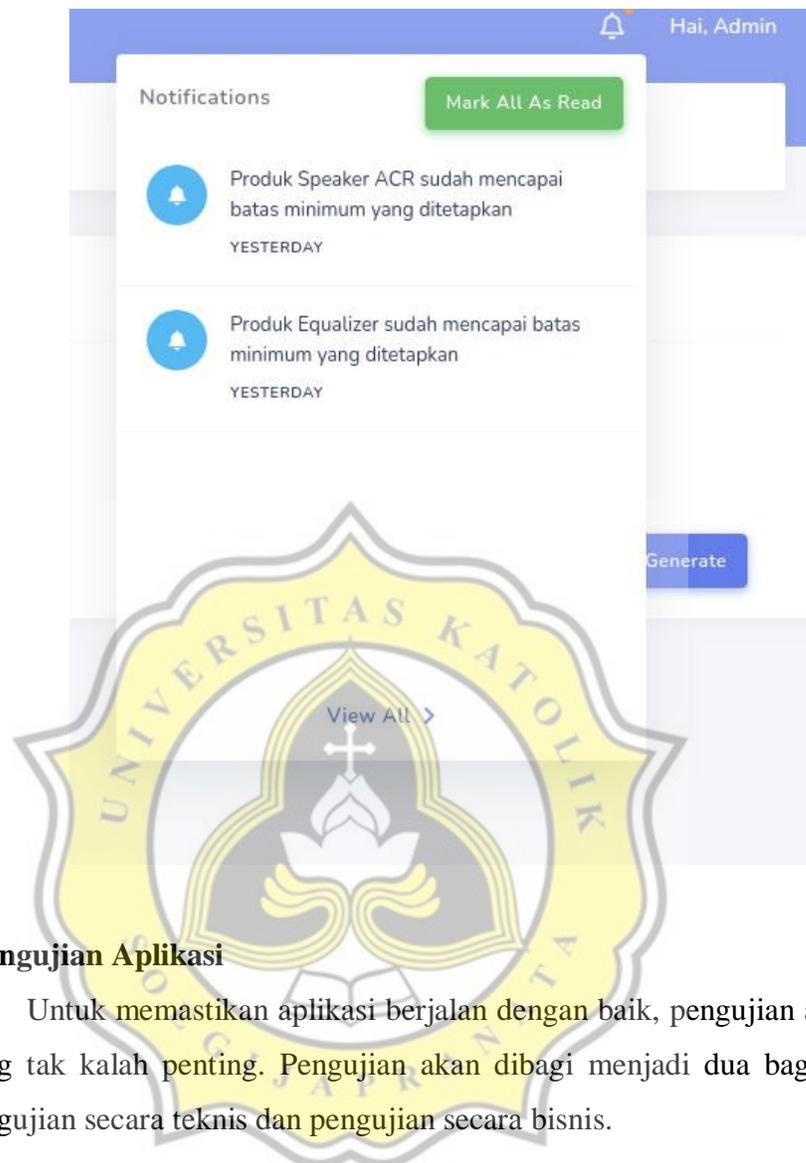
Pada halaman ini, pengguna yang tersimpan datanya dalam aplikasi akan tertampil pada halaman ini. *Role* yang ada dalam aplikasi ini adalah *Owner*, *Inventori* dan *Kasir*.



Gambar 4.30 Halaman Pengguna

4.3.15 Fitur Notifikasi

Pada bagian ini, aplikasi akan memberikan pemberitahuan kepada admin dan atau kasir akan barang-barang yang sudah mencapai batas minimum dari yang ditentukan. Setiap kali ada barang terjual atau berkurang di sistem, sistem akan melakukan pengecekan apakah jumlah barang yang tersisa masih di atas jumlah minimum atau tidak. Jika iya, maka akan membuat notifikasi yang diberikan pada pengguna seperti di bawah ini.



4.4 Hasil Pengujian Aplikasi

Untuk memastikan aplikasi berjalan dengan baik, pengujian adalah hal yang tak kalah penting. Pengujian akan dibagi menjadi dua bagian, yaitu pengujian secara teknis dan pengujian secara bisnis.

4.4.1 Hasil Pengujian Teknis

Dalam pengujian teknis akan mengevaluasi apa saja yang terjadi secara teknis, apakah sudah berjalan dengan baik sesuai dengan harapan ketika masa perencanaan fitur. Setiap fitur akan memiliki *sub* yang berisikan fungsionalitas secara teknisnya untuk diuji.

Tabel 4.1 Daftar Fitur dan Jenis Pengujian

	FITUR	JENIS PENGUJIAN	HASIL PENGUJIAN
1	Login	Aksi masuk	Berjalan dengan baik
2	Order Form	Menambahkan produk ke dalam keranjang	Berjalan dengan baik
		Menghapus produk dari keranjang	Berjalan dengan baik
		Memesan	Koneksi ke whatsapp kadang bermasalah
3	Pengelolaan Produk	Melakukan penambahan produk baru	Berjalan dengan baik
		Melakukan penghapusan produk	Berjalan dengan baik
		Melakukan pengubahan produk	Preview foto sebelumnya kadang tidak muncul
		Melakukan pembacaan produk menggunakan QR-Code	Berjalan dengan baik
4	Kasir	Menambahkan produk ke dalam daftar belanja	Berjalan dengan baik
		Menghapus produk dari daftar belanja	Berjalan dengan baik
		Melakukan penyimpanan data belanja	Berjalan dengan baik
		Melakukan print out nota belanja	Berjalan dengan baik
5	Supply Barang	Menambahkan Supplier	Berjalan dengan baik
		Mengubah Supplier	Berjalan dengan baik
		Menghapus Supplier	Berjalan dengan baik
		Menambahkan data nota supply	Notifikasi nota belum diupload belum ada
		Menghapus data nota supply	Berjalan dengan baik
		Melakukan upload foto nota supply	Berjalan dengan baik
6	User	Melakukan penambahan pengguna	Berjalan dengan baik
		Melakukan penghapusan pengguna	Berjalan dengan baik
		Melakukan pengubahan pengguna	Berjalan dengan baik

4.4.2 Hasil Pengujian Bisnis

Pengujian bisnis dilakukan menggunakan metode wawancara, yang dimana pemilik toko retail diwawancarai akan apa saja dampak yang terjadi setelah implementasi sistem. Selain wawancara, pengamatan secara langsung akan reaksi dari pengguna sistem adalah bagian dari data diambil dalam proses penelitian ini.

Dalam implementasinya, sistem ini secara mendasar dapat mempermudah penggunaannya untuk memantau apa saja yang terjadi lebih baik dan lebih mudah. Dari permasalahan yang ada seperti pemantauan stok, menjadi lebih mudah setelah sistem ini sudah diimplementasi. Namun kendala besar yang terjadi ketika implementasi adalah sulitnya melakukan pelatihan kepada pengguna yang belum terbiasa dalam menggunakan teknologi informasi yang dapat dikategorikan baru di era ini.

Pelatihan dilakukan kepada dua karyawan, setelah kedua karyawan ini dapat mengoperasikan aplikasi ini, kedua karyawan ini akan diminta untuk mengajarkan kepada dua karyawan lain, sehingga pelatihan dapat berjalan lebih efektif tanpa harus mengganggu waktu kerja di dalam toko. Karena menurut penuturan dari karyawan toko ini, mereka merasa lebih mudah mencerna kata-kata yang datang dari sesama karyawan. Hal ini dimungkinkan karena kesamaan latar belakang dan pola pikir, sehingga lebih mudah menyampaikan pesan dan pemahaman kepada sesama karyawan.

Dikarenakan jumlah produk yang terlalu banyak, implementasi sistem ini dimanfaatkan untuk produk yang sifatnya krusial dan memiliki tingkat pembelian yang tinggi. Sehingga implementasi sistem ini akan berjalan seiring waktu. Implementasi yang sepenuhnya belum terlaksana karena kendala ini. Namun secara garis besar, kehadiran sistem ini mempermudah pekerjaan karyawan yang ada di sana.

Penggunaan *gadget* oleh karyawan dapat lebih terkendali karena karyawan cenderung memiliki tugas untuk melakukan input data. Karena pemantauan dapat dilakukan oleh pemilik, sehingga kemungkinan untuk karyawan mangkir dari tugas menjadi lebih kecil. Hal ini dapat dipantau dari banyaknya data yang diinput oleh karyawan dan pencocokannya dengan tugas yang sudah diberikan oleh pemilik toko. Namun, karena penggunaan sistem ini belum sepenuhnya melayani bisnis, sehingga pemantauan hanya terjadi ketika karyawan mendapatkan pekerjaan pada sistem daring.

Dari sudut pandang pemilik toko, dengan adanya kemudahan ini, pemantauan stok barang yang memiliki arus jual beli tinggi menjadi lebih mudah. Pemilik toko merasa masih membutuhkan waktu untuk membiasakan diri dengan kehadiran sistem ini.

Selain kemudahan itu, implementasi *QR-Code* mempermudah pembeli untuk melakukan pengecekan data seputar sebuah barang. Rata-rata pembeli yang datang, menginginkan mengetahui informasi seputar produk yang meliputi harga, jumlah dalam stok dan gambar bentuk barang tersebut. Kemudahan ini ternyata memberikan nilai lebih untuk toko retail sebagai toko yang lebih futuristik dan mengikuti perkembangan zaman. Pembeli yang datang merasa lebih tertarik ketika disuguhkan dengan fitur ini.

Pada fitur toko *online* yang tersedia dalam sistem ini, ditemukan bahwa pengguna lebih nyaman menggunakan sistem pemesanan via *chat* melalui aplikasi *Whatsapp* kepada pemilik toko. Dikarenakan pembeli lebih merasa yakin dengan pesannya ketika mendapatkan respon langsung dari pemilik toko dibanding hanya pesan singkat pada aplikasi. Meskipun implementasi toko online ini terintegrasi dengan *Whatsapp*, pembeli lebih memilih untuk melakukan kontak secara langsung. Hal ini juga dipengaruhi oleh bertambahnya birokrasi ketika melakukan pemesanan pada toko online yang dimana pembeli lebih suka dan lebih merasa dimanjakan dengan pelayanan melalui *chat*.

Selain itu, fitur toko *online* dirasa oleh pembeli lebih tidak meyakinkan mereka karena pembeli tidak dapat melihat barang secara langsung. Dikarenakan toko retail ini menjual barang-barang elektronik yang dimana pembelinya akan merasa lebih yakin terhadap produk yang akan dibeli ketika sudah melihat, menyentuh dan mencoba produk secara langsung terlebih dahulu sebelum membeli.

