



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN APLIKASI SISTEM INVENTORI
BARANG BERBASIS WEB DENGAN TEKNOLOGI *BARCODE*
MENGUNAKAN *FRAMEWORK* LARAVEL DAN REACTJS**

TUGAS AKHIR

JULKARNAIN

0110220250

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN APLIKASI SISTEM INVENTORI
BARANG BERBASIS WEB DENGAN TEKNOLOGI *BARCODE*
MENGUNAKAN *FRAMEWORK* LARAVEL DAN REACTJS**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

JULKARNAIN

0110220250

STT - NF

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



Nama : Julkarnain
NIM : 0110220250

Depok, 23 Juli 2024



Julkarnain

STT - NF

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Julkarnain

NIM : 0110220250

Program Studi : Teknik Informatika


Judul Skripsi : Rancang Bangun Aplikasi Sistem Inventori Barang Berbasis Web dengan Teknologi Barcode Menggunakan *Framework* Laravel dan ReactJS


Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing

Penguji


(Efrizal Zaida, S.Kom., M.M., M.Kom.)


(Henry Saptono, S.Si., M.Kom.)

Ditetapkan di : Depok

Tanggal : 23 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tiffany Nabarian selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Zaki Imaddudin selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama perkuliahan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Efrizal Zaida selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam

penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 23 Juli 2024



Penulis



STT - NF

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Julkarnain

NIM : 0110220250

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*) atas karya ilmiah saya yang berjudul :

Rancang Bangun Aplikasi Sistem Inventori Barang Berbasis Web dengan Teknologi Barcode Menggunakan *Framework* Laravel dan ReactJS beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 23 Juli 2024

Yang Menyatakan



Julkarnain

ABSTRAK

Nama : Julkarnain
NIM : 0110220250
Program Studi : Teknik Informatika
Judul : Rancang Bangun Aplikasi Sistem Inventori Barang Berbasis Web dengan Teknologi Barcode Menggunakan *Framework* Laravel dan ReactJS

Pelaku usaha di Indonesia sebagian besar masih mengandalkan pencatatan keuangan dan stok secara manual. Beberapa kekurangan dari pengelolaan stok secara manual yaitu kurangnya efisiensi dalam pencatatan dan penyimpanan dokumen yang tidak rapi. Mengelola stok secara manual menggunakan media kertas atau dokumen dapat menurunkan efisiensi pembaruan stok yang dilakukan dalam periode tertentu. Tujuan dari penelitian adalah membuat aplikasi sistem inventori berbasis web yang dapat digunakan untuk mencatat stok barang dengan menggunakan *framework* Laravel dan ReactJS. Metode pengembangan perangkat lunak yang digunakan adalah metode Waterfall. Hasil dari penelitian ini adalah aplikasi yang dapat digunakan pengguna untuk mencatat data stok barang. Aplikasi ini juga dapat berjalan dengan baik dan responsif di perangkat *dekstop* maupun *mobile*.

Kata kunci : Inventori, Laravel, Waterfall

STT - NF

ABSTRACT

Name : Julkarnain
NIM : 0110220250
Study Program : Informatics
Title : *Design and Build a Web-Based Goods Inventory System
Application with Barcode Technology Using the Laravel
Framework and ReactJS*

Most business actors in Indonesia still rely on manual financial and stock recording. Some of the disadvantages of manual stock management are the lack of efficiency in recording and sloppy document storage. Managing stock manually using paper or documents can reduce the efficiency of stock updates carried out within a certain period. The aim of the research is to create a web-based inventory system application that can be used to record stock using the Laravel and ReactJS frameworks. The software development method used is the Waterfall method. The result of this research is an application that users can use to record stock data. This application can also run well and is responsive on desktop and mobile devices.

Keywords : *Inventory, Laravel, Waterfall*

STT - NF

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
<i>ABSTRACT</i>	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian.....	2
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	3
BAB II KAJIAN LITERATUR	5
2.1 Tinjauan Pustaka	5
2.1.1 Sistem Inventori	5
2.1.2 <i>Barcode</i>	5
2.1.3 <i>Framework</i>	8
2.1.4 Laravel.....	8
2.1.5 ReactJS.....	8

2.1.6	SPA	9
2.1.7	MySQL.....	10
2.1.8	<i>Waterfall</i>	10
2.1.9	UML.....	11
2.1.10	Black Box Testing.....	12
2.1.11	Skala Likert.....	12
2.2	Penelitian Terkait	14
BAB III METODOLOGI PENELITIAN		16
3.1	Tahapan Penelitian	16
3.2	Rancangan Penelitian	18
3.2.1	Jenis Penelitian.....	18
3.2.2	Metode Analisis Data.....	18
3.2.3	Metode Pengumpulan Data.....	19
3.2.4	Metode Pengujian.....	19
3.2.5	Metode Implementasi dan Evaluasi.....	19
3.2.6	Lingkungan Pengembangan.....	20
BAB IV IMPLEMENTASI DAN EVALUASI.....		21
4.1	Perencanaan Sistem.....	21
4.1.1	Istilah Baku	23
4.1.2	<i>User Requirement</i>	23
4.2	Pemodelan Sistem	25
4.2.1	Desain sistem	25
4.2.2	Arsitektur Sistem.....	44
4.2.3	Antarmuka Sistem.....	44
4.3	Implementasi	49

4.4.1	Implementasi Kode	49
4.4.2	Implentasi Antarmuka.....	54
4.4	Pengujian Sistem	61
4.5.1	Hasil <i>Black Box Testing</i>	61
4.5.2	<i>User Acceptance Test</i>	65
4.5.3	Hasil Kuesioner.....	67
BAB V KESIMPULAN DAN SARAN.....		70
5.1	Kesimpulan.....	70
5.2	Saran.....	70
DAFTAR PUSTAKA		72
LAMPIRAN.....		74
Lampiran 1: Hasil Black Box Testing.....		74
Lampiran 2: Hasil UAT		78
Lampiran 3: Hasil Kuesioner		80

STT - NF

DAFTAR GAMBAR

Gambar 1 : Contoh <i>barcode</i> 39.....	6
Gambar 2 : Contoh <i>barcode</i> UPC-A.....	6
Gambar 3 : Contoh <i>barcode</i> UPC-E	7
Gambar 4 : Contoh <i>barcode</i> EAN-8	7
Gambar 5 : Contoh <i>barcode</i> EAN-13	7
Gambar 6 : Tahapan Penelitian	16
Gambar 7: <i>Use Case Diagram</i>	25
Gambar 8: <i>Activity Diagram</i> mendaftar sebagai <i>user</i>	33
Gambar 9: <i>Activity Diagram</i> login sebagai <i>user</i>	34
Gambar 10: <i>Activity Diagram</i> menambah data barang	35
Gambar 11: <i>Activity Diagram</i> mengelola barang masuk	35
Gambar 12: <i>Activity Diagram</i> mengelola barang keluar.....	36
Gambar 13: <i>Activity Diagram</i> mencari data barang.....	37
Gambar 14: <i>Activity Diagram</i> mengelola akun.....	37
Gambar 15: <i>Activity Diagram</i> scan <i>barcode</i>	38
Gambar 16: <i>Sequence Diagram</i> Register	39
Gambar 17: <i>Sequence Diagram</i> Login	39
Gambar 18: <i>Sequence Diagram</i> tambah barang.....	40
Gambar 19: <i>Sequence Diagram</i> mengelola barang masuk	40
Gambar 20: <i>Sequence Diagram</i> mengelola barang keluar.....	41
Gambar 21: <i>Sequence Diagram</i> mencari barang	41
Gambar 22: <i>Sequence Diagram</i> mengelola akun.....	42
Gambar 23: <i>Sequence Diagram</i> scan barang	42
Gambar 24: <i>Class Diagram</i>	43
Gambar 25: Arsitektur Sistem.....	44
Gambar 26: <i>Wireframe</i> Login	44
Gambar 27: <i>Wireframe</i> Register	45
Gambar 28: <i>Wireframe</i> Data Barang	45
Gambar 29: <i>Wireframe</i> Detail Barang	45

Gambar 30: <i>Wireframe</i> Tambah Barang	46
Gambar 31: <i>Wireframe</i> Data Distributor	46
Gambar 32: <i>Wireframe</i> Tambah Distributor	46
Gambar 33: <i>Wireframe</i> Detail Distributor	47
Gambar 34: <i>Wireframe</i> Barang Masuk	47
Gambar 35: <i>Wireframe</i> Detail Barang Masuk	47
Gambar 36: <i>Wireframe</i> Data Barang Keluar.....	48
Gambar 37: <i>Wireframe</i> Detail Barang Keluar	48
Gambar 38: <i>Wireframe</i> Akun.....	48
Gambar 39: Skema Tabel <i>Shop</i>	49
Gambar 40: Skema Tabel <i>Product Unit</i>	50
Gambar 41: Skema Tabel <i>Product Category</i>	50
Gambar 42: Skema Tabel <i>Product</i>	51
Gambar 43: Schema Tabel <i>Distributor</i>	51
Gambar 44: Skema Tabel <i>Product Out</i>	52
Gambar 45: Skema Tabel <i>Product In</i>	52
Gambar 46: Skema Tabel <i>Product In Detail</i>	53
Gambar 47: Skema Tabel <i>Product Out Detail</i>	53
Gambar 48: Skema Tabel <i>User</i>	54
Gambar 49: Halaman <i>Login</i>	54
Gambar 50: Halaman <i>Register</i>	55
Gambar 51: Halaman data barang.....	55
Gambar 52: Halaman Tambah Barang.....	56
Gambar 53: Halaman Data Distributor	56
Gambar 54: Halaman Tambah Distributor.....	57
Gambar 55: Halaman Data Barang Masuk	57
Gambar 56: Halaman Tambah Barang Masuk.....	58
Gambar 57: Halaman Detail Barang Masuk	58
Gambar 58: Halaman Data Barang Keluar	59
Gambar 59: Halaman Tambah Barang Keluar.....	59
Gambar 60: Halaman Detail Barang Keluar	60

Gambar 61: Halaman Akun 60
Gambar 62: Halaman *Scan Barcode* 61



DAFTAR TABEL

Tabel 1: Penelitian Terkait.....	15
Tabel 2: Fitur Penelitian Sebelumnya.....	21
Tabel 3: Perbandingan Aplikasi.....	22
Tabel 4: Hasil Perbandingan Penelitian Sebelumnya.....	22
Tabel 5: Hasil Perbandingan Aplikasi.....	23
Tabel 6: Istilah Baku.....	23
Tabel 7: <i>User Requirement</i>	24
Tabel 8: Kategori Pengguna.....	24
Tabel 9: Kategori Menu Aplikasi.....	25
Tabel 10: <i>Use Case</i> mendaftar sebagai <i>user</i>	26
Tabel 11: <i>Use Case</i> login sebagai <i>user</i>	27
Tabel 12: <i>Use Case</i> menambah data barang.....	28
Tabel 13: <i>Use Case</i> mengelola barang masuk.....	29
Tabel 14: <i>Use Case</i> mengelola barang keluar.....	30
Tabel 15: <i>Use Case</i> mencari data barang.....	31
Tabel 16: <i>Use Case</i> mengelola akun.....	32
Tabel 17: <i>Use Case scan barcode</i> barang.....	33
Tabel 18: Hasil <i>Black Box Testing</i>	64

STT - NF

BAB I

PENDAHULUAN

1.1 Latar belakang

Meskipun saat ini telah memasuki era teknologi yang maju, masih banyak pengusaha yang menjalankan manajemen keuangan bisnis mereka secara konvensional. Menurut penelitian dari OCBC Indonesia dalam Business Fitness Index tahun 2023, sebanyak 80% dari pelaku usaha di Indonesia masih mengandalkan pencatatan keuangan dan stok secara manual [1].

Beberapa kekurangan dari pengelolaan stok secara manual yaitu kurangnya efisiensi dalam pencatatan dan penyimpanan dokumen yang tidak rapi. Mengelola stok secara manual menggunakan media kertas atau dokumen dapat menurunkan efisiensi pembaruan stok yang dilakukan dalam periode tertentu. Hal ini terjadi karena semua dilakukan dengan cara tradisional. Jika dokumen hilang atau ada kesalahan pencatatan, maka seluruh data harus diperbarui satu per satu, sehingga memperlambat dan mengganggu kegiatan penjualan maupun pembelian.[2].

Berdasarkan laporan We Are Social, per Januari 2023, jumlah pengguna internet di Indonesia mencapai 213 juta orang, yang merupakan 77% dari total populasi Indonesia yang sebanyak 276,4 juta orang pada awal tahun ini. Jumlah ini menunjukkan peningkatan 5,44% dibandingkan tahun sebelumnya, di mana pada Januari 2022, pengguna internet di Indonesia tercatat sebanyak 202 juta orang.[3].

Dengan meningkatnya jumlah pengguna internet di Indonesia, yang mencapai 213 juta orang pada Januari 2023, adopsi teknologi digital dalam berbagai aspek bisnis menjadi semakin relevan. Salah satu area yang dapat memanfaatkan perkembangan ini adalah pengelolaan stok. Beralih dari metode manual yang kurang efisien dan rentan terhadap kesalahan, ke sistem digital yang lebih

terorganisir dan aman serta dapat meningkatkan efisiensi operasional secara signifikan. Integrasi teknologi ini menjadi semakin penting seiring dengan peningkatan akses internet di kalangan masyarakat Indonesia.

Berdasarkan latar belakang masalah di atas, maka dalam tugas akhir ini penulis ingin mengembangkan sebuah aplikasi sistem inventori berbasis web yang dapat digunakan oleh pelaku usaha untuk membantu mengelola produk mereka dengan teknologi *barcode* yang memungkinkan pencatatan stok barang menjadi lebih cepat.

Adapun judul penelitian yang akan penulis lakukan adalah Rancang Bangun Aplikasi Sistem Inventori Produk Berbasis Web dengan Teknologi *Barcode* menggunakan *framework* Laravel dan ReactJS.

1.2 Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan pada latar belakang, maka akan dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimanakah rancangan aplikasi sistem inventori yang dapat digunakan untuk manajemen stok barang menggunakan metode *waterfall*?
2. Apakah aplikasi dapat berjalan dengan baik pada perangkat *desktop* maupun *mobile* dari segi tampilan dan fitur?

1.3 Tujuan dan Manfaat Penelitian

Berdasarkan beberapa permasalahan yang telah dijelaskan pada rumusan masalah, maka akan didapatkan tujuan sebagai berikut:

1. Merancang aplikasi sistem inventori yang dapat digunakan oleh pengguna dalam manajemen stok barang.
2. Membuat fitur dan tampilan yang responsif sehingga dapat berjalan di perangkat *desktop* maupun *mobile*.

Adapun manfaat sebagai berikut:

1. Pengguna dapat mengelola stok barang dengan mudah.
2. Pengguna dapat membuat laporan transaksi barang masuk maupun keluar.
3. Aplikasi sistem inventori ini dapat digunakan di perangkat *desktop* maupun *mobile*.

1.4 Batasan Masalah

Dari penjelasan tujuan dan manfaat penelitian, maka batasan masalah pada penelitian tugas akhir ini adalah:

1. Aplikasi yang dibuat berbasis web dengan menggunakan framework Laravel dengan versi 11 dan tidak mengembangkan aplikasi versi *mobile*.
2. Menggunakan *webserver* Apache versi 2.4 untuk menjalankan aplikasi yang dibuat.
3. Aplikasi yang dibuat hanya berfokus pada fitur mengelola stok produk dan pencatatan laporan transaksi barang masuk maupun keluar.

1.5 Sistematika Penulisan

Dalam penulisan Tugas Akhir ini dibagi menjadi lima bab. Berikut ini adalah penjelasan dari masing-masing bab.

Bab I : Pendahuluan.

Pada bab ini menjelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, dan batasan masalah dalam tugas akhir ini.

Bab II : Kajian Literatur

Pada bab ini menjelaskan teori-teori dasar yang digunakan, teknologi yang digunakan, dan juga penelitian yang berkaitan dengan tugas akhir ini.

Bab III : Metodologi Penelitian

Pada bab ini membahas mengenai metodologi penelitian dan tahapan penelitian yang digunakan dalam tugas akhir ini.

Bab IV : Implementasi dan Evaluasi

Pada bab ini berisi proses rancangan aplikasi yang akan dibangun dan proses pengujian aplikasi apakah semua fitur berjalan dengan baik.

Bab V : Kesimpulan dan Saran

Pada bab ini berisi kesimpulan dari seluruh hasil yang telah diperoleh setelah melakukan penelitian.



BAB II KAJIAN LITERATUR

2.1 Tinjauan Pustaka

2.1.1 Sistem Inventori

Sistem inventori merupakan suatu *platform* yang dipakai untuk menginput informasi mengenai persediaan barang ke dalam suatu basis data, dengan tujuan menghindari kesalahan dalam proses *input* dan *output* data, serta pembuatan laporan yang disusun berdasarkan kebutuhan spesifik. Informasi tersebut diatur sesuai aturan tertentu, disimpan dalam kondisi siap pakai, dan terjaga keberadaannya dalam *database* [4].

Sistem inventori memiliki dampak yang signifikan pada suatu organisasi atau perusahaan. Sistem inventori dapat membantu mengatasi masalah dalam pengolahan data barang dan juga mempermudah pelaporan ketersediaan barang.

2.1.2 Barcode

Barcode merupakan rangkaian garis-garis cetak vertikal yang berwarna hitam dan putih dengan lebar yang bervariasi yang dipergunakan untuk menyimpan informasi tertentu seperti kode produksi, nomor identifikasi, dan lain-lain [5].

Perangkat komputer cenderung lebih cepat dalam mengolah informasi digital daripada analog. Kode *barcode* dengan kontras yang jelas dapat dengan mudah diidentifikasi oleh sensor optik CCD atau laser pada *scanner*, lalu diubah menjadi data angka oleh komputer.

Terdapat beberapa standar jenis *barcode*. Berikut ini adalah 5 jenis *barcode* yang biasanya digunakan dalam industri ritel maupun non-ritel beserta contoh gambarnya [6]:

1. Code 39

Sebagai simbol yang sangat dikenal di seluruh dunia dalam *barcode* non-ritel yang memiliki digit variabel yang panjang, Code 39 saat ini semakin kurang populer dan mulai digantikan oleh Code 128 yang lebih mudah diinterpretasikan oleh mesin pemindai *barcode*.



Gambar 1 : Contoh *barcode* 39

2. UPC-A (*Universal Product Code*)

Barcode EAN-8 adalah jenis *barcode* yang umumnya digunakan dalam industri ritel. Strukturnya terdiri dari total 12 digit, yang terdiri dari 11 digit data serta 1 digit pengecekan validitas di bagian akhir.



Gambar 2 : Contoh *barcode* UPC-A

3. UPC-E

UPC-E merupakan jenis *barcode* yang umum digunakan dalam bisnis ritel kecil. *Barcode* ini terdiri dari 7 digit, dimana 6 digitnya mengandung data, sementara digit terakhir berfungsi sebagai validasi.



Gambar 3 : Contoh *barcode* UPC-E

4. EAN-8

Barcode EAN-8 terbentuk dari sejumlah 8 digit yang mencakup 2 digit untuk kode negara, 5 digit untuk data, dan 1 digit terakhir yang berperan sebagai validasi.



Gambar 4 : Contoh *barcode* EAN-8

5. EAN-13

Mengandung total 13 digit, dengan 12 digit yang memuat data dan 1 digit check. Jenis *barcode* yang umum digunakan di Indonesia adalah EAN-13, yang memiliki 13 digit. Dari 13 digit tersebut, 3 digit pertama mewakili kode negara Indonesia (899). Sementara 4 digit berikutnya adalah kode perusahaan. 5 digit selanjutnya menunjukkan kode produk secara berurutan, dan digit terakhir berfungsi sebagai validasi untuk cek digit.



Gambar 5 : Contoh *barcode* EAN-13

2.1.3 *Framework*

Framework adalah kumpulan dari skrip yang dapat membantu pengembang atau *programmer* dalam menyelesaikan berbagai permasalahan dalam pengkodean, seperti mengatur koneksi ke basis data, pemanggilan variabel, manipulasi berkas, dan sebagainya. Dengan adanya *framework*, pengembang dapat lebih fokus dan efisien dalam membangun sebuah aplikasi karena mereka tidak perlu menghabiskan waktu untuk menulis ulang *script* yang sama untuk melakukan tugas yang sama.

Framework merupakan sekumpulan kode program yang digunakan secara konsisten dalam setiap pembuatan aplikasi. Karena kebutuhan akan kode-kode tersebut terus muncul, maka kode-kode tersebut disusun dan dikelola dengan rapi dalam struktur folder, membentuk suatu *framework* yang siap pakai [7].

2.1.4 **Laravel**

Laravel adalah sebuah kerangka kerja PHP yang tersedia di bawah lisensi MIT dan menggunakan pendekatan MVC (*Model View Controller*). Laravel merupakan platform pengembangan *website* berbasis MVP yang ditulis dalam bahasa pemrograman PHP dengan tujuan memperbaiki kualitas dan pemeliharaan perangkat lunak dengan biaya pemeliharaan yang murah. Selain itu, Laravel juga meningkatkan pengalaman pengguna bagi para pengembang web dengan menggunakan sintaks yang jelas, ekspresif, dan efisien [8].

2.1.5 **ReactJS**

ReactJS adalah sebuah library yang digunakan untuk membuat *user interface* menggunakan bahasa pemrograman JavaScript. ReactJS ini bersifat *open source* dan dibuat oleh Facebook. ReactJS sangat populer digunakan dan selalu dikembangkan baik oleh komunitasnya. ReactJS

menggunakan konsep komponen. Komponen dalam ReactJS ini memiliki beberapa kelebihan dalam membuat tampilan web yaitu:

1. *State Management*

State dapat digunakan ketika membuat komponen yang ditampilkan berdasarkan suatu nilai atau kondisi.

2. *Encapsulated*

Komponen ReactJS bersifat *encapsulated* yang artinya semua data termasuk gambar dan variabel yang dibutuhkan oleh komponen dalam menampilkan *user interface* berada di dalam komponen tersebut.

3. *Reusable*

Komponen ReactJS bersifat *reusable* atau dapat digunakan kembali sehingga, tidak perlu membuat komponen yang sama cukup satu saja.

2.1.6 SPA

SPA (*Single Page Application*) adalah sebuah model konseptual aplikasi web yang dirancang untuk hanya memiliki satu halaman. Berbeda dengan aplikasi web tradisional yang memiliki beberapa halaman terpisah seperti halaman beranda, halaman masuk, dan halaman pendaftaran, SPA menyatukan semua konten dan fungsionalitas dalam satu halaman yang diperbarui secara dinamis sesuai dengan interaksi pengguna. Hal ini memungkinkan pengalaman pengguna yang lebih responsif dan lebih cepat karena tidak perlu memuat ulang halaman baru setiap kali pengguna berpindah antar halaman [9].

Pada *website* tradisional atau yang memiliki beberapa halaman, ketika beralih ke halaman lain, *browser* akan memuat kembali seluruh halaman setelah meminta informasi dari *server*. Namun, dalam SPA, ketika navigasi atau perpindahan ke halaman lain terjadi, JavaScript akan mengambil data dari *server* dan secara dinamis mengubah tampilan ke halaman yang dituju tanpa memuat ulang keseluruhan halaman oleh *browser*.

Menggunakan teknologi web SPA, membuat perpindahan antar halaman menjadi lebih efisien karena *browser* tidak perlu memuat kembali seluruh halaman. Hal ini secara signifikan dapat meningkatkan pengalaman pengguna karena waktu tunggu yang singkat saat berpindah antar halaman.

2.1.7 MySQL

MySQL adalah sebuah sistem untuk manajemen *database* yang bersifat *open source* yang memiliki 2 jenis lisensi dalam penggunaannya, yaitu *free software* dan *shareware*. MySQL dapat digunakan secara gratis atau tanpa biaya untuk keperluan pribadi maupun komersial dengan menggunakan lisensi *free software* tanpa perlu membayar. Dalam lingkup sistem manajemen database, MySQL termasuk ke dalam kategori RDBMS (*Relational Database Management System*). Pada kategori ini, sistem database menggunakan terminologi seperti tabel, kolom, dan baris sehingga, dalam database ini dapat terdiri dari satu atau bahkan beberapa jumlah tabel [10].

MySQL memiliki beberapa kelebihan, seperti kinerja yang cepat, antarmuka yang ramah pengguna, sifatnya yang gratis dan sumber terbuka, dukungan penuh terhadap bahasa kueri, dan kemampuan pengguna untuk mengakses data secara *real-time*. Selain itu, *database* MySQL menyediakan beragam operator yang berguna dalam membuat kueri, yang umumnya digunakan dalam pernyataan *SELECT* dan klausa *WHERE* [11].

2.1.8 Waterfall

Menurut (Trisianto: 2018) metode *waterfall* adalah proses pengembangan perangkat lunak yang berlangsung secara berurutan, seperti air terjun mengalir. Dalam metode ini, proyek dimulai dari perencanaan (mengumpulkan kebutuhan), kemudian dilanjutkan dengan

pemodelan (merancang sistem), diikuti oleh implementasi (menulis kode), dan terakhir pengujian sistem.

Waterfall merupakan pendekatan yang terstruktur dan mengikuti urutan tertentu dalam pembangunan perangkat lunak dan termasuk ke dalam metode klasik. Dalam pengembangan menggunakan metode ini, langkah-langkahnya mencakup analisis kebutuhan, desain sistem, pengkodean, pengujian, penerapan program, dan pemeliharaan [12].

2.1.9 UML

UML (*Unified Modelling Language*) adalah bahasa yang digunakan untuk merancang, menggambarkan, membangun, dan mendokumentasikan berbagai aspek dari sistem perangkat lunak dan non-perangkat lunak. Dikembangkan oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di Rational Software Corps, UML menggunakan konsep orientasi objek dan menyediakan notasi-notasi untuk memodelkan sistem dari berbagai perspektif. Selain pemodelan perangkat lunak, UML juga digunakan secara luas di berbagai bidang lain yang memerlukan pemodelan [13].

Terdapat beberapa diagram UML dalam pengembangan aplikasi antara lain:

1. *Use Case Diagram*

Use Case Diagram adalah salah satu bentuk diagram dalam UML yang mengilustrasikan komunikasi antara sistem dan pihak yang terlibat. Selain itu, *Use Case Diagram* juga memperinci jenis interaksi antara pengguna sistem dengan sistem tersebut.

2. *Activity Diagram*

Activity Diagram adalah salah satu bentuk visualisasi dalam UML yang mampu menggambarkan segala proses yang terjadi di dalam suatu sistem.

3. *Sequence Diagram*

Sequence Diagram adalah salah satu varian dari diagram UML yang menggambarkan interaksi antara objek dalam suatu sistem berdasarkan urutan waktu. *Sequence Diagram* juga mampu menampilkan urutan langkah atau proses yang perlu dilakukan untuk mencapai suatu tujuan, diagram ini mirip dengan yang ditunjukkan dalam *Use Case Diagram*.

4. *Class Diagram*

Class Diagram adalah salah satu bentuk diagram dalam UML yang memperlihatkan kelas-kelas dan paket-paket yang terdapat dalam suatu sistem yang sedang dikembangkan. Diagram ini berguna untuk memberikan gambaran tentang struktur sistem serta hubungan antara elemen-elemen di dalamnya.

2.1.10 Black Box Testing

Salah satu jenis metode pengujian perangkat lunak adalah dengan menggunakan metode Black Box Testing. Black Box Testing atau biasanya dikenal sebagai Behavioral Testing adalah suatu bentuk evaluasi perangkat lunak yang mengamati *input* dan *output* tanpa memperhatikan rincian kode di dalamnya [14].

Biasanya pengujian semacam ini dilakukan pada tahap akhir pembuatan perangkat lunak untuk menentukan apakah fitur dari perangkat lunak tersebut beroperasi dengan baik. Keunggulan dari metode ini adalah siapa saja dapat melakukan pengujian ini tanpa keahlian khusus dalam menulis kode program.

2.1.11 Skala Likert

Skala Likert dikenal sebagai alat pengukuran dalam kuesioner, menjadi salah satu metode yang umum digunakan dalam suatu penelitian survei. Skala ini dinamai dari Rensis Likert, yang memperkenalkannya dalam

laporan karyanya, skala ini digunakan untuk menilai sikap dan pandangan. Responden diminta menyatakan seberapa setuju mereka dengan berbagai pernyataan dalam kuesioner. Skala Likert bisa berbeda-beda tergantung tujuan penelitian [15].

Pada umumnya, skala Likert menggunakan 5 opsi jawaban dan skor sebagai berikut:

1. Sangat Setuju (SS) = 5
2. Setuju (S) = 4
3. Netral (N) = 3
4. Tidak Setuju (TS) = 2
5. Sangat Tidak Setuju (STS) = 1

Untuk menghitung tingkat kepuasan hasil kuisisioner menggunakan skala Likert adalah dengan menjumlahkan hasil dari tiap skor dikalikan dengan frekuensinya. Setelah itu, dibagi dengan jumlah responden yang mengisi kuisisioner. Rumus perhitungannya sebagai berikut:

$$\text{Hasil skor} = \frac{\sum (\text{Skor} * \text{Frekuensi})}{\text{Jumlah responden}}$$

Hasil skor yang diperoleh dari perhitungan tersebut dapat diinterpretasikan ke dalam kategori berikut:

1. Sangat Puas (4.5 – 5.0)
2. Puas (3.5 – 4.4)
3. Cukup Puas (2.5 – 3.4)
4. Tidak Puas (1.5 – 2.4)
5. Sangat Tidak Puas (1.0 – 1.4)

Penulis menggunakan *software* Excel dalam penelitian ini untuk memudahkan analisis data hasil kuisisioner dengan skala Likert.

2.2 Penelitian Terkait

Penelitian terkait membahas mengenai penelitian yang relevan dengan penelitian yang penulis lakukan. Berikut ini adalah beberapa penelitiannya.

1. Pada penelitian dengan judul “Rancang Bangun Sistem Informasi *Event* Keagamaan Berbasis Web Menggunakan *Framework* *Laravel*” menghasilkan sebuah aplikasi event keagamaan berbasis web dengan tujuan menghubungkan dua belah pihak dalam upaya mencari dan menyusun acara keagamaan [12].
2. Pada penelitian dengan judul “Sistem Informasi Inventory Berbasis Web dengan Menggunakan Model *Waterfall*” menghasilkan sebuah aplikasi sistem inventori menggunakan model waterfall dan framework Codeigniter dengan tujuan membantu dalam pengelolaan data inventaris di PT. Rezeki Kencana yang sebelumnya proses pencatatan barang masuk, barang keluar, dan stok masih dilakukan secara manual dalam bentuk catatan tangan di buku. [16].
3. Pada penelitian dengan judul “Aplikasi Kasir Mobile Berbasis Android Untuk Usaha Mikro Kecil Dan Menengah” menghasilkan sebuah aplikasi kasir mobil berbasis Android dengan tujuan membantu UMKM dalam manajemen usahanya meliputi inventarisasi produk hingga transaksi dengan pelanggan. Sistem ini mengurangi kesalahan dalam menghitung total tagihan belanja pelanggan, memungkinkan pemilik usaha melihat total pendapatan harian, dan mengawasi jumlah stok produk [17].

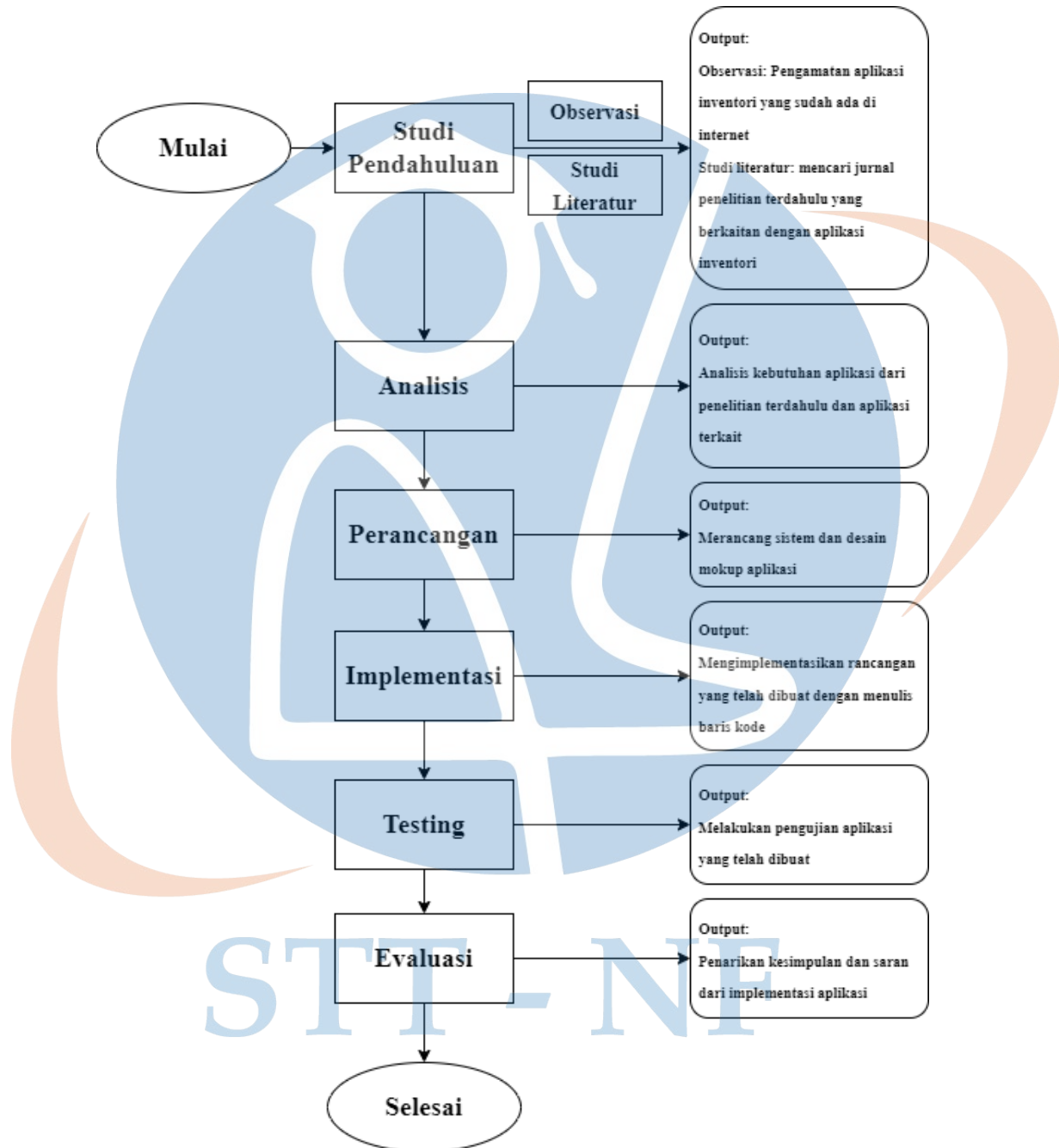
STT - NF

Tabel 1: Penelitian Terkait

No	Nama dan Tahun	Judul	Framework	Hasil
1	Aufa Billah Putra Jazama, 2019	Rancang Bangun Sistem Informasi <i>Event</i> Keagamaan Berbasis Web Menggunakan <i>Framework</i> Laravel	Laravel	Aplikasi <i>event</i> keagamaan
2	Rabiatul Adwiya. 2023	Sistem Informasi Inventory Berbasis Web dengan Menggunakan Model <i>Waterfall</i>	Codeigniter	Apikasi sistem inventori
3	Nova Noor Kamala Sari, 2021	Aplikasi Kasir <i>Mobile</i> Berbasis Android Untuk Usaha Mikro Kecil Dan Menengah	-	Aplikasi kasir <i>mobile</i> berbasis Android
4	Putu Sugiartawan, 2021	Rancang Bangun Sistem Informasi Stok Barang Berbasis <i>Website</i>	-	Aplikasi sistem sistem informasi stok <i>handphone</i>

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian



Gambar 6 : Tahapan Penelitian

1. Studi Pendahuluan

Pada langkah ini, dilakukan 2 pengamatan yaitu secara *online* untuk mengeksplorasi berbagai aplikasi yang berfungsi sebagai sistem inventori dan mencari jurnal penelitian terdahulu yang berkaitan dengan aplikasi sistem inventori. Awalnya, pengamatan dilakukan terhadap aplikasi Sortly dan Olsera yang memiliki manfaat untuk mengelola stok produk, harga, dan lain-lain. Kemudian, pengamatan dilanjutkan dengan melihat jurnal penelitian terdahulu yang berkaitan dengan sistem inventori.

Meskipun telah ditemukan aplikasi yang dapat membantu mengelola produk dan jurnal penelitian terkait, namun aplikasi tersebut belum memiliki fitur penambahan data distributor dan beberapa penelitian belum menggunakan teknologi *barcode* dalam aplikasi.

2. Analisis

Setelah memperoleh gambaran umum dari penelitian sebelumnya dan juga aplikasi yang hendak dibuat, penulis berupaya untuk menganalisis aspek tambahan yang perlu dipertimbangkan dari penelitian sebelumnya. Proses ini bertujuan untuk mengidentifikasi dan mengevaluasi masalah serta hambatan yang ada, sehingga solusi yang diusulkan sesuai dengan hasil analisis. Dengan memahami beberapa kekurangan, diharapkan aplikasi yang dikembangkan selanjutnya dapat mengatasi kelemahan yang ada pada aplikasi sebelumnya.

3. Perancangan

Dalam tahap ini, sistem direncanakan berdasarkan hasil analisis yang telah penulis lakukan sebelumnya. Perancangan sistem menggunakan UML sebagai standar untuk merancang model sistem. Rancangan sistem menggunakan beberapa jenis diagram UML.

4. Implementasi

Langkah berikutnya adalah implementasi berdasarkan rancangan yang sudah disiapkan sebelumnya. Implementasi ini melibatkan proses penulisan kode program untuk membuat tampilan aplikasi dan merancang struktur *database* dengan tujuan menghasilkan aplikasi web yang siap digunakan.

5. Testing

Langkah berikutnya adalah tahapan testing. Aplikasi akan diuji untuk mengecek apakah aplikasi dapat berjalan sesuai fungsionalitasnya. Pada tahap ini, aplikasi web diuji menggunakan Black Box Testing. Pengujian dilakukan dengan langsung menguji fitur-fitur dalam sistem pada hasil akhir, tanpa memeriksa setiap komponen secara terpisah.

6. Evaluasi

Langkah berikutnya adalah tahapan evaluasi. Tahapan ini dilakukan penarikan kesimpulan dan saran dari implementasi aplikasi berdasarkan hasil testing yang telah dilakukan sebelumnya.

3.2 Rancangan Penelitian

3.2.1 Jenis Penelitian

Jenis Penelitian yang digunakan adalah penelitian pengembangan (R&D). Penelitian dan Pengembangan adalah strategi riset yang digunakan untuk memajukan serta mengevaluasi produk yang akan diterapkan dalam konteks pendidikan [18]. Jenis penelitian ini memiliki tujuan untuk mengembangkan sebuah aplikasi yang akan digunakan oleh suatu pihak tertentu. Hasil dari penelitian ini adalah pengembangan aplikasi web yang membantu pengguna dalam mengelola produknya menjadi lebih cepat dan sistematis.

3.2.2 Metode Analisis Data

Dalam penelitian ini, penulis menggunakan metode penelitian kualitatif. Menurut Sugiyono (2019) penelitian kualitatif deskriptif adalah metode penelitian yang didasarkan pada filsafat postpositivme. Metode ini digunakan untuk mempelajari kondisi objek yang alami, dengan peneliti sebagai alat utamanya. Data dikumpulkan dengan berbagai teknik dan berbentuk kata-kata atau gambar, bukan angka. Tujuan penelitian ini adalah untuk mendapatkan gambaran lengkap tentang proses yang sedang terjadi. Analisis data dilakukan dengan menggambarkan data

yang terkumpulnya adanya, tanpa membuat kesimpulan yang berlaku umum.

3.2.3 Metode Pengumpulan Data

Dalam mendukung penulisan Tugas Akhir, penulis menggunakan berbagai metode untuk mengumpulkan data, termasuk observasi dan studi pustaka sebagai proses utama dalam memperoleh informasi yang dibutuhkan.

1. Observasi

Pada tahap ini, dilakukan pengamatan terhadap fitur-fitur dalam aplikasi yang sudah ada. Pengamatan ini bertujuan untuk memahami proses dan mengidentifikasi fitur yang tersedia pada aplikasi tersebut.

2. Studi Pustaka

Pada tahap ini, informasi dan teori-teori yang relevan dengan penelitian yang penulis lakukan dari penelitian terdahulu dikumpulkan. Proses pengumpulan informasi melibatkan pembacaan jurnal, artikel, dan skripsi dari peneliti lain untuk mengevaluasi pengembangan aplikasi berbasis web, terutama yang terkait dengan pengembangan sistem inventori.

3.2.4 Metode Pengujian

Metode pengujian aplikasi dalam penelitian yang penulis lakukan menggunakan metode pengujian *Black Box Testing*. Metode ini merupakan pengujian fungsionalitas sistem langsung di hasil akhir tanpa memeriksa per komponen sistem.

3.2.5 Metode Implementasi dan Evaluasi

Metode implementasi dalam penelitian ini dengan menampilkan hasil tangkapan layar dari semua halaman yang ada di aplikasi. Metode evaluasi dalam penelitian ini dengan melalui 3 tahapan yaitu:

1. Hasil pengujian Black Box Testing
2. UAT (*User Acceptance Test*)
3. Kuisisioner

3.2.6 Lingkungan Pengembangan

Dalam penelitian untuk membangun sebuah aplikasi berbasis web, diperlukan bahan dan peralatan sebagai sarana pengembangan aplikasi. Berikut adalah daftar bahan dan peralatan yang digunakan.

1. Laptop Acer Aspire Lite 14

Laptop ini berperan sebagai perangkat yang mendukung dalam menyusun Tugas Akhir. Laptop ini memiliki spesifikasi sebagai berikut:

- Sistem Operasi Windows 11
- Processor Intel Core i3
- RAM 8 GB

2. PHP

Pada pengembangan aplikasi berbasis web ini, penulis menggunakan PHP dengan versi 8.3.4 sebagai bahasa pemrograman.

3. MySQL

Pada pengembangan aplikasi berbasis web ini, penulis menggunakan MySQL dengan versi 8.0.3 sebagai database yang digunakan untuk menyimpan data aplikasi.

4. Google Chrome

Pada pengembangan aplikasi berbasis web ini, penulis menggunakan browser Google Chrome dengan versi 124.

5. *Framework* Laravel

Pada pengembangan aplikasi web ini, penulis menggunakan *framework* Laravel versi 11.

6. *Library* html5-qrcode

Library JavaScript yang digunakan untuk memproses pemindaian *barcode* melalui browser dalam aplikasi ini menggunakan *html5-qrcode*.

BAB IV

IMPLEMENTASI DAN EVALUASI

Pada metode Waterfall, pengembangan aplikasi terdiri dari beberapa tahapan yaitu perencanaan, pemodelan, implementasi, dan pengujian. Pada tahap perencanaan akan diuraikan dalam bentuk diagram UML. Pada tahap pemodelan akan diuraikan dalam bentuk tampilan *mockup* aplikasi dan arsitektur sistem. Pada tahap implementasi akan diuraikan dalam bentuk implementasi kode dan antarmuka halaman. Pada tahap pengujian diuraikan dalam bentuk hasil pengujian sistem dari Black Box Testing, UAT, dan kuesioner.

4.1 Perencanaan Sistem

Pada penelitian ini dilakukan proses pengumpulan *User Requirement* melalui studi pustaka dari penelitian sebelumnya yang telah dilakukan oleh Rabiatul Adwiyah dengan judul penelitian “Sistem Informasi *Inventory* Berbasis Web dengan Menggunakan Model *Waterfall*”.

Pengumpulan *User Requirement* juga dilakukan melalui proses pengamatan dan perbandingan yang dilakukan pada *website* Olsera (olsera.com) dan aplikasi mobile Sortly. Kedua aplikasi tersebut dapat mencatat stok dan data transaksi dengan *platform* yang berbeda.

Berikut merupakan beberapa poin utama yang didapatkan dari penelitian yang dilakukan oleh Rabiatul Adwiyah dengan judul penelitian “Sistem Informasi *Inventory* Berbasis Web dengan Menggunakan Model *Waterfall*” yaitu:

No	Fitur
1	Mengelola data kategori, barang, distributor
2	Mengelola transaksi pengadaan barang
3	Mengelola laporan stok barang
4	Mengelola akun

Tabel 2: Fitur Penelitian Sebelumnya

Pada tabel di bawah ini merupakan beberapa fitur utama yang didapatkan dari hasil pengamatan dan perbandingan pada website Olsera dan aplikasi Sortly.

No	Perbandingan	Olsera	Sortly
1	Mengelola data supplier	Ya	Tidak
2	<i>Scan barcode</i> produk	Tidak	Ya
3	Transaksi penjualan	Ya	Tidak

Tabel 3: Perbandingan Aplikasi

Melihat kebutuhan *User Requirement* yang didapatkan dari 2 tahapan di atas yaitu pada penelitian sebelumnya dan perbandingan aplikasi Olsera dengan Sortly maka didapatkan hasil analisis sistem sebagai berikut:

1. Hasil analisis sistem dari penelitian sebelumnya.

No	Fitur	Penelitian sebelumnya	Penelitian sekarang
1	Mengelola data kategori, barang, distributor	Ya	Ya
2	Mengelola transaksi barang	Ya	Ya
3	Mengelola laporan stok barang	Ya	Ya
4	Mengelola akun	Ya	Ya
5	Target aplikasi	PT. Rezeki Kencana	Umum
6	<i>Scan barcode</i> produk	Tidak	Ya

Tabel 4: Hasil Perbandingan Penelitian Sebelumnya

2. Hasil analisis sistem dari perbandingan aplikasi.

No	Fitur	Hasil
1	Mengelola data distributor	Ya
2	<i>Scan barcode</i> produk	Ya
3	Transaksi barang masuk dan keluar	Ya

Tabel 5: Hasil Perbandingan Aplikasi

4.1.1 Istilah Baku

Pada aplikasi web yang akan dirancang terdapat beberapa istilah yang akan digunakan. Istilah baku ini dapat dijadikan acuan dari istilah-istilah yang akan digunakan pada pengembangan aplikasi web ini.

Istilah-istilah baku yang akan digunakan adalah sebagai berikut.

No	Istilah	Pengertian
1	<i>Barcode</i>	Rangkaian garis vertikal hitam putih yang berisi kode produksi yang biasanya terdapat di kemasan barang.
2	<i>Register</i>	Mendaftarkan akun di aplikasi.
3	<i>Login</i>	Masuk ke aplikasi menggunakan akun yang sudah dibuat di halaman registrasi..
4	<i>Scan</i>	Memindai suatu gambar menggunakan kamera untuk dijadikan data digital

Tabel 6: Istilah Baku

4.1.2 User Requirement

User Requirement berisi beberapa pernyataan layanan yang disediakan oleh sistem dengan batasan operasional.

1. Kategori Kebutuhan *User*

Kode	Deskripsi
R.001	Mendaftar sebagai <i>User</i>
R.002	Masuk sebagai <i>User</i>
R.003	Menambahkan data barang

R.004	Mengelola barang masuk
R.005	Mengelola barang keluar
R.006	Mencari data barang
R.007	Mengelola akun
R.008	<i>Scan barcode</i> barang

Tabel 7: *User Requirement*

2. Kategori Pengguna Aplikasi

Kategori pengguna yang memiliki hak akses yang berbeda. Kategori pengguna dibedakan menjadi 2 yaitu *Guest* dan *User* dijelaskan dalam tabel berikut.

Pengguna	Deskripsi
<i>Guest</i>	Pengguna aplikasi yang belum <i>login</i> ke sistem
<i>User</i>	Pengguna aplikasi sudah <i>login</i> dapat mengelola barang

Tabel 8: Kategori Pengguna

3. Kategori Menu Aplikasi

Pengelompokan menu dapat dibedakan berdasarkan kategori pengguna yang dapat dilihat dalam tabel berikut:

Pengguna	Menu
<i>Guest</i>	<ul style="list-style-type: none"> • <i>Login</i> • <i>Register</i>
<i>User</i>	<ul style="list-style-type: none"> • Mengelola data barang <ul style="list-style-type: none"> - Menambah barang - Mengedit barang - Menghapus barang • Mengelola data distributor <ul style="list-style-type: none"> - Menambah distributor - Mengedit distributor - Menghapus distributor • Mengelola data barang masuk

	<ul style="list-style-type: none"> • Mengelola data barang keluar • Mengelola akun • <i>Logout</i>
--	---

Tabel 9: Kategori Menu Aplikasi

4.2 Pemodelan Sistem

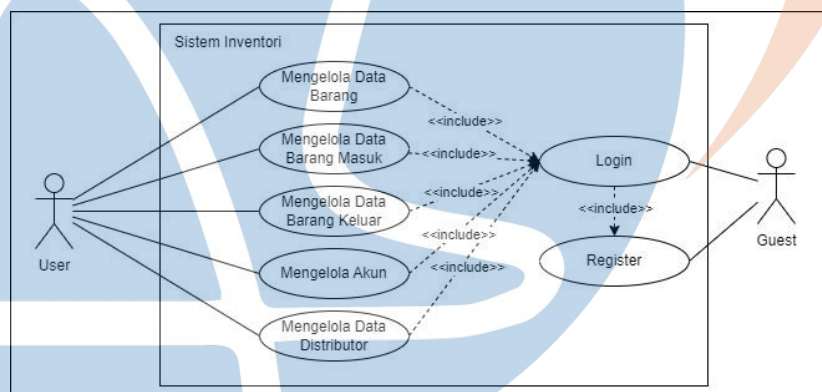
4.2.1 Desain sistem

1. Use Case Diagram

Pada bagian ini menjelaskan mengenai *Use Case Diagram* dalam aplikasi sistem inventori berbasis web yang diperoleh dari hasil analisis.

1. Use Case Diagram

Pada gambar di bawah ini menjelaskan tentang *Use Case Diagram* aplikasi sistem inventori berbasis web yang akan dikembangkan.



Gambar 7: Use Case Diagram

2. Use Case Description

Berikut ini adalah daftar *Use Case Description* dari *Use Case Diagram* di atas.

1. Use Case mendaftar sebagai user

Nama Use Case	Use Case Scenario UC-001 Mendaftar sebagai user
Author	Julkarnain
Requirement Terkait	R-001 Mendaftar sebagai user

Tujuan	Pengguna <i>guest</i> dapat mendaftarkan akun
Kondisi Awal	Pengguna <i>guest</i> membuka halaman <i>register</i>
Kondisi Sukses	Pengguna <i>guest</i> dapat mendaftar
Kondisi Gagal	Pengguna <i>guest</i> tidak dapat mendaftarkan akun
<i>Primary Actor</i>	<i>Guest</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol daftar
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna <i>guest</i> mengklik tombol daftar - Pengguna <i>guest</i> mengisi form pendaftaran - Data pengguna disimpan
<i>Extension</i>	

Tabel 10: *Use Case* mendaftar sebagai *user*

2. *Use Case* login sebagai *user*

Nama <i>Use Case</i>	<i>Use Case Scenario</i> UC-002 <i>login</i> sebagai <i>user</i>
<i>Author</i>	Julkarnain
<i>Requirement</i> Terkait	R-002 <i>login</i> sebagai <i>user</i>
Tujuan	Pengguna <i>guest</i> dapat masuk ke aplikasi
Kondisi Awal	Pengguna <i>guest</i> membuka halaman <i>login</i>
Kondisi Sukses	Pengguna <i>guest</i> dapat masuk ke aplikasi
Kondisi Gagal	Pengguna <i>guest</i> tidak dapat masuk ke aplikasi

<i>Primary Actor</i>	<i>Guest</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol masuk
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna <i>guest</i> mengklik tombol masuk - Pengguna <i>guest</i> mengisi formulir <i>login</i> - <i>User guest</i> masuk ke aplikasi dan menjadi pengguna <i>user</i>
<i>Extension</i>	

Tabel 11: *Use Case login sebagai user*

3. *Use Case* menambah data barang

Nama <i>Use Case</i>	<i>Use Case Scenario UC-002 login sebagai user</i>
<i>Author</i>	Julkarnain
<i>Requirement</i> Terkait	R-003 menambahkan data barang
Tujuan	Pengguna dapat menambahkan data barang
Kondisi Awal	Pengguna sudah <i>login</i> ke sistem
Kondisi Sukses	Pengguna berhasil menambahkan barang
Kondisi Gagal	Pengguna tidak dapat menambahkan barang
<i>Primary Actor</i>	<i>User</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol tambah barang
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna mengklik tombol tambah barang

	<ul style="list-style-type: none"> - Pengguna mengisi data informasi barang - Barang tersimpan dan pengguna dapat melihat barang yang telah ditambahkan
<i>Extension</i>	<ul style="list-style-type: none"> - Pengguna memasukkan data tidak sesuai format - Pengguna mendapatkan pesan kesalahan

Tabel 12: *Use Case* menambah data barang

4. *Use Case* mengelola barang masuk

<i>Nama Use Case</i>	<i>Use Case Scenario</i> UC-004 mengelola barang masuk
<i>Author</i>	Julkarnain
<i>Requirement</i> Terkait	R-004 mengelola barang masuk
Tujuan	Pengguna dapat menambahkan data barang masuk untuk menambah stok
Kondisi Awal	Pengguna sudah <i>login</i> ke system dan telah menambahkan data barang dan distributor
Kondisi Sukses	Pengguna berhasil menambahkan data barang masuk dan stok barang bertambah
Kondisi Gagal	Pengguna tidak dapat menambahkan data barang masuk
<i>Primary Actor</i>	<i>User</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol tambah
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna mengklik tombol tambah

	<ul style="list-style-type: none"> - Pengguna mengisi data barang masuk dengan data barang yang telah ditambahkan - Pengguna berhasil menambahkan data barang masuk
<i>Extension</i>	<ul style="list-style-type: none"> - Pengguna diminta untuk menambahkan data barang/distributor jika belum ada - Pengguna mendapatkan pesan kesalahan jika tidak mengisi data sesuai format

Tabel 13: *Use Case* mengelola barang masuk

5. *Use Case* mengelola barang keluar

<i>Nama Use Case</i>	<i>Use Case Scenario</i> UC-005 mengelola barang keluar
<i>Author</i>	Julkarnain
<i>Requirement</i> Terkait	R-005 mengelola barang keluar
Tujuan	Pengguna dapat menambahkan data barang keluar
Kondisi Awal	Pengguna sudah <i>login</i> ke system dan telah menambahkan data barang dan distributor
Kondisi Sukses	Pengguna berhasil menambahkan data barang keluar dan stok barang berkurang
Kondisi Gagal	Pengguna tidak dapat menambahkan data barang keluar
<i>Primary Actor</i>	<i>User</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol tambah

<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna mengklik tombol tambah - Pengguna mengisi data barang keluar dengan data barang yang telah ditambahkan sebelumnya - Pengguna berhasil menambahkan data barang keluar
<i>Extension</i>	<ul style="list-style-type: none"> - Pengguna diminta untuk menambahkan data barang/distributor jika belum ada - Pengguna mendapatkan pesan kesalahan jika tidak mengisi data sesuai format

Tabel 14: *Use Case* mengelola barang keluar

6. *Use Case* mencari data barang

Nama <i>Use Case</i>	<i>Use Case Scenario</i> UC-006 mencari data barang
<i>Author</i>	Julkarnain
<i>Requirement</i> Terkait	R-006 mencari data barang
Tujuan	Pengguna dapat mencari data barang berdasarkan nama, ID, atau kode produksi
Kondisi Awal	Pengguna sudah <i>login</i> ke sistem dan telah menambahkan data barang
Kondisi Sukses	Pengguna mendapatkan barang yang dicari sesuai kata kunci
Kondisi Gagal	Pengguna tidak menemukan barang yang dicari
<i>Primary Actor</i>	<i>User</i>
<i>Secondary Actor</i>	

<i>Trigger</i>	Pengguna memasukkan kata kunci di kolom pencarian
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna memasukkan kata kunci di kolom pencarian - Sistem mencari barang sesuai kata kunci - Pengguna mendapatkan barang yang dicari
<i>Extension</i>	<ul style="list-style-type: none"> - Menampilkan pesan barang tidak ditemukan jika kata kunci tidak cocok dengan semua data barang - Menampilkan pesan barang masih kosong jika barang belum ditambahkan

Tabel 15: *Use Case* mencari data barang

7. *Use Case* mengelola akun

Nama <i>Use Case</i>	<i>Use Case Scenario</i> UC-007 mengelola akun
Author	Julkarnain
Requirement Terkait	R-007 mengelola akun
Tujuan	Pengguna dapat mengubah nama dan kata sandi
Kondisi Awal	Pengguna sudah <i>login</i> ke sistem
Kondisi Sukses	Pengguna berhasil mengubah data akun
Kondisi Gagal	Pengguna gagal mengubah data akun
Primary Actor	User
Secondary Actor	
Trigger	Pengguna masuk ke menu akun
Main Flow	<ul style="list-style-type: none"> - Pengguna mengklik menu akun

	<ul style="list-style-type: none"> - Pengguna mengklik tombol edit - Pengguna memasukkan data akun yang baru - Data pengguna berhasil diubah
<i>Extension</i>	<ul style="list-style-type: none"> - Pengguna mendapatkan pesan kesalahan jika memasukkan data tidak sesuai format

Tabel 16: *Use Case* mengelola akun

8. *Use Case scan barcode barang*

<i>Nama Use Case</i>	<i>Use Case Scenario UC-008 scan barcode barang</i>
<i>Author</i>	Julkarnain
<i>Requirement Terkait</i>	R-008 <i>scan barcode barang</i>
<i>Tujuan</i>	Pengguna dapat menambahkan atau mencari barang dengan memindai <i>barcode barang</i>
<i>Kondisi Awal</i>	Pengguna sudah <i>login</i> ke sistem
<i>Kondisi Sukses</i>	Pengguna berhasil menemukan barang jika sudah ditambahkan atau beralih ke halaman tambah barang jika belum ditambahkan
<i>Kondisi Gagal</i>	Aplikasi tidak dapat memindai <i>barcode barang</i>
<i>Primary Actor</i>	<i>User</i>
<i>Secondary Actor</i>	
<i>Trigger</i>	Pengguna mengklik tombol <i>scan</i>
<i>Main Flow</i>	<ul style="list-style-type: none"> - Pengguna mengklik tombol <i>scan</i> - Pengguna diminta untuk memberikan akses kamera

	<ul style="list-style-type: none"> - Pengguna mengarahkan kamera ke <i>barcode</i> barang - Pengguna menemukan barang jika sudah ditambahkan atau beralih ke halaman tambah barang jika belum
<i>Extension</i>	

Tabel 17: *Use Case scan barcode* barang

2. *Activity Diagram*

Activity Diagram adalah salah satu bentuk visualisasi dalam UML yang mampu menggambarkan segala proses yang terjadi di dalam suatu sistem dalam bentuk model aliran dari aktivitas ke aktivitas lainnya.

a) *Activity Diagram mendaftar sebagai user*

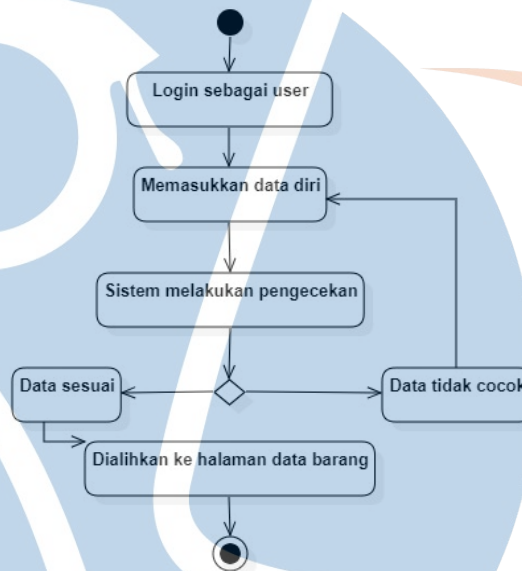
Pada *Activity Diagram mendaftar sebagai user* menggambarkan proses pendaftaran akun yang dilakukan oleh pengguna yang belum memiliki akun. Lalu, pengguna memasukkan data dirinya berupa nama, *email*, dan kata sandi. Setelah itu, pengguna akan dialihkan ke halaman *login*.



Gambar 8: *Activity Diagram mendaftar sebagai user*

b) *Activity Diagram login sebagai user*

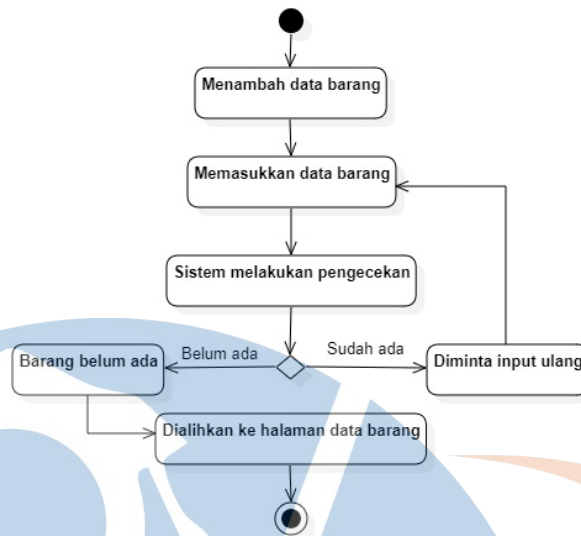
Pada *Activity Diagram login sebagai user* menggambarkan proses *user* masuk ke sistem. Sistem akan mengecek apakah data *email* dan kata sandi cocok, jika cocok maka *user* akan diarahkan ke halaman data barang dan jika tidak maka *user* akan diminta untuk mengulangi pengisian data.



Gambar 9: *Activity Diagram login sebagai user*

c) *Activity Diagram menambah data barang*

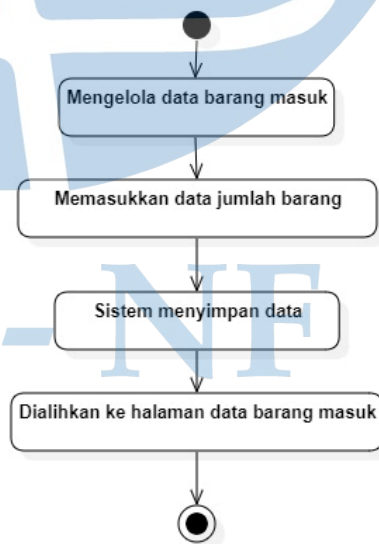
Pada *Activity Diagram menambah data barang* mendeskripsikan proses *user* memasukkan data barang baru ke sistem. Sistem akan melakukan pengecekan apakah barang yang akan ditambahkan sudah ada atau belum jika belum maka sistem akan menyimpan data barang, tetapi jika sudah *user* akan diminta untuk menginput data barang yang baru.



Gambar 10: *Activity Diagram* menambah data barang

d) *Activity Diagram* mengelola barang masuk

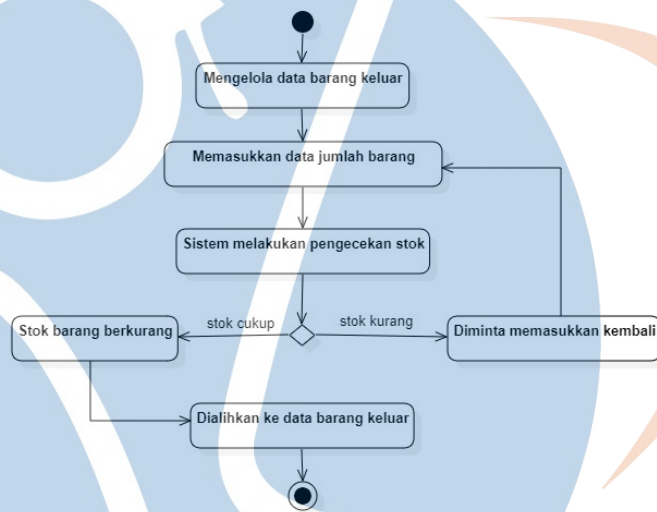
Pada *Activity Diagram* mengelola barang masuk menggambarkan proses *user* menambahkan stok barang ke sistem. *User* diminta untuk mengisi data barang, jumlah, asal distributor, dan tanggal pemasukan. Sistem akan menyimpan data dan mengubah stok setiap barang yang masuk.



Gambar 11: *Activity Diagram* mengelola barang masuk

e) Activity Diagram mengelola barang keluar

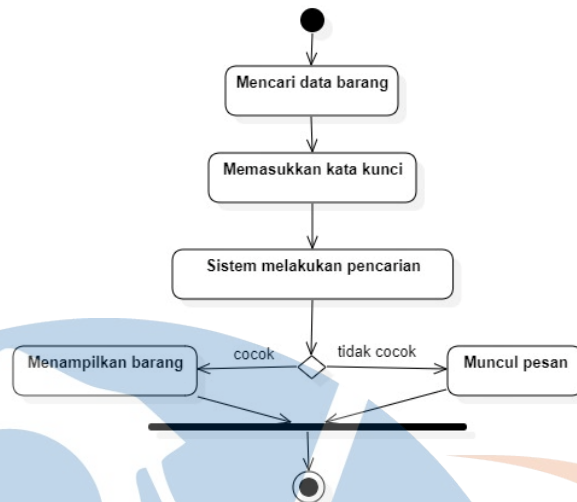
Pada *Activity Diagram* mengelola barang keluar menggambarkan proses *user* mengurangi stok barang di sistem. *User* diminta untuk mengisi data barang, jumlah dan tanggal pengeluaran. Sistem akan melakukan pengecekan jika jumlah barang keluar melebihi stok maka *user* diminta untuk mengubah jumlah barang keluar. Lalu, sistem menyimpan data dan mengubah stok setiap barang yang keluar.



Gambar 12: *Activity Diagram* mengelola barang keluar

f) Activity Diagram mencari data barang

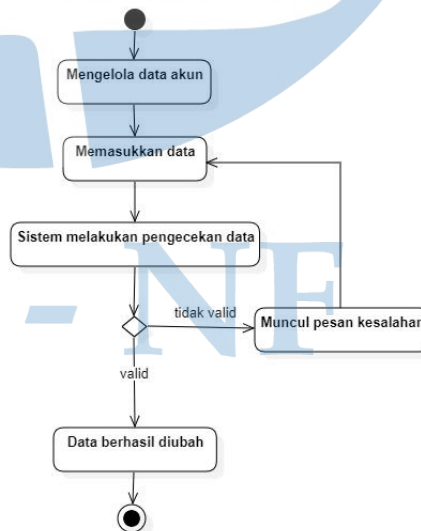
Pada *Activity Diagram* mencari data barang menggambarkan proses *user* mencari barang yang sebelumnya sudah ditambahkan. *User* memasukkan kata kunci barang berupa ID atau nama. Sistem akan mencari data barang yang cocok dengan kata kunci tersebut. Lalu, sistem akan menampilkan barang yang cocok jika tidak maka sistem akan menampilkan pesan barang tidak ditemukan.



Gambar 13: *Activity Diagram* mencari data barang

g) *Activity Diagram* mengelola akun

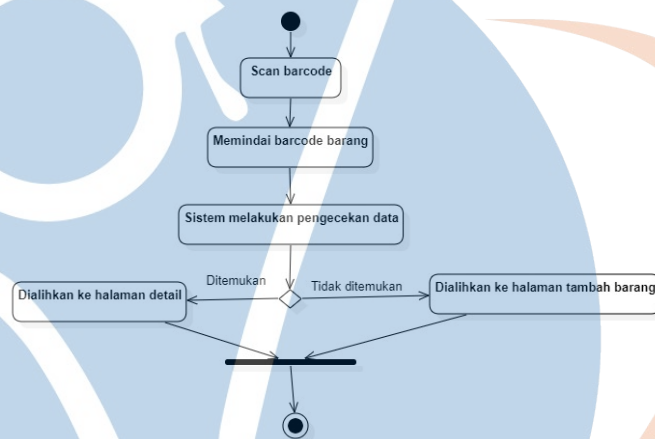
Pada *Activity Diagram* mengelola akun menggambarkan proses *user* mengubah data profilnya seperti nama, email, dan kata sandi. Sistem akan melakukan pengecekan apakah data yang dikirimkan sudah valid, jika iya maka data profil berubah dan jika tidak valid maka user akan diminta untuk memasukkan ulang datanya.



Gambar 14: *Activity Diagram* mengelola akun

h) *Activity Diagram scan barcode barang*

Pada *Activity Diagram scan barcode* barang menggambarkan proses *user* melakukan pemindaian *barcode* menggunakan kamera. Sistem akan mengecek apakah barang yang dipindai sudah terdata di sistem, jika iya maka sistem akan mengalihkan ke halaman detail barang dan apabila tidak ditemukan maka sistem akan mengalihkan ke halaman tambah barang.

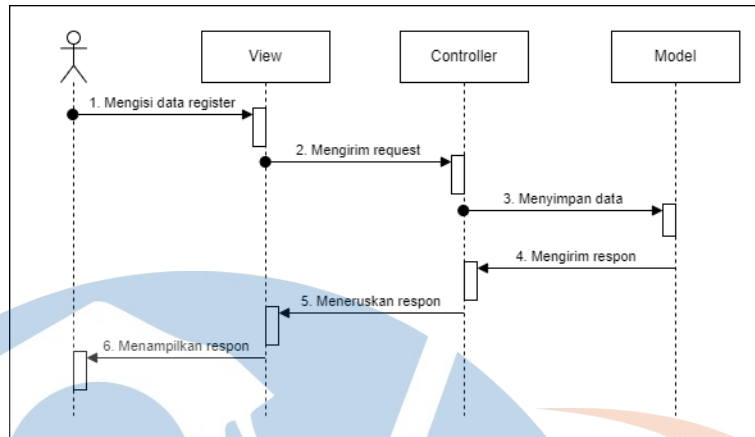


Gambar 15: *Activity Diagram scan barcode*

3. *Sequence Diagram*

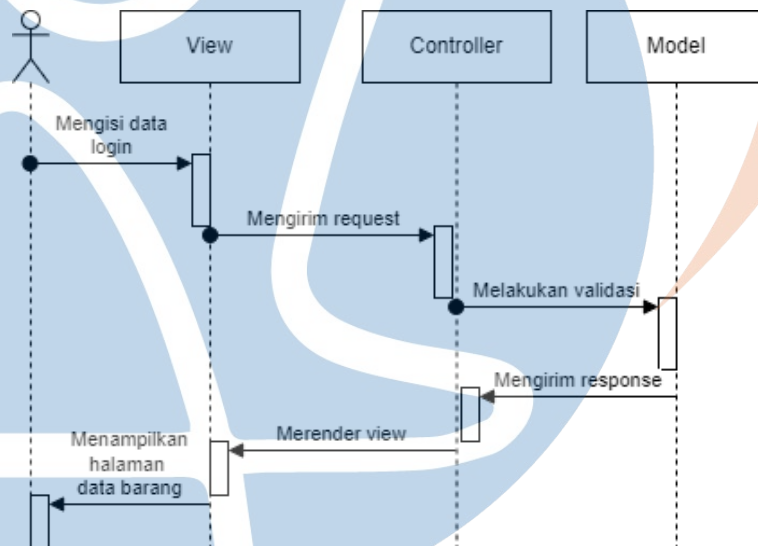
Sequence Diagram adalah salah satu varian dari diagram UML yang menggambarkan interaksi antara objek dalam suatu sistem berdasarkan urutan waktu. *Sequence Diagram* dapat menampilkan urutan langkah atau proses yang perlu dilakukan untuk mencapai suatu tujuan, diagram ini mirip dengan yang ditunjukkan dalam *Use Case Diagram*, sehingga *Use Case Diagram* dapat dijadikan acuan untuk diagram ini.

a) *Sequence Diagram* mendaftar sebagai user



Gambar 16: *Sequence Diagram Register*

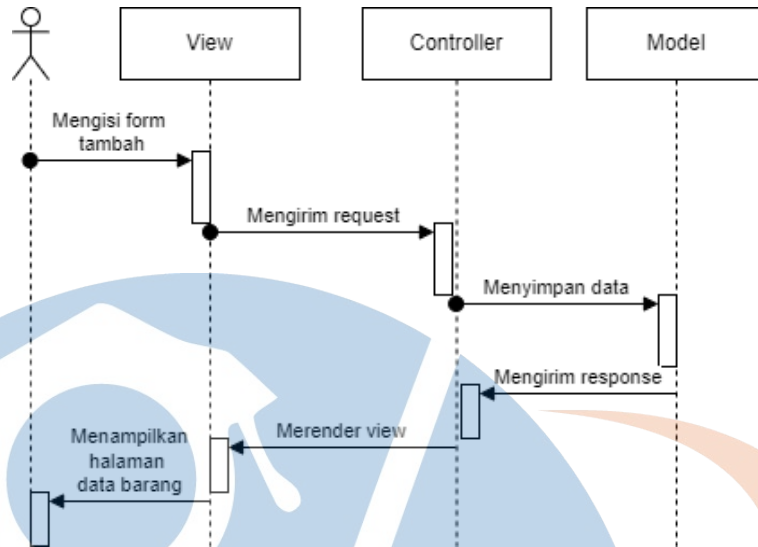
b) *Sequence Diagram* masuk sebagai user



Gambar 17: *Sequence Diagram Login*

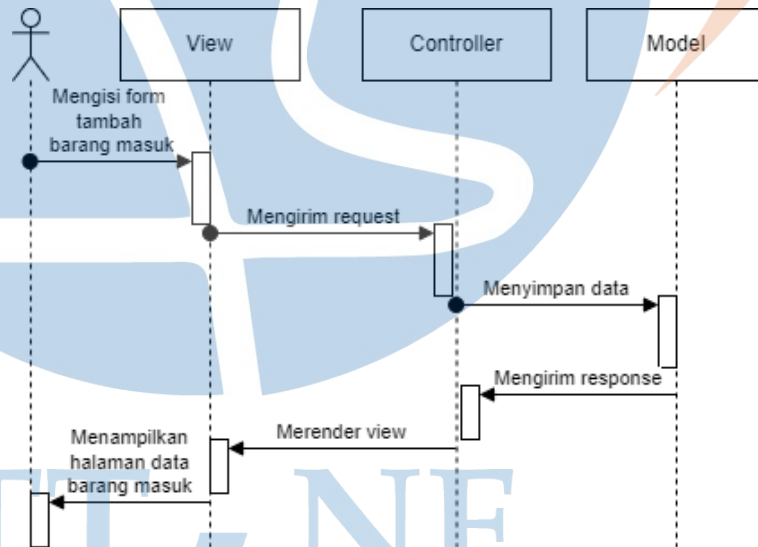
STT - NF

c) *Sequence Diagram* menambahkan barang



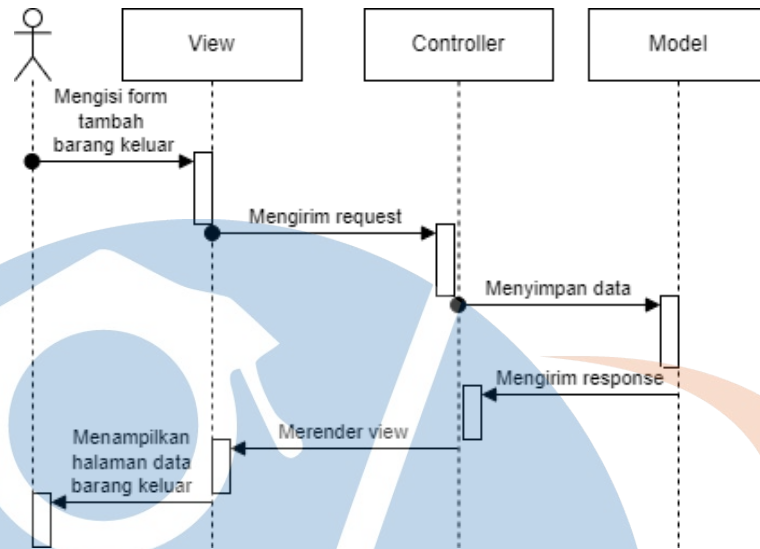
Gambar 18: *Sequence Diagram* tambah barang

d) *Sequence Diagram* mengelola barang masuk



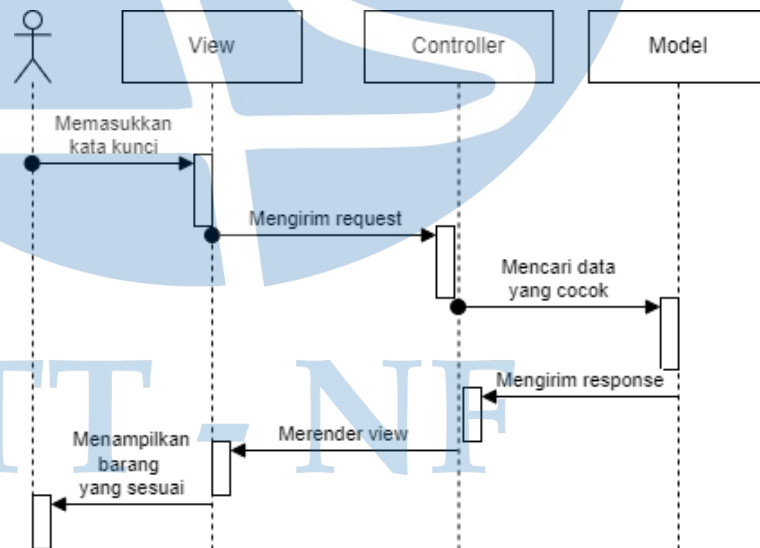
Gambar 19: *Sequence Diagram* mengelola barang masuk

e) *Sequence Diagram* mengelola barang keluar



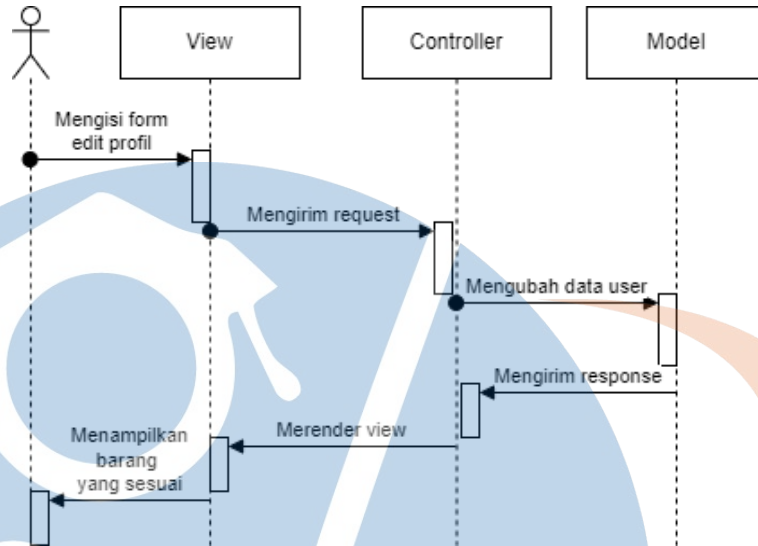
Gambar 20: *Sequence Diagram* mengelola barang keluar

f) *Sequence Diagram* mencari data barang



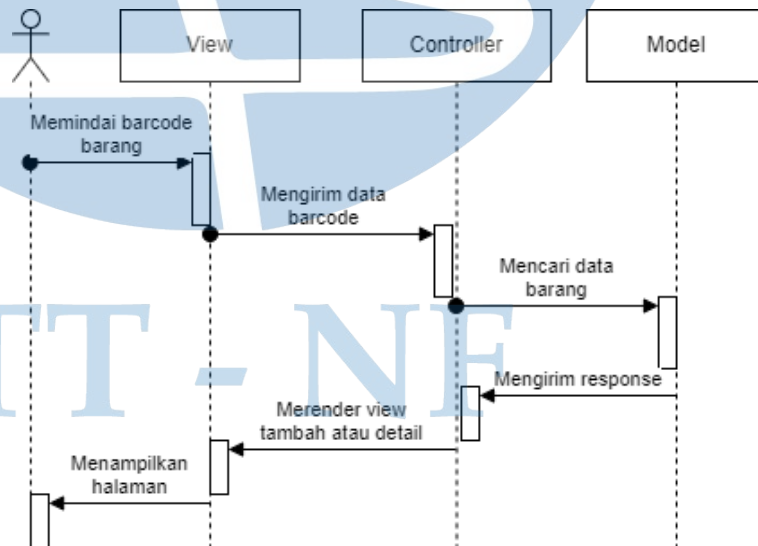
Gambar 21: *Sequence Diagram* mencari barang

g) Sequence Diagram mengelola akun



Gambar 22: *Sequence Diagram* mengelola akun

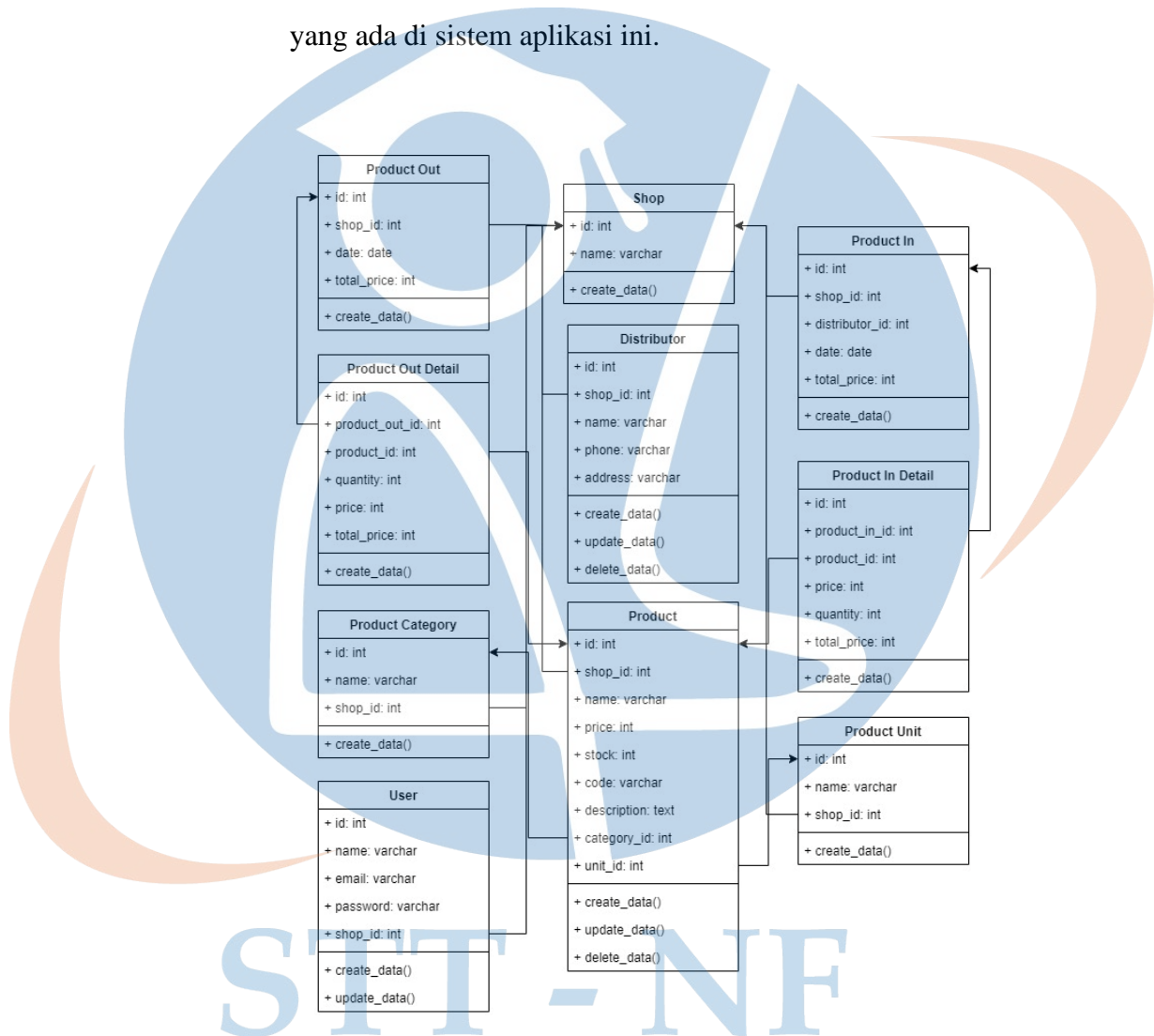
h) Sequence Diagram scan barcode barang



Gambar 23: *Sequence Diagram* scan barang

4. Class Diagram

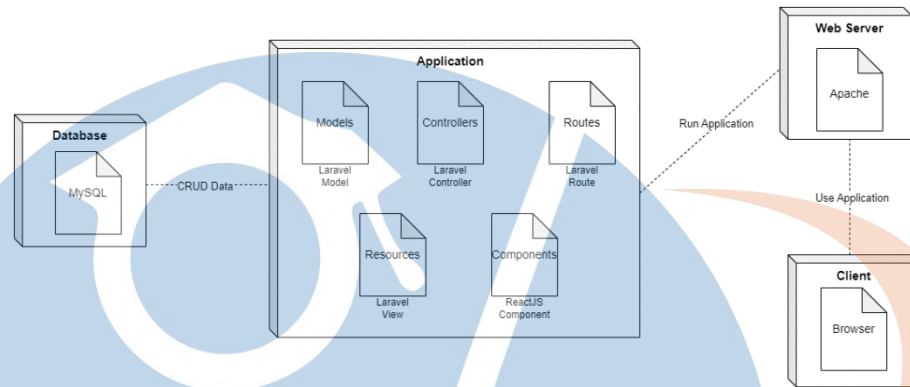
Class Diagram adalah salah satu bentuk diagram dalam UML yang memperlihatkan kelas-kelas dan paket-paket yang terdapat dalam suatu sistem. Diagram ini berguna untuk memberikan gambaran tentang struktur sistem serta hubungan antara elemen-elemen atau tabel di dalamnya. Berikut ini adalah gambaran mengenai relasi tabel yang ada di sistem aplikasi ini.



Gambar 24: *Class Diagram*

4.2.2 Arsitektur Sistem

Setelah melakukan perancangan desain sistem selanjutnya adalah membuat arsitektur sistem yang mendeskripsikan komponen dan proses apa saja yang terjadi di dalam aplikasi yang akan dirancang berikut ini adalah gambar arsitektur sistem aplikasi.

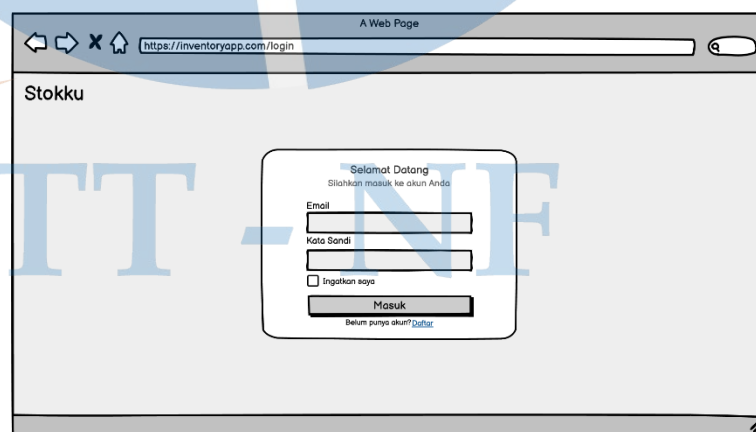


Gambar 25: Arsitektur Sistem

4.2.3 Antarmuka Sistem

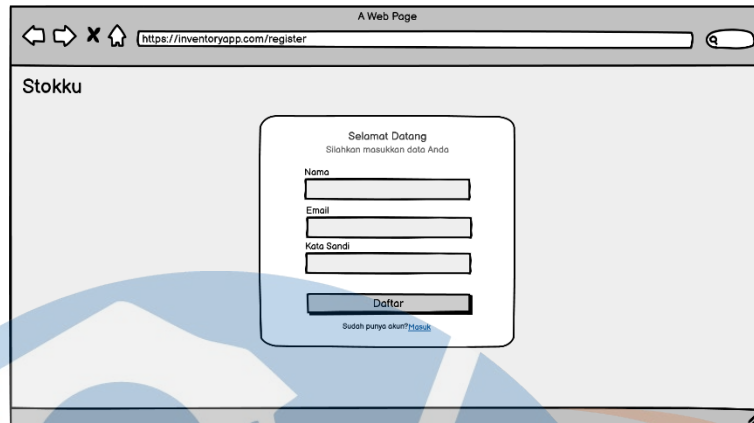
Setelah melakukan perancangan arsitektur sistem selanjutnya adalah membuat wireframe antarmuka aplikasi yang mendeskripsikan tampilan antarmuka aplikasi yang akan dirancang. Berikut ini adalah beberapa antarmuka sistem inventori.

1. Halaman *Login* Aplikasi



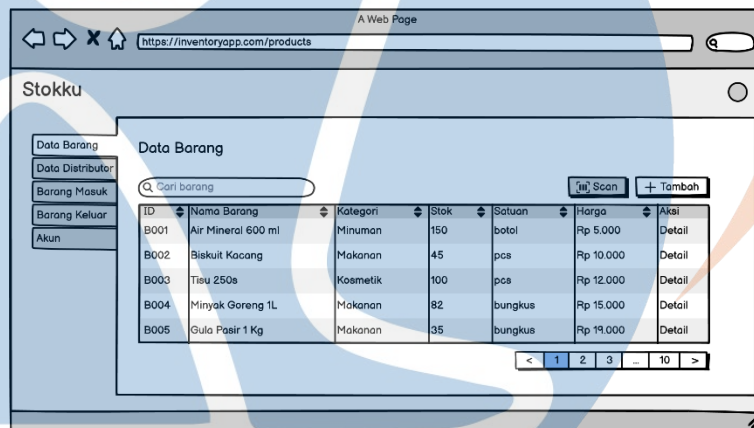
Gambar 26: Wireframe Login

2. Halaman Register Aplikasi



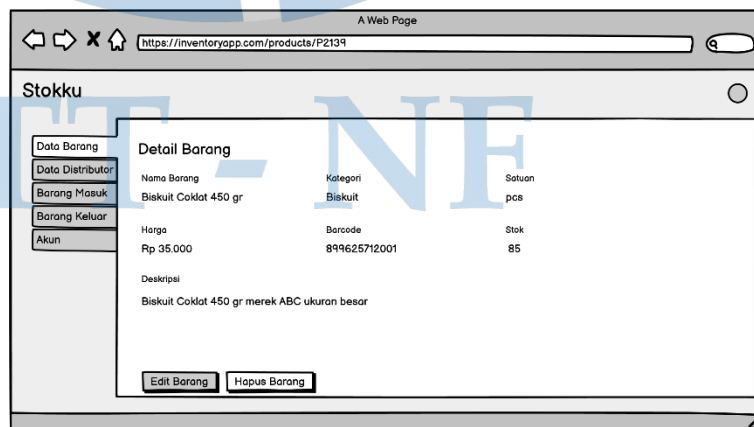
Gambar 27: Wireframe Register

3. Halaman Data Barang



Gambar 28: Wireframe Data Barang

4. Halaman Detail Barang



Gambar 29: Wireframe Detail Barang

5. Halaman Tambah Barang

Stokku

Data Barang

Data Distributor

Barang Masuk

Barang Keluar

Akun

Tambah Barang Baru

Nama Barang*

Kategori*

Satuan*

Harga (Rp)*

Barcode

Deskripsi

Kembali Simpan

Gambar 30: Wireframe Tambah Barang

6. Halaman Data Distributor

Stokku

Data Barang

Data Distributor

Barang Masuk

Barang Keluar

Akun

Data Distributor

Cari distributor

+ Tambah

No	Nama	Telepon	Alamat	Aksi
D001	Toko Makmur	081148575623	Jl. Swadaya RT.02/RW.05 No.24 Jakarta Timur	Detail
D002	Toko Melati	081288657574	Jl. Manggis RT.07/RW.03 No.62 Jakarta Barat	Detail
D003	Toko Cahaya	081904048061	Jl. Kecapi RT.01/RW.02 No.3 Jakarta Pusat	Detail
D004	Toko Mawar	089555810188	Jl. Manggis RT.04/RW.08 No.10 Jakarta Timur	Detail
D005	Toko Mekar	085801283475	Jl. Manggis RT.09/RW.01 No.30 Jakarta Selatan	Detail

< 1 2 3 >

Gambar 31: Wireframe Data Distributor

7. Halaman Tambah Distributor

Stokku

Data Barang

Data Distributor

Barang Masuk

Barang Keluar

Akun

Tambah Distributor

Nama Distributor*

Telepon

Alamat*

Kembali Simpan

Gambar 32: Wireframe Tambah Distributor

8. Halaman Detail Distributor

Stokku

Data Distributor

Detail Distributor

Nama Distributor* Toko Melati

Telepon 08114389012

Alamat* Jl. Raya Bogor RT03/RW05 No.45

Edit Distributor Hapus Distributor

Gambar 33: Wireframe Detail Distributor

9. Halaman Barang Masuk

Stokku

Data Barang Masuk

Cari Transaksi + Tambah

No	ID Transaksi	Tanggal	Distributor	Total Harga	Aksi
1	BM4792	12-05-2024	Toko Makmur	Rp 2.328.500	Detail
2	BM4037	25-04-2024	Toko Melati	Rp 376.100	Detail
3	BM4023	18-04-2024	Toko Cahaya	Rp 823.000	Detail
4	BM3219	11-02-2024	Toko Mawar	Rp 200.500	Detail
5	BM3109	04-02-2024	Toko Mekar	Rp 90.400	Detail

< 1 2 3 ... 10 >

Gambar 34: Wireframe Barang Masuk

10. Halaman Detail Barang Masuk

Stokku

Detail Barang Masuk

ID Transaksi: TRM-2847

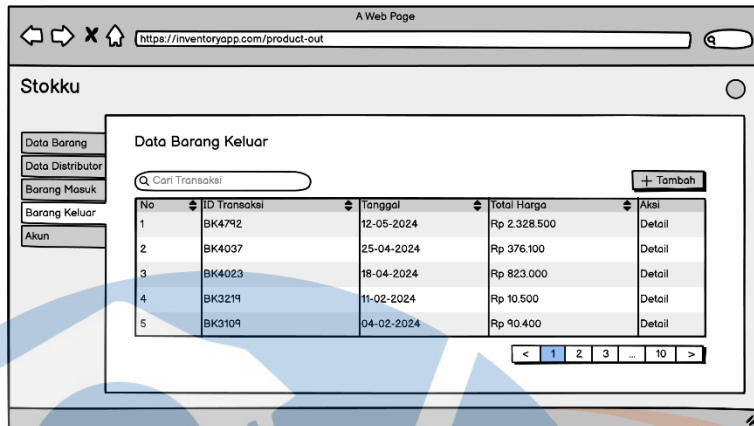
Tanggal: 12 Januari 2024

Distributor: Toko Melati

Nama Barang	Jumlah	Harga	Total Harga
Air Mineral 600 ml	120	Rp 3.600	Rp 432.000
Biskuit Kacang	60	Rp8.400	Rp 504.000
Tisu 250s	500	Rp 9.800	Rp 4.900.000
		Total	Rp 5.836.000

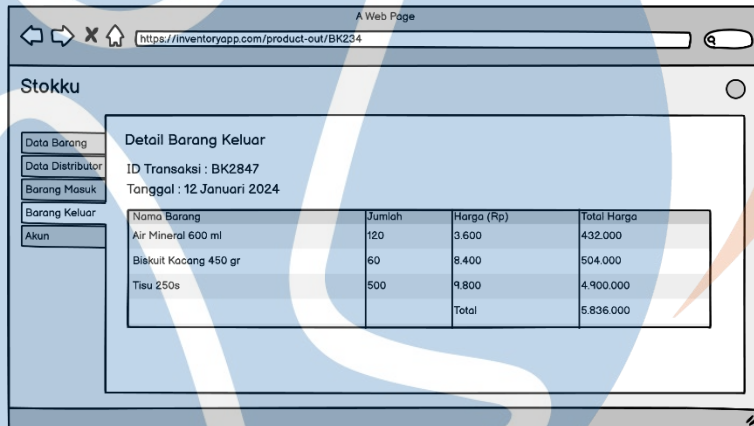
Gambar 35: Wireframe Detail Barang Masuk

11. Halaman Barang Keluar



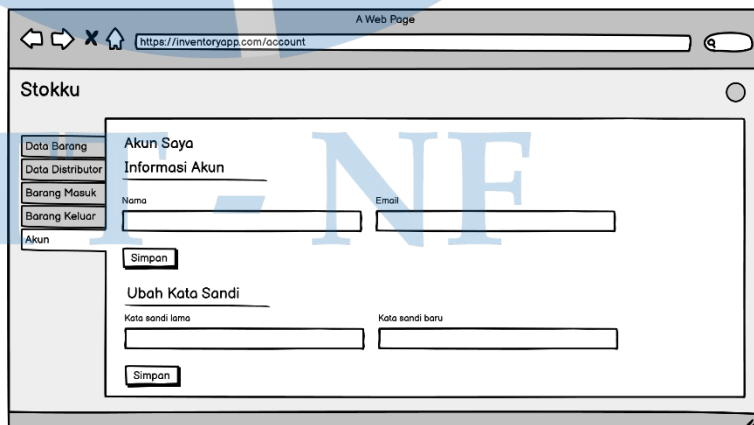
Gambar 36: Wireframe Data Barang Keluar

12. Halaman Detail Barang Keluar



Gambar 37: Wireframe Detail Barang Keluar

13. Halaman Akun



Gambar 38: Wireframe Akun

4.3 Implementasi

Implementasi merupakan hasil dari tahapan perancangan aplikasi yang telah dibuat sebelumnya, sehingga menghasilkan sebuah aplikasi yang sesuai dengan *requirement* yang sudah ada sebelumnya.

4.4.1 Implementasi Kode

Pada implementasi kode menampilkan hasil kode dari membuat skema *database* pada setiap tabel.

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('shops', function (Blueprint $table) {
15             $table->id();
16             $table->string('name')->nullable();
17             $table->timestamps();
18         });
19     }
20
21     /**
22     * Reverse the migrations.
23     */
24     public function down(): void
25     {
26         Schema::dropIfExists('shops');
27     }
28 };
```

Gambar 39: Skema Tabel *Shop*

STT - NF

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_units', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->foreignId('shop_id');
18             $table->timestamps();
19
20             $table->unique(['name', 'shop_id', 'shop_product_unit']);
21             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('product_units');
31     }
32 };
33
```

Gambar 40: Skema Tabel *Product Unit*

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_categories', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->foreignId('shop_id');
18             $table->timestamps();
19
20             $table->unique(['name', 'shop_id', 'shop_product_category']);
21             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('product_categories');
31     }
32 };
33
```

Gambar 41: Skema Tabel *Product Category*

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('products', function (Blueprint $table) {
15             $table->id();
16             $table->string('code')->nullable();
17             $table->string('name');
18             $table->text('description')->nullable();
19             $table->unsignedInteger('price');
20             $table->unsignedInteger('stock')->default(0);
21             $table->foreignId('shop_id');
22             $table->foreignId('unit_id')->nullable();
23             $table->foreignId('category_id')->nullable();
24             $table->timestamps();
25
26             $table->unique(['name', 'shop_id'], 'shop_product');
27             $table->unique(['code', 'shop_id'], 'shop_product_code');
28             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
29             $table->foreign('unit_id')->references('id')->on('product_units')->onDelete('cascade');
30             $table->foreign('category_id')->references('id')->on('product_categories')->onDelete('cascade');
31         });
32     }
33
34     /**
35      * Reverse the migrations.
36      */
37     public function down(): void
38     {
39         Schema::dropIfExists('products');
40     }
41 };
42
```

Gambar 42: Skema Tabel Product

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('distributors', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('address');
18             $table->string('phone');
19             $table->foreignId('shop_id');
20             $table->timestamps();
21
22             $table->unique(['name', 'shop_id'], 'shop_distributor');
23             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('distributors');
33     }
34 };
35
```

Gambar 43: Schema Tabel Distributor

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_outs', function (Blueprint $table) {
15             $table->id();
16             $table->date('date');
17             $table->unsignedInteger('total_price');
18             $table->foreignId('shop_id');
19             $table->timestamps();
20
21             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('product_outs');
31     }
32 };
33
```

Gambar 44: Skema Tabel Product Out

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_ins', function (Blueprint $table) {
15             $table->id();
16             $table->date('date');
17             $table->unsignedInteger('total_price');
18             $table->foreignId('distributor_id');
19             $table->foreignId('shop_id');
20             $table->timestamps();
21
22             $table->foreign('distributor_id')->references('id')->on('distributors')->onDelete('cascade');
23             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('product_ins');
33     }
34 };
35
```

Gambar 45: Skema Tabel Product In

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_in_details', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('product_in_id');
17             $table->foreignId('product_id');
18             $table->unsignedInteger('quantity');
19             $table->unsignedInteger('price');
20             $table->unsignedInteger('total_price');
21             $table->timestamps();
22
23             $table->foreign('product_in_id')->references('id')->on('product_ins')->onDelete('cascade');
24             $table->foreign('product_id')->references('id')->on('products')->onDelete('cascade');
25         });
26     }
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('product_in_details');
33     }
34 };
35
36
```

Gambar 46: Skema Tabel Product In Detail

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('product_out_details', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('product_out_id');
17             $table->foreignId('product_id');
18             $table->unsignedInteger('quantity');
19             $table->unsignedInteger('price');
20             $table->unsignedInteger('total_price');
21             $table->timestamps();
22
23             $table->foreign('product_out_id')->references('id')->on('product_outs')->onDelete('cascade');
24             $table->foreign('product_id')->references('id')->on('products')->onDelete('cascade');
25         });
26     }
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('product_out_details');
33     }
34 };
35
36
```

Gambar 47: Skema Tabel Product Out Detail

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->string('password');
19             $table->foreignId('shop_id');
20             $table->rememberToken();
21             $table->timestamps();
22             $table->foreign('shop_id')->references('id')->on('shops')->onDelete('cascade');
23         });
24     }
25
26     /**
27     * Reverse the migrations.
28     */
29     public function down(): void
30     {
31         Schema::dropIfExists('users');
32     }
33 };
34
35

```

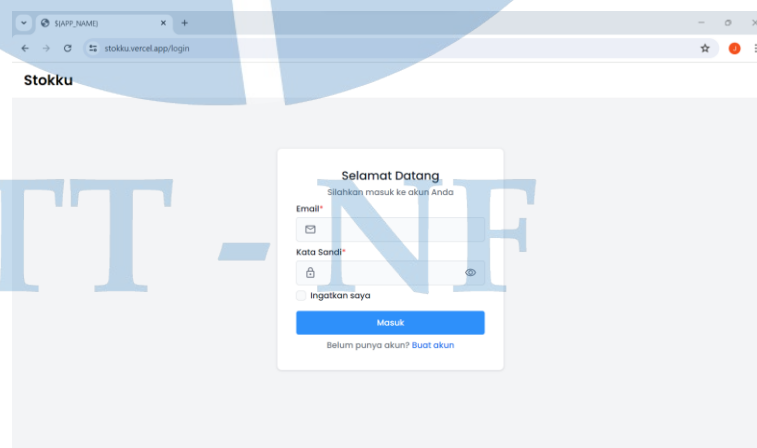
Gambar 48: Skema Tabel User

4.4.2 Implementasi Antarmuka

Pada implementasi antarmuka menampilkan hasil dari antarmuka aplikasi yang difoto pada setiap halaman beserta penjelasannya.

1. Halaman Login

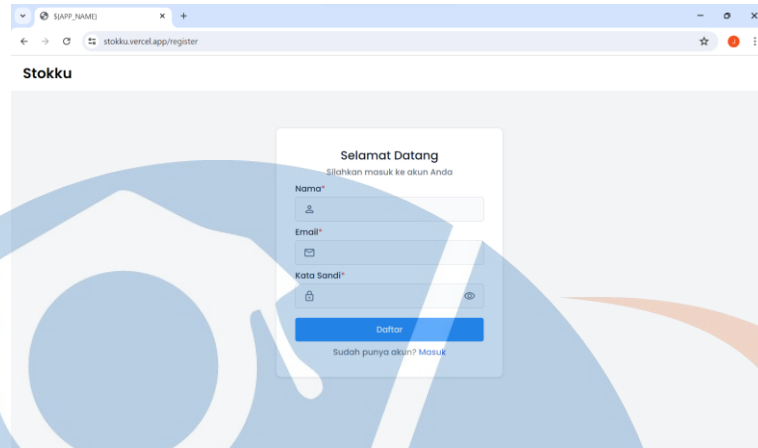
Pada halaman ini *user* dapat memasukkan *email* dan kata sandi yang telah dibuat sebelumnya



Gambar 49: Halaman *Login*

2. Halaman Register

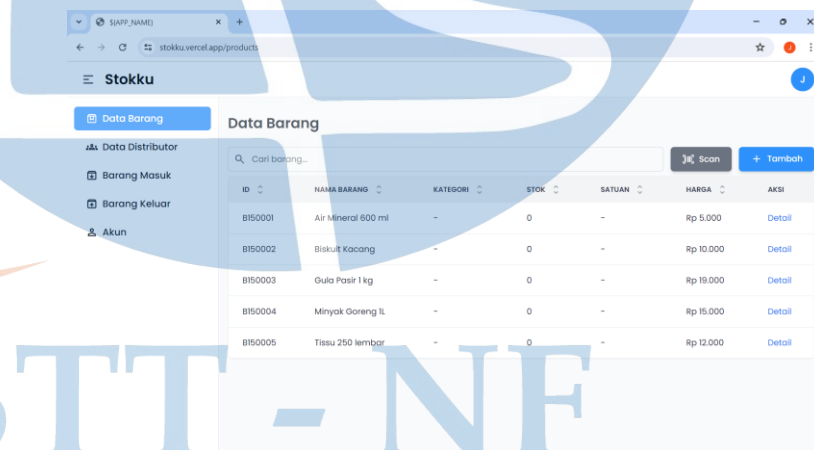
Pada halaman ini *user* bisa membuat akun dengan mengisi nama, *email*, dan kata sandi.



Gambar 50: Halaman *Register*

3. Halaman Data Barang

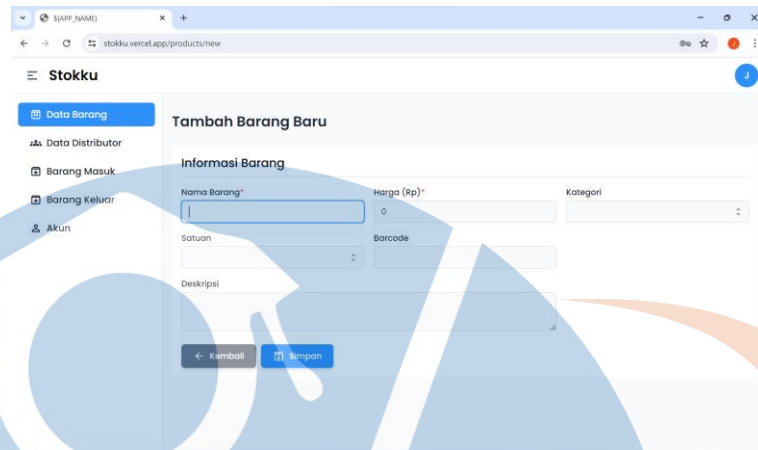
Pada halaman ini *user* dapat melihat data barang yang sudah ditambahkan sebelumnya dan juga *user* dapat mencari data barang di kolom pencarian serta mengurutkan datanya



Gambar 51: Halaman data barang

4. Halaman Tambah Data Barang

Pada halaman ini *user* dapat menambahkan data barang baru dengan mengisi nama barang, harga, kategori, satuan, *barcode*, dan deskripsi

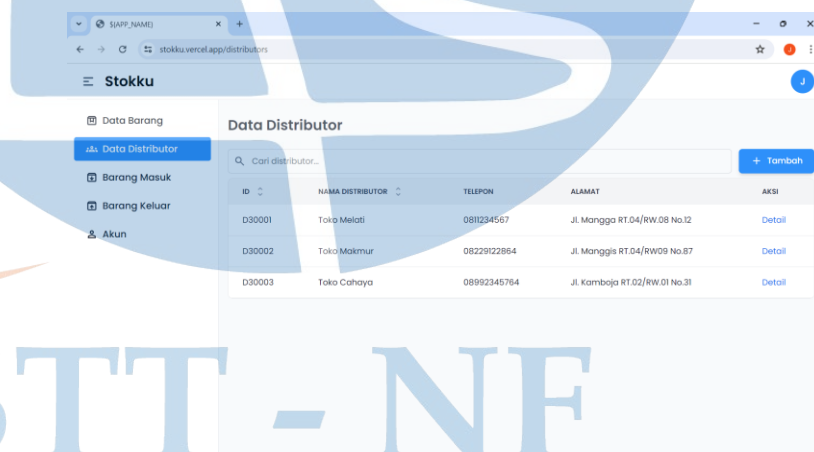


The screenshot shows a web browser window with the URL 'stokku.ventel.app/products/new'. The application header is 'Stokku'. A sidebar menu on the left includes 'Data Barang', 'Data Distributor', 'Barang Masuk', 'Barang Keluar', and 'Akun'. The main content area is titled 'Tambah Barang Baru' and contains a form with the following fields: 'Nama Barang' (text input), 'Harga (Rp)' (text input), 'Kategori' (dropdown menu), 'Satuan' (dropdown menu), 'Barcode' (text input), and 'Deskripsi' (text area). At the bottom of the form are two buttons: 'Kembali' (Back) and 'Simpan' (Save).

Gambar 52: Halaman Tambah Barang

5. Halaman Data Distributor

Pada halaman ini *user* dapat melihat data distributor yang sudah ditambahkan sebelumnya dan dapat mencari data distributor berdasarkan nama serta mengurutkan datanya.



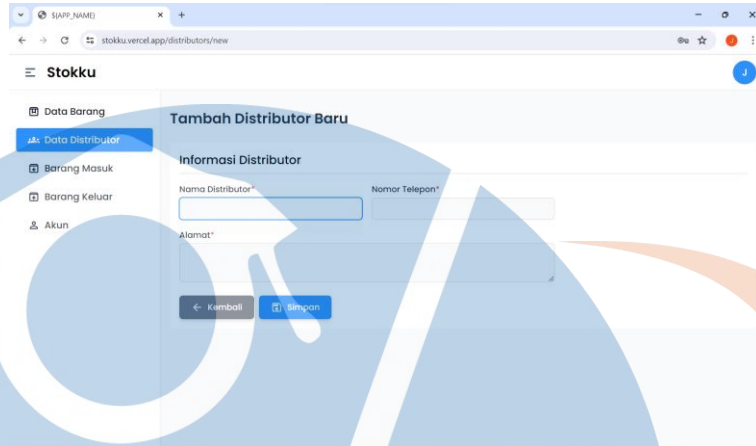
The screenshot shows a web browser window with the URL 'stokku.ventel.app/distributors'. The application header is 'Stokku'. A sidebar menu on the left includes 'Data Barang', 'Data Distributor', 'Barang Masuk', 'Barang Keluar', and 'Akun'. The main content area is titled 'Data Distributor' and features a search bar with the placeholder 'Cari distributor...' and a '+ Tambah' (Add) button. Below the search bar is a table with the following data:

ID	NAMA DISTRIBUTOR	TELEPON	ALAMAT	AKSI
D30001	Toko Melati	0811234567	Jl. Mangga RT.04/RW.08 No.12	Detail
D30002	Toko Makmur	08229122864	Jl. Manggis RT.04/RW09 No.87	Detail
D30003	Toko Cahaya	08992345794	Jl. Kambaja RT.02/RW.01 No.31	Detail

Gambar 53: Halaman Data Distributor

6. Halaman Tambah Data Distributor

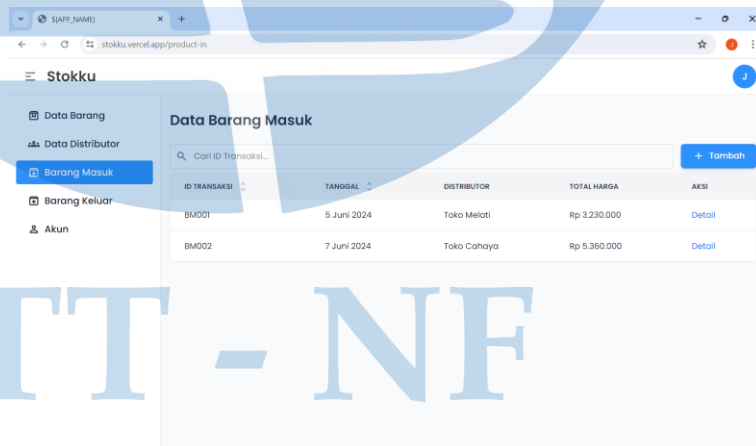
Pada halaman ini *user* dapat menambahkan data distributor baru dengan mengisi informasi distributor seperti nama, nomor telepon, dan alamat.



Gambar 54: Halaman Tambah Distributor

7. Halaman Data Barang Masuk

Pada halaman ini *user* dapat melihat semua data barang masuk dan juga dapat mencari data barang masuk berdasarkan ID transaksi serta dapat mengurutkan datanya.



Gambar 55: Halaman Data Barang Masuk

8. Halaman Tambah Data Barang Masuk

Pada halaman ini *user* dapat menambahkan data barang masuk dengan memasukkan data tanggal, nama distributor, nama barang, harga beli dan jumlah.

NAMA BARANG	JUMLAH	HARGA	TOTAL HARGA
Minyak Goreng IL	30	Rp 16.000	Rp 480.000
Biskuit Kacang	25	Rp 10.000	Rp 250.000
Tisu 250 lembar	250	Rp 10.000	Rp 2.500.000

Gambar 56: Halaman Tambah Barang Masuk

9. Halaman Detail Barang Masuk

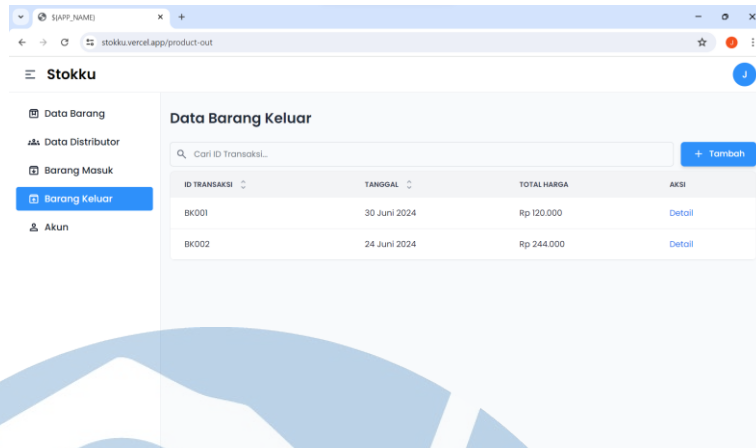
Pada halaman ini *user* dapat melihat rincian data barang masuk dengan tanggal dan nama distributornya.

NAMA BARANG	JUMLAH	HARGA	TOTAL HARGA
Minyak Goreng IL	30	Rp 16.000	Rp 480.000
Biskuit Kacang	25	Rp 10.000	Rp 250.000
Tisu 250 lembar	250	Rp 10.000	Rp 2.500.000
Jumlah			Rp 3.230.000

Gambar 57: Halaman Detail Barang Masuk

10. Halaman Data Barang Keluar

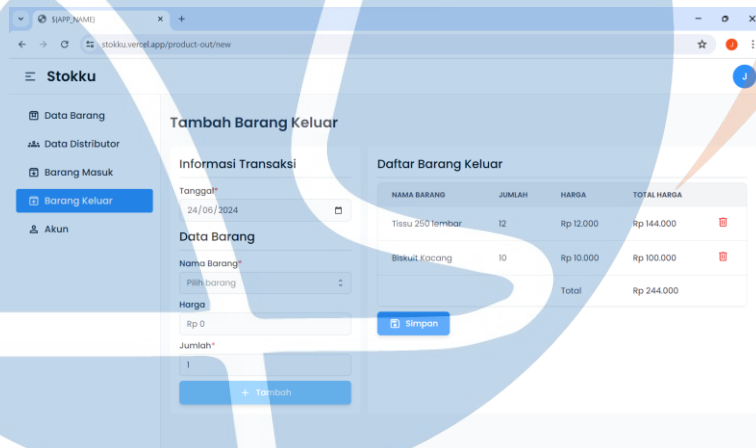
Pada Halaman ini *user* dapat melihat semua data barang keluar lengkap dengan tanggal dan total harganya.



Gambar 58: Halaman Data Barang Keluar

11. Halaman Tambah Data Barang Keluar

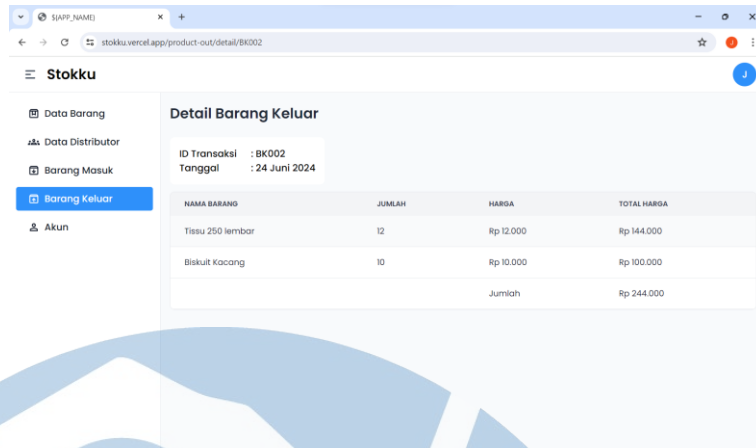
Pada halaman ini user dapat menambahkan data barang keluar dengan memasukkan data tanggal, nama barang, harga, dan jumlahnya.



Gambar 59: Halaman Tambah Barang Keluar

12. Halaman Detail Barang Keluar

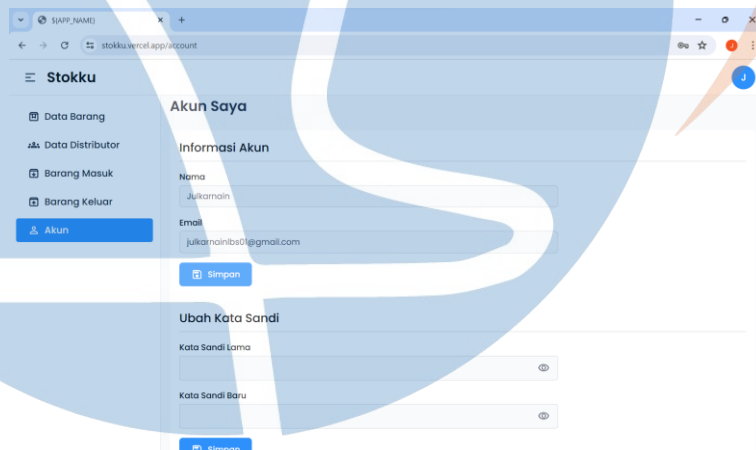
Pada halaman ini *user* dapat melihat rincian data barang keluar lengkap dengan tanggal, nama barang, jumlah, dan total harganya.



Gambar 60: Halaman Detail Barang Keluar

13. Halaman Akun

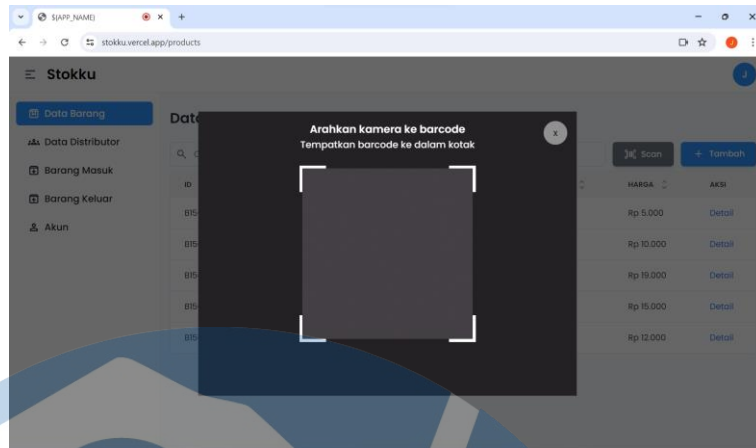
Pada halaman ini *user* dapat mengubah data akun dan kata sandi dengan mengisi kolom nama, *email*, kata sandi lama, dan kata sandi baru.



Gambar 61: Halaman Akun

14. Halaman *Scan Barcode*

Pada halaman ini *user* dapat memindai *barcode* barang dengan mengarahkan kamera ke *barcode* barang. Setelah itu, aplikasi akan mengarahkan ke detail barang jika data barang cocok atau mengarahkan ke halaman tambah barang jika barang tidak ditemukan.



Gambar 62: Halaman *Scan Barcode*

4.4 Pengujian Sistem

Evaluasi sistem merupakan sebuah proses untuk menguji aplikasi yang telah dikembangkan untuk mengetahui apakah fitur yang ada dalam aplikasi tersebut sudah berjalan dengan baik.

Dalam penelitian ini penulis melakukan evaluasi sistem melalui 3 tahapan yaitu:

1. *Black Box Testing*: Pengujian ini dilakukan oleh pengembang aplikasi
2. User Acceptance: Pengujian ini dilakukan oleh pengguna aplikasi
3. Kuesioner: Penilaian dari pengguna

4.5.1 Hasil *Black Box Testing*

No	Skenario Pengujian	Butir Uji	Hasil yang diharapkan	Hasil Pengujian
1	Mendaftar akun	Mendaftar akun dengan tidak mengisi salah satu kolom	Mendapatkan pesan <i>error</i> untuk mengisi semua kolom	Sesuai
		Mendaftar akun dengan <i>email</i> yang sama	Mendapatkan pesan <i>error email</i> sudah terdaftar	

		Mendaftar akun dengan mengisi semua kolom	Berhasil membuat akun	Sesuai
2	<i>Login</i> akun	<i>Login</i> akun dengan tidak mengisi salah satu kolom	Mendapatkan pesan <i>error</i> untuk mengisi semua kolom	Sesuai
		<i>Login</i> akun dengan <i>email</i> dan kata sandi salah	Mendapatkan pesan <i>error email</i> atau kata sandi salah	Sesuai
		<i>Login</i> akun dengan mengisi <i>email</i> dan kata sandi yang benar	Berhasil <i>login</i>	Sesuai
3	Tambah data barang	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai
		Mengisi nama barang yang sudah ada	Mendapatkan pesan <i>error</i> nama barang sudah ada	Sesuai
		Mengisi data barang yang baru	Berhasil menambah data barang	Sesuai
4	Mengelola data barang masuk	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai

		Mengisi data dengan lengkap	Berhasil menambah data barang masuk	Sesuai
5	Mengelola data barang keluar	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai
		Memasukan jumlah barang lebih dari jumlah stok yang ada	Mendapatkan pesan <i>error</i> untuk mengisi jumlah barang tidak melebihi stok	Sesuai
		Mengisi data dengan lengkap	Berhasil menambah data barang keluar	Sesuai
6	Mencari data barang	Tidak mengisi kolom pencarian	Menampilkan seluruh data barang	Sesuai
		Mengisi kolom pencarian dengan nama barang yang belum ada	Menampilkan pesan barang tidak ditemukan	Sesuai
		Mengisi kolom pencarian dengan nama barang yang sudah ada	Menampilkan data barang yang sesuai dengan nama	Sesuai
7	Mengelola akun	Tidak mengisi nama dan <i>email</i>	Mendapatkan pesan <i>error</i> untuk mengisi kolom nama dan email	Sesuai

		Mengisi kolom email dengan email yang sudah terdaftar	Mendapatkan pesan <i>error email</i> sudah terdaftar	Sesuai
		Mengisi nama dan email yang belum terdaftar	Berhasil mengedit data profil	Sesuai
		Tidak mengisi kata sandi lama dan kata sandi baru	Mendapatkan pesan <i>error</i> untuk mengisi kolom kata sandi lama dan kata sandi baru	Sesuai
		Mengisi kata sandi lama dengan salah	Mendapatkan pesan <i>error</i> kata sandi salah	Sesuai
		Mengisi kata sandi lama dan kata sandi baru dengan benar	Berhasil mengubah kata sandi lama dengan kata sandi baru	Sesuai
8	Scan barcode barang	Memindai <i>barcode</i> yang belum tersimpan di data barang	Mengarah ke halaman tambah barang dengan kolom <i>barcode</i> sesuai	Sesuai
		Memindai <i>barcode</i> yang sudah tersimpan di data barang	Mengarah ke detail barang dengan <i>barcode</i> yang sama	Sesuai

Tabel 18: Hasil *Black Box Testing*

Dalam pengujian dengan Black Box Testing di atas, terdapat 8 skenario pengujian. Setiap skenario pengujian terdapat 2 sampai 6 butir uji. Dari 8 skenario pengujian dengan Black Box Testing yang dilakukan oleh penulis, semua fitur telah berjalan dengan baik sesuai *requirement*. **Hasil dari skenario pengujian *Black Box Testing* dapat disimpulkan bahwa semua fitur sudah sesuai dengan *requirement*.**

4.5.2 User Acceptance Test

User Acceptance Test merupakan suatu proses pengujian aplikasi yang dilakukan oleh *user* yang bertujuan apakah aplikasi sudah sesuai dengan kebutuhan.

No	Fitur	Deskripsi	Hasil		
			User 1	User 2	User 3
1	<i>Register</i>	<i>User</i> dapat membuat akun	Sesuai	Sesuai	Sesuai
2	<i>Login</i>	<i>User</i> dapat masuk ke aplikasi dengan akun yang telah dibuat	Sesuai	Sesuai	Sesuai
3	Tambah Barang	<i>User</i> dapat menambahkan data barang	Sesuai	Sesuai	Sesuai
4	Cari Barang	<i>User</i> dapat mencari barang yang telah ditambahkan sebelumnya	Sesuai	Sesuai	Sesuai
5	<i>Scan Barcode</i> (mencari)	<i>User</i> dapat memindai <i>barcode</i> barang untuk mencari barang yang sudah ditambahkan	Sesuai	Sesuai	Sesuai
6	<i>Scan Barcode</i> (menambah)	<i>User</i> dapat memindai <i>barcode</i> barang untuk	Sesuai	Sesuai	Sesuai

		menambahkan barang yang belum ditambahkan			
7	Mengurutkan Barang	<i>User</i> dapat mengurutkan data barang berdasarkan ID, nama, harga, stok, kategori, dan satuan	Sesuai	Sesuai	Sesuai
8	Ubah data barang	Mengubah data barang yang sudah ada	Sesuai	Tidak sesuai	Sesuai
9	Hapus barang	Menghapus barang yang sudah ada	Sesuai	Sesuai	Sesuai
10	Tambah distributor	Menambah data distributor	Sesuai	Sesuai	Sesuai
11	Cari distributor	Mencari data distributor	Sesuai	Sesuai	Sesuai
12	Ubah distributor	Mengubah informasi distributor	Sesuai	Sesuai	Sesuai
13	Hapus distributor	Menghapus data distributor	Sesuai	Sesuai	Sesuai
14	Tambah barang masuk	Menambah stok barang	Sesuai	Sesuai	Sesuai
15	Detail barang masuk	Melihat rincian daftar barang masuk	Sesuai	Sesuai	Sesuai
16	Tambah barang keluar	Mengurangi stok barang	Sesuai	Sesuai	Sesuai
17	Detail barang keluar	Melihat rincian daftar barang keluar	Sesuai	Sesuai	Sesuai
18	Akun	Melihat informasi akun	Sesuai	Sesuai	Sesuai
19	Ubah data akun	Mengubah nama atau <i>email user</i>	Sesuai	Sesuai	Sesuai
20	Ubah kata sandi	Mengubah kata sandi <i>user</i>	Sesuai	Sesuai	Sesuai

Pengujian UAT memiliki 20 fitur yang diujikan oleh *user*, setiap fitur memiliki 2 hasil yaitu sesuai dan tidak sesuai. Dari 3 *user* yang melakukan tes ini, terdapat 1 *user* memberikan hasil tidak sesuai pada salah satu fitur. **Maka dapat disimpulkan dari hasil pengujian UAT bahwa 98,33% fitur sesuai dengan *requirement*.**

4.5.3 Hasil Kuesioner

Hasil Kuesioner dalam penelitian ini adalah hasil yang didapatkan dari 3 responden yang mengisi kuesioner untuk memberikan penilaian pada aplikasi yang dibuat. Penilaian pada kuesioner ini menggunakan skala *likert* untuk mendapatkan persentase dari hasil kuesioner. Kuesioner terdiri dari 7 pertanyaan berikut ini adalah pertanyaannya.

No	Pertanyaan
1	Apakah Anda dapat mendaftar dan masuk di aplikasi ini?
2	Apakah Anda dapat menambahkan, mengubah, dan menghapus data barang?
3	Apakah Anda dapat menambahkan, mengubah, dan menghapus data distributor?
4	Apakah Anda dapat menambahkan, mencari, dan melihat rincian barang masuk?
5	Apakah Anda dapat menambahkan, mencari, dan melihat rincian barang keluar?
6	Apakah Anda dapat mengubah data akun Anda?
7	Apakah fungsionalitas aplikasi sudah berjalan dengan baik?
8	Apakah tampilan aplikasi sudah terlihat baik?

Berikut ini adalah nilai dari skala likert yang digunakan untuk mendapatkan persentase dari hasil kuesioner

1. Sangat Setuju (SS) = 5
2. Setuju (S) = 4

3. Netral (N) = 3
4. Tidak Setuju (TS) = 2
5. Sangat Tidak Setuju (STS) = 1

Kuesioner diberikan kepada 3 responden yang menggunakan perangkat *desktop* dan *mobile* untuk memperoleh jawaban terkait aplikasi yang dikembangkan. Berikut adalah hasil kuesionernya.

No	Pertanyaan	Jawaban Responden				
		SS	S	RG	TS	STS
1	Apakah Anda dapat mendaftar dan masuk di aplikasi ini?	3				
2	Apakah Anda dapat menambahkan, mengubah, dan menghapus data barang?	2			1	
3	Apakah Anda dapat menambahkan, mengubah, dan menghapus data distributor?	3				
4	Apakah Anda dapat menambahkan, mencari, dan melihat rincian barang masuk?	3				
5	Apakah Anda dapat menambahkan, mencari, dan melihat rincian barang keluar?	3				

6	Apakah Anda dapat mengubah data akun Anda?	3				
7	Apakah fungsionalitas aplikasi sudah berjalan dengan baik?	2			1	
8	Apakah tampilan aplikasi sudah terlihat baik?	3				

Berikut ini adalah rumus perhitungan skala *likert* untuk kuesioner ini

$$\text{Hasil skor} = \frac{\sum(\text{Skor} * \text{Frekuensi})}{\text{Jumlah responden}}$$

$$\begin{aligned} \text{Hasil skor} &= (5 * 22 + 2 * 2) / 3 \\ &= 38 \end{aligned}$$

Hasil skor total yang didapatkan sebesar 38 lalu dibagi dengan 8 pertanyaan, maka hasilnya sebesar 4.75.

Terdapat rentang kepuasan dari skor yang didapatkan sebagai berikut:

1. Sangat Puas (4.5 – 5.0)
2. Puas (3.5 – 4.4)
3. Cukup Puas (2.5 – 3.4)
4. Tidak Puas (1.5 – 2.4)
5. Sangat Tidak Puas (1.0 – 1.4)

Menurut hasil skor yang diperoleh yaitu sebesar 4.75 maka nilai ini masuk ke kagetori sangat puas. **Maka dapat disimpulkan nilai user untuk aplikasi ini dalam hasil kuesioner termasuk ke kategori sangat puas.**

BAB V

KESIMPULAN DAN SARAN

Pada bab ini berisi mengenai penarikan kesimpulan dan saran dari penulis terhadap seluruh proses yang telah dilakukan untuk mendapatkan pengembangan aplikasi yang lebih baik.

5.1 Kesimpulan

Berdasarkan hasil pengujian pada penelitian ini maka didapatkan kesimpulan sebagai berikut:

1. Perancangan aplikasi sistem inventori menggunakan metode *waterfall* terdiri dari beberapa tahapan yaitu perencanaan, pemodelan, implementasi, dan pengujian. Tahap perencanaan dilakukan untuk mengidentifikasi kebutuhan pengguna, fungsionalitas sistem serta menentukan fitur. Tahap pemodelan dilakukan untuk mendesain arsitektur aplikasi seperti antarmuka dalam bentuk *wireframe*, struktur basis data, dan logika aplikasi. Tahap implementasi dilakukan untuk membuat aplikasi sesuai model yang sudah dibuat sebelumnya dengan membuat kode program bagian *frontend* menggunakan *framework* ReactJS dan *backend* menggunakan *framework* Laravel. Tahap pengujian dilakukan untuk memastikan semua fitur yang ada di aplikasi dapat berjalan dengan baik menggunakan Black Box Testing.
2. Berdasarkan hasil pengujian Black Box dan UAT yang telah dilakukan, aplikasi sistem inventori dapat berjalan dengan baik dan responsif di perangkat *desktop* maupun *mobile*.

5.2 Saran

Penelitian ini masih terdapat banyak kekurangan sehingga masih diperlukan pengembangan lebih lanjut untuk kedepannya. Terdapat beberapa saran yang penulis ajukan untuk pengembangan aplikasi pada penelitian yang akan datang sebagai berikut:

1. Aplikasi sistem inventori dapat dibuat dalam aplikasi *mobile* sehingga dapat diakses secara *offline*.

2. Dalam pengembangan berikutnya disarankan menggunakan metode perancangan lainnya seperti metode Agile.
3. Aplikasi dapat dikembangkan lebih lanjut seperti memiliki lebih dari satu *role* dan terhubung ke perangkat *barcode scanner*.



STT - NF

DAFTAR PUSTAKA

- [1] Busthomi, “Sebanyak 80% Pelaku UMKM Masih Catat Keuangan Secara Manual, Ini Masukan Bank OCBC.” [Online]. Available: <https://www.topbusiness.id/86319/sebanyak-80-pelaku-umkm-masih-catat-keuangan-secara-manual-ini-masukan-bank-ocbc.html>
- [2] A. Tiarasandy, “Inilah Kekurangan Mengatur Stock secara Manual.” [Online]. Available: <https://www.kompasiana.com/clodeo/5db7deb2097f3643fe6b85d2/inilah-kekurangan-mengatur-stock-secara-manual>
- [3] C. M. Annur, “Pengguna Internet di Indonesia Tembus 213 Juta Orang hingga Awal 2023.” [Online]. Available: <https://databoks.katadata.co.id/datapublish/2023/09/20/pengguna-internet-di-indonesia-tembus-213-juta-orang-hingga-awal-2023>
- [4] Renaldy and A. Rustam, “Perancangan Sistem Informasi Inventory Berbasis Web Pada Gudang Di Pt. Spin Warriors,” *J. Homepage*, vol. 4, no. 1, pp. 27–32, 2020, [Online]. Available: <http://jti.aisyahuniversity.ac.id/index.php/AJIEE>
- [5] B. Haqi and H. S. Setiawan, *Aplikasi Absensi Dosen dengan Java dan Smartphone sebagai Barcode Reader*. Jakarta: PT Elex Media Koputindo, 2019.
- [6] “1-D Linear Barcodes,” Cognex Corporation. [Online]. Available: <https://www.cognex.com/resources/symbologies/1-d-linear-barcodes>
- [7] A. Subagia, *Kolaborasi Laravel dan Database PostgreSQL di Linux*. Jakarta: PT Elex Media Koputindo, 2019.
- [8] Y. Yudhanto and H. A. Prasetyo, *Mudah Menguasai Framework Laravel*. Jakarta: PT Elex Media Koputindo, 2019.
- [9] M. Rizqy, “Apa itu SPA (Single Page Application),” Codepolitan. [Online]. Available: <https://www.codepolitan.com/blog/apa-itu-spa-single-page-application>
- [10] R. Fitri, *Pemrograman Basis Data Menggunakan MySQL*. Poliban Press, 2020.
- [11] M. P. Putri et al., *Sistem Manajemen Basis Data Menggunakan MYSQL*. 2013.

- [Online]. Available: <http://www.nber.org/papers/w16019>
- [12] A. B. P. Jazama, "Rancang Bangun Sistem Informasi Event Berbasis Web Menggunakan Framework Laravel," Depok, 2019.
- [13] R. Destriana, S. Maulana Husein, N. Handayani, and A. Tegar Prahara Siswanto, *Diagram UML Dalam Membuat Aplikasi Android Firebase "Studi Kasus Aplikasi Bank Sampah."* Deepublish, 2021.
- [14] R. Setiawan, "Black Box Testing Untuk Menguji Perangkat Lunak," Dicoding. [Online]. Available: <https://www.dicoding.com/blog/black-box-testing/>
- [15] "MEMAHAMI SKALA LIKERT DALAM PENELITIAN ILMIAH," Bina Nusantara. [Online]. Available: <https://accounting.binus.ac.id/2021/08/13/memahami-skala-likert-dalam-penelitian-ilmiah/>
- [16] R. Adwiya, "SISTEM INFORMASI INVENTORY BERBASIS WEB DENGAN MENGGUNAKAN MODEL WATERFALL," vol. 7, no. 2, pp. 277–285, 2023.
- [17] N. N. K. Sari and F. F. Purba, "Aplikasi Kasir Mobile Berbasis Android Untuk Usaha Mikro Kecil Dan Menengah," *J. Inf. Technol. Comput. Sci.*, vol. 1, no. 3, pp. 256–256, 2021, doi: 10.47111/jointecom.v1i3.8820.
- [18] A. Maydiantoro, "Model Penelitian Pengembangan," *Chem. Educ. Rev.*, vol. 3, no. 2, p. 185, 2020.

STT - NF

LAMPIRAN

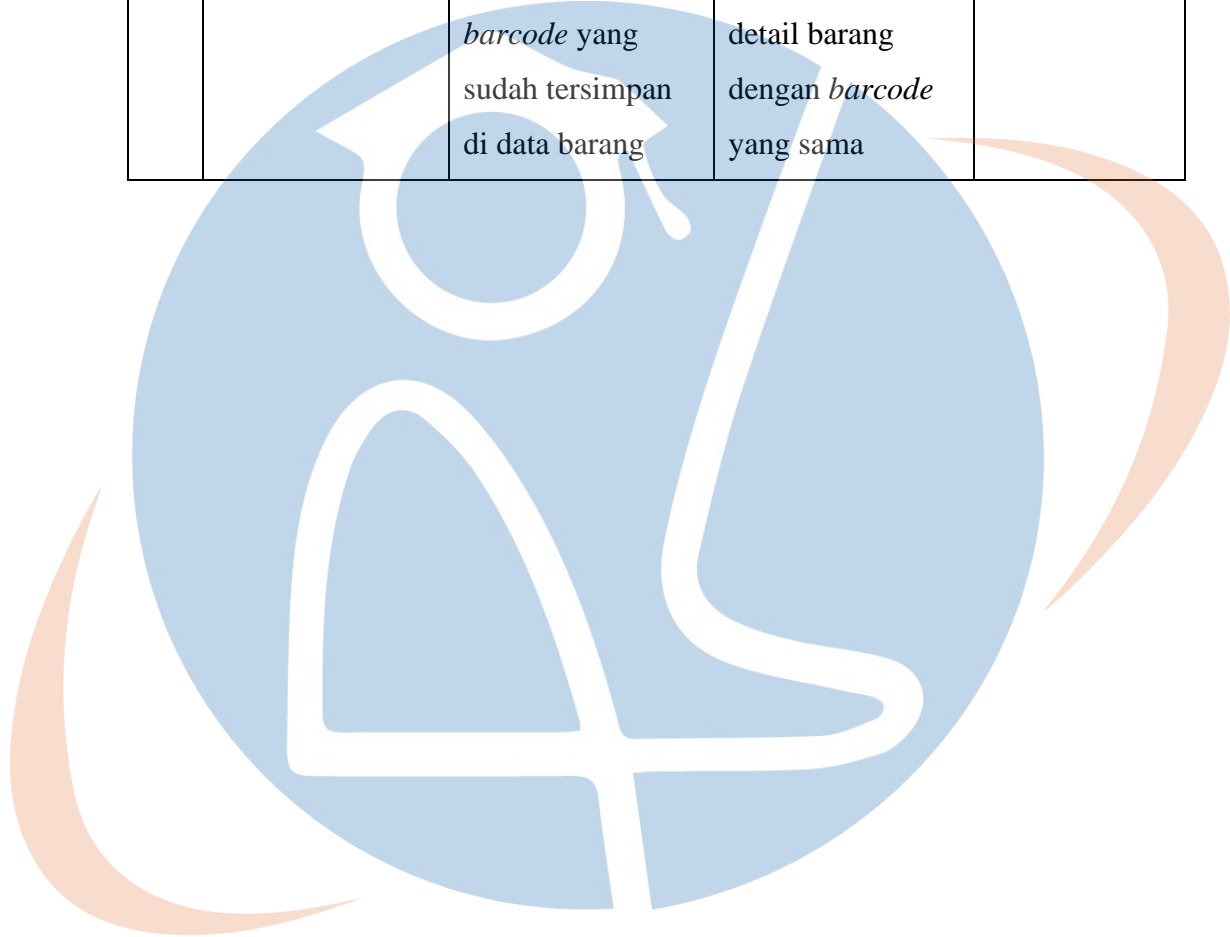
Lampiran 1: Hasil Black Box Testing

No	Skenario Pengujian	Butir Uji	Hasil yang diharapkan	Hasil Pengujian
1	Mendaftar akun	Mendaftar akun dengan tidak mengisi salah satu kolom	Mendapatkan pesan <i>error</i> untuk mengisi semua kolom	Sesuai
		Mendaftar akun dengan <i>email</i> yang sama	Mendapatkan pesan <i>error email</i> sudah terdaftar	Sesuai
		Mendaftar akun dengan mengisi semua kolom	Berhasil membuat akun	Sesuai
2	<i>Login</i> akun	<i>Login</i> akun dengan tidak mengisi salah satu kolom	Mendapatkan pesan <i>error</i> untuk mengisi semua kolom	Sesuai
		<i>Login</i> akun dengan <i>email</i> dan kata sandi salah	Mendapatkan pesan <i>error email</i> atau kata sandi salah	Sesuai
		<i>Login</i> akun dengan mengisi <i>email</i> dan kata sandi yang benar	Berhasil <i>login</i>	Sesuai
3	Tambah data barang	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai

		Mengisi nama barang yang sudah ada	Mendapatkan pesan <i>error</i> nama barang sudah ada	Sesuai
		Mengisi data barang yang baru	Berhasil menambah data barang	Sesuai
4	Mengelola data barang masuk	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai
		Mengisi data dengan lengkap	Berhasil menambah data barang masuk	Sesuai
5	Mengelola data barang keluar	Tidak mengisi kolom yang wajib diisi	Mendapatkan pesan <i>error</i> untuk mengisi kolom yang wajib diisi	Sesuai
		Memasukan jumlah barang lebih dari jumlah stok yang ada	Mendapatkan pesan <i>error</i> untuk mengisi jumlah barang tidak melebihi stok	Sesuai
		Mengisi data dengan lengkap	Berhasil menambah data barang keluar	Sesuai
6	Mencari data barang	Tidak mengisi kolom pencarian	Menampilkan seluruh data barang	Sesuai
		Mengisi kolom pencarian dengan	Menampilkan pesan barang tidak ditemukan	Sesuai

		nama barang yang belum ada		
		Mengisi kolom pencarian dengan nama barang yang sudah ada	Menampilkan data barang yang sesuai dengan nama	Sesuai
7	Mengelola akun	Tidak mengisi nama dan <i>email</i>	Mendapatkan pesan <i>error</i> untuk mengisi kolom nama dan email	Sesuai
		Mengisi kolom email dengan email yang sudah terdaftar	Mendapatkan pesan <i>error email</i> sudah terdaftar	Sesuai
		Mengisi nama dan email yang belum terdaftar	Berhasil mengedit data profil	Sesuai
		Tidak mengisi kata sandi lama dan kata sandi baru	Mendapatkan pesan <i>error</i> untuk mengisi kolom kata sandi lama dan kata sandi baru	Sesuai
		Mengisi kata sandi lama dengan salah	Mendapatkan pesan <i>error</i> kata sandi salah	Sesuai
		Mengisi kata sandi lama dan kata sandi baru dengan benar	Berhasil mengubah kata sandi lama dengan kata sandi baru	Sesuai

8	<i>Scan barcode barang</i>	Memindai <i>barcode</i> yang belum tersimpan di data barang	Mengarah ke halaman tambah barang dengan kolom <i>barcode</i> sesuai	Sesuai
		Memindai <i>barcode</i> yang sudah tersimpan di data barang	Mengarah ke detail barang dengan <i>barcode</i> yang sama	Sesuai



STT - NF

Lampiran 2: Hasil UAT

No	Fitur	Deskripsi	Hasil		
			<i>User 1</i>	<i>User 2</i>	<i>User 3</i>
1	<i>Register</i>	<i>User</i> dapat membuat akun	Sesuai	Sesuai	Sesuai
2	<i>Login</i>	<i>User</i> dapat masuk ke aplikasi dengan akun yang telah dibuat	Sesuai	Sesuai	Sesuai
3	Tambah Barang	<i>User</i> dapat menambahkan data barang	Sesuai	Sesuai	Sesuai
4	Cari Barang	<i>User</i> dapat mencari barang yang telah ditambahkan sebelumnya	Sesuai	Sesuai	Sesuai
5	<i>Scan Barcode</i> (mencari)	<i>User</i> dapat memindai <i>barcode</i> barang untuk mencari barang yang sudah ditambahkan	Sesuai	Sesuai	Sesuai
6	<i>Scan Barcode</i> (menambah)	<i>User</i> dapat memindai <i>barcode</i> barang untuk menambahkan barang yang belum ditambahkan	Sesuai	Sesuai	Sesuai
7	Mengurutkan Barang	<i>User</i> dapat mengurutkan data barang berdasarkan ID, nama, harga, stok, kategori, dan satuan	Sesuai	Sesuai	Sesuai
8	Ubah data barang	Mengubah data barang yang sudah ada	Sesuai	Tidak sesuai	Sesuai
9	Hapus barang	Menghapus barang yang sudah ada	Sesuai	Sesuai	Sesuai

10	Tambah distributor	Menambah data distributor	Sesuai	Sesuai	Sesuai
11	Cari distributor	Mencari data distributor	Sesuai	Sesuai	Sesuai
12	Ubah distributor	Mengubah informasi distributor	Sesuai	Sesuai	Sesuai
13	Hapus distributor	Menghapus data distributor	Sesuai	Sesuai	Sesuai
14	Tambah barang masuk	Menambah stok barang	Sesuai	Sesuai	Sesuai
15	Detail barang masuk	Melihat rincian daftar barang masuk	Sesuai	Sesuai	Sesuai
16	Tambah barang keluar	Mengurangi stok barang	Sesuai	Sesuai	Sesuai
17	Detail barang keluar	Melihat rincian daftar barang keluar	Sesuai	Sesuai	Sesuai
18	Akun	Melihat informasi akun	Sesuai	Sesuai	Sesuai
19	Ubah data akun	Mengubah nama atau <i>email user</i>	Sesuai	Sesuai	Sesuai
20	Ubah kata sandi	Mengubah kata sandi <i>user</i>	Sesuai	Sesuai	Sesuai

STT - NF

Lampiran 3: Hasil Kuesioner

Nama	Perangkat	Apakah Anda dapat mendaftar dan masuk di aplikasi ini?	Apakah Anda dapat menambah, mengubah, dan menghapus data barang?	Apakah Anda dapat menambah, mengubah, dan menghapus data distributor ?	Apakah Anda dapat menambah, mencari, dan melihat rincian barang masuk?	Apakah Anda dapat menambah, mencari, dan melihat rincian barang keluar?	Apakah Anda dapat mengubah data akun Anda?	Apakah fungsionalitas aplikasi sudah berjalan dengan baik?	Apakah tampilan aplikasi sudah terlihat baik?
Robiul Awal	<i>Desktop</i>	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
Muhammad Qiyam Bil Muzakki	<i>Desktop</i>	Sangat Setuju	Tidak Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Tidak Setuju	Sangat Setuju
Rizal Abimanyu	<i>Mobile</i>	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju