



**KERJA PRAKTIK - IF184801**

**Perancangan dan Implementasi Sistem Informasi Inventaris  
Barang Perusahaan GUMUGU**

Jl. Gabus No.40B Gp. Bandar Baru Kec. Kuta Alam, Banda Aceh  
Indonesia, 23245

Periode: 1 September 2021 – 30 November 2021

**Oleh:**

Kana Rekha

0511184000001

**Pembimbing Jurusan**

Adhatus Solichah Ahmadiyah S.Kom.,M.Sc

**Pembimbing Lapangan**

Aslan Saputra

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2022



**KERJA PRAKTIK - IF184801**

**Perancangan dan Implementasi Sistem Informasi Inventaris  
Barang Perusahaan GUMUGU**

Jl. Gabus No.40B Gp. Bandar Baru Kec. Kuta Alam, Banda Aceh  
Indonesia, 23245

Periode: 1 September 2021 – 30 November 2021

Oleh:

Kana Rekha                      0511184000001

**Pembimbing Jurusan**

Adhatus Solichah Ahmadiyah S.kom., M.Sc

**Pembimbing Lapangan**

Aslan Saputra

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2022

## DAFTAR ISI

DAFTAR ISI.....	iii
DAFTAR GAMBAR .....	vii
DAFTAR KODE SUMBER .....	ix
DAFTAR TABEL .....	xi
LEMBAR PENGESAHAN.....	xiii
KATA PENGANTAR.....	xvii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Tujuan.....	2
1.3. Manfaat.....	2
1.4. Rumusan Masalah .....	2
1.5. Lokasi dan Waktu Kerja Praktik .....	3
1.6. Metodologi Kerja Praktik.....	3
1.6.1. Perumusan Masalah.....	3
1.6.2. Studi Literatur.....	4
1.6.3. Pengujian dan Evaluasi.....	4
1.6.4. Analisis dan Perancangan Sistem .....	4
1.6.5. Implementasi Sistem .....	4
1.6.6. Kesimpulan dan Saran.....	5
1.7. Sistematika Laporan .....	5
1.7.1. Bab I Pendahuluan.....	5

1.7.2.	Bab II Profil Perusahaan.....	5
1.7.3.	Bab III Tinjauan Pustaka .....	5
1.7.4.	Bab IV Analisis dan Perancangan Infrastruktur Sistem.....	6
1.7.5.	Bab V Implementasi Sistem .....	6
1.7.6.	Bab VI Pengujian dan Evaluasi .....	6
1.7.7.	Bab VII Kesimpulan dan Saran.....	6
<b>BAB II PROFIL PERUSAHAAN .....</b>		<b>7</b>
2.1.	Profil PT. Gumugu Teknologi Kreatif.....	7
<b>BAB III TINJAUAN PUSTAKA.....</b>		<b>9</b>
3.1.	Visual Studio code.....	9
3.2.	CodeIgniter 4.....	9
3.3.	HTML.....	10
3.4.	PHP.....	10
3.5.	CSS.....	10
3.6.	JavaScript .....	11
3.7.	MySQL.....	11
<b>BAB IV ANALISIS DAN DESAIN SISTEM.....</b>		<b>13</b>
4.1.	Analisis Sistem .....	13
4.1.1.	Definisi Umum Aplikasi.....	14
4.1.2.	Proses Bisnis.....	15
4.1.3.	Deskripsi Fungsional .....	16
4.2.	Perancangan Sistem.....	32

4.2.1. Perancangan Tabel dalam Sistem .....	32
<b>BAB V IMPLEMENTASI SISTEM.....</b>	<b>37</b>
5.1. Implementasi Source Code.....	37
5.1.1. Routes.....	37
5.1.2. Controllers .....	39
5.1.3 Folder Models.....	50
5.1.4 Folder View .....	57
5.2. Implementasi Antarmuka Pengguna.....	89
5.2.1. Sisi Umum.....	89
5.2.2. Sisi Administrator.....	99
5.2.3 Sisi Karyawan.....	105
<b>BAB VI PENGUJIAN DAN EVALUASI.....</b>	<b>107</b>
6.1. Tujuan Pengujian.....	107
6.2. Kriteria Pengujian.....	107
6.3. Skenario Pengujian.....	107
6.4. Evaluasi Pengujian .....	110
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>113</b>
7.1. Kesimpulan.....	113
7.2. Saran.....	113
<b>DAFTAR PUSTAKA.....</b>	<b>115</b>
<b>LAMPIRAN .....</b>	<b>117</b>
<b>BIODATA PENULIS I .....</b>	<b>123</b>

*[Halaman sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 4.1 Use Case Sistem Informasi Inventaris Barang .....	16
Gambar 5. 1 Halaman Login.....	90
Gambar 5. 2 Halaman Data Barang.....	90
Gambar 5. 3 Halaman Tambah Data barang.....	91
Gambar 5. 4 Halaman Ubah Data Barang.....	92
Gambar 5. 5 Halaman Detail Data Barang.....	92
Gambar 5. 6 Halaman Hapus Data Barang .....	93
Gambar 5. 7 Halaman Cari Data Barang .....	93
Gambar 5. 8 Halaman Pemasukan Barang.....	94
Gambar 5. 9 Halaman Tambah Pemasukan Barang .....	95
Gambar 5. 10 Halaman Ubah Pemasukan Barang .....	95
Gambar 5. 11 Halaman Hapus Pemasukan Barang.....	96
Gambar 5. 12 Halaman Cari Pemasukan Barang .....	97
Gambar 5. 13 Halaman Penjualan Barang .....	98
Gambar 5. 14 Halaman Tambah Penjualan Barang .....	98
Gambar 5. 15 Halaman Cari Penjualan Barang .....	99
Gambar 5. 16 Halaman Dashboard.....	100
Gambar 5. 17 Halaman Edit Penjualan Barang .....	100
Gambar 5. 18 Halaman Hapus Penjualan Barang.....	101
Gambar 5. 19 Halaman Data Karyawan.....	102
Gambar 5. 20 Halaman Tambah Karyawan.....	102
Gambar 5. 21 Halaman Edit Karyawan.....	103
Gambar 5. 22 Halaman Hapus Karyawan.....	104
Gambar 5. 23 Halaman Cari Karyawan.....	104
Gambar 5. 24 Halaman Dashboard.....	105
Gambar 5. 25 Halaman Penjualan Barang.....	106

*[Halaman sengaja dikosongkan]*



## DAFTAR KODE SUMBER

Kode Sumber 5. 1 Halaman Routes .....	38
Kode Sumber 5. 2 Halaman Auth.....	39
Kode Sumber 5. 3 Halaman Home .....	40
Kode Sumber 5. 4 Halaman Item .....	42
Kode Sumber 5. 5 Halaman Pages .....	44
Kode Sumber 5. 6 Halaman Pemasukan.....	47
Kode Sumber 5. 7 Halaman ItemModel .....	51
Kode Sumber 5. 8 Halaman SalesModel .....	53
Kode Sumber 5. 9 Halaman StockModel .....	54
Kode Sumber 5. 10 Halaman SupplyModel.....	55
Kode Sumber 5. 11 Halaman KaryawanModel .....	56
Kode Sumber 5. 12 Halaman View Dashboard .....	58
Kode Sumber 5. 13 Halaman View Login .....	61
Kode Sumber 5. 14 Halaman View ItemCreate.....	63
Kode Sumber 5. 15 Halaman View ItemEdit .....	65
Kode Sumber 5. 16 Halaman View ItemIndex .....	67
Kode Sumber 5. 17 Halaman View Itemview .....	68
Kode Sumber 5. 18 Halaman View SalesCreate.....	71
Kode Sumber 5. 19 Halaman View SalesEdit .....	72
Kode Sumber 5. 20 Halaman View SalesIndex .....	75
Kode Sumber 5. 21 Halaman View SupplyCreate .....	76
Kode Sumber 5. 22 Halaman View SupplyEdit.....	78
Kode Sumber 5. 23 Halaman View Supplyindex.....	80
Kode Sumber 5. 24 Halaman View KaryawanCreate.....	83
Kode Sumber 5. 25 Halaman View KaryawanEdit .....	85
Kode Sumber 5. 26 Halaman View KaryawanIndex .....	87
Kode Sumber 5. 27 Halaman View Layout .....	89



## **DAFTAR TABEL**

Tabel 1. Spesifikasi kasus menambah karyawan .....	17
Tabel 2. Spesifikasi Kasus Mengubah Karyawan .....	18
Tabel 3. Spesifikasi Kasus Menghapus Karyawan.....	19
Tabel 4. Spesifikasi Kasus Mencari Karyawan .....	20
Tabel 5. Spesifikasi Kasus Menambah Barang.....	21
Tabel 6. Spesifikasi Kasus Mengubah Barang .....	22
Tabel 7. Spesifikasi Kasus Menghapus Barang .....	23
Tabel 8. Spesifikasi Kasus Melihat Detail Barang .....	24
Tabel 9. Spesifikasi Kasus Mencari Barang .....	24
Tabel 10. Spesifikasi Kasus Menambah Pemasukan Barang .....	25
Tabel 11. Spesifikasi Kasus Megubah Pemasukan Barang .....	26
Tabel 12. Spesifikasi Kasus Menghapus Pemasukan Barang ...	27
Tabel 13. Spesifikasi Kasus Menghapus Pemasukan Barang ....	28
Tabel 14. Spesifikasi Kasus Mencari Pemasukan Barang .....	29
Tabel 15. Spesifikasi Kasus Mengubah Penjualan Barang .....	30
Tabel 16. Spesifikasi Kasus Menghapus Penjualan Barang.....	31
Tabel 17. Spesifikasi Kasus Mencari Penjualan Barang .....	32
Tabel 18. Tabel Barang.....	33
Tabel 19 Tabel Stok .....	33
Tabel 20 Tabel Pemasukan Barang .....	34
Tabel 21 Tabel Penjualan Barang.....	34
Tabel 22 Tabel Status Barang .....	35
Tabel 23 Tabel Jenis Pengguna .....	35
Tabel 24 Tabel Karyawan .....	36
Tabel 25. Penjelasan Kode Sumber 5.1 .....	38
Tabel 26 Penjelasan Kode Sumber 5.2.....	39
Tabel 27. Penjelasan Kode sumber 5.3 .....	40
Tabel 28. Penjelasan Kode sumber 5.4 .....	42
Tabel 29. Penjelasan Kode Sumber 5.5 .....	44

Tabel 30. Penjelasan Kode Sumber 5.6.....	45
Tabel 31. Penjelasan Kode Sumber 5.6.....	47
Tabel 32. Penjelasan Kode Sumber 5.8.....	49
Tabel 33. Penjelasan Kode sumber 5.7 .....	51
Tabel 34. Penjelasan Kode sumber 5.8 .....	53
Tabel 35. Penjelasan Kode Sumber 5.9.....	54
Tabel 36. Penjelasan Kode Sumber 5.10 .....	55
Tabel 37. Penjelasan Kode Sumber 5.11 .....	56
Tabel 38. Daftar Skenario Pengujian .....	110
Tabel 39. Daftar Pengujian berdasarkan kebutuhan fungsional	111
Tabel 40. Hasil Evaluasi Pengujian .....	112

**LEMBAR PENGESAHAN  
KERJA PRAKTIK**

**Perancangan dan Implementasi Sistem Informasi  
Inventaris Barang Perusahaan GUMUGU**

Oleh:

Kana Rekha

0511184000001

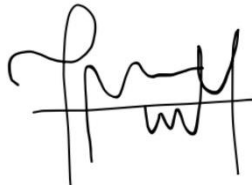
Disetujui oleh Pembimbing Kerja Praktik:

1. Adhatus Solichah  
Ahmadiyah S.Kom., M.Sc.  
NIP. 198508262015042002



(Pembimbing Departemen)

2. Aslan Saputra



(Pembimbing Lapangan)

*[Halaman ini sengaja dikosongkan]*

## **Perancangan dan Implementasi Sistem Informasi Inventaris Barang Perusahaan GUMUGU**

Nama Mahasiswa : Kana Rekha  
NRP : 0511184000001  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Departemen : Adhatus Solichah Ahmadiyah,  
S.Kom., M.Sc.  
Pembimbing Lapangan : Aslan Saputra

### **ABSTRAK**

*Masa era digital semakin berkembang, pemanfaatan teknologi khususnya dalam berbagi kebutuhan informasi yang dapat membantu suatu organisasi maupun individu lainnya. Pemanfaatan teknologi khususnya dalam membuat website sistem informasi inventaris barang yang dapat membantu perusahaan Gumugu dalam berkembang. Tampilan antarmuka yang memiliki fitur dalam mengelola barang, mengelola pemasukan dan pengeluaran barang. Tampilan yang menarik membantu pengguna dalam memahami dan memakai website tersebut dengan baik dan mudah. Sistem yang kami kembangkan menggunakan framework Codeigniter 4, database MySQL, memanfaatkan HTML, CSS, Javascript dan Boostraps dalam membuat tampilan website. Dengan adanya sistem informasi inventaris barang ini diharapkan dapat membantu perusahaan Gumugu dalam mengelola barang dengan efektif dan efisien.*

***Kata Kunci : Website, Codeigniter4, Inventaris Barang***

*[Halaman ini sengaja dikosongkan]*



## **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Perancangan dan Implementasi Sistem Informasi Inventaris Barang di perusahaan GUMUGU.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Orang tua penulis.
2. Ibu Adhatus Solichah Ahmadiyah, S.Kom., M.Sc. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Aslan Saputra selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan kerja praktik.

Surabaya, 10 Juni 2022

Kana Rekha

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Masa era digital semakin berkembang, pemanfaatan teknologi khususnya dalam berbagi kebutuhan informasi dapat berlangsung dengan cepat, efisien dan efektif. Website merupakan salah satu dari hasil perkembangan teknologi informasi yang bisa digunakan sebagai sarana dalam mendapatkan pertukaran informasi pada suatu instansi. Penggunaan website juga sangat mudah digunakan oleh orang awam dan dapat diakses kapanpun dan dimanapun. Website dapat digunakan dalam berbagai perangkat keras seperti komputer, laptop, Ipad maupun handphone.

Inventaris merupakan daftar persediaan barang yang dimiliki oleh sebuah perusahaan dan dikelola untuk nantinya dijual. Tentu dalam mengelola inventaris barang, sebuah perusahaan perlu memiliki catatan pengelolaan pemasukan barang dan pengeluaran barang dalam kurung waktu tertentu supaya pemimpin perusahaan mengetahui alur masuk dan keluarnya suatu barang. Salah satu faktor gangguan pencatatan inventaris yang dimiliki perusahaan adalah pencatatan inventaris barang masih dilakukan secara manual. Hal ini dapat menyebabkan kesalahan pencatatan yang dapat dilakukan oleh manusia secara tidak sadar.

Pada perusahaan Gumugu pencatatan barang masih dilakukan secara manual yaitu menggunakan tools excel ataupun buku kertas. Jika terjadi penambahan data barang, karyawan harus mengkaji semua data tersebut secara rinci sehingga proses pengerjaan yang dilakukan membutuhkan waktu yang lama dan tidak akurat. Penyimpanan data barang juga menjadi bagian penting

dalam mengelola data secara efisien dan efektif. Untuk mengatasi masalah tersebut, perusahaan membutuhkan sebuah sistem informasi inventaris barang berbasis website untuk membantu para karyawan dalam mengelola data barang yang bisa dilakukan pada setiap entri data barang atau melakukan perubahan data dengan cepat dan akurat.

## **1.2. Tujuan**

Tujuan kerja praktik ini adalah membantu PT. Gumugu Teknologi Kreatif untuk mengembangkan sebuah sistem informasi inventaris barang berbasis website secara sistematis dan tersruktur sehingga dapat digunakan oleh karyawan dalam mengatasi sistem rekapitulasi data secara manual. Sistem ini diharapkan dapat mempercepat pengolahan dan penyimpanan data barang, data pemasukan barang dan data pengeluaran barang.

## **1.3. Manfaat**

Adapun manfaat dari pembuatan situs web sekolah dalam pelaksanaan kerja praktik ini adalah sebagai berikut :

1. Manajemen informasi bagi karyawan perusahaan dalam mengelola informasi inventaris barang secara efektif dan efisien.
2. Data inventaris barang dapat tersimpan dalam sistem dengan jelas dan dapat diakses dengan mudah

## **1.4. Rumusan Masalah**

Adapun Rumusan masalah pada kerja praktik ini adalah sebagai berikut:

1. Bagaimana sistem informasi inventaris barang dapat berguna untuk memudahkan pengolahan data barang pada perusahaan PT. Gumugu Teknologi Kreatif ?

## **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja Praktik dilaksanakan pada waktu dan tempat sebagai berikut :

Lokasi : Kantor PT Gumugu Teknologi Informasi

Alamat : Jl. Gabus No.40B Gp. Bandar Baru Kec. Kuta Alam, Banda Aceh Indonesia, 23245

Waktu : 1 September 2021 - 30 November 2021

Hari Kerja : Senin – Jumat

Jam Kerja : 09.00 WIB – 17.00 WIB

## **1.6. Metodologi Kerja Praktik**

Tahapan pengerjaan kerja praktik meliputi dapat dijabarkan sebagai berikut :

### **1.6.1. Perumusan Masalah**

Untuk mengetahui kebutuhan fungsional dan non fungsional dari sistem, saya mengikut rapat dengan tim perusahaan. Pada rapat tersebut karyawan menyampaikan permasalahan yang dihadapi saat mencatat barang barang. Setelah itu, saya dan tim pengembang website dari perusahaan melakukan diskusi untuk mengetahui fitur apa saja yang dibutuhkan dari permasalahan tersebut untuk ditampilkan pada website.

### **1.6.2. Studi Literatur**

Setelah ditentukan kebutuhan sistem dari website seperti kerangka kerja yang digunakan, database, tools yang digunakan, dilakukan studi literatur mengenai cara implementasinya, dalam hal ini tujuan dilakukan terhadap kerangka kerja CodeIgniter 4, database MySQL dengan bahasa pemrograman PHP, HTML, CSS dan Javascript. Pada tahap ini dilakukan proses pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam mengimplementasikan code dan query data.

### **1.6.3. Pengujian dan Evaluasi**

Setelah website yang telah direncanakan telah jadi, perlu adanya evaluasi untuk menguji apakah website sesuai dengan harapan client. Pembimbing lapangan akan memeriksa apakah fitur yang dibuat sudah sesuai dengan tujuan dan kebutuhan sistem yang sudah dianalisis. Hal ini akan menentukan hasil evaluasi terhadap kinerja sistem.

### **1.6.4. Analisis dan Perancangan Sistem**

Setelah melakukan tinjauan pustaka, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem. Pada website ini saya menggunakan arsitektur desain MVC (Model - View - Controller).

### **1.6.5. Implementasi Sistem**

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini saya merancang desain dengan kerangka kerja baru dan merancang database baru.

### **1.6.6. Kesimpulan dan Saran**

Pengujian yang dilakukan ini telah memenuhi syarat dan sesuai dengan yang diinginkan. Fitur aplikasi berjalan dengan lancar dan sesuai dengan kebutuhan. Dalam menjalankan website ini, diperlukan penjelasan secara detail terkait alur kerja website. Sehingga pengguna dapat memahami alur kerja website dengan baik.

## **1.7. Sistematika Laporan**

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut :

### **1.7.1. Bab I Pendahuluan**

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

### **1.7.2. Bab II Profil Perusahaan**

Bab ini berisi gambaran umum mengenai profil perusahaan PT. Gumugu Teknologi Kreatif yaitu tempat pelaksanaan kerja praktik.

### **1.7.3. Bab III Tinjauan Pustaka**

Bab ini berisi dasar teori tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di PT. Gumugu Teknologi Kreatif.

**1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem**

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

**1.7.5. Bab V Implementasi Sistem**

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

**1.7.6. Bab VI Pengujian dan Evaluasi**

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

**1.7.7. Bab VII Kesimpulan dan Saran**

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.



## **BAB II**

### **PROFIL PERUSAHAAN**

#### **2.1. Profil PT. Gumugu Teknologi Kreatif**

PT. Gumugu Teknologi Kreatif adalah perusahaan IT yang bergerak untuk membangun ekosistem paperless dan green economy di Indonesia yang terletak di Kota Banda Aceh tepatnya di Jalan Gabus no.40 Gp. Bandar Baru , kec. Kuta alam, Kota Banda Aceh, 23245. Perusahaan ini didirikan sejak tahun 2015. Terdapat 5 orang yang bekerja di perusahaan ini antara lain terdiri dari CEO, CMO dan 2 karyawan IT dan 1 sekretaris. Perusahaan ini memiliki tujuan dalam meningkatkan perekonomian Indonesia dan mensejahterakan masyarakat melalui penggunaan teknologi yang lebih mudah dan lebih cepat. Perusahaan ini dapat membantu pelanggan yang membutuhkan kustomisasi website dalam waktu yang cepat dan harga yang terjangkau.

Perusahaan ini memiliki 4 produk utama yang dikembangkan yaitu Gu-CMS, Gu-Edu, Gu-Spark, Gu-Docs. Produk Gu-CMS merupakan wadah yang dapat digunakan oleh klien dalam mengkustomisasi *template* website yang diinginkan dengan didukung beberapa plugin khusus. Produk Gu-Edu merupakan paket edukasi lengkap untuk kebutuhan lembaga pendidikan seperti LMS, *test online*, administrasi sekolah, PPDB hingga aplikasi pendaftaran beasiswa. Produk Gu-Spark merupakan aplikasi *budgeting* yang terintegrasi dengan pelaporan keuangan, manajemen stok, kepegawaian hingga sistem invoice. Produk Gu-Docs merupakan aplikasi *paperless* untuk pembuatan surat online, pendaftaran izin, sertifikat online, hingga sistem *approval*.

*[Halaman sengaja dikosongkan]*

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1. Visual Studio code**

Visual Studio Code adalah text editor perangkat lunak yang digunakan dalam menulis atau merubah code dalam berbagai sistem operasi seperti Windows, Linux dan MacOS[1]. Bahasa pemrograman yang dapat digunakan beragam, seperti Java, JavaScript, Go, Node.js, Python dan C++. Penggunaan Visual Studio Code digunakan sebagai teks editor dapat membantu pengembang dalam debugging, penyorotan sintaksis, penyelesaian kode cerdas, mengontrol Git, pintasan keyboard dan preferensi.

#### **3.2. CodeIgniter 4**

CodeIgniter 4 adalah salah satu framework php yang bersifat open source, dapat mengembangkan aplikasi berbasis website dengan menghadirkan fitur yang lengkap dan bisa bekerja dengan cepat[2]. Kelebihan yang dimiliki bisa lebih hemat tempat server dan tidak menghabiskan banyak resource dibandingkan dengan framework lain. Framework ini mendukung bahasa pemrograman berbasis PHP dengan versi 7.3 ke atas dan mendukung arsitektur Model-View-Controller (MVC). CodeIgniter 4 memiliki dokumentasi lengkap yang dapat memudahkan pengembang web dalam mengimplementasikan kode. Penggunaan CodeIgniter 4 digunakan sebagai kerangka kerja dalam mengembangkan sebuah website karena memiliki sintaks yang ekspresif, jelas dan bisa menghemat waktu.

### **3.3. HTML**

HTML ( Hypertext Markup Language) adalah bahasa markup standar yang berfungsi untuk membuat struktur halaman website agar dapat ditampilkan di halaman web[3]. Penggunaan HTML dalam mengembakan website dapat diartikan sebagai pondasi awal dalam menyusun kerangka halaman web secara terstruktur sebelum berlanjut ke tampilan desain dan sisi fungsionalitas website.

### **3.4. PHP**

PHP adalah bahasa pemrograman server-side yang dapat digunakan dalam mengelola konten dinamis dan database lalu dapat mengirimkannya kembali ke web browser menjadi kode HTML[4]. Jenis server yang sering digunakan dengan PHP seperti Apache, Nginx, dan Lite Speed. Database yang dapat diintegrasikan dengan PHP antara lain MySQL, PostgreSQL, Oracle, Sybase, Informix, dan Microsoft SQL Server. Dokumentasi PHP dapat dengan mudah dipelajari oleh pengembang website dan digunakan secara gratis sesuai dengan kebutuhan masing masing. Penggunaan PHP pada pengembangan website yang dilakukan dapat menghubungkan database MySQL dengan data masukan yang didapatkan dari website.

### **3.5. CSS**

CSS adalah aturan atau format yang digunakan untuk mengatur elemen elemen dalam sebuah web sehingga akan lebih terstruktur dan seragam dalam bahasa markup. CSS memiliki peran dalam mengatur styling seperti warna font, alignment, background, border, ukuran gambar dan style lainnya secara terpisah. Sehingga

pengembang tidak perlu menuliskan kode secara berulang dan dapat lebih mudah dalam me-maintain kode website.

### **3.6. JavaScript**

JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan website untuk membuat konten lebih menarik dan menciptakan interaksi yang lebih dinamis seperti untuk menampilkan animasi, mengubah visibilitas teks, dan membuat menu drop-down[5]. Penggunaan Javascript digunakan dalam memanipulasi dan mengirim data pada browser pengguna.

### **3.7. MySQL**

MySQL adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server. MySQL termasuk jenis RDBMS yang mengelola data dalam database berdasarkan model relasional[6]. Penggunaan MySQL digunakan sebagai bahasa pemrograman manajemen basis data dalam mengelola data yang didapatkan dari website.

*[Halaman sengaja dikosongkan]*

## **BAB IV**

### **ANALISIS DAN DESAIN SISTEM**

#### **4.1. Analisis Sistem**

Pada bab ini akan dijelaskan mengenai tahapan dalam pembangunan aplikasi sistem informasi inventaris barang , yaitu analisis dari sistem yang berjalan dan analisis sistem yang dibangun.

Alur sistem yang berjalan pada perusahaan gumugu meliputi staf yang bertugas sebagai penjual barang dan sekretaris yang bertugas dalam membuat rekapitulasi data barang. Staf akan mencatat jumlah pemasukan barang, mencatat jumlah barang terjual dan kadaluarsa. Selanjutnya staff akan memberikan data tersebut kepada sekretaris yang nantinya akan dicatat data tersebut kedalam tools excel. Proses tersebut dilakukan antara staff dan sekretaris secara harian, sehingga dapat memudahkan mencari dan membuat laporan barang berdasarkan keadaan dalam waktu tertentu. Alur proses sistem yang berjalan masih bersifat manual. Terdapat beberapa kekurangan yang perlu diperbaiki dari sistem yang berjalan, antara lain seperti :

1. Terdapat data yang tidak akurat karena rekapitulasi masih dilakukan secara manual sehingga menghasilkan data yang tidak sesuai atau tidak akurat
2. Terdapat keterlambatan pengolahan data barang masuk, barang terjual dan barang kadaluarsa.
3. Menyulitkan dalam melakukan pengolahan data jika terjadi kesalahan dalam mencatat barang ke sistem.

Dari kekurangan tersebut perlu dilakukan perancangan sistem informasi, mengingat

perkembangan perusahaan akan meningkat seiring dengan berjalannya waktu. Produk penjualan dan karyawan tentu akan meningkat. Pengolahan data yang bersifat manual, tidak mungkin diterapkan dalam kondisi tersebut. Pengolahan data menggunakan sistem informasi tentu dapat membantu pihak perusahaan dalam mengontrol dan mengelola data barang secara efektif dan efisien.

Adapaun analisis perancangan sistem yang akan akan dijelaskan dalam tiga bagian, yaitu definisi umum aplikasi, proses bisnis, dan deskripsi fungsional.

#### **4.1.1. Definisi Umum Aplikasi**

Secara umum, aplikasi sistem informasi inventaris barang (SIIB) merupakan situs web yang berfungsi untuk mengelola barang barang seperti mencatat, mengubah, menghapus serta mencari data dari pemasukan barang dan penjualan barang. Karyawan kantor GUMUGU dapat memanfaatkan sistem tersebut untuk mempermudah pengelolaan inventaris atau rekapitulasi data barang. Sistem yang akan dirancang tidak mengakomodasikan untuk sistem penjualan barang melainkan hanya berfungsi sebagai rekapitulasi data barang harian.

Sistem Informasi Inventaris Barang mempunyai yang akan dirancang memiliki beberapa fungsi utama, antara lain :

1. (F001) Menambah Karyawan
2. (F002) Mengubah Karyawan
3. (F003) Menghapus Karyawan
4. (F004) Mencari Karyawan
5. (F005) Menambah Barang
6. (F006) Mengubah Barang
7. (F007) Menghapus Barang



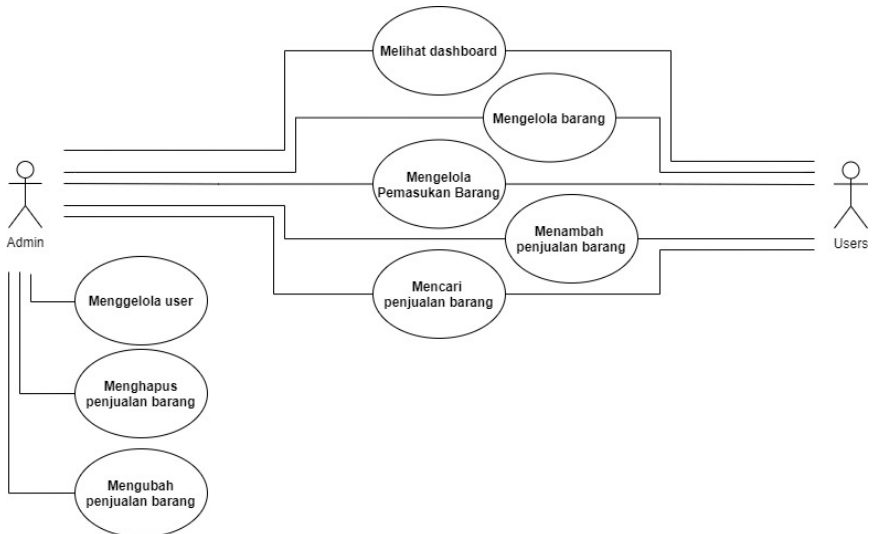
8. (F008) Melihat Detail Barang
9. (F009) Mencari Barang
10. (F010) Menambah Pemasukan Barang
11. (F011) Mengubah Pemasukan Barang
12. (F012) Menghapus Pemasukan Barang
13. (F013) Mencari Pemasukan Barang
14. (F014) Menambah Penjualan Barang
15. (F015) Mengubah Penjualan Barang
16. (F016) Menghapus Penjualan Barang
17. (F017) Mencari Penjualan Barang

Adapaun pengguna aplikasi sistem informasi inventaris barang antara lain adalah sebagai berikut :

1. Karyawan  
Karyawan adalah orang yang bekerja dalam mengelola barang dikantor yang memiliki akses terbatas dalam mengelola website. Hak akses yang diberikan berupa mengelola barang, mengelola pemasukan barang dan menambahkan penjualan barang.
2. Pemimpin perusahaan/Administrator  
Pemimpin perusahaan adalah orang yang berperan dalam mengoperasikan keseluruhan website.

#### **4.1.2. Proses Bisnis**

Berikut adalah gambaran proses bisnis dari aplikasi Sistem Informasi Inventaris Barang (SIIB) :



*Gambar 4.1 Use Case Sistem Informasi Inventaris Barang*

### **4.1.3. Deskripsi Fungsional**

Berikut adalah deskripsi fungsional pada sistem informasi inventaris barang berbasis website yaitu

#### **4.1.3.1. F001: Menambah Karyawan**

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan mengisi formulir penambahan data karyawan. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 1

Tabel 1. Spesifikasi kasus menambah karyawan

<b>Kode Use Case</b>	<b>F001</b>
Nama Use Case	Menambah Karyawan
Aktor	Administrator
Deskripsi	Administrator menambahkan Karyawan yang dapat mengakses
Kondisi Awal	Administrator belum menambahkan Karyawan
Kondisi Akhir	Karyawan sudah ditambahkan kedalam sistem
<b>Alur Normal</b>	
Aktor	Sistem
1.Administrator membuka halaman "Karyawan". 2. Administrator menekan tombol "Tambah Karyawan". 3. Administrator mengisi formulir yang disediakan 4.Administrator melakukan submit formulir.	1.1 Sistem menampilkan formulir login 2.1Sistem menampilkan daftar pengguna. 3.1 Sistem menampilkan formulir. 4.1 Sistem mengecek validitas isi formulir 4.2 Sistem menyimpan data formulir 4.3 Sistem menampilkan halaman "Karyawan" beserta flash message.
<b>Alur Alternatif</b>	
-	
4.1a 1. Administrator memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.2. F002: Mengubah Karyawan

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan mengisi formulir perubahan data karyawan. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 2

Tabel 2. Spesifikasi Kasus Mengubah Karyawan

<b>Kode Use Case</b>	<b>F002</b>
Nama Use Case	Mengubah Karyawan
Aktor	Administrator
Deskripsi	Administrator mengubah data Karyawan yang telah terdaftar
Kondisi Awal	Data Karyawan telah terdaftar pada sistem.
Kondisi Akhir	Data Karyawan telah berubah pada sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "Karyawan" 2. Administrator menekan tombol "Edit" pada salah satu data Karyawan. 3. Administrator mengubah data isian formulir.	1.1 Sistem menampilkan daftar Karyawan. 2.1 Sistem menampilkan detail dari Karyawan yang telah dipilih. 3.1 Sistem mengecek validitas data yang telah diisikan oleh Administrator. 3.2 Sistem menyimpan data pengguna yang telah diubah.
<b>Alur Alternatif</b>	
-	
4.1a 1. Administrator memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.3. F003: Menghapus Karyawan

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan menghapus data karyawan. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 3

*Tabel 3. Spesifikasi Kasus Menghapus Karyawan*

<b>Kode Use Case</b>	<b>F003</b>
Nama Use Case	Menghapus Karyawan
Aktor	Administrator
Deskripsi	Administrator menghapus Karyawan yang telah terdaftar pada
Kondisi Awal	Data Karyawan telah terdaftar pada sistem
Kondisi Akhir	Data Karyawan telah dihapus oleh sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "Karyawan"	1.1 Sistem menampilkan daftar Karyawan.
2. Administrator menekan "Delete" salah satu data pengguna.	2.1 Sistem menampilkan pop up konfirmasi untuk menghapus data pengguna
3. Administrator menekan tombol "Ya"	3.1 Sistem menghapus data pengguna.
<b>Alur Alternatif</b>	
4.1a	
1. Administrator memilih "Tidak"	
2. Sistem menampilkan halaman daftar dari pengguna.	

#### 4.1.3.4. F004: Mencari Karyawan

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan mencari data karyawan. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 4

Tabel 4. Spesifikasi Kasus Mencari Karyawan

<b>Kode Use Case</b>	<b>F004</b>
Nama Use Case	Mencari Karyawan
Aktor	Administrator
Deskripsi	Administrator mencari Karyawan yang telah terdaftar pada sistem
Kondisi Awal	Administrator belum menemukan data Karyawan yang ingin dicari
Kondisi Akhir	Administrator sudah menemukan data Karyawan yang dicari pada
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "Karyawan"	1.1 Sistem menampilkan daftar Karyawan.
2. Administrator menulis data Karyawan pada kolom "Cari".	2.1 Sistem menampilkan data Karyawan yang dicari.
<b>Alur Alternatif</b>	
3.1a	
1. Administrator memasukkan data yang tidak terdaftar dalam sistem pencarian.	
2. Sistem menampilkan pemberitahuan tidak ada Karyawan yang terdaftar pada	

#### 4.1.3.5. F005: Menambah Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mengisi formulir penambahan data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 5

Tabel 5. Spesifikasi Kasus Menambah Barang

<b>Kode Use Case</b>	<b>F005</b>
Nama Use Case	Menambah Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan menambahkan barang pada website
Kondisi Awal	Administrator atau Karyawan belum menambahkan barang
Kondisi Akhir	Barang sudah ditambahkan kedalam sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Barang". 2. Administrator atau Karyawan menekan tombol "Tambah Barang". 3. Administrator atau Karyawan mengisi formulir yang disediakan 4. Administrator atau Karyawan melakukan submit formulir.	1.1 Sistem menampilkan daftar barang. 2.1 Sistem menampilkan formulir. 4.1 Sistem mengecek validitas isi formulir 4.2 Sistem menyimpan data formulir 4.3 Sistem menampilkan halaman "barang" beserta flash message.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.6. F006: Mengubah Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mengisi formulir perubahan data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 6

Tabel 6. Spesifikasi Kasus Mengubah Barang

<b>Kode Use Case</b>	<b>F006</b>
Nama Use Case	Mengubah barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan mengubah data barang yang telah
Kondisi Awal	Data barang telah terdaftar pada sistem.
Kondisi Akhir	Data barang telah berubah pada sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Barang"	1.1 Sistem menampilkan daftar barang.
2. Administrator atau Karyawan menekan tombol "Edit" pada salah satu data barang.	2.1 Sistem menampilkan detail dari barang yang telah dipilih.
3. Administrator atau Karyawan mengubah data isian formulir.	3.1 Sistem mengecek validitas data yang telah diisikan oleh Administrator atau Karyawan.
	3.2 Sistem menyimpan data barang yang telah diubah.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator atau Karyawan memasukkan data yang tidak sesuai dengan format formulir.	
2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.7. F007: Menghapus Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan menghapus data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 7



Tabel 7. Spesifikasi Kasus Menghapus Barang

<b>Kode Use Case</b>	<b>F007</b>
Nama Use Case	Menghapus Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan menghapus barang yang telah
Kondisi Awal	Data barang telah terdaftar pada sistem
Kondisi Akhir	Data barang telah dihapus oleh sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Barang" 2. Administrator menekan "Delete" salah satu data barang. 3. Administrator menekan tombol "Ya"	1.1 Sistem menampilkan daftar barang. 2.1 Sistem menampilkan pop up konfirmasi untuk menghapus data barang 3.1 Sistem menghapus data barang.
<b>Alur Alternatif</b>	
4.1a 1. Administrator memilih "Tidak" 2. Sistem menampilkan halaman daftar dari barang.	

#### 4.1.3.8. F008: Melihat Detail Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan melihat detail data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 8

*Tabel 8. Spesifikasi Kasus Melihat Detail Barang*

<b>Kode Use Case</b>	<b>F008</b>
Nama Use Case	Melihat Detail Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan melihat detail barang yang telah
Kondisi Awal	Sistem belum menampilkan detail barang
Kondisi Akhir	Sistem menampilkan detail barang
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "Barang" 2. Administrator menekan "detail" salah satu data barang.	1.1 Sistem menampilkan daftar barang. 2.1 Sistem menampilkan detail data barang
<b>Alur Alternatif</b>	
-	

#### 4.1.3.9. F009: Mencari Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan mencari data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 9

*Tabel 9. Spesifikasi Kasus Mencari Barang*

<b>Kode Use Case</b>	<b>F009</b>
Nama Use Case	Mencari Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan mencari barang yang telah terdaftar
Kondisi Awal	Administrator atau Karyawan belum menemukan data barang
Kondisi Akhir	Administrator atau Karyawan sudah menemukan data barang
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Barang" 2. Administrator atau Karyawan menulis data barang pada kolom	1.1 Sistem menampilkan daftar barang. 2.1 Sistem menampilkan data barang yang dicari.
<b>Alur Alternatif</b>	
3.1a 1. Administrator atau Karyawan memasukkan data yang tidak terdaftar dalam sistem pencarian.	

#### 4.1.3.10. F010: Menambah Pemasukan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mengisi formulir penambahan pemasukan data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 10

*Tabel 10. Spesifikasi Kasus Menambah Pemasukan Barang*

<b>Kode Use Case</b>	<b>F010</b>
Nama Use Case	Menambah Pemasukan Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan menambahkan data pemasukan
Kondisi Awal	Administrator atau Karyawan belum menambahkan data pemasukan
Kondisi Akhir	Data pemasukan barang sudah ditambahkan kedalam sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Pemasukan Barang". 2. Administrator atau Karyawan menekan tombol "Tambah Pemasukan Barang". 3. Administrator atau Karyawan mengisi formulir yang disediakan 4. Administrator atau Karyawan melakukan submit formulir.	1.1 Sistem menampilkan daftar pemasukan barang. 2.1 Sistem menampilkan formulir.  4.1 Sistem mengecek validitas isi formulir 4.2 Sistem menyimpan data formulir 4.3 Sistem menampilkan halaman "pemasukan barang" beserta flash message.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.11. F011: Mengubah Pemasukan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mengisi

formulir pengubahan pemasukan data barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 11

*Tabel 11. Spesifikasi Kasus Mengubah Pemasukan Barang*

<b>Kode Use Case</b>	<b>F011</b>
<b>Nama Use Case</b>	Mengubah Pemasukan Barang
<b>Aktor</b>	Administrator/Karyawan
<b>Deskripsi</b>	Administrator atau Karyawan mengubah data pemasukan barang
<b>Kondisi Awal</b>	Data pemasukan barang telah terdaftar pada sistem.
<b>Kondisi Akhir</b>	Data pemasukan barang telah berubah pada sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Pemasukan barang"	1.1 Sistem menampilkan daftar pemasukan barang.
2. Administrator atau Karyawan menekan tombol "Edit" pada salah satu data pemasukan barang.	2.1 Sistem menampilkan detail dari pemasukan barang yang telah dipilih.
3. Administrator atau Karyawan mengubah data isian formulir.	3.1 Sistem mengecek validitas data yang telah diisikan oleh Administrator. 3.2 Sistem menyimpan data pemasukan barang yang telah diubah.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator memasukkan data yang tidak sesuai dengan format formulir.	
2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.12. F012: Menghapus Pemasukan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan menghapus data pemasukan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 12

Tabel 12. Spesifikasi Kasus Menghapus Pemasukan Barang

<b>Kode Use Case</b>	<b>F012</b>
Nama Use Case	Menghapus Pemasukan Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan menghapus pemasukan barang yang
Kondisi Awal	Data pemasukan barang telah terdaftar pada sistem
Kondisi Akhir	Data pemasukan barang telah dihapus oleh sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Pemasukan barang"	1.1 Sistem menampilkan daftar pemasukan barang.
2. Administrator atau Karyawan menekan "Delete" salah satu data pemasukan barang.	2.1 Sistem menampilkan pop up konfirmasi untuk menghapus data pemasukan barang.
3. Administrator atau Karyawan	3.1 Sistem menghapus data pemasukan barang.
<b>Alur Alternatif</b>	
4.1a	
1. Administrator atau Karyawan memilih "Tidak"	
2. Sistem menampilkan halaman daftar dari pemasukan barang.	

#### 4.1.3.13. F013: Mencari Pemasukan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mencari data pemasukan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 13

Tabel 13. Spesifikasi Kasus Menghapus Pemasukan Barang

<b>Kode Use Case</b>	<b>F013</b>
Nama Use Case	Mencari Pemasukan Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan mencari pemasukan barang yang
Kondisi Awal	Administrator atau Karyawan belum menemukan data pemasukan
Kondisi Akhir	Administrator atau Karyawan sudah menemukan data pemasukan
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Pemasukan Barang" 2. Administrator atau Karyawan menulis data pemasukan barang	1.1 Sistem menampilkan daftar pemasukan barang. 2.1 Sistem menampilkan data pemasukan barang yang dicari.
<b>Alur Alternatif</b>	
3.1a	1. Administrator atau Karyawan memasukkan data yang tidak terdaftar dalam sistem pencarian.

#### 4.1.3.14. F014: Menambah Penjualan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mengisi formulir penambahan data penjualan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 14

Tabel 14. Spesifikasi Kasus Mencari Pemasukan Barang

<b>Kode Use Case</b>	<b>F014</b>
Nama Use Case	Menambah Penjualan Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan menambahkan data penjualan
Kondisi Awal	Administrator atau Karyawan belum menambahkan penjualan barang
Kondisi Akhir	Data penjualan barang sudah ditambahkan kedalam sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Penjualan Barang". 2. Administrator atau Karyawan menekan tombol "Tambah penjualan barang". 3. Administrator atau Karyawan mengisi formulir yang disediakan 4. Administrator melakukan submit formulir.	1.1 Sistem menampilkan daftar penjualan barang. 2.1 Sistem menampilkan formulir.  4.1 Sistem mengecek validitas isi formulir 4.2 Sistem menyimpan data formulir 4.3 Sistem menampilkan halaman "pemasukan barang" beserta flash message.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator atau Karyawan memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.15. F015: Mengubah Penjualan Barang

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan mengisi formulir perubahan data penjualan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 15

Tabel 15. Spesifikasi Kasus Mengubah Penjualan Barang

<b>Kode Use Case</b>	<b>F015</b>
Nama Use Case	Mengubah Penjualan Barang
Aktor	Administrator
Deskripsi	Administrator mengubah data Karyawan yang telah terdaftar
Kondisi Awal	Data penjualan barang telah terdaftar pada sistem.
Kondisi Akhir	Data penjualan barang telah berubah pada sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "penjualan barang" 2. Administrator menekan tombol "Edit" pada salah satu data penjualan barang. 3. Administrator mengubah data isian formulir.	1.1 Sistem menampilkan daftar penjualan barang. 2.1 Sistem menampilkan detail dari penjualan barang yang telah dipilih. 3.1 Sistem mengecek validitas data yang telah diisikan oleh Administrator. 3.2 Sistem menyimpan data penjualan barang yang telah diubah.
<b>Alur Alternatif</b>	
-	
4.1a	
1. Administrator memasukkan data yang tidak sesuai dengan format formulir. 2. Sistem menampilkan pemberitahuan kesalahan format.	

#### 4.1.3.16. F016: Menghapus Penjualan Barang

Kasus penggunaan ini dapat diakses oleh Administrator ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator akan menghapus data penjualan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 16



*Tabel 16. Spesifikasi Kasus Menghapus Penjualan Barang*

<b>Kode Use Case</b>	<b>F016</b>
Nama Use Case	Menghapus Penjualan Barang
Aktor	Administrator
Deskripsi	Administrator menghapus penjualan barang yang telah terdaftar
Kondisi Awal	Data penjualan barang telah terdaftar pada sistem
Kondisi Akhir	Data penjualan barang telah dihapus oleh sistem
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator membuka halaman "Penjualan Barang"	1.1 Sistem menampilkan daftar Penjualan Barang.
2. Administrator menekan "Delete" salah satu data Penjualan Barang.	2.1 Sistem menampilkan pop up konfirmasi untuk menghapus data Penjualan Barang
3. Administrator menekan tombol "Ya"	3.1 Sistem menghapus data Penjualan Barang.
<b>Alur Alternatif</b>	
4.1a	
1. Administrator memilih "Tidak"	
2. Sistem menampilkan halaman daftar dari Penjualan Barang.	

#### 4.1.3.17. F017: Mencari Penjualan Barang

Kasus penggunaan ini dapat diakses oleh Administrator atau Karyawan ketika sudah melakukan login pada sistem. Pada kasus penggunaan ini, Administrator atau Karyawan akan mencari data penjualan barang. Penjelasan skenario yang lebih rinci dapat dilihat pada tabel 17

*Tabel 17. Spesifikasi Kasus Mencari Penjualan Barang*

<b>Kode Use Case</b>	<b>F017</b>
Nama Use Case	Mencari Penjualan Barang
Aktor	Administrator/Karyawan
Deskripsi	Administrator atau Karyawan mencari penjualan barang yang telah
Kondisi Awal	Administrator atau Karyawan belum menemukan data penjualan
Kondisi Akhir	Administrator atau Karyawan sudah menemukan data penjualan
<b>Alur Normal</b>	
Aktor	Sistem
1. Administrator atau Karyawan membuka halaman "Penjualan Barang" 2. Administrator atau Karyawan menulis data penjualan barang	1.1 Sistem menampilkan daftar penjualan barang. 2.1 Sistem menampilkan data penjualan barang yang dicari.
<b>Alur Alternatif</b>	
3.1a 1. Administrator atau Karyawan memasukkan data yang tidak terdaftar dalam sistem pencarian.	

## **4.2. Perancangan Sistem**

Perancangan sistem meliputi perancangan tabel yang akan digunakan untuk database website. Database berfungsi dalam menyimpan data yang dimasukkan ke dalam sistem dan dapat diakses kembali untuk ditampilkan kepada pengguna website.

### **4.2.1. Perancangan Tabel dalam Sistem**

Adapun perancangan tabel yang akan digunakan dalam membuat sistem informasi inventaris barang.

#### 4.2.1.1. Tabel Barang

Sistem rekapitulasi barang yang terjadi dilakukan pencatatan nama barang, deksripsi, harga awal barang, harga jual barang, tanggal dan nama pencatat. Maka dari itu untuk merancang sistem yang mencatat barang diperlukan tabel barang yang berisi informasi seperti pada tabel 18.

*Tabel 18. Tabel Barang*

	Nama Kolom
PK	<b>id_item : int(11)</b>
FK	id_user : int(11)
	name : varchar(225)
	description : varchar(225)
	buying_price : float(2,8)
	selling_price : float(2,8)
	created_at : datetime
	updated_at : datetime

#### 4.2.1.2. Tabel Stok Barang

Pada saat melakukan pencatatan pemasukan dan penjualan barang kedalam sistem tentu jumlah stok barang akan berubah. Jumlah stok barang akan bertambah saat memasukkan barang dan jumlah stok barang akan berkurang saat barang terjual atau kadaluarsa. Maka dari itu perlu membuat tabel stok yang berisi informasi seperti pada tabel 19

*Tabel 19 Tabel Stok*

	Nama Kolom
PK	<b>id_stock : int(11)</b>
FK	id_item : int(11)
	jumlah : int(11)
	updated_at : datetime

#### 4.2.1.3. Tabel Pemasukan Barang

Pada saat mencatat pemasukan barang, sistem memerlukan informasi nama barang, nama pencatat, jumlah barang serta tanggal masuk barang. Maka dari itu diperlukan tabel pemasukan barang yang berisi informasi pemasukan barang seperti pada tabel 20.

*Tabel 20 Tabel Pemasukan Barang*

	Nama Kolom
<b>PK</b>	<b>id_supply : int(11)</b>
FK	id_item : int(11)
FK	id_user : int(11)
	quantity : int(11)
	created_at : datetime
	updated_at : datetime

#### 4.2.1.4. Tabel Penjualan Barang

Pada saat mencatat penjualan barang, sistem memerlukan informasi nama barang, nama pencatat, jumlah barang, status barang yang meliputi kadaluarsa atau terjual, total harga barang serta tanggal terjual barang. Maka dari itu diperlukan tabel penjualan barang yang berisi informasi penjualan barang seperti pada tabel 21.

*Tabel 21 Tabel Penjualan Barang*

	Nama Kolom
<b>PK</b>	<b>id_sale : int(11)</b>
FK	id_item : int(11)
FK	id_user : int(11)
FK	id_status : int(4)
	quantity : int(11)
	total_price : float(2,8)
	created_at : datetime
	updated_at : datetime

#### 4.2.1.5. Tabel Status Barang

Pada saat mencatat barang kadaluarsa atau terjual, diperlukan tabel status yang mengakomodasikan informasi tersebut seperti pada tabel 22.

*Tabel 22 Tabel Status Barang*

	Nama Kolom
<b>PK</b>	<b>id_status : int(4)</b>
	name : varchar(50)

#### 4.2.1.6. Tabel Jenis Pengguna

Sistem yang dirancang memiliki jenis pengguna yang berbeda. Setiap jenis pengguna memiliki hak akses yang berbeda. Maka dari itu diperlukan tabel jenis pengguna yang berisi informasi seperti pada tabel 23.

*Tabel 23 Tabel Jenis Pengguna*

	Nama Kolom
<b>PK</b>	<b>id_role : int(4)</b>
	name : varchar(50)

#### 4.2.1.7. Tabel Karyawan

Pada saat mencatat informasi barang kedalam sistem dibutuhkan nama yang bertanggung jawab dalam memasukkan data barang tersebut. Informasi data karyawan dapat dirancang seperti pada tabel 24.

*Tabel 24 Tabel Karyawan*

	Nama Kolom
<b>PK</b>	<b>id_user : int(11)</b>
<b>FK</b>	id_role : int(4)
	name : varchar(225)
	phone : varchar(15)
	address : varchar(225)
	email : varchar(225)
	password : varchar(225)
	created_at : datetime
	updated_at : datetime

## BAB V IMPLEMENTASI SISTEM

### 5.1. Implementasi Source Code

Pada bagian ini menjelaskan implementasi sistem yang dibuat menggunakan kerangka kerja Codeigniter 4. Bahasa pemrograman yang digunakan menggunakan php, html, css, javascript dan mySql untuk database sistem. Teks editor yang digunakan menggunakan Visual Studio Code dan menggunakan XAMPP sebagai server.

#### 5.1.1. Routes

Kode sumber 5.1 mengimplementasi *Routes* yang mengatur atau memetakan URL dalam bentuk nama lain. *Routes* memiliki peran dalam menentukan arah rute yang harus dilalui melalui controller. Penjelasan kode sumber dapat dilihat pada tabel 25

```
1. <?php
2. namespace Config;
3. $routes->setDefaultNamespace('App\Controllers');
4. $routes->setDefaultController('Home');
5. $routes->setDefaultMethod('index');
6. $routes->get('/', 'Home::login');
7. $routes->post('/auth/login', 'Auth::login');
8. $routes->get('/dashboard', 'Pages::index');
9.
10. // User
11. $routes->get('/user', 'User::index');
12. $routes->add('/user/edit/{:segment}', 'User::edit/$1');
13. $routes->post('/user/update/{:segment}', 'User::update/$1');
14. $routes->delete('/user/{:num}', 'User::delete/$1');
15.
16. // Items
```

```

17. $routes->get('/item', 'Item::index');
18. $routes->add('/item/view/{:segment}', 'Item::view/$1');
19. $routes->add('/item/edit/{:segment}', 'Item::edit/$1');
20. $routes->post('/item/update/{:segment}', 'Item::update/$1');
21. $routes->delete('/item/{:num}', 'Item::delete/$1');
22.
23. // Supply
24. $routes->get('/supply', 'Supply::index');
25. $routes->add('/supply/edit/{:segment}', 'Supply::edit/$1');
26. $routes->post('/supply/update/{:segment}', 'Supply::update/$1');
27. $routes->delete('/supply/{:num}', 'Supply::delete/$1');
28.
29. // Sales
30. $routes->get('/sales', 'Sales::index');
31. $routes->add('/sales/edit/{:segment}', 'Sales::edit/$1');
32. $routes->post('/sales/update/{:segment}', 'Sales::update/$1');
33. $routes->delete('/sales/{:num}', 'Sales::delete/$1');
34.
35.

```

*Kode Sumber 5.1 Halaman Routes*

*Tabel 25. Penjelasan Kode Sumber 5.1*

No.Baris	Kegunaan
4	Mengarahkan ke halaman dashboard
6-8	Mengarahkan ke halaman login
11-14	Mengarahkan ke halaman mengelola user
17-21	Mengarahkan ke halaman mengelola barang
24-27	Mengarahkan ke halamana mengelola pemasukan barang
30-33	Mengarahkan ke halamana mengelola penjualan barang



## 5.1.2. Controllers

### 5.1.2.1 Auth.php

Kode sumber 5.2 merupakan implementasi controller Auth.php yang menangani login dan logout pengguna saat masuk ke halaman web. Penjelasan kode sumber dapat dilihat pada tabel 26

```
1. <?php
2. namespace App\Controllers;
3. use App\Models\UserModel;
4.
5. class Auth extends BaseController
6. {
7.     public function index()
8.     {
9.         return view('login');
10.    }
11.    public function logout()
12.    {
13.        $this->session->destroy();
14.        return redirect()->to('/');
15.    }
16. }
```

*Kode Sumber 5. 2 Halaman Auth*

*Tabel 26 Penjelasan Kode Sumber 5.2*

No.Baris	Kegunaan
7-9	Mengarahkan ke halaman login
11-14	Mengarahkan ke halaman logout

### 5.1.2.2 Home.php

Kode sumber 5.3 mengimplementasi kode sumber controller Home.php yang menangani arahan halaman awal website pada saat pengguna sukses login web. Penjelasan kode sumber dapat dilihat pada tabel 27

```
1. <?php
2. namespace App\Controllers;
3. class Home extends BaseController
4. {
5.     public function home()
6.     {
7.         $data = [
8.             'title' => 'Home'
9.         ];
10.        return view('home', $data);
11.    }
12.
13.    public function login()
14.    {
15.        $data = [
16.            'title' => 'login'
17.        ];
18.        return view('login', $data);
19.    }
20. }
```

*Kode Sumber 5. 3 Halaman Home*

*Tabel 27. Penjelasan Kode sumber 5.3*

No.Baris	Kegunaan
5-10	Mengarahkan ke halaman dashboard
13-19	Mengarahkan ke halaman login

### 5.1.2.3 Item.php

Kode sumber 5.4 mengimplementasi kode sumber controller Item.php yang menangani proses create, read, update dan delete pada halaman item. Penjelasan kode sumber dapat dilihat pada tabel 28

```
1. <?php
2. namespace App\Controllers;
3. use App\Models\ItemModel;
4. use App\Models\StockModel;
5. use App\Models\SalesModel;
6. use CodeIgniter\HTTP\IncomingRequest;
7. class Item extends BaseController
8. {
9.     protected $itemModel;
10.    protected $stockModel;
11.    protected $salesModel;
12.    public function index()
13.    {
14.        $item = $this->itemModel->getAllItems();
15.        $data = [
16.            'title' => 'Home | barang',
17.            'item' => $item
18.        ];
19.        return view('items/index', $data);
20.    }
21.    public function view($id_item)
22.    {
23.        $stock = $this->stockModel->getStockByIdItem($id_item);
24.        $salesItem=$this->itemModel ->getSalesItemByIdItem ($id_item);
25.        $itemName = $this->itemModel->getItem($id_item);
26.        return view('items/view', $data);
27.    }
28.    public function create()
29.    {
30.        $item = $this->itemModel->findAll();
31.        $data = [
```

```

32.     'title' => 'Form Tambah Barang',
33.     'validation' => \Config\Services::validation(),
34.     'item' => $item
35. ];
36.     return view('items/create', $data);
37. }
38. public function update($id_item)
39. {
40.     $insert = $this->itemModel->getUpdateItem($data, $id_item);
41.     session()->setFlashdata('pesan', 'Data berhasil diupdate!');
42.     return redirect()->to('/item/index');
43. }
44. }
45. public function delete($id_item)
46. {
47.     $this->itemModel->delete($id_item);
48.     session()->setFlashdata('pesan', 'Data berhasil dihapus!');
49.     return redirect()->to('/item/index');
50. }
51. }

```

*Kode Sumber 5. 4 Halaman Item*

*Tabel 28. Penjelasan Kode sumber 5.4*

No.Baris	Kegunaan
12-20	Menampilkan halaman data barang
21-27	Menampilkan halaman detail barang
28-37	Menambahkan barang baru dan menyimpan ke database
38-44	Mengubah data barang
45-50	Menghapus data barang

#### 5.1.2.4 Pages.php

Kode sumber 5.5 mengimplementasi kode sumber controller Pages.php yang menangani proses tampilan data barang seperti jumlah barang, jumlah keuntungan, jumlah kerugian, total pengguna pada halaman dashboard. Penjelasan kode sumber dapat dilihat pada tabel 29

```
1. <?php
2. namespace App\Controllers;
3. use App\Models\SalesModel;
4. use App\Models\UserModel;
5. use App\Models\StockModel;
6. use App\Models\ItemModel;
7.
8. class Pages extends BaseController
9. {
10.     public function index()
11.     {
12.         $get_item = $this->itemModel->getAllItems();
13.         $total_items = $this->SalesModel->getTotalItems();
14.         $total_price = $this->SalesModel->getTotalPrice();
15.         $total_users = $this->UserModel->getTotalUsers();
16.         $total_stock = $this->StockModel->getTotalStock();
17.         $total_price_exp = $this->SalesModel->getTotalPriceExp();
18.         $total_stock_exp = $this->SalesModel->getTotalItemExp();
19.         $total_keuntungan = $this->SalesModel->getTotalKeuntungan();
20.
21.         $data = [
22.             'title' => 'Home',
23.             'total_items' => $total_items->total_items,
24.             'total_users' => $total_users->total_users,
25.             'total_price' => $total_price->total_price,
26.             'total_stock' => $total_stock->total_stock,
27.             'total_price_exp' => $total_price_exp->total_rugi,
28.             'total_stock_exp' => $total_stock_exp->barang_exp,
29.             'total_keuntungan' => $total_keuntungan->untung,
30.         ];
```

```

31.     echo view('dashboard', $data);
32.   }
33.

```

*Kode Sumber 5.5 Halaman Pages*

*Tabel 29. Penjelasan Kode Sumber 5.5*

No.Baris	Kegunaan
10-33	Menampilkan halaman data dashboard

### 5.1.2.5 Sales.php

Kode sumber 5.6 merupakan implementasi dari controller Sales.php yang menangani tampilan data penjualan dan data stok seperti memasukkan data barang terjual, menghapus data barang terjual, memperbarui data barang terjual pada halaman penjualan dan memperbarui jumlah stok barang pada halaman stok. Penjelasan kode sumber dapat dilihat pada tabel 31

```

1.  <?php
2.  namespace App\Controllers;
3.  use App\Models\SalesModel;
4.  use App\Models\ItemModel;
5.  use App\Models\StockModel;
6.  class Sales extends BaseController
7.  {
8.      protected $SalesModel;
9.      protected $ItemModel;
10.     protected $StockModel;
11.     public function index()
12.     {
13.         $sales = $this->SalesModel->getAllSales();
14.         $data = [
15.             'title' => 'Home | Penjualan Barang',
16.             'sales' => $sales
17.         ];

```

```

18.     return view('sales/index', $data);
19.   }
20.   public function create()
21.   {
22.       $items = $this->ItemModel->findAll();
23.       $data = [
24.           'items' => $items,
25.           'title' => 'Form Pencatatan Penjualan',
26.           'validation' => \Config\Services::validation()
27.       ];
28.       return view('sales/create', $data);
29.   }
30.   public function update($id_sale)
31.   {
32.       $validation = \Config\Services::validation();
33.       $validation->setRules([
34.           'id_item' => 'required',
35.           'quantity' => 'required',
36.           'created_at' => 'required',
37.           'id_status' => 'required',
38.       ]);
39.       $isValid = $validation->withRequest($this->request)->run();
40.       $insert = $this->SalesModel->getUpdateSale($data, $id_sale);
41.       session()->setFlashdata('pesan', 'Data berhasil diupdate!');
42.       return redirect()->to('/sales/index');
43.   }
44. }
45. }
46. public function delete($id_sale)
47. {
48.     $this->SalesModel->delete($id_sale);
49.     session()->setFlashdata('pesan', 'Data berhasil dihapus!');
50.     return redirect()->to('/sales/index');
51. }
52. }

```

*Kode Sumber 5.6 Halaman Sales*

*Tabel 30. Penjelasan Kode Sumber 5.6*

No.Baris	Kegunaan
11-19	Menampilkan halamanan penjualan barang
20-29	Menambahkan data penjualan baru dan menyimpan ke database
30-45	Mengubah data penjualan barang
46-51	Menghapus data penjualan barang

#### 5.1.4.6 Supply.php

Kode sumber 5.7 mewakili contoller Supply.php yang menanggapi proses tampilan data pemasukan barang seperti menambahkan data pemasukan barang, memperbarui data pemasukan barang, menghapus pemasukan barang pada halaman pemasukan. Penjelasan kode sumber dapat dilihat pada tabel 31

<pre> 1. &lt;?php 2. namespace App\Controllers; 3. use App\Models\SupplyModel; 4. use App\Models\StockModel; 5. use App\Models\ItemModel; 6. class Supply extends BaseController 7. { 8.     protected \$SupplyModel; 9.     protected \$StockModel; 10.    protected \$ItemModel; 11.    public function index() 12.    { 13.        \$supply = \$this-&gt;SupplyModel-&gt;getAllSupply(); 14.        \$data = [ 15.            'title' =&gt; 'Home   Daftar Pemasukan Barang', 16.            'supply' =&gt; \$supply 17.        ]; 18.        return view('supply/index', \$data); 19.    } </pre>
---



```

20. public function create()
21. {
22.     $items = $this->ItemModel->findAll();
23.     $data = [
24.         'items' => $items,
25.         'title' => 'Form Pemasukan Barang',
26.         'validation' => \Config\Services::validation()
27.     ];
28.     return view('supply/create', $data);
29. }
30. public function update($id_supply)
31. {
32.     $validation = \Config\Services::validation();
33.     $validation->setRules([
34.         'id_item' => 'required',
35.         'quantity' => 'required',
36.         'created_at' => 'required',
37.     ]);
38.     $isDataValid = $validation->withRequest($this->request)->run();
39.     $insert = $this->SupplyModel->getUpdateSupply($data, $id_supply);
40.     session()->setFlashdata('pesan', 'Data berhasil diupdate!');
41.     return redirect()->to('/supply/index');
42. }
43. }
44. }
45. public function delete($id_supply)
46. {
47.     $this->SupplyModel->delete($id_supply);
48.     session()->setFlashdata('pesan', 'Data berhasil dihapus!');
49.     return redirect()->to('/supply/index');
50. }
51. }

```

*Kode Sumber 5. 6 Halaman Pemasukan*

*Tabel 31. Penjelasan Kode Sumber 5.6*

No.Baris	Kegunaan
----------	----------

11-19	Menampilkan halaman pemasukan barang
20-28	Menambahkan data pemasukan barang dan menyimpan ke database
30-44	Mengubah data pemasukan barang
45-51	Menghapus data pemasukan barang

#### 5.1.2.6 Karyawan.php

Kode sumber 5.8 mewakili controler Karyawan.php menangani proses pembuatan karyawan, memperbarui data karyawan serta menghapus data karyawan. Penjelasan kode sumber dapat dilihat pada tabel 32

```

1. <?php
2. namespace App\Controllers;
3. use App\Models\UserModel;
4. use CodeIgniter\HTTP\IncomingRequest;
5. class User extends BaseController
6. {
7.     protected $userModel;
8.     public function index()
9.     {
10.         $user = $this->userModel->getAllUserRole();
11.         $data = [
12.             'title' => 'Home | Daftar User',
13.             'user' => $user
14.         ];
15.         return view('user/index', $data);
16.     }
17.
18.     public function create()
19.     {

```

```

20.     $data = [
21.         'title' => 'Form Tambah Data User',
22.         'validation' => \Config\Services::validation()
23.     ];
24.     return view('user/create', $data);
25. }
26.
27. public function update($id_user)
28. {
29.     $request = service('request');
30.     $db      = \Config\Database::connect();
31.
32.     $validation = \Config\Services::validation();
33.     $isDataValid = $validation->withRequest($this->request)->run();
34.     $insert = $this->userModel->getUpdateUser($data, $id_user);
35.     session()->setFlashdata('pesan', 'Data berhasil diupdate!');
36.     return redirect()->to('/user/index');
37. }
38. }
39.
40. public function delete($id_user)
41. {
42.     $this->userModel->delete($id_user);
43.     session()->setFlashdata('pesan', 'Data berhasil dihapus!');
44.     return redirect()->to('/user/index');
45. }
46. }

```

*Kode Sumber 5.8 Halaman Karyawan*

*Tabel 32. Penjelasan Kode Sumber 5.8*

No.Baris	Kegunaan
8-15	Menampilkan halaman karyawan
18-25	Menambahkan data karyawan dan menyimpan ke database
27-38	Mengubah data karyawan

### 5.1.3 Folder Models

#### 5.1.3.1 ItemsModel.php

Kode sumber 5.7 merupakan implementasi dari ItemsModel menghubungkan website dengan tabel items yang berada di dalam database. Kode sumber ini juga menampilkan data, menyediakan fitur pencarian, melakukan perhitungan jumlah data, dan mengambil data dengan jumlah tertentu. Penjelasan kode sumber dapat dilihat pada tabel 33

```
1. <?php
2. namespace App\Models;
3. use CodeIgniter\Model;
4. class ItemModel extends Model
5. {
6.     public function getItem($id_item)
7.     {
8.         return $this->db->query("SELECT * FROM items WHERE id_item=" .
9.             $id_item)->getrow();
10.    }
11.    public function getUpdateItem($data, $id_item)
12.    {
13.        $db = \Config\Database::connect();
14.        $builder = $db->table('items');
15.        $builder->where('id_item', $id_item);
16.        $builder->update($data);
17.        return $this->db->affectedRows();
18.    }
19.    public function getKeuntunganByItem($id_item)
20.    {
21.        return $this->db->query("SELECT sales.total_price AS
22.            harga_jual ,items.name AS nama_barang ,sales.id_status,
```

```

21.      (sales.quantity * items.buying_price) AS harga_beli ,(sales.total_price-
(sales.quantity * items.buying_price)) AS keuntungan
22.      FROM sales
23.      JOIN items ON sales.id_status=1 AND sales.id_item=items.id_item AND
items.id_item=" . $id_item)->getRow();
24.  }
25.  public function getKerugianByItem($id_item)
26.  {
27.      return $this->db->query("
28.      SELECT items.name , sum(sales.total_price) AS total_rugi
29.      FROM items
30.      INNER JOIN sales ON items.id_item=sales.id_item
31.      WHERE sales.id_status=0 AND sales.id_item=$id_item")->getRow();
32.  }
33.  public function getSalesItemByIdItem($id_item)
34.  {
35.      return $this->db->query("SELECT * FROM sales
36.      INNER JOIN items ON sales.id_item = items.id_item
37.      WHERE sales.id_item = $id_item ")->getResultArray();
38.  }
39.  }

```

*Kode Sumber 5. 7 Halaman ItemModel*

*Tabel 33. Penjelasan Kode sumber 5.7*

No.Baris	Kegunaan
6-9	Menampilkan semua data barang
10-17	Mengubah data barang dan menyimpan ke database
18-24	Mengambil data keuntungan barang
25-32	Mengambil data kerugian barang
33-38	Mengambil data penjualan berdasarkan daftar barang

### 5.1.3.2 SalesModel.php

Kode sumber 5.8 merupakan implementasi dari SalesModel menghubungkan website dengan tabel sales yang berada di dalam database. Kode sumber ini juga menampilkan data, menyediakan fitur pencarian, melakukan perhitungan jumlah data, dan mengambil data dengan jumlah tertentu. Penjelasan kode sumber dapat dilihat pada tabel 34

```
1. <?php
2. namespace App\Models;
3. use CodeIgniter\Model;
4. class SalesModel extends Model
5. {
6.     public function getSale($id_sale)
7.     { return $this->db->query("SELECT * FROM sales WHERE id_sale=" .
   $id_sale->getrow());
8.     }
9.     public function getUpdateSale($data, $id_sale)
10.    {
11.        $db = \Config\Database::connect();
12.        $builder = $db->table('sales');
13.        $builder->where('id_sale', $id_sale);
14.        $builder->update($data);
15.        return $this->db->affectedRows();
16.    }
17.    public function getTotalItemSoldByIdItem($id_item)
18.    {
19.        return $this->db->query("SELECT items.name AS nama_barang ,
   sum(sales.quantity) AS barang_terjual
20.        FROM items
21.        INNER JOIN sales ON items.id_item=sales.id_item
22.        WHERE sales.id_status=1 AND sales.id_item=$id_item
23.        GROUP BY sales.id_item")->getRow();
24.    }
25.    public function getTotalItemExpByIdItem($id_item)
26.    {
27.        return $this->db->query("SELECT items.name AS nama_barang ,
   sum(sales.quantity) AS barang_exp
```

```

28.     FROM items
29.     INNER JOIN sales ON items.id_item=sales.id_item
30.     WHERE sales.id_status=0 AND sales.id_item=$id_item
31.     GROUP BY sales.id_item")->getRow();
32.     }

```

*Kode Sumber 5. 8 Halaman SalesModel*

*Tabel 34. Penjelasan Kode sumber 5.8*

No.Baris	Kegunaan
6-8	Menampilkan seluruh data penjualan barang
9-16	Memasukkan data pengubaha penjualan barang ke database
19-24	Menampilkan jumlah data terjual

### 5.1.3.3 StockModel.php

Kode sumber 5.9 merupakan implementasi dari StockModel menghubungkan website dengan tabel stock yang berada di dalam database. Kode sumber ini juga menampilkan data, menyediakan fitur pencarian, melakukan perhitungan jumlah data, dan mengambil data dengan jumlah tertentu. Penjelasan kode sumber dapat dilihat pada tabel 35

```

1.  <?php
2.  namespace App\Models;
3.  use CodeIgniter\Model;
4.  class StockModel extends Model
5.  {
6.      public function getStockByIdItem($id_item)
7.      {
8.          return $this->db->query("SELECT * FROM stock WHERE id_item=" .
          $id_item)->getRow();
9.      }
10.
11.     public function updateStock($data, $id_stock)

```

```

12.  {
13.    $db = \Config\Database::connect();
14.    $builder = $db->table('stock');
15.
16.    $builder->where('id_stock', $id_stock);
17.    $builder->update($data);
18.    return $this->db->affectedRows();
19.  }
20.
21.  public function insertStock($data)
22.  {
23.    $db = \Config\Database::connect();
24.    $builder = $db->table('stock');
25.
26.    $builder->insert($data);
27.    return $this->db->affectedRows();
28.  }
29. }

```

*Kode Sumber 5. 9 Halaman StockModel*

*Tabel 35. Penjelasan Kode Sumber 5.9*

No.Baris	Kegunaan
6-9	Menampilkan semua data stok
11-19	Mengubah jumlah stok barang
21-28	Menambahkan stok barang kedalam database

### 5.1.3.4 SupplyModel.php

Kode sumber 5.10 merupakan implementasi dari SupplyModel menghubungkan website dengan tabel supply yang berada di dalam database. Kode sumber ini juga menampilkan data, menyediakan fitur pencarian, melakukan perhitungan jumlah data, dan mengambil data dengan jumlah tertentu. Penjelasan kode sumber dapat dilihat pada tabel 36

```

1.  <?php

```



```

2. namespace App\Models;
3. use CodeIgniter\Model;
4. class SupplyModel extends Model
5. {
6.     public function getSupply($id_supply)
7.     {
8.         return $this->db->query("SELECT * FROM supply WHERE id_supply=" .
           $id_supply)->getrow();
9.     }
10.
11.    public function getUpdatesupply($data, $id_supply)
12.    {
13.        $db = \Config\Database::connect();
14.        $builder = $db->table('supply');
15.
16.        $builder->where('id_supply', $id_supply);
17.        $builder->update($data);
18.        return $this->db->affectedRows();
19.    }
20. }
21.

```

*Kode Sumber 5.10 Halaman SupplyModel*

*Tabel 36. Penjelasan Kode Sumber 5.10*

No.Baris	Kegunaan
6-9	Menampilkan semua data pemasukan barang
11-19	Mengubah data pemasukan barang dan menyimpan ke database

### 5.1.3.5 KaryawanModel.php

Kode sumber 5.11 implementasi KaryawanModel menghubungkan website dengan tabel Karyawan yang berada di dalam database. Kode sumber ini juga menampilkan data,

menyediakan fitur pencarian, mengambil data dengan jumlah tertentu dan melakukan hashing password. Penjelasan kode sumber dapat dilihat pada tabel 37

```

1. <?php
2. namespace App\Models;
3. use CodeIgniter\Model;
4. use Backpack\CRUD\app\Models\Traits\CrudTrait;
5. class UserModel extends Model
6. {
7.     public function getUser($id_user)
8.     {
9.         return $this->db->query("SELECT * FROM users WHERE id_user=" .
           $id_user->getrow());
10.    }
11.    public function getUpdateUser($data, $id_user)
12.    {
13.        $db = \Config\Database::connect();
14.        $builder = $db->table('users');
15.
16.        $builder->where('id_user', $id_user);
17.        $builder->update($data);
18.        return $this->db->affectedRows();
19.    }
20.
21.    public function getTotalUsers()
22.    {
23.        return $this->db->query("SELECT count(*) as total_users FROM users")-
           >getRow();
24.    }
25. }

```

*Kode Sumber 5.11 Halaman KaryawanModel*

*Tabel 37. Penjelasan Kode Sumber 5.11*

No.Baris	Kegunaan
7-10	Menampilkan data karyawan
11-19	Mengubah data karyawan

21-24	Menampilkan jumlah karyawan
-------	-----------------------------

## 5.1.4 Folder View

### 5.1.4.1 View Dashboard

Kode sumber 5.12 merupakan implementasi view Dashboard menanggapi dalam menampilkan tampilan dashboard.

```

1. <?=$this->extend('layout/template') ?>
2. <?=$this->section('content') ?>
3.
4. <!-- Main Content -->
5. <div id="content">
6.     <?php
7.         $session = session();
8.     ?>
9.     <div class="container-fluid">
10.         <div class="d-sm-flex align-items-center justify-content-between mb-4">
11.             <h1 class="h3 text-gray-800">Dashboard</h1>
12.             <form action="/pages/searchDate" method="post">
13.                 </form>
14.         </div>
15.         <div class="row">
16.             <div class="col-xl-3 col-md-6 mb-4">
17.                 <div class="card border-left-info shadow h-100 py-2">
18.                     <div class="card-body">
19.                         <div class="row no-gutters align-items-center">
20.                             <div class="col mr-2">
21.                                 <div class="text-xs font-weight-bold text-info text-uppercase
mb-1">Stok Terjual</div>
22.                                 <div class="h5 mb-0 font-weight-bold text-gray-800"><?=$
$total_items; ?></div>
23.                             </div>
24.                         <div class="col-auto">
25.                             <i class="fas fa-clipboard-list fa-2x text-gray-300"></i>

```

```

26.         </div>
27.         </div>
28.     </div>
29. </div>
30. <div class="col-xl-3 col-md-6 mb-4">
31.     <div class="card border-left-success shadow h-100 py-2">
32.         <div class="card-body">
33.             <div class="row no-gutters align-items-center">
34.                 <div class="col mr-2">
35.                     <div class="text-xs font-weight-bold text-success text-
uppercase mb-1">Stok Barang</div>
36.                 <div class="row no-gutters align-items-center">
37.                     <div class="col-auto">
38.                         <div class="h5 mb-0 mr-3 font-weight-bold text-gray-
800"><?= $total_stock; ?></div>
39.                     </div>
40.                 </div>
41.             <div class="col-auto">
42.                 <i class="fas fa-clipboard-list fa-2x text-gray-300"></i>
43.             </div>
44.         </div>
45.     <div class="col-xl-3 col-md-6 mb-4">
46.         <div class="card border-left-danger shadow h-100 py-2">
47.             <div class="card-body">
48.                 <div class="row no-gutters align-items-center">
49.                     <div class="col mr-2">
50.                         <div class="text-xs font-weight-bold text-danger text-
uppercase mb-1">Stock Expired</div>
51.                     <div class="h5 mb-0 font-weight-bold text-gray-800"><?=
$total_stock_exp; ?></div>
52.                 </div>
53.             <div class="col-auto">
54.                 <i class="fas fa-clipboard-list fa-2x text-gray-300"></i>
55.             </div>
56.         </div>
<?= $this->endSection() ?>

```

*Kode Sumber 5.12 Halaman View Dashboard*

#### 5.1.4.2 View Login

Kode sumber 5.13 merupakan implementasi view Login menangani dalam menampilkan tampilan login saat pengguna mengakses website.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.
6. <meta charset="utf-8">
7. <meta http-equiv="X-UA-Compatible" content="IE=edge">
8. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
9. <meta name="description" content="">
10. <meta name="author" content="">
11.
12. <title>Login</title>
13.
14. <!-- Custom fonts for this template-->
15. <link href="<? = base_url('vendor/fontawesome-free/css/all.min.css'); ?>"
rel="stylesheet" type="text/css">
16. <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">
17.
18. <!-- Custom styles for this template-->
19. <link href="<? = base_url('css/sb-admin-2.min.css'); ?>" rel="stylesheet">
20. <style>
21. .center {
22.     display: flex;
23.     justify-content: center;
24.     align-items: center;
25. }
26. </style>
27. </head>
28.
```

```

29. <body class="bg-gradient-primary">
30.
31.   <div class="container">
32.     <div class="row justify-content-center">
33.       <div class="col-lg-5">
34.         <div class="card o-hidden border-0 shadow-lg my-5">
35.           <div class="card-body p-0">
36.             <!-- Nested Row within Card Body -->
37.             <div class="row">
38.               <div class="col-lg">
39.                 <div class="p-5">
40.                   <div class="text-center">
41.                     <h1 class="h4 text-gray-900 mb-4">Welcome
Back!</h1>
42.                   </div>
43.                   <?php if (session()->getFlashdata('pesan')) : ?>
44.                     <div class="alert alert-danger" role="alert">
45.                       <?= session()->getFlashdata('pesan'); ?>
46.                     </div>
47.                   <?php endif; ?>
48.                   <form class="user" method="POST" action="<?=
base_url('auth/login'); ?>">
49.                     <div class="form-group">
50.                       <input type="email" class="form-control form-
control-user" id="exampleInputEmail" name="email" aria-
describedby="emailHelp" placeholder="Enter Email Address" required>
51.                     </div>
52.                     <div class="form-group">
53.                       <input type="password" class="form-control form-
control-user" id="exampleInputPassword" name="password" placeholder="Enter
Password" required>
54.                     </div>
55.                     <div class="center">
56.                       <button type="submit" class="btn btn-
primary">login</button>
57.                     </div>
58.                   </form>
59.                   <hr>
60.                 </div>

```

```

61.         </div>
62. </body>
63. </html>

```

*Kode Sumber 5. 13 Halaman View Login*

#### 5.1.4.3 View Items

Kode sumber View Items(Barang) menangani proses dalam penambahan Barang seperti pada Kode sumber 5.14, mengubah data Barang seperti pada Kode sumber 5.15 , melihat daftar Barang seperti pada Kode sumber 5.16 dan menghapus Barang seperti pada Kode sumber 5.17

##### 5.1.4.3.1 create.php

```

1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3.
4. <div class="container-fluid">
5.     <div class="card shadow mb-4">
6.         <div class="card-header py-3">
7.             <h6 class="m-0 font-weight-bold text-primary">Form Data Barang</h6>
8.         </div>
9.         <div class="card-body">
10.            <form action="/item/save" method="POST">
11.                <?= csrf_field(); ?>
12.                <div class="form-group row">
13.                    <label for="inputEmail3" class="col-sm-2 col-form-label">Nama
Barang</label>
14.                    <div class="col-sm-10">
15.                        <input type="text" class="form-control <?= ($validation-
>hasError('name') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="name"
placeholder="Nama Barang">
16.                    <div class="invalid-feedback">
17.                        <?= $validation->getError('name'); ?>
18.                    </div>
19.                </div>
20.            </form>

```

```

21.         <div class="form-group row">
22.             <label for="inputHargaAsli" class="col-sm-2 col-form-label">Harga
Asli</label>
23.             <div class="col-sm-10">
24.                 <input type="text" class="form-control <?=$validation-
>hasError('buying_price') ? 'is-invalid' : ''; ?>" id="inputPassword3"
name="buying_price" placeholder="Harga Asli">
25.                 <div class="invalid-feedback">
26.                     <?=$validation->getError('buying_price'); ?>
27.                 </div>
28.             </div>
29.         </div>
30.         <div class="form-group row">
31.             <label for="inputHargaJual" class="col-sm-2 col-form-label">Harga
Jual</label>
32.             <div class="col-sm-10">
33.                 <input type="text" class="form-control <?=$validation-
>hasError('selling_price') ? 'is-invalid' : ''; ?>" id="inputEmail3"
name="selling_price" placeholder="Harga Jual">
34.                 <div class="invalid-feedback">
35.                     <?=$validation->getError('selling_price'); ?>
36.                 </div>
37.             </div>
38.         </div>
39.         <div class="form-group row">
40.             <label for="inputDeskripsi" class="col-sm-2 col-form-
label">Deskripsi</label>
41.             <div class="col-sm-10">
42.                 <input type="text" class="form-control <?=$validation-
>hasError('description') ? 'is-invalid' : ''; ?>" id="inputEmail3"
name="description" placeholder="Deskripsi">
43.                 <div class="invalid-feedback">
44.                     <?=$validation->getError('description'); ?>
45.                 </div>
46.             </div>
47.         </div>
48.         <div class="form-group row">
49.             <div class="col-sm-10">
50.                 <button type="submit" class="btn btn-primary">Submit</button>

```



```

51.         </div>
52.     </div>
53. </form>
54.
55. <?= $this->endSection() ?>

```

*Kode Sumber 5. 14 Halaman View ItemCreate*

### 5.1.4.3.2 edit.php

```

1. <?= $this->extend('layout/templatee') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Data Barang</h6>
7.         </div>
8.         <div class="card-body">
9.             <form action="<?= base_url('item/update/' . $item->id_item) ?>"
method="POST">
10.                 <?= csrf_field(); ?>
11.                 <div class="form-group row">
12.                     <label for="inputEmail3" class="col-sm-2 col-form-label">Nama
Pencatat</label>
13.                     <div class="col-sm-10">
14.                         <input type="text" class="form-control <?= ($validation-
>hasError('id_user') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="id_user"
value="<?= $item->id_user; ?>" placeholder="Nama Pencatat">
15.                         <div class="invalid-feedback">
16.                             <?= $validation->getError('id_user'); ?>
17.                         </div>
18.                     <div class="form-group row">
19.                         <label for="inputEmail3" class="col-sm-2 col-form-label">Nama
Barang</label>
20.                         <div class="col-sm-10">
21.                             <input type="text" class="form-control <?= ($validation-
>hasError('name') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="name"
value="<?= $item->name; ?>" placeholder="Nama Barang">
22.                             <div class="invalid-feedback">
23.                                 <?= $validation->getError('name'); ?>

```

```

24.         </div>
25.         <div class="form-group row">
26.             <label for="inputHargaAsli" class="col-sm-2 col-form-label">Harga
Asli</label>
27.             <div class="col-sm-10">
28.                 <input type="text" class="form-control <?=$validation-
>hasError('buying_price') ? 'is-invalid' : '';" id="inputPassword3"
name="buying_price" value="<?=$item->buying_price; ?>"
placeholder="Harga Asli">
29.                 <div class="invalid-feedback">
30.                     <?=$validation->getError('buying_price'); ?>
31.                 </div>
32.             <div class="form-group row">
33.                 <label for="inputHargaJual" class="col-sm-2 col-form-label">Harga
Jual</label>
34.                 <div class="col-sm-10">
35.                     <input type="text" class="form-control <?=$validation-
>hasError('selling_price') ? 'is-invalid' : '';" id="inputEmail3"
name="selling_price" value="<?=$item->selling_price; ?>" placeholder="Harga
Jual">
36.                     <div class="invalid-feedback">
37.                         <?=$validation->getError('selling_price'); ?>
38.                     </div>
39.                 <div class="form-group row">
40.                     <label for="inputDeskripsi" class="col-sm-2 col-form-
label">Deskripsi</label>
41.                     <div class="col-sm-10">
42.                         <input type="text" class="form-control <?=$validation-
>hasError('description') ? 'is-invalid' : '';" id="inputEmail3"
name="description" value="<?=$item->description ?>"
placeholder="Deskripsi">
43.                         <input type="hidden" value="<?php $item->id_item ?>">
44.                         <div class="invalid-feedback">
45.                             <?=$validation->getError('description'); ?>
46.                         </div>
47.                     <div class="form-group row">
48.                         <div class="col-sm-10">
49.                             <button type="submit" class="btn btn-primary">Submit</button>
50.                         </div>
51.                     </div>

```

```
52.     </form>
53. <?= $this->endSection() ?>
```

*Kode Sumber 5.15 Halaman View ItemEdit*

### 5.1.4.3.2 index.php

```
1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <!-- Page Heading -->
5.     <h1 class="h3 mb-2 text-gray-800">Table Data Barang</h1>
6.     <?php if (session()->getFlashdata('pesan')) : ?>
7.         <div class="alert alert-success" role="alert">
8.             <?= session()->getFlashdata('pesan'); ?>
9.         </div>
10.    <?php endif; ?>
11.    <!-- button tambah barang -->
12.    <a href="<?= base_url('item/create'); ?>" class="btn btn-primary btn-icon-split
mb-3">
13.        <span class="icon text-white-50">
14.            <i class="fas fa-plus"></i>
15.        </span>
16.        <span class="text">Tambah Barang</span>
17.    </a>
18.    <!-- DataTales Example -->
19.    <div class="card shadow mb-4">
20.        <div class="card-header py-3">
21.            <h6 class="m-0 font-weight-bold text-primary">Catatan Data Pemasukan
Barang</h6>
22.        </div>
23.        <div class="card-body">
24.            <div class="table-responsive">
25.                <table class="table table-bordered" id="dataTable" width="100%"
cellspacing="0">
26.                    <thead>
27.                        <tr>
28.                            <th>No</th>
29.                            <th>Pencatat </th>
```

```

30.         <th>Nama Barang </th>
31.         <th>Harga Asli</th>
32.         <th>Harga Jual</th>
33.         <th>Aksi</th>
34.     </tr>
35. </thead>
36. <tbody>
37.     <?php $i = 1; ?>
38.     <?php foreach ($item a $b): ?>
39.         <tr>
40.             <th scope="row"><?=$ $i++; ?></th>
41.             <td><?=$ $b->username; ?></td>
42.             <td><?=$ $b->name; ?></td>
43.             <td><?="Rp " . number_format($b->buying_price, 0, ',',
44.             '.) ?></td>
45.             <td><?="Rp " . number_format($b->selling_price, 0, ',',
46.             '.) ?></td>
47.             <td>
48.                 <div class="row">
49.                     <i class="fas fa-eye"></i>
50.                 </span>
51.                 <span class="text">View</span>
52.             </a>
53.             <a href="<?=$ base_url('item/view/' . $b->id_item) ?>"
54.             class="btn btn-info btn-icon-split btn-sm mx-1">
55.                 <a href="<?=$ base_url('item/edit/' . $b->id_item) ?>"
56.                 class="btn btn-warning btn-icon-split btn-sm mx-1">
57.                     <span class="icon text-white-50">
58.                         <i class="fas fa-edit"></i>
59.                     </span>
60.                     <span class="text">Edit</span>
61.                 </a>
62.                 <form action="<?=$ $b->id_item; ?>"
63.                 method="POST" class="d-inline">
64.                     <?=$ csrf_field(); ?>
65.                     <input type="hidden" name="_method"
66.                     value="DELETE">

```

```

62.         <button type="submit" class="btn btn-danger btn-
            icon-split btn-sm " onclick="return confirm('apakah anda yakin?');">
63.         <span class="icon text-white-50">
64.             <i class="fas fa-trash"></i>
65.         </span>
66.         <span class="text">Delete</span></button>
67.     </form>
68.

```

*Kode Sumber 5. 16 Halaman View ItemIndex*

#### 5.1.4.3.3 view.php

```

1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Data Barang</h6>
7.         </div>
8.         <div class="card-body">
9.             <!-- jika untung ada dan stock ada -->
10.            <?php if ($keuntungan != null && $stock != null) : ?>
11.                <div class="form-group row">
12.                    <div class="col-sm-3">
13.                        <div class="mx-2" style="color:Black;">Nama Barang </div>
14.                    </div>
15.                    <div class="col-sm-3">
16.                        <div class="text-secondary mx-2 text-uppercase font-weight-
                            bold"><?= $name_item; ?></div>
17.                    </div>
18.                </div>
19.                <div class="form-group row">
20.                    <div class="col-sm-3">
21.                        <div class="mx-2" style="color:Black;">Jumlah stok</div>
22.                    </div>
23.                    <div class="col-sm-3">

```

```

24.         <div class="text-secondary mx-2"><?= $stock->jumlah; ?></div>
25.         </div>
26.     </div>
27.     <div class="form-group row">
28.         <div class="col-sm-3">
29.             <div class="mx-2" style="color:Black;">Jumlah barang
                terjual</div>
30.         </div>
31.         <div class="col-sm-3">
32.             <div class="text-secondary mx-2"><?= $sold; ?></div>
33.         </div>
34.     </div>
35.     <div class="form-group row">
36.         <div class="col-sm-3">
37.             <div class="mx-2" style="color:Black;">Keuntungan</div>
38.         </div>
39.         <div class="col-sm-3">
40.             <div class="text-secondary mx-2" style="color:Black;"><?= "Rp
                " . number_format($keuntungan, 0, ',', '.') ?></div>
41.         </div>
42.     </div>
43.     <?php if ($kerugian != null) : ?>
44.         <div class="form-group row">
45.             <div class="col-sm-3">
46.                 <div class="mx-2" style="color:Black;">Kerugian</div>
47.             </div>
48.             <div class="col-sm-3">
49.                 <div class="text-secondary mx-2"><?= "Rp " .
                number_format($kerugian, 0, ',', '.') ?></div>
50.             <?= $this->endSection() ?>

```

*Kode Sumber 5. 17 Halaman View Itemview*

#### 5.1.4.4 View Sales

Kode sumber View Sales (Penjualan Barang) menangani proses dalam penambahan Penjualan Barang seperti pada Kode sumber 5.18, mengubah data Penjualan Barang seperti pada Kode

sumber 5.19 dan melihat daftar Penjualan Barang seperti pada Kode sumber 5.20.

#### 5.1.4.4.1 create.php

```
1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <?php if (session()->getFlashdata('pesan')) : ?>
5.         <div class="alert alert-danger" role="alert">
6.             <?= session()->getFlashdata('pesan'); ?>
7.         </div>
8.     <?php endif; ?>
9.     <div class="card shadow mb-4">
10.        <div class="card-header py-3">
11.            <h6 class="m-0 font-weight-bold text-primary">Form Pencatatan
                Penjualan</h6>
12.        </div>
13.        <div class="card-body">
14.            <form action="/sales/save" method="POST">
15.                <?= csrf_field(); ?>
16.                <div class="form-group row">
17.                    <label for="inputUsername" class="col-sm-2 col-form-label">Nama
                        Barang</label>
18.                    <div class="col-sm-10">
19.                        <select name="id_item" class="form-control">
20.                            <?php foreach ($items as $item) : ?>
21.                                <option value=?= $item['id_item']; ?><?=
                                    $item['name']; ?></option>
22.                            <?php endforeach; ?>
23.                        </select>
24.                        <!-- <input type="text" class="form-control <?= ($validation
                            >hasError('id_item') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="id_item"
                            placeholder="Nama Barang" -->
25.                    </div>
26.                </div>
27.            <div class="form-group row">
```

```

28.         <label for="inputJumlah" class="col-sm-2 col-form-label">Jumlah
Barang</label>
29.         <div class="col-sm-10">
30.             <input type="text" class="form-control <?=$validation-
>hasError('quantity') ? 'is-invalid' : ''; ?>" id="inputEmail3" name="quantity"
placeholder="Jumlah Barang">
31.         </div>
32.     </div>
33.     <!-- <div class="form-group row">
34.         <label for="inputHargaAsli" class="col-sm-2 col-form-
label">Tanggal Masuk</label>
35.         <div class="col-sm-10">
36.             <input type="datetime-local" class="form-control <?=$validation-
>hasError('created_at') ? 'is-invalid' : ''; ?>" id="inputPassword3"
name="created_at" placeholder="Tanggal Masuk">
37.         </div>
38.     </div -->
39.     <div class="form-group row">
40.         <label for="inputHargaAsli" class="col-sm-2 col-form-
label">Status</label>
41.         <div class="col-sm-10 col-md-4">
42.             <select id="inputState" class="form-control <?=$validation-
>hasError('id_status') ? 'is-invalid' : ''; ?>" name="id_status" required>
43.                 <option selected>Choose Status</option>
44.                 <option value=0>Expired</option>
45.                 <option value=1>Sold</option>
46.             </select>
47.             <div class="invalid-feedback">
48.                 <?=$validation->getError('id_status'); ?>
49.             </div>
50.         </div>
51.     </div>
52.     <div class="form-group row">
53.         <div class="col-sm-10">
54.             <button type="submit" class="btn btn-primary">Submit</button>
55.         </div>
56.     </div>
57. </form>
58.

```



```
59. <?= $this->endSection() ?>
```

*Kode Sumber 5.18 Halaman View SalesCreate*

#### 5.1.4.4.2 edit.php

```
1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Pencatatan
Penjualan</h6>
7.         </div>
8.         <div class="card-body">
9.             <form action="<?= base_url('sales/update/' . $sale->id_sale); ?>"
method="POST">
10.                 <?= csrf_field(); ?>
11.                 <div class="form-group row">
12.                     <label for="inputEmail" class="col-sm-2 col-form-label">Nama
Barang</label>
13.                     <div class="col-sm-10">
14.                         <select name="id_item" class="form-control">
15.                             <?php foreach ($items as $it) : ?>
16.                                 <option value=<?= $it['id_item']; ?>><?=
$it['name']; ?></option>
17.                             <?php endforeach; ?>
18.                         </select>
19.                         <!-- <input type="text" class="form-control <?= ($validation-
>hasError('id_item') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="id_item"
value="<?= $sale->id_item; ?>" placeholder="Nama Barang"> -->
20.                     </div>
21.                 </div>
22.                 <div class="form-group row">
23.                     <label for="inputJumlah" class="col-sm-2 col-form-label">Jumlah
Barang</label>
24.                     <div class="col-sm-10">
25.                         <input type="text" class="form-control <?= ($validation-
>hasError('quantity') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="quantity"
value="<?= $sale->quantity; ?>" placeholder="Jumlah Barang">
26.                     </div>
```

```

27.         </div>
28.         <!-- <div class="form-group row">
29.             <label for="inputHargaAsli" class="col-sm-2 col-form-
label">Tanggal Masuk</label>
30.             <div class="col-sm-10">
31.                 <input type="datetime-local" class="form-control "
id="inputPassword3" name="created_at" value="<?= $sale->created_at; ?>"
placeholder="Tanggal Masuk">
32.                 <div class="invalid-feedback">
33.                     <?= $validation->getError('created_at'); ?>
34.                 </div>
35.             </div>
36.         </div> -->
37.         <div class="form-group row">
38.             <label for="inputHargaAsli" class="col-sm-2 col-form-
label">Status</label>
39.             <div class="col-sm-10 col-md-4">
40.                 <select id="inputState" class="form-control " name="id_status"
value="<?= $sale->id_status; ?>" required>
41.                     <option selected>Choose Status</option>
42.                     <option value=0>Expired</option>
43.                     <option value=1>Sold</option>
44.                 </select>
45.             </div>
46.         </div>
47.         <div class="form-group row">
48.             <div class="col-sm-10">
49.                 <button type="submit" class="btn btn-primary">Submit</button>
50.             </div>
51.         </div>
52.     </form>
53. <?= $this->endSection() ?>

```

*Kode Sumber 5.19 Halaman View SalesEdit*

#### 5.1.4.4.3 index.php

```

1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">

```

```

4. <!-- Page Heading -->
5. <h1 class="h3 mb-2 text-gray-800">Table Data Penjualan</h1>
6. <?php if (session()->getFlashdata('pesan')) : ?>
7.     <div class="alert alert-success" role="alert">
8.         <?= session()->getFlashdata('pesan'); ?>
9.     </div>
10. <?php endif; ?>
11. <a href="<?= base_url('sales/create'); ?>" class="btn btn-primary btn-icon-split
mb-3">
12.     <span class="icon text-white-50">
13.         <i class="fas fa-plus"></i>
14.     </span>
15.     <span class="text">Tambah Data</span>
16. </a>
17. <!-- DataTales Example -->
18. <div class="card shadow mb-4">
19.     <div class="card-header py-3">
20.         <h6 class="m-0 font-weight-bold text-primary">Catatan Data
Penjualan</h6>
21.     </div>
22.     <div class="card-body">
23.         <div class="table-responsive">
24.             <table class="table table-bordered" id="dataTable" width="100%"
cellspacing="0">
25.                 <thead>
26.                     <tr>
27.                         <th>No</th>
28.                         <th>Item</th>
29.                         <th>Pencatat </th>
30.                         <th>Jumlah</th>
31.                         <th>Total</th>
32.                         <th>Status</th>
33.                         <th>Tanggal Masuk</th>
34.                         <?php if (session()->get('id_role') == 1) : ?>
35.                             <th>Aksi</th>
36.                         <?php endif ?>
37.                     </tr>
38.                 </thead>

```

```

39.         <tbody>
40.             <?php $i = 1; ?>
41.             <?php foreach ($sales as $s) : ?>
42.                 <tr>
43.                     <th scope="row"><?= $i++; ?></th>
44.                     <td><?= $s->item_name; ?></td>
45.                     <td><?= $s->user_name; ?></td>
46.                     <td><?= $s->quantity; ?></td>
47.                     <td><?= "Rp " . number_format($s->total_price, 0, ',',
'.' ) ?></td>
48.                     <td class="text-uppercase font-weight-bold"><?= $s-
>status_name; ?></td>
49.                     <td><?= $s->created_at; ?></td>
50.                 <div>
51.                     <a href="<?= base_url('sales/edit/' . $s->id_sale) ?>"
class="btn btn-warning btn-icon-split btn-sm mx-1">
52.                         <span class="icon text-white-50">
53.                             <i class="fas fa-edit"></i>
54.                         </span>
55.                         <span class="text">Edit</span>
56.                     </a>
57.                     <form action="/sales/<?= $s->id_sale; ?>"
method="POST" class="d-inline">
58.                         <?= csrf_field(); ?>
59.                         <input type="hidden" name="_method"
value="DELETE">
60.                         <button type="submit" class="btn btn-danger btn-
icon-split btn-sm" onclick="return confirm('apakah anda yakin?');">
61.                             <span class="icon text-white-50">
62.                                 <i class="fas fa-trash"></i>
63.                         <button class="text">Delete</span></button>
64.                     </form>
65.                 </div>
66.             </td>
67.         <?php endif ?>
68.     </tr>
69. <?php endforeach; ?>
70. </tbody>
71. </table>

```

```
72. <?= $this->endSection() ?>
```

*Kode Sumber 5. 20 Halaman View SalesIndex*

#### 5.1.4.5 View Supply

Kode sumber View Supply(Pemasukan Barang) menangani proses dalam penambahan Pemasukan Barang seperti pada Kode sumber 5.21, mengubah data Pemasukan Barang seperti pada Kode sumber 5.22 dan melihat daftar Pemasukan Barang seperti pada Kode sumber 5.23.

##### 5.1.4.5.1 create.php

```
1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Data Barang</h6>
7.         </div>
8.         <div class="card-body">
9.             <form action="/supply/save" method="POST">
10.                 <?= csrf_field(); ?>
11.                 <div class="form-group row">
12.                     <label for="inputUsername" class="col-sm-2 col-form-label">Nama
Barang</label>
13.                     <div class="col-sm-10">
14.                         <!-- bikin dropdown , foreach item -->
15.                         <select name="id_item" class="form-control">
16.                             <?php foreach ($items as $it) : ?>
17.                                 <option         value=<?=         $it['id_item'];         ?>><?=
$it['name']; ?></option>
18.                             <?php endforeach; ?>
19.                         </select>
20.                         <!-- <input type=" text" class="form-control <?= ($validation-
hasError('id_item') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="id_item"
placeholder="Nama Barang" -->
21.                         <div class="invalid-feedback">
22.                             <?= $validation->getError('id_item'); ?>
```

```

23.         </div>
24.     </div>
25. </div>
26.     <div class="form-group row">
27.         <label for="inputJumlah" class="col-sm-2 col-form-label">Jumlah
Barang</label>
28.         <div class="col-sm-10">
29.             <input type="text" class="form-control <?=( $validation-
>hasError('quantity') ? 'is-invalid' : '); ?>" id="inputEmail3" name="quantity"
placeholder="Jumlah Barang">
30.             <div class="invalid-feedback">
31.                 <?=( $validation->getError('quantity'); ?>
32.             </div>
33.         </div>
34.     </div>
35.     <div class="form-group row">
36.         <label for="inputTanggalMasuk" class="col-sm-2 col-form-
label">Tanggal Masuk</label>
37.         <div class="col-sm-10">
38.             <input type="datetime-local" class="form-control <?=(
$validation->hasError('created_at') ? 'is-invalid' : '); ?>" id="inputEmail3"
name="created_at" placeholder="Tanggal Masuk">
39.             <div class="invalid-feedback">
40.                 <?=( $validation->getError('created_at'); ?>
41.             </div>
42.         </div>
43.     </div>
44.     <div class="form-group row">
45.         <div class="col-sm-10">
46.             <button type="submit" class="btn btn-primary">Submit</button>
47.         </div>
48.     </div>
49. </form>
50. </div>
51. </div>
52. </div>
53.
54. <?=( $this->endSection() ?>

```

*Kode Sumber 5. 21 Halaman View SupplyCreate*

### 5.1.4.5.2 edit.php

```
1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Data Barang</h6>
7.         </div>
8.         <div class="card-body">
9.             <form action="<?= base_url('supply/update/' . $supply->id_supply); ?>"
method="POST">
10.                 <?= csrf_field(); ?>
11.                 <div class="form-group row">
12.                     <label for="inputUsername" class="col-sm-2 col-form-label">Nama
Barang</label>
13.                     <div class="col-sm-10">
14.                         <select name="id_item" class="form-control">
15.                             <?php foreach ($items as $it) : ?>
16.                                 <option
value=<?= $it['id_item']; ?>><?=
$it['name']; ?></option>
17.                             <?php endforeach; ?>
18.                         </select>
19.                         <div class="invalid-feedback">
20.                             <?= $validation->getError('id_item'); ?>
21.                         </div>
22.                     </div>
23.                 </div>
24.                 <div class="form-group row">
25.                     <label for="inputJumlah" class="col-sm-2 col-form-label">Jumlah
Barang</label>
26.                     <div class="col-sm-10">
27.                         <input type="text" class="form-control <?= ($validation-
>hasError('quantity') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="quantity"
value="<?= $supply->quantity; ?>" placeholder="Jumlah Barang">
28.                         <div class="invalid-feedback">
29.                             <?= $validation->getError('quantity'); ?>
30.                         </div>
31.                     </div>
```

```

32.         </div>
33.         <div class="form-group row">
34.             <label for="inputTanggalMasuk" class="col-sm-2 col-form-
label">Tanggal Masuk</label>
35.             <div class="col-sm-10">
36.                 <input type="datetime-local" class="form-control <?=
($validation->hasError('created_at') ? 'is-invalid' : ''); ?>" id="inputEmail3"
name="created_at" value="<?= $supply->created_at; ?>" placeholder="Tanggal
Masuk">
37.                 <div class="invalid-feedback">
38.                     <?= $validation->getError('created_at'); ?>
39.                 </div>
40.             </div>
41.         </div>
42.         <div class="form-group row">
43.             <div class="col-sm-10">
44.                 <button type="submit" class="btn btn-primary">Submit</button>
45.             </div>
46.         </div>
47.     </form>
48. </div>
49. </div>
50. </div>
51.
52. <?= $this->endSection() ?>

```

*Kode Sumber 5. 22 Halaman View SupplyEdit*

### 5.1.4.5.3 index.php

```

1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>
3. <div class="container-fluid">
4.     <!-- Page Heading -->
5.     <h1 class="h3 mb-2 text-gray-800">Table Data Pemasukan Barang</h1>
6.     <?php if (session()->getFlashdata('pesan')) : ?>
7.         <div class="alert alert-success" role="alert">
8.             <?= session()->getFlashdata('pesan'); ?>
9.         </div>
10.    <?php endif; ?>

```



```

11. <!-- button tambah user -->
12. <a href="<?= base_url('supply/create') ?>" class="btn btn-primary btn-icon-split mb-3">
13.     <span class="icon text-white-50">
14.         <i class="fas fa-plus"></i>
15.     </span>
16.     <span class="text">Tambah Data</span>
17. </a>
18. <div class="card shadow mb-4">
19.     <div class="card-header py-3">
20.         <h6 class="m-0 font-weight-bold text-primary">Catatan Data Pemasukan
Barang</h6>
21.     </div>
22.     <div class="card-body">
23.         <div class="table-responsive">
24.             <table class="table table-bordered" id="dataTable" width="100%"
cellspacing="0">
25.                 <thead>
26.                     <tr>
27.                         <th>No</th>
28.                         <th>Nama Item</th>
29.                         <th>Nama Pencatat </th>
30.                         <th>Jumlah</th>
31.                         <th>Tanggal Masuk</th>
32.                         <th>Aksi</th>
33.                     </tr>
34.                 </thead>
35.                 <tbody>
36.                     <?php $i = 1; ?>
37.                     <?php foreach ($supply as $s) : ?>
38.                         <tr>
39.                             <th scope="row"><?= $i++; ?></th>
40.                             <td><?= $s->item_name; ?></td>
41.                             <td><?= $s->user_name; ?></td>
42.                             <td><?= $s->quantity; ?></td>
43.                             <td><?= $s->created_at; ?></td>
44.                             <td>
45.                                 <div class="row">

```

```

46.             <!-- edit -->
47.             <a href="<?= base_url('supply/edit/' . $s->id_supply) ?>"
class="btn btn-warning btn-icon-split btn-sm mx-1">
48.                 <span class="icon text-white-50">
49.                     <i class="fas fa-edit"></i>
50.                 </span>
51.                 <span class="text">Edit</span>
52.             </a>
53.             <form action="/supply/<?= $s->id_supply; ?>" method="POST">
54.                 <?= csrf_field(); ?>
55.                 <input type="hidden" name="_method"
value="DELETE">
56.                 <button type="submit" class="btn btn-danger btn-
icon-split btn-sm mx-1" onclick="return confirm('apakah anda yakin?');">
57.                     <span class="icon text-white-50">
58.                         <i class="fas fa-trash"></i>
59.                     class="text">Delete</span>
60.                 </form>
61.             </div>
62.         </td>
63.     </tr>
64. <?php endforeach; ?>
65. </tbody>
66. </table>
67. <?= $this->endSection() ?>

```

*Kode Sumber 5. 23 Halaman View Supplyindex*

### 5.1.4.6 View Karyawan

Kode sumber View Karyawan menangani proses dalam pembuatan Karyawan baru seperti pada Kode sumber 5.24, mengubah data Karyawan seperti pada Kode sumber 5.25 dan melihat daftar Karyawan seperti pada Kode sumber 5.6.

#### 5.1.4.6.1 create.php

```

1. <?= $this->extend('layout/template') ?>
2. <?= $this->section('content') ?>

```

```

3. <div class="container-fluid">
4.   <div class="card shadow mb-4">
5.     <div class="card-header py-3">
6.       <h6 class="m-0 font-weight-bold text-primary">Form Data User</h6>
7.     </div>
8.     <div class="card-body">
9.       <form action="/user/save" method="POST">
10.        <? csrf_field(); ?>
11.        <div class="form-group row">
12.          <label for="inputUsername" class="col-sm-2 col-form-
label">Username</label>
13.          <div class="col-sm-10">
14.            <input type="text" class="form-control <? = ($validation-
>hasError('name') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="name"
placeholder="Username">
15.            <div class="invalid-feedback">
16.              <? = $validation->getError('name'); ?>
17.            </div>
18.          </div>
19.        </div>
20.        <div class="form-group row">
21.          <label for="inputEmail" class="col-sm-2 col-form-
label">Email</label>
22.          <div class="col-sm-10">
23.            <input type="email" class="form-control <? = ($validation-
>hasError('email') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="email"
placeholder="Email">
24.            <div class="invalid-feedback">
25.              <? = $validation->getError('email'); ?>
26.            </div>
27.          </div>
28.        </div>
29.        <div class="form-group row">
30.          <label for="inputPassword" class="col-sm-2 col-form-
label">Password</label>
31.          <div class="col-sm-10">
32.            <input type="password" class="form-control <? = ($validation-
>hasError('password') ? 'is-invalid' : ''); ?>" id="inputPassword3"
name="password" placeholder="Password">

```

```

33.         <div class="invalid-feedback">
34.             <?=$validation->getError('password'); ?>
35.         </div>
36.     </div>
37. </div>
38.     <div class="form-group row">
39.         <label for="inputEmail" class="col-sm-2 col-form-
label">Email</label>
40.         <div class="col-sm-10">
41.             <input type="text" class="form-control <?=$validation-
>hasError('email') ? 'is-invalid' : "; ?>" id="inputEmail" name="email"
placeholder="Email">
42.             <div class="invalid-feedback">
43.                 <?=$validation->getError('email'); ?>
44.             </div>
45.         </div>
46.     </div>
47.     <div class="form-group row">
48.         <label for="inputAddress" class="col-sm-2 col-form-
label">Address</label>
49.         <div class="col-sm-10">
50.             <input type="text" class="form-control <?=$validation-
>hasError('address') ? 'is-invalid' : "; ?>" id="inputAddress" name="address"
placeholder="Address">
51.             <div class="invalid-feedback">
52.                 <?=$validation->getError('address'); ?>
53.             </div>
54.         </div>
55.     </div>
56.     <div class="form-group row">
57.         <label for="inputRole" class="col-sm-2 col-form-
label">Role</label>
58.         <div class="col-sm-10 col-md-4">
59.             <select id="inputRole" class="form-control <?=$validation-
>hasError('id_role') ? 'is-invalid' : "; ?>" name="id_role" required>
60.                 <option selected>Choose role</option>
61.                 <option value=0>User</option>
62.                 <option value=1>Admin</option>
63.             </select>

```

```

64.         <div class="invalid-feedback">
65.             <?=$validation->getError('id_role'); ?>
66.         <div class="form-group row">
67.             <div class="col-sm-10">
68.                 <button type="submit" class="btn btn-
primary">Submit</button>
69.             </div>
70.         </div>
71.     </form>
72. <?=$this->endSection() ?>

```

*Kode Sumber 5. 24 Halaman View KaryawanCreate*

#### 5.1.4.6.2 edit.php

```

1. <?=$this->extend('layout/template') ?>
2. <?=$this->section('content') ?>
3. <div class="container-fluid">
4.     <div class="card shadow mb-4">
5.         <div class="card-header py-3">
6.             <h6 class="m-0 font-weight-bold text-primary">Form Data User</h6>
7.         </div>
8.         <div class="card-body">
9.             <form action="<?=$base_url('user/update/' . $user->id_user) ?>"
method="POST">
10.                 <?=$csrf_field(); ?>
11.                 <div class="form-group row">
12.                     <label for="inputEmail" class="col-sm-2 col-form-
label">Username</label>
13.                     <div class="col-sm-10">
14.                         <input type="text" class="form-control <?=$validation-
>hasError('name') ? 'is-invalid' : ''; ?>" id="inputEmail3" name="name"
value="<?=$user->name; ?>" placeholder="Username">
15.                     </div>
16.                 </div>
17.                 <div class="form-group row">
18.                     <label for="inputEmail" class="col-sm-2 col-form-
label">Email</label>
19.                     <div class="col-sm-10">

```

```

20.         <input type="email" class="form-control <?= ($validation-
>hasError('email') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="email"
value="<?= $user->email; ?>" placeholder="Email">
21.             </div>
22.         </div>
23.         <div class="form-group row">
24.             <label for="inputPassword" class="col-sm-2 col-form-
label">Password</label>
25.             <div class="col-sm-10">
26.                 <input type="password" class="form-control <?= ($validation-
>hasError('password') ? 'is-invalid' : ''); ?>" id="inputPassword3"
name="password" value="<?= $user->password; ?>" placeholder="Password">
27.                 </div>
28.             </div>
29.         <div class="form-group row">
30.             <label for="inputPhone" class="col-sm-2 col-form-
label">Phone</label>
31.             <div class="col-sm-10">
32.                 <input type="text" class="form-control <?= ($validation-
>hasError('phone') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="phone"
value="<?= $user->phone; ?>" placeholder="Phone">
33.                 </div>
34.             </div>
35.         <div class="form-group row">
36.             <label for="inputAddress" class="col-sm-2 col-form-
label">Address</label>
37.             <div class="col-sm-10">
38.                 <input type="text" class="form-control <?= ($validation-
>hasError('address') ? 'is-invalid' : ''); ?>" id="inputEmail3" name="address"
value="<?= $user->address; ?>" placeholder="Address">
39.                 </div>
40.             </div>
41.         <div class="form-group row">
42.             <label for="inputHargaAsli" class="col-sm-2 col-form-
label">Role</label>
43.             <div class="col-sm-10 col-md-4">
44.                 <select id="inputState" class="form-control <?= ($validation-
>hasError('id_role') ? 'is-invalid' : ''); ?>" name="id_role" required>
45.                     <option selected>Choose role</option>
46.                     <option value=0>User</option>

```

```

47.         <option value=1>Admin</option>
48.         </select>
49.     </div>
50. </div>
51.
52.     <div class="form-group row">
53.         <div class="col-sm-10">
54.             <button type="submit" class="btn btn-
primary">Submit</button>
55.         </div>
56.     </div>
57. </form>
58.
59. <?=$this->endSection() ?>

```

*Kode Sumber 5. 25 Halaman View KaryawanEdit*

#### 5.1.4.6.3 index.php

```

1. <?=$this->extend('layout/templatee') ?>
2. <?=$this->section('content') ?>
3. <div class="container-fluid">
4.     <h1 class="h3 mb-2 text-gray-800">Table Data User</h1>
5.     <?php if (session()->getFlashdata('pesan')) : ?>
6.         <div class="alert alert-success" role="alert">
7.             <?=$session()->getFlashdata('pesan'); ?>
8.         </div>
9.     <?php endif; ?>
10.    <a href="<?=$base_url('user/create') ?>" class="btn btn-primary btn-
icon-split mb-3">
11.        <span class="icon text-white-50">
12.            <i class="fas fa-plus"></i>
13.        </span>
14.        <span class="text">Tambah User</span>
15.    </a>
16. <!-- DataTales Example -->
17. <div class="card shadow mb-4">
18.     <div class="card-header py-3">

```

```

19.     <h6 class="m-0 font-weight-bold text-primary">Catatan Data
      User</h6>
20.     </div>
21.     <div class="card-body">
22.         <div class="table-responsive">
23.             <table class="table table-bordered" id="dataTable"
      width="100%" cellspacing="0">
24.                 <thead>
25.                     <tr>
26.                         <th>No</th>
27.                         <th>Nama</th>
28.                         <th>Email </th>
29.                         <th>Phone</th>
30.                         <th>Address</th>
31.                         <th>Role</th>
32.                         <th>Tanggal</th>
33.                         <th>Aksi</th>
34.                     </tr>
35.                 </thead>
36.                 <tbody>
37.                     <?php $i = 1; ?>
38.                     <?php foreach ($user as $u) : ?>
39.                         <tr>
40.                             <th scope="row"><?=$i++; ?></th>
41.                             <td><?=$u->name; ?></td>
42.                             <td><?=$u->email; ?></td>
43.                             <td><?=$u->phone; ?></td>
44.                             <td><?=$u->address; ?></td>
45.                             <td class="text-uppercase"><?=$u-
      >name_role; ?></td>
46.                             <td><?=$u->created_at; ?></td>
47.                             <td>
48.                                 <div class="col">
49.                                     <a href="<?=$base_url('user/edit/' . $u-
      >id_user) ?>" class="btn btn-warning btn-icon-split btn-sm my-1">
50.                                         <span class="icon text-white-50">
51.                                             <i class="fas fa-edit"></i>
52.                                         </span>

```



```

53.         <span class="text">Edit</span>
54.     </a>
55.     <form action="/user/<?= $u->id_user; ?>"
method="POST">
56.         <?= csrf_field(); ?>
57.         <input type="hidden" name="_method"
value="DELETE">
58.         <button type="submit" class="btn btn-danger
btn-icon-split btn-sm" onclick="return confirm('apakah anda yakin?');">
59.             <span class="icon text-white-50">
60.                 <i class="fas fa-trash"></i>
61.             </span>
62.             <span class="text">Delete</span>
63.         </button>
64.     </form>
65. </div>
66. </td>
67. </tr>
68. <?php endforeach; ?>
69. </tbody>
70. </table>
71. <?= $this->endSection() ?>

```

*Kode Sumber 5. 26 Halaman View KaryawanIndex*

#### 5.1.4.7 View Layout

Kode sumber 5.27 mengimplementasikan view Layout yang menyediakan template halaman yang digunakan di seluruh halaman website. Selain tampilan, kode sumber ini juga menangani plugin maupun script yang akan digunakan. Kode sumber view layout lengkap dapat dilihat pada lampiran 1

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="utf-8">
5.     <meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```

6. <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
7. <meta name="description" content="">
8. <meta name="author" content="">
9. <title><?=$title; ?></title>
10. <link href="<?=$base_url('vendor/fontawesome-
free/css/all.min.css') ?>" rel="stylesheet" type="text/css">
11. <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i
,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">
12. <link href="<?=$base_url('css/sb-admin-2.min.css') ?>"
rel="stylesheet">
13. <link href="<?=$base_url('vendor/datatables/dataTables.bootstrap4.min.css') ?>"
rel="stylesheet">
14.
15. </head>
16. <body id="page-top">
17. <?php
18. $session = session();
19. ?>
<script src="<?=$base_url('vendor/jquery/jquery.min.js') ?>"></script>
20. <script src="<?=$base_url('vendor/bootstrap/js/bootstrap.bundle.min.js') ?>"></script>
21. <!-- Core plugin JavaScript-->
22. <script src="<?=$base_url('vendor/jquery-
easing/jquery.easing.min.js') ?>"></script>
23.
24. <!-- Custom scripts for all pages-->
25. <script src="<?=$base_url('js/sb-admin-2.min.js') ?>"></script>
26.
27. <!-- Page level plugins -->
28. <script src="<?=$base_url('vendor/chart.js/Chart.min.js') ?>"></script>
29. <script src="<?=$base_url('vendor/datatables/jquery.dataTables.min.js') ?>"></script>
30. <script src="<?=$base_url('vendor/datatables/dataTables.bootstrap4.min.js') ?>"></script>
31.
32. <!-- Page level custom scripts -->
33. <script src="<?=$base_url('js/demo/chart-area-demo.js') ?>"></script>

```

```
34. <script src="<?= base_url('js/demo/chart-pie-demo.js') ?"></script>
35. <script src="<?= base_url('js/demo/datatables-demo.js') ?"></script>
36. </body>
37. </html>
```

*Kode Sumber 5. 27 Halaman View Layout*

## **5.2. Implementasi Antarmuka Pengguna**

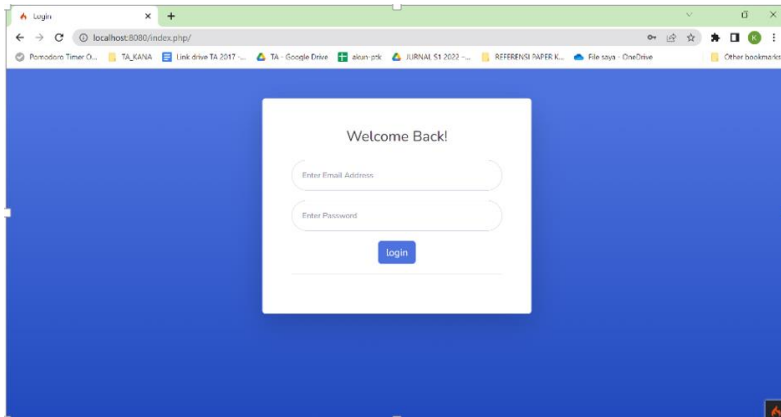
Pada bagian ini akan ditampilkan antarmuka halaman aplikasi sistem informasi inventaris barang yang terbagi menjadi sisi umum (Administrator dan Karyawan), sisi Administrator dan sisi Karyawan.

### **5.2.1. Sisi Umum**

Berikut adalah tampilan antarmuka untuk aplikasi dari sisi umum.

#### **5.3.1.1. Halaman login**

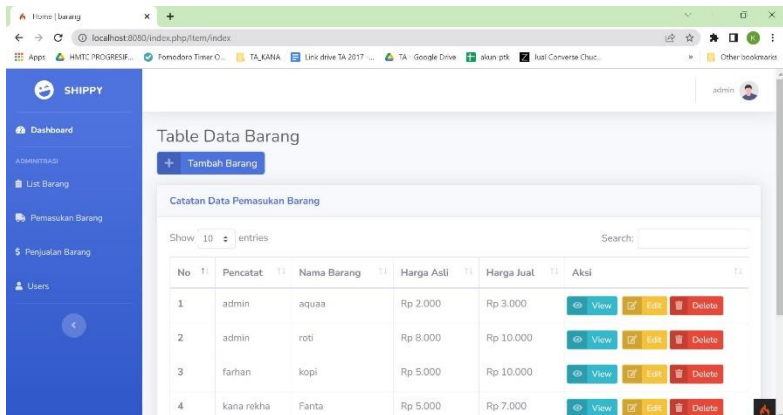
Pada halaman login, Administrator atau Karyawan dapat memasukkan email dan password untuk masuk ke halaman dashboard. Bagian dari implementasi antarmuka pengguna login ditampilkan pada gambar 5.1.



Gambar 5. 1 Halaman Login

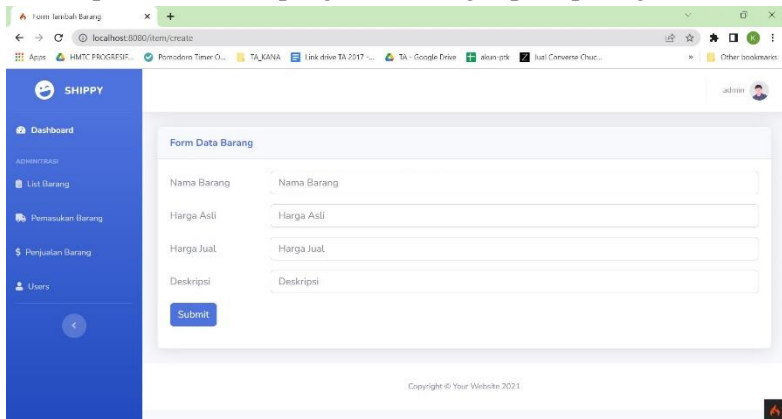
### 5.2.1.2. Halaman Barang

Pada halaman Barang, Administrator atau Karyawan dapat melihat daftar barang. Halaman ini juga menyediakan aksi untuk menambah barang, mengubah barang, melihat detail barang serta menghapus barang seperti pada gambar 5.2.



Gambar 5. 2 Halaman Data Barang

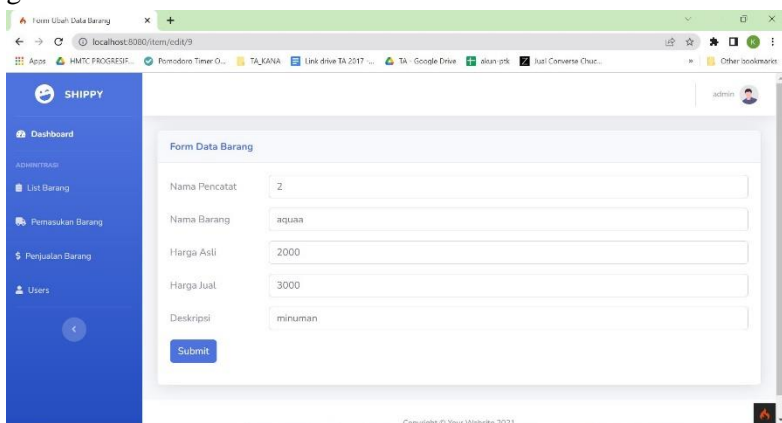
Ketika Administrator atau Karyawan ingin menambahkan barang, dapat memilih tombol Tambah Barang. Sistem akan menampilkan formulir pengisian barang seperti pada gambar 5.3.



The screenshot shows a web browser window with the URL `localhost:8080/item/create`. The page title is "Form Data Barang". On the left is a blue sidebar with the SHIPPY logo and navigation links: Dashboard, ADMINSTRASI, List Barang, Pemasukan Barang, Penjualan Barang, and Users. The main content area contains a form with the following fields: "Nama Barang" (empty), "Harga Asli" (empty), "Harga Jual" (empty), and "Deskripsi" (empty). A blue "Submit" button is located below the form. The footer of the page reads "Copyright © Your Website 2021".

Gambar 5.3 Halaman Tambah Data barang

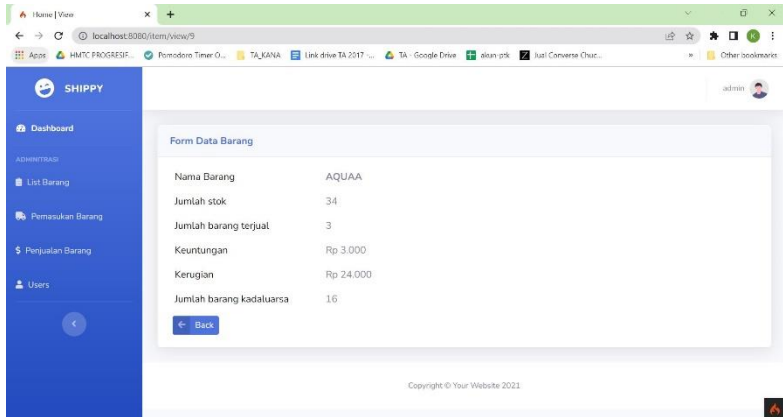
Jika barang ingin dirubah, Administrator atau Karyawan bisa memilih tombol Edit. Sistem akan menampilkan formulir barang yang sudah terisi. Lalu bisa merubah data barang yang diinginkan. Tampilan antarmuka Ubah Data Barang seperti pada gambar 5.4.



The screenshot shows a web browser window with the URL `localhost:8080/item/edit/9`. The page title is "Form Data Barang". The sidebar is identical to the previous image. The form fields are now pre-filled: "Nama Pencatat" contains "Z", "Nama Barang" contains "aqua", "Harga Asli" contains "2000", "Harga Jual" contains "3000", and "Deskripsi" contains "minuman". The blue "Submit" button remains at the bottom of the form. The footer of the page reads "Copyright © Your Website 2021".

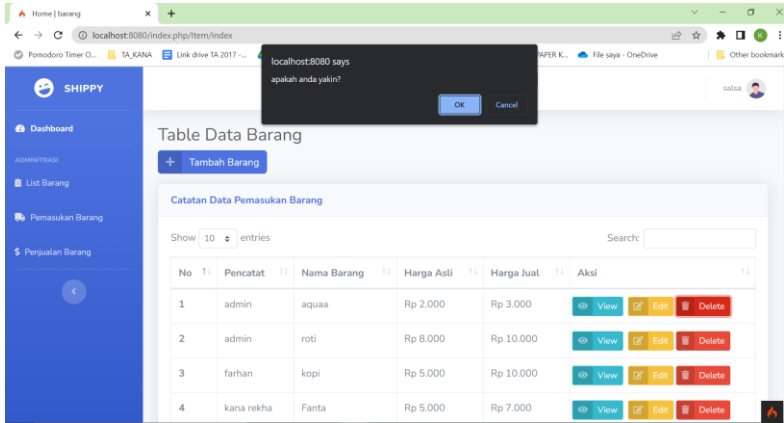
*Gambar 5. 4 Halaman Ubah Data Barang*

Jika ingin melihat detail data barang, Administrator atau Karyawan dapat memilih tombol View. Sistem akan menampilkan detail data barang seperti pada gambar 5.5.



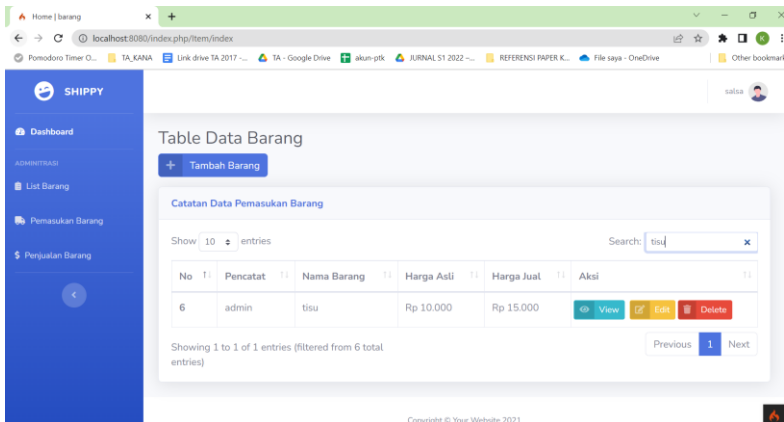
*Gambar 5. 5 Halaman Detail Data Barang*

Jika ingin menghapus data barang, Administrator atau Karyawan dapat memilih tombol hapus. Sistem akan menampilkan pesan konfirmasi berisi peringatan pesan untuk menghapus barang. Implementasi antarmuka hapus barang dapat dilihat pada gambar 5.6.



Gambar 5. 6 Halaman Hapus Data Barang

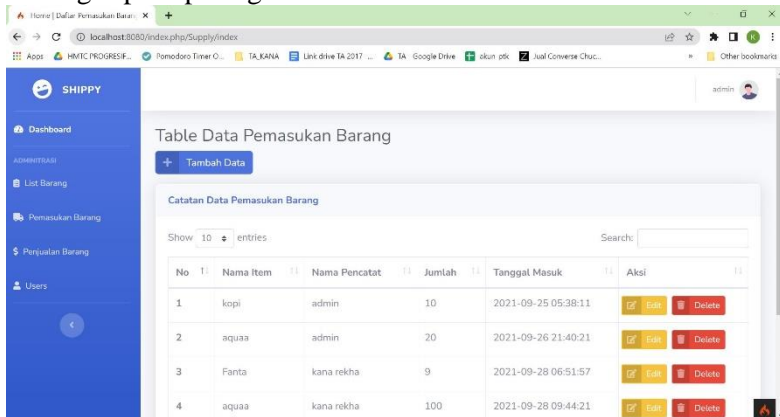
Jika ingin mencari data barang, Administrator atau Karyawan dapat menuliskan keyword barang yang ingin dicari pada kolom search. Sistem akan menampilkan hasil pencaharian data barang. Tampilan antarmuka cari barang seperti pada gambar 5.7.



Gambar 5. 7 Halaman Cari Data Barang

### 5.2.1.3. Halaman Pemasukan Barang

Pada halaman Pemasukan Barang, Administrator atau Karyawan dapat melihat daftar Pemasukan Barang. Halaman ini juga menyediakan aksi untuk menambah Pemasukan Barang, mengubah Pemasukan Barang, serta menghapus Pemasukan Barang seperti pada gambar 5.8.

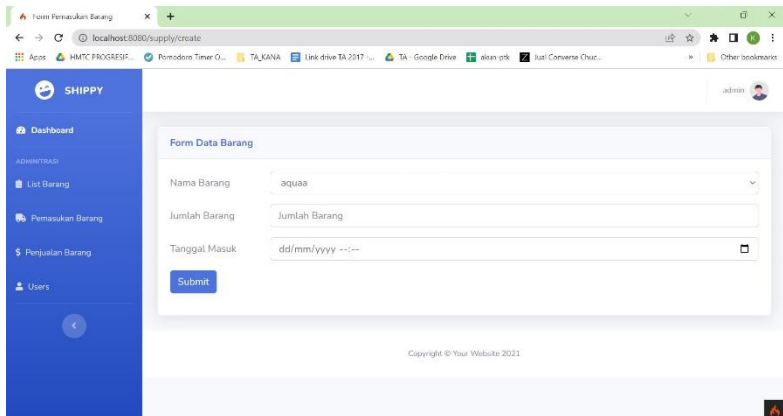


No	Nama Item	Nama Pencatat	Jumlah	Tanggal Masuk	Aksi
1	kopi	admin	10	2021-09-25 05:38:11	Edit Delete
2	aquaa	admin	20	2021-09-26 21:40:21	Edit Delete
3	Fanta	kana rekha	9	2021-09-28 06:51:57	Edit Delete
4	aquaa	kana rekha	100	2021-09-28 09:44:21	Edit Delete

Gambar 5. 8 Halaman Pemasukan Barang

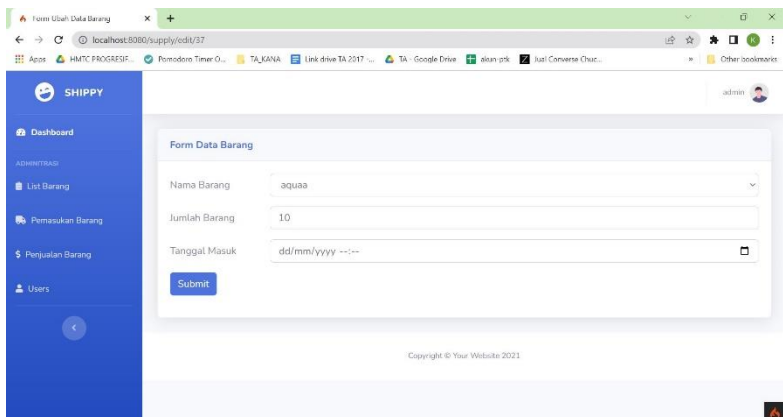
Ketika Administrator atau Karyawan ingin menambahkan Pemasukan Barang, dapat memilih tombol Tambah Pemasukan Barang. Sistem akan menampilkan formulir pengisian Pemasukan Barang seperti pada gambar 5.9.





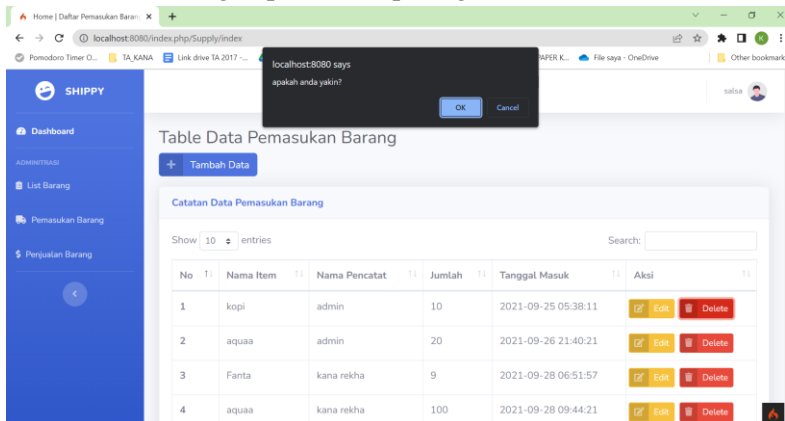
*Gambar 5. 9 Halaman Tambah Pemasukan Barang*

Jika Data Pemasukan Barang ingin dirubah, Administrator atau Karyawan bisa memilih tombol Edit. Sistem akan menampilkan formulir Pemasukan Barang yang sudah terisi. Lalu bisa merubah data Pemasukan Barang yang diinginkan. Tampilan antarmuka Ubah Data Pemasukan Barang seperti pada gambar 5.10.



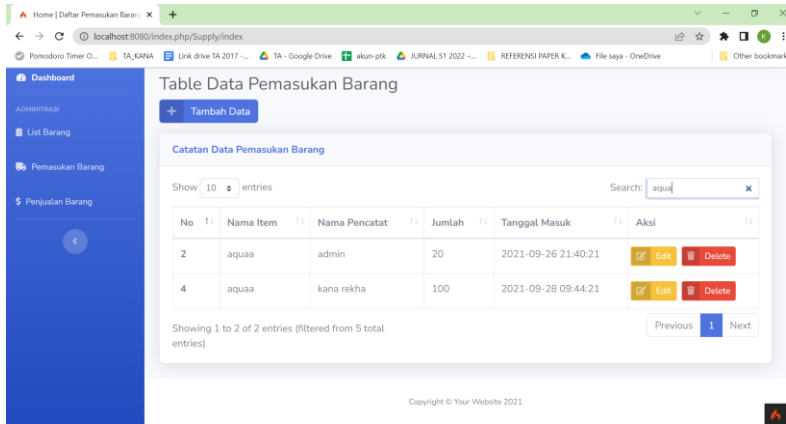
*Gambar 5. 10 Halaman Ubah Pemasukan Barang*

Jika ingin menghapus data Pemasukan barang, Administrator atau Karyawan dapat memilih tombol hapus. Sistem akan menampilkan pesan konfirmasi berisi peringatan pesan untuk menghapus Pemasukan Barang. Implementasi antarmuka hapus Pemasukan Barang dapat dilihat pada gambar 5.11.



*Gambar 5. 11 Halaman Hapus Pemasukan Barang*

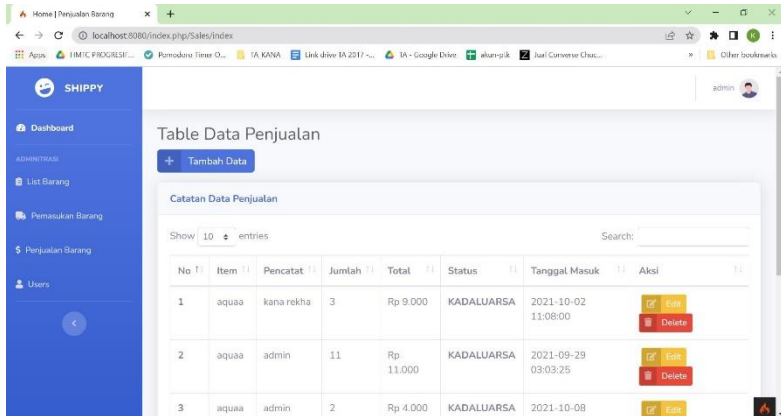
Jika ingin mencari data Pemasukan barang, Administrator atau Karyawan dapat menuliskan keyword data Pemasukan barang yang ingin dicari pada kolom search. Sistem akan menampilkan hasil pencaharian data Pemasukan Barang. Tampilan antarmuka cari Pemasukan Barang seperti pada gambar 5.12.



Gambar 5. 12 Halaman Cari Pemasukan Barang

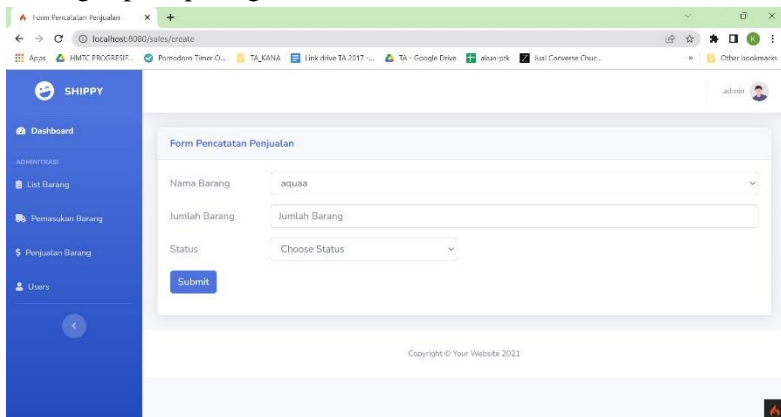
#### 5.2.1.4. Halaman Penjualan Barang

Pada halaman Penjualan Barang, Administrator atau Karyawan dapat melihat daftar Penjualan Barang. Pada halaman ini Administrator memiliki akses dalam menambah Penjualan Barang, mengubah Penjualan Barang, serta menghapus barang seperti pada gambar 5.13. Sedangkan Karyawan hanya memiliki akses pada penambahan Data Penjualan. Jika Karyawan ingin mengubah atau menghapus data barang, Karyawan bisa melakukan hal tersebut melalui Administrator.



Gambar 5. 13 Halaman Penjualan Barang

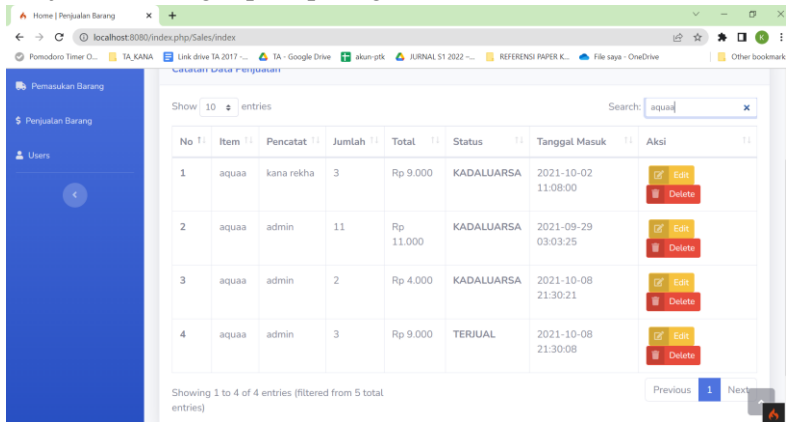
Ketika Administrator atau Karyawan ingin menambahkan Penjualan Barang, dapat memilih tombol Tambah Penjualan Barang. Sistem akan menampilkan formulir pengisian Penjualan Barang seperti pada gambar 5.14.



Gambar 5. 14 Halaman Tambah Penjualan Barang

Jika ingin mencari data Penjualan Barang, Administrator atau Karyawan dapat menuliskan keyword Penjualan barang yang ingin dicari pada kolom search. Sistem akan menampilkan hasil

pencaharian data Penjualan Barang. Tampilan antarmuka cari Penjualan Barang seperti pada gambar 5.15.



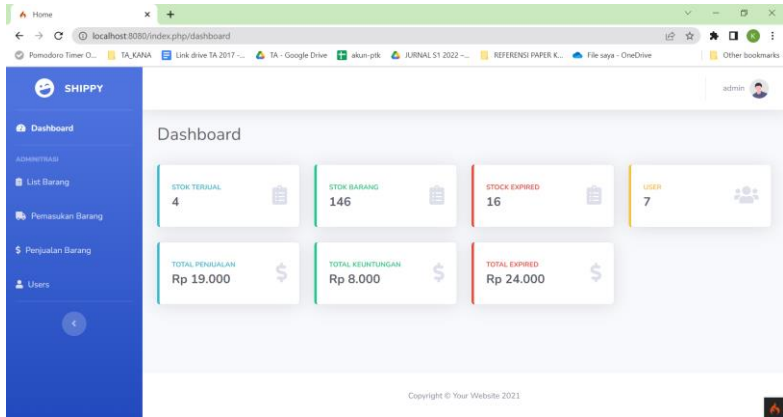
Gambar 5. 15 Halaman Cari Penjualan Barang

## 5.2.2. Sisi Administrator

Berikut adalah tampilan antarmuka untuk aplikasi dari sisi administrator.

### 5.2.2.1. Halaman Dashboard

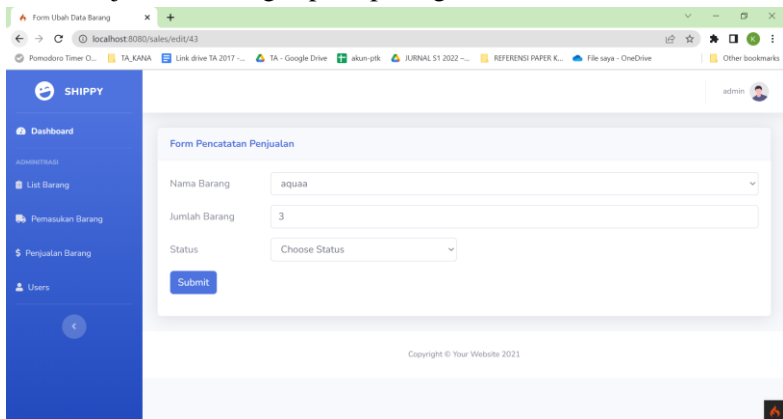
Pada halaman dashboard, sistem menampilkan jumlah stok barang, jumlah stok barang terjual, jumlah stok barang kadaluarsa, total penjualan barang, total keuntungan barang dan total Karyawan.



Gambar 5. 16 Halaman Dashboard

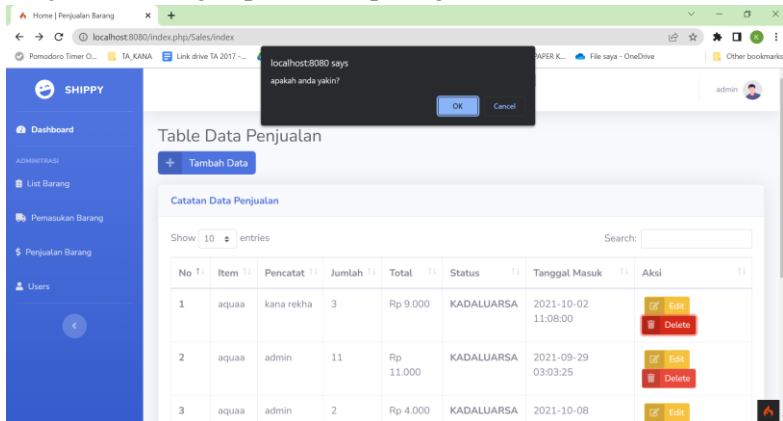
### 5.2.2.2. Halaman Penjualan Barang

Jika Data Penjualan Barang ingin dirubah, Administrator bisa memilih tombol Edit. Sistem akan menampilkan formulir Penjualan barang yang sudah terisi. Lalu bisa merubah data Penjualan Barang yang diinginkan. Tampilan antarmuka Ubah Data Penjualan Barang seperti pada gambar 5.17.



Gambar 5. 17 Halaman Edit Penjualan Barang

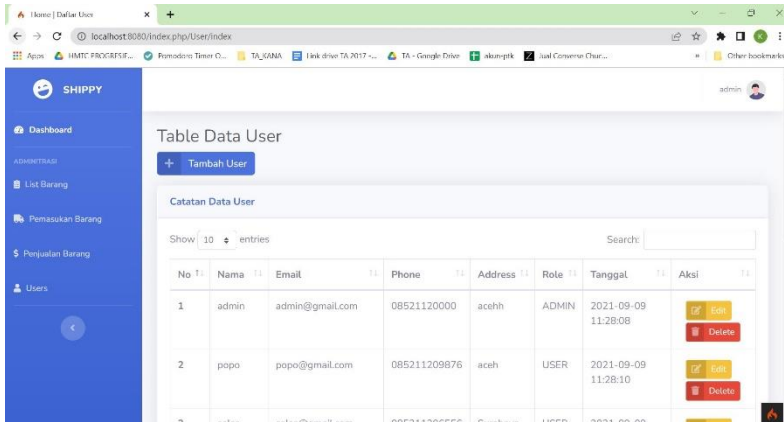
Jika ingin menghapus data Penjualan Barang, Administrator dapat memilih tombol Delete. Sistem akan menampilkan pesan konfirmasi berisi peringatan pesan untuk menghapus Penjualan barang. Implementasi antarmuka hapus Penjualan barang dapat dilihat pada gambar 5.6.



Gambar 5. 18 Halaman Hapus Penjualan Barang

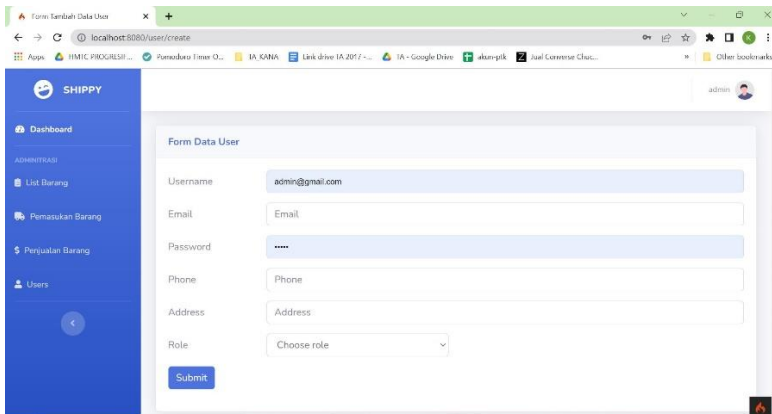
### 5.2.2.3. Halaman Daftar Karyawan

Pada halaman Karyawan, Administrator dapat melihat daftar Karyawan. Halaman ini juga menyediakan aksi untuk menambah Karyawan, mengubah Karyawan, serta menghapus Karyawan seperti pada gambar 5.19.



*Gambar 5. 19 Halaman Data Karyawan*

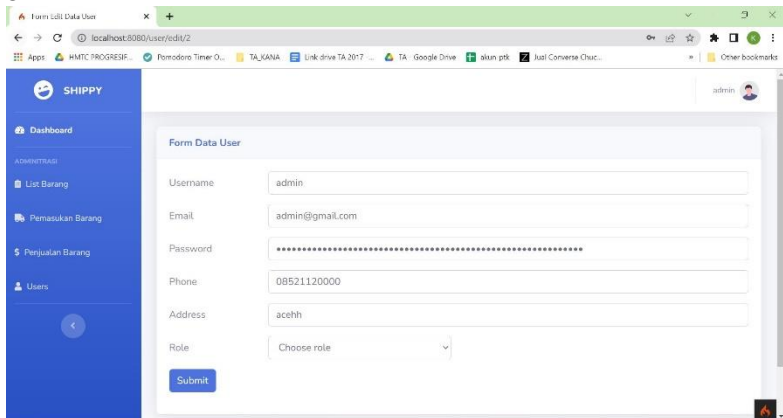
Ketika Administrator ingin menambahkan Karyawan, dapat memilih tombol Tambah Karyawan. Sistem akan menampilkan formulir pengisian Karyawan seperti pada gambar 5.20.



*Gambar 5. 20 Halaman Tambah Karyawan*

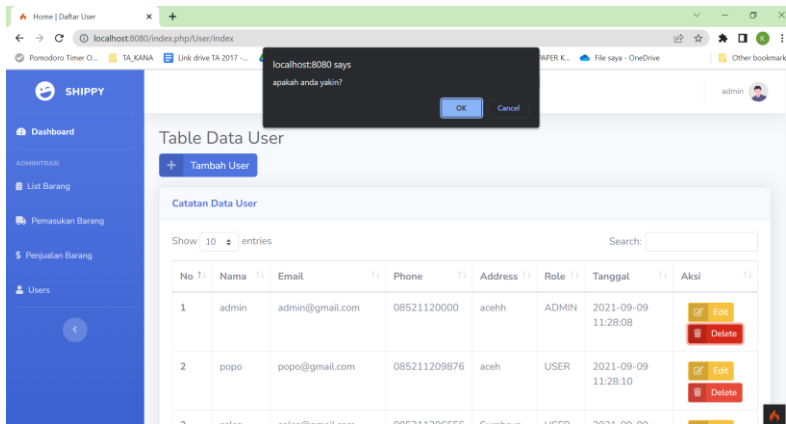


Jika Data Karyawan ingin dirubah, Administrator bisa memilih tombol Edit. Sistem akan menampilkan formulir Karyawan yang sudah terisi. Lalu bisa merubah data Karyawan yang diinginkan. Tampilan antarmuka Ubah Data Karyawan seperti pada gambar 5.21.



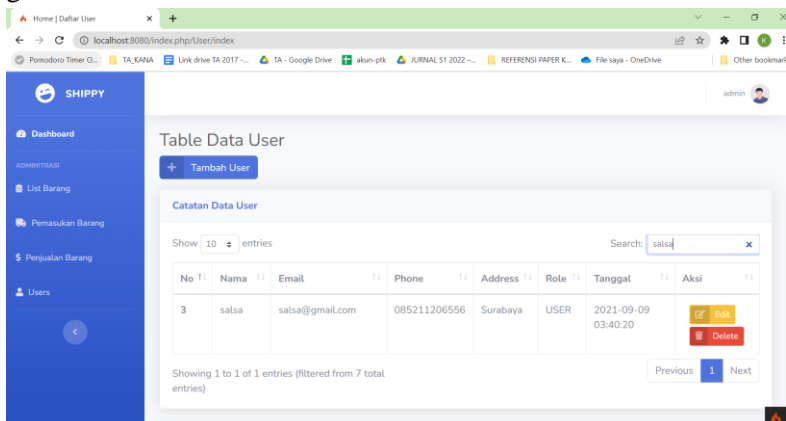
*Gambar 5. 21 Halaman Edit Karyawan*

Jika ingin menghapus data Karyawan, Administrator dapat memilih tombol Delete. Sistem akan menampilkan pesan konfirmasi berisi peringatan pesan untuk menghapus Karyawan. Implementasi antarmuka hapus Karyawan dapat dilihat pada gambar 5.22.



Gambar 5. 22 Halaman Hapus Karyawan

Jika ingin mencari data Karyawan, Administrator dapat menuliskan keyword/User name yang ingin dicari pada kolom search. Sistem akan menampilkan hasil pencaharian data Karyawan. Tampilan antarmuka cari Karyawan seperti pada gambar 5.23.



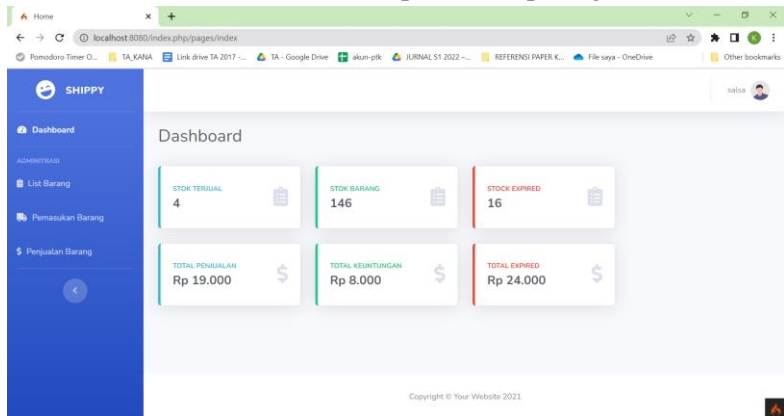
Gambar 5. 23 Halaman Cari Karyawan

### 5.2.3 Sisi Karyawan

Berikut adalah tampilan antarmuka untuk aplikasi dari sisi karyawan.

#### 5.2.3.1. Halaman Dashboard

Pada halaman dashboard, sistem menampilkan jumlah stok barang, jumlah stok barang terjual, jumlah stok barang kadaluarsa, total penjualan barang, dan total keuntungan barang. Implementasi antarmuka halaman dashboard dapat dilihat pada gambar 5.24.



Gambar 5. 24 Halaman Dashboard

#### 5.2.3.1. Halaman Penjualan Barang

Pada halaman Penjualan Barang, Karyawan dapat melihat daftar barang dan menambah data Penjualan Barang. Implementasi antarmuka halaman penjualan barang untuk Karyawan dapat dilihat pada gambar 5.25.

The screenshot shows a web application interface for 'SHIPPY'. The main content area is titled 'Table Data Penjualan' and contains a table with the following data:

No	Item	Pencatat	Jumlah	Total	Status	Tanggal Masuk
1	aquaa	kana rekha	3	Rp 9.000	KADALUARSA	2021-10-02 11:08:00
2	aquaa	admin	11	Rp 11.000	KADALUARSA	2021-09-29 03:03:25
3	aquaa	admin	2	Rp 4.000	KADALUARSA	2021-10-08 21:30:21
4	aquaa	admin	3	Rp 9.000	TERJUAL	2021-10-08 21:30:08

*Gambar 5. 25 Halaman Penjualan Barang*

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap beberapa bagian menu aplikasi website Sistem Informasi Inventaris Barang (SIIB). Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dibangun dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap Aplikasi Sistem Informasi Inventaris Barang (SIIB) untuk mengetahui hasil pengujian terhadap kesesuaian dan ketepatan fungsionalitas dari sistem aplikasi website tersebut.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut ini:

- a. Kemampuan aplikasi memenuhi kebutuhan fungsionalitas lainnya, yaitu memasukkan data, mengubah data, dan menghapus data.
- b. Kemampuan aplikasi untuk melakukan pencarian berdasarkan keyword yang dimasukkan oleh pengguna.
- c. Kesesuaian dalam memenuhi kebutuhan non-fungsional aplikasi web, yaitu memiliki antarmuka yang responsif dan mudah dipahami oleh penggunanya.

#### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan melakukan peran sebagai Administrator dan Karyawan yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Administrator atau Karyawan membuka aplikasi Sistem Informasi Inventaris Barang (SIIB) kemudian login
2. Administrator atau Karyawan melihat dashboard pada aplikasi Sistem Informasi Inventaris Barang (SIIB)
3. Administrator atau Karyawan membuka halaman barang dan menekan tombol “tambah”
4. Administrator atau Karyawan mengisi formulir penambahan barang kemudian menekan tombol “simpan”
5. Administrator atau Karyawan menekan tombol “edit” pada halaman barang
6. Administrator atau Karyawan mengisi formulir perubahan barang kemudian menekan tombol “simpan”
7. Administrator atau Karyawan menekan tombol “delete” kemudian pada halaman barang memilih “ya” pada pesan konfirmasi
8. Administrator atau Karyawan memasukkan keyword dalam mencari barang yang diinginkan
9. Administrator atau Karyawan membuka halaman pemasukan barang dan menekan tombol “tambah”
10. Administrator atau Karyawan mengisi formulir penambahan pemasukan barang kemudian menekan tombol “simpan”
11. Administrator atau Karyawan menekan tombol “edit” pada halaman pemasukan barang
12. Administrator atau Karyawan mengisi formulir perubahan pemasukan barang kemudian menekan tombol “simpan”
13. Administrator atau Karyawan menekan tombol “delete” pada pemasukan barang kemudian memilih “ya” pada pesan konfirmasi

14. Administrator atau Karyawan memasukkan keyword dalam mencari pemasukan barang yang diinginkan
15. Administrator atau Karyawan membuka halaman penjualan barang dan menekan tombol “tambah”
16. Administrator atau Karyawan mengisi formulir penjualan barang kemudian menekan tombol “simpan”
17. Administrator menekan tombol “edit” pada halaman penjualan barang
18. Administrator mengisi formulir perubahan penjualan barang kemudian menekan tombol “simpan”
19. Administrator menekan tombol “delete” pada penjualan barang kemudian memilih “ya” pada pesan konfirmasi
20. Administrator atau Karyawan memasukkan keyword dalam mencari penjualan barang yang diinginkan
21. Administrator membuka halaman Karyawan dan menekan tombol “tambah”
22. Administrator mengisi formulir Karyawan barang kemudian menekan tombol “simpan”
23. Administrator menekan tombol “edit” pada halaman Karyawan
24. Administrator mengisi formulir perubahan Karyawan kemudian menekan tombol “simpan”
25. Administrator menekan tombol “delete” pada halaman Karyawan kemudian memilih “ya” pada pesan konfirmasi
26. Administrator memasukkan keyword dalam mencari Karyawan yang diinginkan

Adapun skenario pengujian yang akan di uji coba pada sistem informasi inventaris barang dapat dilihat pada tabel 38

Tabel 38. Daftar Skenario Pengujian

<b>Kode Pengujian</b>	<b>Skenario Pengujian</b>
P001	Melihat dashboard
P002	Menambah, mengubah, mencari dan menghapus data barang sebagai Administrator dan Karyawan
P003	Menambah, mengubah, mencari dan menghapus data pemasukan barang sebagai Administrator dan Karyawan
P004	Menambah dan mencari data penjualan data barang sebagai Administrator dan Karyawan
P005	Menambah, mengubah, mencari dan menghapus data penjualan barang sebagai Administrator
P006	Menambah, mengubah, mencari dan menghapus data Karyawan sebagai Administrator

#### **6.4. Evaluasi Pengujian**

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi sistem informasi inventaris barang terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan. Pengujian fungsionalitas website sudah diuji coba pada fase *development*, dan belum uji coba pada server production karena adanya faktor teknis dari perusahaan. Tabel 39 daftar pengujian yang diuji coba berdasarkan kebutuhan fungsionalitas sistem informasi inventaris barang. Tabel 40 menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.



*Tabel 39. Daftar Pengujian berdasarkan kebutuhan fungsional*

<b>Kebutuhan Fungsionalitas</b>	<b>Desain</b>	<b>Implementasi</b>	<b>Pengujian</b>
Melihat Dashboard	Tabel Karyawan, Tabel Barang, Tabel Penjualan	routes.php, dashboard.php, Item.php	P001
Menambah Karyawan	Tabel Karyawan	User.php, UserModel.php	P006
Mengubah Karyawan	Tabel Karyawan	User.php, UserModel.php	P006
Menghapus Karyawan	Tabel Karyawan	User.php, UserModel.php	P006
Mencari Karyawan	Tabel Karyawan	User.php, UserModel.php	P006
Menambah Barang	Tabel Barang	Item.php, ItemModel.php	P002
Mengubah Barang	Tabel Barang	Item.php, ItemModel.php	P002
Menghapus Barang	Tabel Barang	Item.php, ItemModel.php	P002
Melihat Detail Barang	Tabel Barang	Item.php, ItemModel.php	P002
Mencari Barang	Tabel Barang	Item.php, ItemModel.php	P002
Menambah Pemasukan Barang	Tabel Pemasukan Barang	Supply.php, SupplyModel.php, ItemModel.php, StockModel.php	P003
Mengubah Pemasukan Barang	Tabel Pemasukan Barang	Supply.php, SupplyModel.php, ItemModel.php, StockModel.php	P003
Menghapus Pemasukan Barang	Tabel Pemasukan Barang	Supply.php, SupplyModel.php, ItemModel.php, StockModel.php	P003
Mencari Pemasukan Barang	Tabel Pemasukan Barang	Supply.php, SupplyModel.php, ItemModel.php	P003
Menambah Penjualan Barang	Tabel Penjualan Barang	Sales.php, SalesModel.php, ItemModel.php,	P004 dan P005

		StockModel.php	
Mengubah Penjualan Barang	Tabel Penjualan Barang	Sales.php, SalesModel.php, StockModel.php	P005
Menghapus Penjualan Barang	Tabel Penjualan Barang	Sales.php, SalesModel.php, StockModel.php	P005
Mencari Penjualan Barang	Tabel Penjualan Barang	Sales.php, SalesModel.php	P004 dan P005

*Tabel 40. Hasil Evaluasi Pengujian*

<b>Kode Pengujian</b>	<b>Kriteria Pengujian</b>	<b>Hasil Pengujian</b>
P001	Melihat dashboard	Terpenuhi
P002	Menambah, mengubah, dan menghapus data barang sebagai Administrator dan Karyawan	Terpenuhi
P003	Menambah, mengubah, dan menghapus data pemasukan barang sebagai Administrator dan Karyawan	Terpenuhi
P004	Menambah data penjualan data barang sebagai Administrator dan Karyawan	Terpenuhi
P005	Mengubah, dan menghapus data penjualan barang sebagai Administrator	Terpenuhi
P006	Menambah, mengubah, dan menghapus data Karyawan sebagai Administrator	Terpenuhi

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah merancang aplikasi Sistem Informasi Inventaris Barang adalah sebagai berikut :

- a. Dengan adanya aplikasi ini dapat memudahkan karyawan kantor dalam mengefisienkan waktu dan dapat mengelola barang secara mudah setiap ada penambahan atau perubahan data.
- b. Penyimpanan data menggunakan database sehingga data lebih terjamin dan dapat diakses dengan mudah.
- c. *Framework* yang digunakan dalam merancang aplikasi ini bersifat *open source*, sehingga mudah apabila ingin memperbarui aplikasi tersebut.

#### **7.2. Saran**

Dalam pengerjaannya, sistem yang dibuat masih memiliki kekurangan. Saran untuk pengembangan sistem aplikasi Sistem Informasi Inventaris Barang adalah sebagai berikut :

- a. Adanya fitur mencari barang berdasarkan tanggal sehingga memudahkan Administrator dalam melihat barang berdasarkan waktu perbulan ataupun pertahun.
- b. Tata letak desain sistem perlu dibuat secara dinamis sehingga dapat membuat tampilan lebih menarik dan memudahkan dalam *maintainace* website.
- c. Diperlukan pengujian pada lingkungan permasalahan perusahaan. Terdapat hambatan dari perusahaan saat menindaklanjuti dalam taha *deployment* website.



## DAFTAR PUSTAKA

- [1] B. Johnson, *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. John Wiley & Sons, 2019.
- [2] T. Myer, *Professional CodeIgniter*. John Wiley & Sons, 2008.
- [3] K. Hayhow, *Basic HTML Teacher Resources*. Payne Gallway, 2005.
- [4] “PHP: Hypertext Preprocessor.” <https://www.php.net/> (accessed Jun. 10, 2022).
- [5] “JavaScript | MDN.” <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (accessed Jun. 10, 2022).
- [6] C. Mehta, A. K. Bhavsar, H. Oza, and S. Shah, *MySQL 8 Administrator's Guide: Effective guide to Administratoristering high-performance MySQL 8 solutions*. Packt Publishing Ltd, 2018.

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN

### LAMPIRAN 1. Kode sumber view Layout

Kode untuk *template layout semua halaman* website

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.
6.     <meta charset="utf-8">
7.     <meta http-equiv="X-UA-Compatible"
8.     content="IE=edge">
9.     <meta name="viewport" content="width=device-
10.     width, initial-scale=1, shrink-to-fit=no">
11.     <meta name="description" content="">
12.     <meta name="author" content="">
13.
14.     <title><?= $title; ?></title>
15.
16.     <!-- Custom fonts for this template-->
17.     <link href="<?= base_url('vendor/fontawesome-
18.     free/css/all.min.css') ?>" rel="stylesheet"
19.     type="text/css">
20.     <link
21.     href="https://fonts.googleapis.com/css?family=Nuni
22.     to:200,200i,300,300i,400,400i,600,600i,700,700i,80
23.     0,800i,900,900i" rel="stylesheet">
24.
25.     <!-- Custom styles for this template-->
26.     <link href="<?= base_url('css/sb-admin-
27.     2.min.css') ?>" rel="stylesheet">
28.
29.     <!-- Custom styles for pemasukan page -->
30.     <link href="<?=
31.     base_url('vendor/datatables/dataTables.bootstrap4.
32.     min.css') ?>" rel="stylesheet">
33. </head>
34. <body id="page-top">
35.     <?php
36.     $session = session();
37.     ?>
38.     <!-- Page Wrapper -->
39.     <div id="wrapper">
40.
41.         <!-- Sidebar -->
```

```

34.         <ul class="navbar-nav bg-gradient-primary
           sidebar sidebar-dark accordion"
           id="accordionSidebar">
35.
36.             <!-- Sidebar - Brand -->
37.             <a class="sidebar-brand d-flex align-
           items-center justify-content-center" href="<?=
           site_url('home/index') ?>">
38.                 <div class="sidebar-brand-icon
           rotate-n-15">
39.                     <i class="fas fa-laugh-
           wink"></i>
40.                 </div>
41.                 <div class="sidebar-brand-text mx-
           3">Shippy </div>
42.             </a>
43.
44.             <!-- Divider -->
45.             <hr class="sidebar-divider my-0">
46.
47.             <!-- Nav Item - Dashboard -->
48.             <li class="nav-item active">
49.                 <a class="nav-link" href="<?=
           site_url('pages/index') ?>">
50.                     <i class="fas fa-fw fa-
           tachometer-alt"></i>
51.                     <span>Dashboard</span></a>
52.                 </li>
53.
54.             <!-- Divider -->
55.             <hr class="sidebar-divider">
56.
57.             <!-- Heading -->
58.             <div class="sidebar-heading">
59.                 Adminitrasi
60.             </div>
61.
62.             <!-- Nav Item - Pages Collapse Menu --
           >
63.             <li class="nav-item">
64.                 <a class="nav-link collapsed"
           href="<?= site_url('Item/index') ?>">
65.                     <i class="fas fa-clipboard-
           list fa-2x text-gray-300"></i>
66.                     <span>List Barang</span>
67.                 </a>
68.             </li>
69.
70.             <li class="nav-item">

```



```

71.         <a class="nav-link collapsed"
href="<? = site_url('Supply/index') ?>">
72.         <i class="fas fa-truck fa-2x
text-gray-300"></i>
73.         <span>Pemasukan Barang</span>
74.         </a>
75.     </li>
76.
77.     <!-- Nav Item - Utilities Collapse
Menu -->
78.     <li class="nav-item">
79.         <a class="nav-link collapsed"
href="<? = site_url('Sales/index') ?>">
80.         <i class="fas fa-dollar-sign
fa-2x text-gray-300"></i>
81.         <span>Penjualan Barang</span>
82.         </a>
83.     </li>
84.
85.     <?php if (session()->get('id_role') ==
1) : ?>
86.     <!-- Nav Item - Pages Collapse
Menu -->
87.     <li class="nav-item">
88.         <a class="nav-link collapsed"
href="<? = site_url('User/index') ?>">
89.         <i class="fas fa-user fa-
2x text-gray-300"></i>
90.         <span>Users</span>
91.         </a>
92.     </li>
93. <?php endif ?>
94.
95.     <!-- Divider -->
96.     <hr class="sidebar-divider d-none d-
md-block">
97.
98.     <!-- Sidebar Toggler (Sidebar) -->
99.     <div class="text-center d-none d-md-
inline">
100.         <button class="rounded-
circle border-0" id="sidebarToggle"></button>
101.     </div>
102. </ul>
103. <!-- End of Sidebar -->
104.
105. <!-- Content Wrapper -->
106. <div id="content-wrapper" class="d-
flex flex-column">

```

```

107.
108.         <!-- Main Content -->
109.         <?= $this-
>renderSection('content') ?>
110.         <!-- End of Main Content -->
111.
112.         <!-- Footer -->
113.         <footer class="sticky-footer
bg-white">
114.             <div class="container my-
auto">
115.                 <div class="copyright
text-center my-auto">
116.                     <span>Copyright
&copy; Your Website 2021</span>
117.                     </div>
118.                 </div>
119.             </footer>
120.         <!-- End of Footer -->
121.
122.     </div>
123.     <!-- End of Content Wrapper -->
124.
125. </div>
126. <!-- End of Page Wrapper -->
127.
128. <!-- Scroll to Top Button-->
129. <a class="scroll-to-top rounded"
href="#page-top">
130.     <i class="fas fa-angle-up"></i>
131. </a>
132.
133. <!-- Logout Modal-->
134. <div class="modal fade"
id="logoutModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
135.     <div class="modal-dialog"
role="document">
136.         <div class="modal-content">
137.             <div class="modal-header">
138.                 <h5 class="modal-title"
id="exampleModalLabel">Ready to Leave?</h5>
139.                 <button class="close"
type="button" data-dismiss="modal" aria-
label="Close">
140.                     <span aria-
hidden="true">x</span>
141.                 </button>
142.             </div>

```

```

143.             <div class="modal-
body">Select "Logout" below if you are ready to
end your current session.</div>
144.             <div class="modal-footer">
145.                 <button class="btn btn-
secondary" type="button" data-
dismiss="modal">Cancel</button>
146.                 <a class="btn btn-
primary" href="<?=  

base_url('/auth/logout'); ?>">Logout</a>
147.             </div>
148.         </div>
149.     </div>
150. </div>
151.
152.     <!-- Bootstrap core JavaScript-->
153.     <script src="<?=  

base_url('vendor/jquery/jquery.min.js') ?>"></scri  

pt>
154.         <script src="<?=  

base_url('vendor/bootstrap/js/bootstrap.bundle.min  

.js') ?>"></script>
155.
156.         <!-- Core plugin JavaScript-->
157.         <script src="<?=  

base_url('vendor/jquery-  

easing/jquery.easing.min.js') ?>"></script>
158.
159.         <!-- Custom scripts for all pages-->
160.         <script src="<?=  

base_url('js/sb-admin-  

2.min.js') ?>"></script>
161.
162.         <!-- Page level plugins -->
163.         <script src="<?=  

base_url('vendor/chart.js/Chart.min.js') ?>"></scr  

ipt>
164.         <script src="<?=  

base_url('vendor/datatables/jquery.dataTables.min.  

js') ?>"></script>
165.         <script src="<?=  

base_url('vendor/datatables/dataTables.bootstrap4.  

min.js') ?>"></script>
166.
167.         <!-- Page level custom scripts -->
168.         <script src="<?=  

base_url('js/demo/chart-area-  

demo.js') ?>"></script>

```

```
169.         <script src="<?=
base_url('js/demo/chart-pie-
demo.js') ?>"></script>
170.         <script src="<?=
base_url('js/demo/datatables-
demo.js') ?>"></script>
171.
172.         </body>
173.
174.         </html>
```

## **BIODATA PENULIS I**

Nama : Kana Rekha  
Tempat, Tanggal Lahir : Bambi, 23 September 2000  
Jenis Kelamin : Perempuan  
Telepon : +6285211205121  
Email : kanarekha8@gmail.com

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika –  
FTEIC , ITS  
Angkatan : 2018  
Semester : 8 (Delapan)

