

**MANAJEMEN RANTAI PASOK TERINTEGRASI DENGAN KONSEP *VENDOR
MANAGED INVENTORY* MENGGUNAKAN ALGORITMA *FUZZY TSUKAMOTO*
UNTUK MENINGKATKAN EFEKTIFITAS PENJUALAN
(Studi Kasus: *Retailer* Tradisional Di Kecamatan Selo)**

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana
Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta



DISUSUN OLEH :
Windyasmanti Nur Faiz
123150060

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2022**

HALAMAN PENGESAHAN PEMBIMBING

**MANAJEMEN RANTAI PASOK TERINTEGRASI DENGAN KONSEP *VENDOR
MANAGED INVENTORY* MENGGUNAKAN ALGORITMA *FUZZY TSUKAMOTO*
UNTUK MENINGKATKAN EFEKTIFITAS PENJUALAN
(Setudi Kasus: *Retailer* Tradisional Di Kecamatan Selo)**

Disusun oleh:

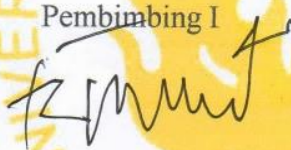
Windyasmanti Nur Faiz

123150060

Telah diuji dan dinyatakan lulus oleh Pembimbing
pada tanggal:

Menyetujui,

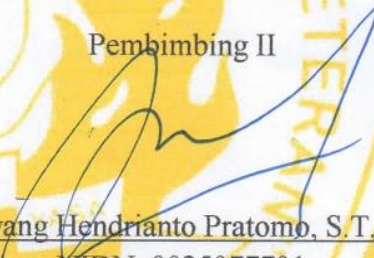
Pembimbing I



Frans Richard Kodong, S.T, M.Kom.

NIDN. 0523026201

Pembimbing II



Dr. Awang Hendrianto Pratomo, S.T, M.T.

NIDN. 0025077701

Mengetahui,
Koordinator Program Studi

Dr. Heriyanto, A. Md, S. Kom., M. Cs.

NIDN. 0508067703

HALAMAN PENGESAHAN PENGUJI

**MANAJEMEN RANTAI PASOK TERINTEGRASI DENGAN KONSEP *VENDOR
MANAGED INVENTORY* MENGGUNAKAN ALGORITMA *FUZZY TSUKAMOTO*
UNTUK MENINGKATKAN EFEKTIFITAS PENJUALAN
(Setudi Kasus: *Retailer Tradisional Di Kecamatan Selo*)**

Disusun oleh:
Windyasmanti Nur Faiz
123150060

Telah diuji dan dinyatakan lulus oleh Penguji
pada tanggal:

Menyetujui,

Penguji I

Penguji II

Frans Richard Kodong, S.T, M.Kom.
NIDN. 0523026201

Dr. Awang Hendrianto Pratomo, S.T, M.T.
NIDN. 0025077701

Penguji III

Penguji IV

Dr. Heriyanto, A. Md., S. Kom., M. Cs.
NIDN. 0508067703

Mangaras Yanu F., S.T, M. Eng.
NIDN. 0521018201

SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan dibawah ini, saya :

Nama : Windyasmanti Nur Faiz
NIM : 123150060

Menyatakan bahwa karya ilmiah saya yang berjudul : **MANAJEMEN RANTAI PASOK TERINTEGRASI DENGAN KONSEP *VENDOR MANAGED INVENTORY* MENGGUNAKAN ALGORITMA *FUZZY TSUKAMOTO* UNTUK MENINGKATKAN EFEKTIFITAS PENJUALAN (Setudi Kasus: *Retailer Tradisional Di Kecamatan Selo*)** Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila di kemudian hari karya saya didinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apapun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : Oktober 2022

Yang Menyatakan,



Windyasmanti Nur Faiz
NIM. 123150060

SURAT PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Windyasmanti Nur Faiz
Nim : 123150060
Fakultas/Prodi : Teknik Industri / Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir **MANAJEMEN RANTAI PASOK TERINTEGRASI DENGAN KONSEP *VENDOR MANAGED INVENTORY* MENGGUNAKAN ALGORITMA *FUZZY TSUKAMOTO* UNTUK MENINGKATKAN EFEKTIFITAS PENJUALAN (Setudi Kasus: *Retailer Tradisional Di Kecamatan Selo*)** Adalah hasil kerja saya sendiri dan benar bebas dari plagiasi kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No. 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : Oktober 2022

Yang Menyatakan,



Windyasmanti Nur Faiz
NIM. 123150060

ABSTRAK

Peningkatan infrastruktur dan teknologi manufaktur mengakibatkan meningkatnya pertumbuhan industri *retail* modern dengan konsep waralaba atau *franchise* seperti *Indomaret* dan *Alfamart*. Hal ini memicu persaingan yang cukup ketat bagi *retail* tradisional dan berakibat menurunnya omset pendapatan. Dikuasainya bisnis *retail* modern oleh suatu perusahaan tertentu mengakibatkan persaingan dan monopoli di suatu daerah sehingga menimbulkan dampak negatif bagi pengusaha lokal yang berada di daerah tersebut terutama bagi *retailer* tradisional seperti toko kelontong dan waserda. Pada penelitian terkait Peranan Pengendalian Internal Persediaan mencatat adanya penumpukan barang atau *over stock* yang mengakibatkan kerusakan dan kadaluarsa pada barang sehingga tidak diminati lagi oleh konsumen. Penyebabnya adalah pengadaan barang yang berlebihan dan tidak sesuai, sementara kebutuhan konsumen tidak dapat diprediksi. Sementara itu ketika barang terlalu sedikit atau *out of stock* menyebabkan terhambatnya proses penjualan. Serta pembelian barang yang sedikit juga meningkatkan frekuensi pembelian ke pemasok yang tidak efektif dan meningkatkan biaya pengiriman. Sementara bagi distributor permintaan pasar yang lebih cepat mengalami perubahan menuntut untuk lebih detail dalam perencanaan strategi untuk meningkatkan penjualan setiap harinya.

Penelitian ini dilakukan dengan menerapkan konsep manajemen rantai pasok dengan model VMI (*Vendor Managed Inventory*) satu pemasok dan banyak *retailer* yang terintegrasi antara gudang utama yang berperan sebagai distributor dengan pedagang-pedagang *retail* tradisional untuk meningkatkan efektifitas manajemen pasok. Serta melakukan analisis penjualan dengan menggunakan algoritma *fuzzy* untuk penentuan rekomendasi jumlah distribusi yang sesuai bagi distributor dan algoritma *k-means* untuk mengetahui tingkat penjualan barang yang akan dibagi menjadi tiga *cluster* pada tiap-tiap *Retail*.

Hasil dari penelitian ini diketahui bahwa sistem yang dibangun mampu mengintegrasikan data barang dan penjualan milik masing-masing *retailer* kepada pihak distributor yang berperan sebagai gudang utama. Pada pengujiannya yang dilakukan dengan metode *black box tasting* menghasilkan nilai 100% berhasil yang berarti sistem dapat bekerja dengan baik. Algoritma *fuzzy tsukamoto* diketahui mampu bekerja dengan layak untuk melakukan rekomendasi jumlah restok oleh distributor kepada pihak *retailer* pada data penjualan *retailer* Arasyid dalam data kurun waktu lima minggu atau lebih, namun bekerja dengan buruk pada data transaksi kurang dari lima minggu. Hal ini dibuktikan dengan hasil pengujian MAPE yang telah dilakukan pada data transaksi selama lima minggu dan enam minggu yang masing-masing mendapatkan hasil perhitungan nilai rata-rata kesalahan sebesar 43,988% dan 26,411% yang masuk dalam kategori layak serta mendapatkan nilai rata-rata kesalahan lebih dari 50% pada pengujian data transaksi dengan kurun waktu kurang dari lima minggu yang masuk dalam kategori buruk berdasarkan kriteria MAPE.

Kata Kunci: Manajemen Rantai Pasok, *Vendor Managed Inventory*, Algoritma *fuzzy Tsukamoto*, Algoritma *K-Means*, Pedagang Eceran, Distributor

KATA PENGANTAR

Assalamu 'alaikum warahmatullahi wabarakatuh.

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan inayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Manajemen Rantai Pasok Terintegrasi Dengan Konsep *Vendor Managed Inventory* Menggunakan Algoritma *Fuzzy Tsukamoto* Untuk Meningkatkan Efektifitas Penjualan (Setudi Kasus: *Retailer Tradisional Di Kecamatan Selo*)” dapat diselesaikan.

Tugas akhir ini merupakan salah satu syarat kelulusan untuk menyelesaikan pendidikan pada jenjang stratasatu (S1) dan memperoleh gelar sarjana di Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, dan dalam proses penyusunan tugas akhir ini, penulis mendapatkan banyak sekali bantuan, bimbingan serta dukungan dari berbagai pihak, sehingga dalam kesempatan ini penulis juga bermaksud menyampaikan rasa terima kasih kepada:

1. Allah SWT yang senantiasa memberikan petunjuk, kemudahan selama penyelesaian tugas akhir ini.
2. Keluarga terkasih, kedua Orang Tua serta adik saya yang selalu sabar memberikan doa dan dukungan bail mental bagi penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Frans Richard Kodong, S.T., M.T., selaku Dosen Pembimbing I atas kesediaan, bimbingan, saran, dukungan dan ilmu yang telah diberikan kepada penulis untuk selama melaksanakan tugas akhir ini.
4. Bapak Dr.Awang Hendrianto Pratomo,S.T, M.T. selaku Dosen Pembimbing II atas kesediaan, bimbingan, saran, dukungan dan ilmu yang telahdiberikan kepada penulis untuk selama melaksanakan tugas akhir ini.
5. Bapak Budi Dr. Heriyanto, A.Md, S.Kom, M.Cs, dan Bapak Mangaras Yanu F.,S.T, M.Eng, selaku Dosen Penguji atas kesediaan, bimbingan, saran, dukungan dan ilmu yang telah diberikan kepada penulis untuk selama melaksanakan tugas akhir ini.
6. Seluruh dosen dan staf tata usaha Jurusan Teknik Informatika yang telah membantu selama menempuh pendidikan di Universitas Pembangunan Nasional “Veteran” Yogyakarta.
7. Teman terbaik Khasnaul Muniroh dan Dwi Muawanah yang telah memberi dukungan dan doa selama penulis selama pengerjaan tugas akhir.
8. Teman-teman yang kebersamai semasa penyusunan laporan tugas akhir Tirta, Yola, Novia, Dhita, Nisa, Mifta, Novita, Linda, Theresia dan seluruh keluarga besar besar teknik informatika khususnya angkatan 2015 yang telah memberikan dukungan, motivasi, dan pengalaman menarik dari awal perkuliahan hingga tugas akhir ini diselesaikan.
9. Seluruh pihak yang telah memberikan semangat dan motivasi serta bantuan dalam penulisan tugas akhir. Serta pihak-pihak lain yang tidak mungkin penulis sebutkan satu-persatu.

Semoga Allah SWT memberi balasan yang setimpal kepada semuanya. Penulis berharapkripsi yang telah disusun ini bisa memberikan sumbangsih untuk menambah pengetahuan para pembaca, dan akhir kata, dalam rangka perbaikan selanjutnya, penulis

akan terbuka terhadap saran dan masukan dari semua pihak karena penulis menyadari skripsi yang telah disusun ini memiliki banyak sekali kekurangan.

Wassalamu 'alaikum Wr.Wb.

DAFTAR ISI

HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR	iv
SURAT PERNYATAAN BEBAS PLAGIASI.....	v
ABSTRAK.....	vi
KATA PENGANTAR	vii
DAFTAR ISI	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR	v
DAFTAR MODUL PROGRAM	vii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Tugas Akhir.....	3
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi Penelitian.....	3
1.6.1 Metode Pengumpulan Data.....	3
1.6.2 Metode Pengembangan Sistem	4
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Retail.....	6
2.2 Distributor	6
2.3 Warehouse.....	7
2.4 Supply Chain Management (SCM)	8
2.5 Vendor Managed Inventory (VMI)	8
2.6 Point Of Sale	10
2.7 Algoritma K-Means.....	11
2.7.1 Contoh Kalkulasi Algoritma K-means	11
2.7.2 Penelitian Terkait	13
2.8 Fuzzy.....	13
2.8.1 Contoh Kalkulasi Algoritma Fuzzy Tsukamoto.....	14
2.8.2 Penelitian Terkait	16
2.8.3 Pengujian MAPE.....	16
2.9 Penelitian Terkait	17
BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM	19
3.1 Metodologi Penelitian.....	19
3.2 Requirement	19
3.2.1 Studi Pustaka.....	20
3.2.2 Observasi.....	20
3.2.3 Analisis Permasalahan	20
3.2.4 Analisis Data	21
3.2.5 Analisis Kebutuhan Sistem	22
3.3 Proses Desain	22
3.3.1 Perancangan Arsitektur	22
3.3.2 Perancangan Basis Data	30
3.3.3 Perancangan Perangkat Lunak	40

3.4	Perancangan Pengujian.....	57
3.4.1	Perancangan Pengujian Fungsional Aplikasi.....	57
3.4.2	Perancangan Pengujian Algoritma.....	57
BAB IV HASIL DAN PEMBAHASAN		59
4.1	Hasil Penelitian	59
4.1.1	Hasil Perancangan Sistem	59
4.2	Pengujian	87
4.2.1	Pengujian Fungsional Sistem	88
4.2.2	Pengujian MAPE	89
4.3	Pembahasan.....	98
BAB V PENUTUP		100
5.1	Kesimpulan.....	100
5.2	Saran	100
DAFTAR PUSTAKA		101
LAMPIRAN		104

DAFTAR TABEL

Tabel 2.1 Tabel Data Sample (Mandala et al., 2018)	12
Tabel 2.2 Titik Awal Cluster (Mandala et al., 2018).....	12
Tabel 2.3 Tabel Jarak Item (M) ke C1 dan C2 (Mandala et al., 2018).....	12
Tabel 2.4 Tabel Kelompok Data dengan Masing-masing Cluster (Mandala et al., 2018)...	12
Tabel 2.5 Rangkuman Penelitian Terkait	17
Tabel 2.6 Lanjutan Rangkuman Penelitian Terkait	18
Tabel 3.1 Tabel Data Primer	21
Tabel 3.2 Lanjutan Tabel Data Primer	22
Tabel 3.3 Tabel Data Penjualan Produk Aqua	25
Tabel 3.4 Titik Awal Cluster	25
Tabel 3.5 Tabel Jarak Item (M) ke C1 dan C2	25
Tabel 3.6 Tabel Kelompok Data dengan Masing-masing Cluster.....	26
Tabel 3.7 Tabel Data Penjualan Produk Aqua	26
Tabel 3. 8 Tabel Data Penjualan Produk Gula Pasir.....	27
Tabel 3.9 Tabel Variabel Fuzzy Data Penjualan Produk Gula Pasir	28
Tabel 3.10 Tabel produk	34
Tabel 3.11 Tabel satuan	34
Tabel 3.12 Tabel kategori.....	34
Tabel 3.13 Tabel diskon.....	34
Tabel 3.14 Tabel suplier.....	35
Tabel 3.15 Tabel distributor_restok.....	35
Tabel 3.16 Tabel rincianDetailSuplay	35
Tabel 3.17 Tabel retailer_restok.....	35
Tabel 3.18 Tabel rincian_retailer_restok	36
Tabel 3.19 Tabel produk_retailer	36
Tabel 3.20 Tabel transaksi.....	36
Tabel 3.21 Tabel rincian_transaksi.....	36
Tabel 3.22 Tabel pegawai	37
Tabel 3.23 Tabel retailer	37
Tabel 3.24 Tabel laporan_retailer	37
Tabel 3. 25 Tabel pengiriman.....	38
Tabel 3.26 Tabel rekomendasi_fuzzy	38
Tabel 3.27 Tabel cluster.....	38
Tabel 3.28 Tabel Rancangan Pengujian Fungsional Aplikasi.....	57
Tabel 3.29 Tabel Rancangan Pengujian Algoritma Fuzzy Tsukamoto	58
Tabel 4.1 Tabel Responden Pengujian Aplikasi.....	88
Tabel 4.2 Tabel Pengujian Aplikasi.....	88
Tabel 4.3 Lanjutan Tabel Pengujian Aplikasi	89
Tabel 4.4 Tabel Data Sample Produk	90
Tabel 4.5 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Satu Minggu.....	91
Tabel 4.6 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Dua Minggu	92
Tabel 4.7 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Tiga Minggu	94
Tabel 4.8 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Empat Minggu	95
Tabel 4.9 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Lima Minggu	96
Tabel 4.10 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Lima Minggu	98

DAFTAR GAMBAR

Gambar 1.1 Tahapan Metode Agile (Runyan, 2015).....	4
Gambar 2.1 Pemodelan VMI (Aritonang et al., 2017)	9
Gambar 2.2 Pemodelan VMI Multi Retailer (Yosefa et al., 2015).....	10
Gambar 2.3 Fungsi Keanggotaan Variable Penjualan (Adithama et al., 2020)	14
Gambar 2.4 Fungsi Keanggotaan Variable Persediaan (Adithama et al., 2020).....	15
Gambar 2.5 Fungsi Keanggotaan Variable Pembelian (Adithama et al., 2020).....	15
Gambar 3.1 Kerangka Kerja Pengembangan Sistem.....	19
Gambar 3.2 Arsitektur Sistem Model VMI Satu Pemasok Multi Retailer	23
Gambar 3.3 Arsitektur Sistem	23
Gambar 3.4 Flowchart Penentuan Kelompok Cluster dengan Algoritma K-Means	24
Gambar 3.5 Flowchart Rekomendasi Penyetokan Produk dengan Algoritma Fuzzy Tsukamoto	27
Gambar 3.6 Fungsi Keanggotaan Variable Penjualan	28
Gambar 3.7 Fungsi Keanggotaan Variable Persediaan.....	29
Gambar 3.8 Fungsi Keanggotaan Variable Pembelian	29
Gambar 3.9 Entity Relationship Diagram (ERD).....	31
Gambar 3.10 Relasi Antar Tabel	39
Gambar 3.11 Context Diagram.....	40
Gambar 3.12 DFD level 1	42
Gambar 3.13 DFD level 2	44
Gambar 3.14 Perancangan Struktur Menu	47
Gambar 3.15 Halaman Awal	47
Gambar 3.16 Halaman Login Distributor	48
Gambar 3.17 Halaman Lupa Password.....	48
Gambar 3.18 Halaman Dashboard.....	49
Gambar 3.19 Halaman Tabel List.....	49
Gambar 3.20 Halaman Data Karyawan	50
Gambar 3.21 Halaman Data Retailer	51
Gambar 3.22 Halaman Rekomendasi	51
Gambar 3.23 Halaman Pengiriman.....	52
Gambar 3.24 Halaman Riwayat Restok.....	52
Gambar 3.25 Data Pelaporan.....	53
Gambar 3.26 Halaman Login Retailer	53
Gambar 3.27 Halaman Registrasi Retailer	54
Gambar 3.28 Halaman Dashboard Retailer.....	54
Gambar 3.29 Halaman Produk Retailer	55
Gambar 3.30 Halaman Transaksi Retailer	55
Gambar 3.31 Halaman Pelaporan Retailer	56
Gambar 3.32 Halaman Riwayat Pelaporan Retailer	56
Gambar 3.33 Halaman Reports Retailer	57
Gambar 4.1 Tampilan Halaman Utama	59
Gambar 4.2 Tampilan Halaman Login	60
Gambar 4.3 Tampilan Halaman Dashboard Distributor	61
Gambar 4.4 Tampilan Halaman Dashboard Distributor	62
Gambar 4.5 Tampilan Halaman Tambah Produk	63
Gambar 4.6 Tampilan Halaman Input Produk	64
Gambar 4.7 Tampilan Halaman Edit Produk	65

Gambar 4.8 Tampilan Halaman Tabel Kryawan.....	67
Gambar 4.9 Tampilan Halaman Tabel Retailer.....	67
Gambar 4.10 Tampilan Halaman Edit Data Retailer.....	68
Gambar 4.11 Tampilan Halaman Tabel Produk Retailer.....	68
Gambar 4.12 Tampilan Halaman Rekomendasi Restok.....	69
Gambar 4.13 Tampilan Halaman Rekomendasi Distribusi Produk.....	70
Gambar 4.14 Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto.....	74
Gambar 4.15 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto.....	74
Gambar 4.16 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto.....	75
Gambar 4.17 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto.....	76
Gambar 4.18 Tampilan Halaman Tabel Pengiriman.....	76
Gambar 4.19 Tampilan Halaman Riwayat Restok.....	77
Gambar 4.20 Tampilan Halaman Produk Retailer All.....	77
Gambar 4.21 Tampilan Halaman Detail Perhitungan Algoritma K-Means.....	80
Gambar 4.22 Lanjutan Tampilan Halaman Detail Perhitungan Algoritma K-Means.....	81
Gambar 4.23 Tampilan Halaman Data Pelaporan.....	82
Gambar 4.24 Tampilan Halaman Pengujian MAPE.....	82
Gambar 4.25 Lanjutan Tampilan Halaman Pengujian MAPE.....	83
Gambar 4.26 Tampilan Halaman Retailer Dashboard.....	84
Gambar 4.27 Tampilan Halaman Retailer Produk.....	84
Gambar 4.28 Tampilan Halaman Retailer Transaksi.....	85
Gambar 4.29 Tampilan Halaman Retailer Riwayat Transaksi.....	86
Gambar 4.30 Tampilan Halaman Retailer Pelaporan.....	86
Gambar 4.31 Tampilan Halaman Retailer Riwayat Pelaporan.....	87
Gambar 4.32 Tampilan Halaman Retailer Reports.....	87
Gambar 4.33 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Satu Minggu.....	90
Gambar 4.34 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Dua Minggu.....	92
Gambar 4.35 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Tiga Minggu.....	93
Gambar 4.36 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Empat Minggu.....	94
Gambar 4.37 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Lima Minggu.....	96
Gambar 4.38 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Enam Minggu.....	97

DAFTAR MODUL PROGRAM

Modul Program 4.1 Pseudocode Proses Login Karyawan.....	60
Modul Program 4.2 Pseudo Code Proses Login Retailer.....	60
Modul Program 4.3 Pseudo Code Tampilan Informasi Data Dashboard.....	61
Modul Program 4.4 Pseudo Code Tambah Jumlah Data Produk	63
Modul Program 4.5 Pseudo Code Input Data Produk.....	65
Modul Program 4.6 Pseudo Code Edit Data Produk	66
Modul Program 4.7 Pseudo Code Rekomendasi dengan Algoritma Fuzzy Tsukamoto.....	72
Modul Program 4.8 Lanjutan Pseudo Code Rekomendasi dengan Algoritma Fuzzy Tsukamoto	73
Modul Program 4.9 Pseudo Code Algoritma K-Means	78
Modul Program 4.10 Lanjutan Pseudo Code Algoritma K-Means	79
Modul Program 4.11 Pengujian MAPE	83

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Peningkatan infrastruktur dan teknologi manufaktur mengakibatkan meningkatnya pertumbuhan industri *retail* modern dengan konsep waralaba atau *franchise* seperti *Indomaret* dan *Alfamart*. Hal ini memicu persaingan yang cukup ketat bagi *retail* tradisional dan berakibat menurunnya omset pendapatan (Lestario, 2017). Waralaba dengan konsep terintegrasi memiliki keunggulan dalam pengelolaan dan analisis data sehingga dapat memberikan kualitas produk yang terjamin serta memenuhi kebutuhan pasar dengan harga yang relatif stabil. Hal ini akan menarik minat konsumen dalam bertransaksi dibanding dengan *retail* tradisional (Triyawan & Jayanti, 2018). Dikuasainya bisnis *retail* modern oleh suatu perusahaan tertentu mengakibatkan persaingan dan monopoli di suatu daerah sehingga menimbulkan dampak negatif bagi pengusaha lokal yang berada di daerah tersebut terutama bagi *retailer* tradisional seperti toko kelontong dan waserda (Jumaidi et al., 2019). Pada penelitian terkait Peranan Pengendalian Internal Persediaan yang dilakukan pada suatu bisnis eceran. Mencatat adanya penumpukan barang atau *over stock* yang mengakibatkan kerusakan dan kadaluarsa pada barang sehingga tidak diminati lagi oleh konsumen. Penyebabnya adalah pengadaan barang yang berlebihan dan tidak sesuai, sementara kebutuhan konsumen tidak dapat diprediksi (Maulina & Rahmi, 2016). Sementara itu ketika barang terlalu sedikit atau *out of stock* menyebabkan terhambatnya proses penjualan. Serta pembelian barang yang sedikit juga meningkatkan frekuensi pembelian ke pemasok yang tidak efektif dan meningkatkan biaya pengiriman (Zain et al., 2015).

Bagi distributor permintaan pasar yang lebih cepat mengalami perubahan menuntut untuk lebih detail dalam perencanaan strategi untuk meningkatkan penjualan setiap harinya (Hayati, 2014). Beberapa hal yang perlu dilakukan oleh pihak distributor adalah melakukan analisis dan manajemen pasokan untuk menentukan strategi tepat agar dapat menentukan jumlah dan lokasi yang tepat pada barang yang akan. Sehingga diperlukan informasi penjualan produk termasuk diantaranya mengenai sisa stok barang yang telah di distribusikan kepada *retailer* (Yosefa et al., 2015). Untuk mengatasi masalah ini maka perlu dibuatnya suatu sistem terintegrasi antara distributor dan pihak *retail* tradisional yang mencakup berbagai proses transaksi, analisis serta koordinasi. Analisis penjualan diperlukan untuk menghasilkan informasi yang berguna agar dapat mendukung keputusan penjualan maupun penyediaan stok barang. Informasi menjadi salah satu komponen yang sangat penting bagi pedagang *retail*. Cara suatu badan usaha mengelola informasinya merupakan kunci suksesnya suatu kegiatan bisnis di era modern (Indrajat, 2016).

Pada penelitian yang dilakukan oleh Nuha pada tahun 2017 menjelaskan pentingnya saluran informasi yang terintegrasi ke dalam sistem *point of sale* bagi pedagang *retail* multi-cabang yang tersebar diberbagai wilayah dan *manager* serta mobilitas yang tinggi dengan menggunakan *internet* sehingga dapat menekan biaya pemasangan jaringan serta berguna untuk pengintegrasian data meskipun berbeda *device*. Pada penelitian lain juga telah banyak membahas mengenai pengoptimalan bisnis dengan melakukan analisis penjualan dan

manajemen pasokan diantaranya oleh Yosefa et al. pada tahun 2015. mengenai Perancangan Model *Vendor Managed Inventory* (VMI) dengan Satu Pemasok dan Banyak *Retailer*. Metode ini mampu mengurangi biaya rantai pasokan dengan melakukan analisis dan koordinasi antara distributor dengan pedagang *retail* yang telah diintegrasikan. Salah satunya dengan melakukan pengiriman data *Point of Sale* (POS) dari pihak *retail* kepada distributor.

Penelitian terkait analisis penjualan dengan algoritma *K-Means* dilakukan oleh Mandala et al. Pada tahun 2018 yang menghasilkan analisis mengelompokkan penjualan barang serta dapat mengurangi penumpukan barang di masing-masing toko retail yang telah terintegrasi. Penelitian serupa juga dilakukan oleh Muningsih & Kiswati pada tahun 2015 mengenai penentuan stok barang dengan menggunakan algoritma *K-Means* pada produk *online shop* yang menghasilkan tiga *cluster* yaitu *cluster* pertama merupakan produk yang paling diminati untuk jumlah stok banyak, *cluster* kedua merupakan produk diminati untuk jumlah stok sedang dan *cluster* ketiga merupakan produk kurang diminati untuk jumlah stok sedikit.

Penelitian lain terkait pengoptimalan jumlah pengiriman barang telah banyak dilakukan diantaranya melalui pengembangan sistem rekomendasi dengan menggunakan algoritma *Fuzzy* untuk perencanaan stok barang yang dapat membantu memberikan referensi dalam pengambilan keputusan terkait jumlah pengadaan barang dan menghasilkan rekomendasi yang optimal sehingga mengurangi terjadinya kesalahan pengadaan jumlah barang yang mengakibatkan barang *out of stock* maupun *over stock* (Kosasih, & Setiyawati, 2020; Adithama et al., 2020; Wibawa et al., 2018). Variabel yang digunakan untuk menerapkan metode *Fuzzy* dalam penentuan rekomendasi pengiriman agar persediaan sesuai dengan kebutuhan pasar diantaranya jumlah persediaan, jumlah penjualan, dan jumlah pembelian (Harefa, 2017).

Penelitian yang akan dilakukan mengusulkan suatu manajemen rantai pasok dengan model *Vendor Managed Inventory* (VMI) dengan satu pemasok dan banyak *retailer* yang terintegrasi antara gudang utama yang berperan sebagai distributor dengan pedagang-pedagang *retail* tradisional. Serta melakukan analisis penjualan dengan menggunakan algoritma *Fuzzy* untuk penentuan rekomendasi jumlah distribusi yang sesuai bagi distributor dan algoritma *K-Means* untuk mengetahui tingkat penjualan barang yang akan dibagi menjadi beberapa *cluster* pada tiap-tiap *retail*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan pada subbab latar belakang masalah, maka perumusan masalah dalam penelitian ini adalah:

1. Bagaimana implementasi manajemen rantai pasok terintegrasi menggunakan model VMI dengan satu pemasok dan banyak *retailer*?
2. Bagaimana implementasi algoritma *Fuzzy Tsukamoto* untuk penentuan rekomendasi jumlah restok pada proses distribusi multi *retailer*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Objek penelitian berupa data penjualan pedagang *retail* dan data produk yang dijual.

2. Data yang digunakan untuk analisis adalah data penjualan selama bulan Januari 2022.
3. Analisa penjualan mencakup pembagian kategori penjualan menjadi tiga *cluster* dengan menggunakan algoritma *K-Means* serta penentuan rekomendasi jumlah distribusi produk dengan menggunakan algoritma *Fuzzy*.
4. Pada penelitian ini algoritma *K-Means* hanya berfungsi untuk memberikan informasi pengelompokan produk menjadi tiga *cluster* yaitu laris, cukup laris dan kurang laris.

1.4 Tujuan Tugas Akhir

Berdasarkan permasalahan yang dijelaskan pada latar belakang, maka tujuan penelitian ini adalah:

1. Menerapkan konsep sistem terintegrasi dengan model VMI satu pemasok dan banyak *retailer* untuk meningkatkan efektivitas manajemen pasok.
2. Mengetahui bagaimana implementasi penggunaan algoritma *Fuzzy Tsukamoto* dalam melakukan rekomendasi jumlah restok pada proses distribusi multi *retailer*.

1.5 Manfaat Tugas Akhir

Dengan adanya penelitian ini manfaat yang diharapkan adalah:

1. Dengan diterapkannya model VMI satu pemasok banyak *retailer* yang terintegrasi maka akan memudahkan analisis bagi distributor sehingga menghasilkan informasi baru yang dapat dijadikan rekomendasi untuk menentukan kebijakan dan keputusan strategis lebih lanjut dalam pendistribusian, seperti penentuan lokasi, jenis, dan jumlah barang yang akan didistribusikan. Selain itu penelitian ini diharapkan akan membantu mengurangi penumpukan barang yang tidak terjual atau *over stock*, tidak tersedianya barang atau *out of stock*, serta mempercepat aliran barang.
2. Mengetahui tingkat akurasi penggunaan algoritma *Fuzzy Tsukamoto* untuk melakukan analisis penjualan pada data multi *retail*.

1.6 Metodologi Penelitian

Tahapan-tahapan dilakukan serta metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1.6.1 Metode Pengumpulan Data

1. Observasi

Observasi dilakukan untuk menambah informasi serta mendapatkan data yang akan digunakan dalam penelitian. Observasi dilakukan dengan melakukan pengamatan kepada pihak distributor dan *retailer* serta pengambilan data yang dibutuhkan dari pihak *retailer*.

2. Wawancara

Wawancara dilakukan dengan melakukan komunikasi secara langsung kepada pihak distributor dan *retailer* untuk mendapatkan informasi dan data yang diperlukan terkait penelitian.

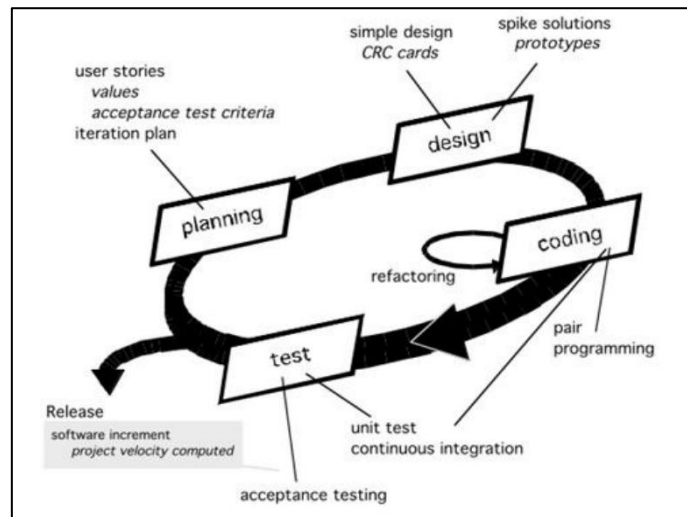
3. Studi literatur

Pada tahap ini dilakukan studi terkait penelitian, jurnal, dan buku yang

berhubungan dengan *Supply Chain Management* (SCM), VMI, algoritma *clustering* serta pendukung keputusan yang diterapkan dalam berbagai studi kasus. Dilakukan studi mengenai perbandingan metode-metode yang akan digunakan melalui jurnal dan penelitian yang pernah dilakukan sebelumnya sebagai bahan pertimbangan untuk menentukan metode yang akan digunakan. Serta mempelajari mengenai teori-teori pemrograman melalui materi kuliah, buku, jurnal maupun literatur lain yang dapat dijadikan referensi.

1.6.2 Metode Pengembangan Sistem

Metode digunakan untuk membangun sistem ini menggunakan metode *Agile*. Pada metode *Agile* merupakan metode pengembangan incremental yang berfokus pada pengembangan yang cepat dan bertahap serta mengurangi *overhead* proses (Sommerville, 2011). Sistematisa metode *Agile* terdapat pada Gambar 1.1 memperlihatkan tahapan pada metode *Agile*.



Gambar 1.1 Tahapan Metode Agile (Runyan, 2015)

Dalam pengembangannya metode *Agile* memiliki beberapa tahapan yang berurut yaitu sebagai berikut:

1. *Planning* dan *Requirement* (Perencanaan dan Pengumpulan Kebutuhan)

Tahapan perencanaan dan pengumpulan kebutuhan bertujuan untuk mengetahui dan memahami berbagai macam kebutuhan suatu sistem serta untuk mendapatkan data yang lengkap terkait kebutuhan sistem maupun kebutuhan pengguna akan sistem yang akan dikembangkan.

2. *Design* (Desain)

Proses desain adalah proses perancangan dari sistem yang akan dibangun sesuai dengan data yang telah diperoleh secara mendetail dan terperinci mulai dari tahapan perancangan arsitektur sistem, perancangan *database*, hingga perancangan perangkat lunak dengan tujuan memberikan gambaran mengenai apa saja yang dikerjakan.

3. *Development* (Pembangunan Sistem)

Selanjutnya tahapan pembangunan sistem adalah proses pengembangan sistem menggunakan bahasa pemrograman tertentu untuk menghasilkan suatu sistem informasi yang sesuai dengan rancangan yang dibuat serta menguji bagaimana fungsionalitas dari suatu sistem.

4. *Testing* (Pengujian)

Tahapan selanjutnya ialah melakukan pengujian sistem secara keseluruhan untuk memeriksa dan mengidentifikasi kemungkinan adanya kesalahan dalam suatu sistem yang telah dikembangkan.

5. *Release* (Penerapan)

Tahapan *release* atau penerapan merupakan tahapan penyebaran sistem yang telah dikembangkan oleh pengembang.

1.7 Sistematika Penulisan

Adapun sistematika penulisan penelitian ini adalah sebagai berikut:

Bab I Pendahuluan

Dalam bab ini dijelaskan tentang latar belakang penelitian, rumusan masalah, batasan-batasan masalah dalam penelitian yang dilakukan, tujuan penelitian berdasarkan permasalahan yang dijelaskan pada latar belakang, manfaat penelitian, metodologi penelitian yang berisi tahapan-tahapan dilakukan serta metodologi yang digunakan dalam penelitian, serta sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi uraian tinjauan pustaka dari beberapa penelitian yang sudah dilakukan sebelumnya sebagai landasan yang mendasari dan menunjang penelitian yang akan dilakukan dalam permasalahan yang diangkat. Diantaranya mengenai *retail*, distributor, *warehouse*, *Supply Chain Management (SCM)*, *Vendor Managed Inventory (VMI)*, *point of sale*, algoritma *K-Means*, algoritma *Fuzzy Tsukamoto* serta pengujian MAPE.

Bab III Analisis dan Perancangan Sistem

Bab ini menjelaskan mengenai metodologi penelitian, *requiremen* atau pengumpulan kebutuhan yang bertujuan untuk mengetahui dan memahami berbagai macam kebutuhan suatu sistem serta untuk mendapatkan data yang lengkap terkait kebutuhan sistem yang akan dikembangkan, proses desain mulai dari tahapan perancangan arsitektur sistem, perancangan *database*, hingga perancangan perangkat lunak dengan tujuan memberikan gambaran mengenai apa yang akan dikerjakan, serta perancangan pengujian dalam melakukan penelitian serta pengembangan sistem yang dibagi menjadi dua macam pengujian diantaranya pengujian terkait fungsional aplikasi, dan pengujian terkait algoritma yang digunakan.

Bab IV Hasil, Pengujian dan Pembahasan

Bab ini menyajikan pembahasan dari hasil penelitian yang dilakukan yaitu berisi implementasi serta analisis dari perencanaan yang telah dibuat pada bab sebelumnya terkait hasil penelitian, hasil perancangan sistem, pengujian fungsional sistem, pengujian mape serta pembahasan yang berisi hasil dari penyelesaian masalah yang telah dirumuskan serta hasil pengujian yang telah dilakukan.

Bab V Penutup

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilaksanakan mengenai manajemen rantai pasok terintegrasi dengan konsep VMI menggunakan algoritma *Fuzzy Tsukamoto*, serta saran- saran untuk penelitian selanjutnya agar memberikan pengembangan penelitian yang lebih baik.

BAB II

TINJAUAN PUSTAKA

2.1 Retail

Ritel atau *retail* merupakan istilah yang berasal dari bahasa Perancis “*retailer*” yang memiliki arti memecah atau memotong sesuatu. *Retail* juga dapat diartikan eceran atau pedagang eceran sementara *retailer* adalah pengecer yaitu sebuah lembaga yang melakukan usaha menjual barang kepada konsumen akhir untuk keperluan pribadi (Fauza, 2017). Berdasarkan sifatnya *retail* dikategorikan menjadi dua macam yaitu *retail* tradisional dan *retail* modern. *Retail* tradisional memiliki beberapa ciri utama diantaranya yaitu, konsumen menengah kebawah, bersekala kecil, modal kecil, manajemen belum profesional, transaksi tunai serta jarang ada program promosi. Sedangkan ciri utama *retail* modern adalah manajemen modern, bermodal kuat menggunakan teknologi modern, harga sudah pasti, pembayaran dapat menggunakan kartu atau *e-money*, banyak kegiatan promosi, diskon, dan hadiah, konsumen menengah atas, umumnya dikelola pihak swasta serta kebanyakan berada di daerah perkotaan (Hikmawati & Nuryakin, 2018).

Retail memiliki kontribusi yang sangat besar dalam bidang perdagangan yaitu 19–31% terhadap Pendapatan Domestik Regional Bruto (PDRB) tiap daerah yang ada di Indonesia. Menurut Biro Perekonomian DKI Jakarta (2015) di Indonesia bisnis *retail* mengalami perkembangan yang cukup pesat yang mana *retail* modern mengalami pertumbuhan rata-rata 19% sementara *retail* tradisional mengalami pertumbuhan 6% pada tahun 2005 hingga 2013. Namun pada 3 tahun setelahnya tercatat pertumbuhan *retail* modern sebanyak 13% dari jumlah terakhir 2011 yang mana berbanding terbalik dengan *retail* tradisional yang tidak mengalami pertumbuhan.

Dalam penelitiannya (Hikmawati & Nuryakin, 2018) menjelaskan bahwa meningkatnya keberadaan *retail* modern memiliki pengaruh signifikan terhadap kinerja *retail* tradisional yang diwakili proksi pendekatan jumlah pedagang. Atau dapat dikatakan bahwa meningkatnya *retail* modern menjadi salah satu faktor penyebab menurunnya jumlah *retail* tradisional hingga *retail* tradisional yang perlahan mulai ditinggalkan. Perubahan pada masyarakat dalam memilih tempat berbelanja yang mana cenderung memilih *retail* modern dipengaruhi oleh faktor produk, harga, distribusi, proses, pelayanan hingga promosi pada *retail* modern lebih menarik. Melihat kondisi ini *retail* tradisional perlu untuk segera berbenah agar dapat mempertahankan eksistensinya agar tidak tergeser oleh *retail* modern (Mardiono & Suyanto, 2016).

Adapun beberapa tindakan yang dapat dilakukan untuk mengatasi permasalahan ini yaitu selain pengkajian ulang kebijakan mengenai peraturan *retail* juga perlunya peningkatan manajemen pengelolaan *retail* tradisional secara sistematis (Hikmawati & Nuryakin, 2018).

2.2 Distributor

Distributor merupakan orang atau perusahaan yang mengirim barang dari manufaktur ke *retailer* atau langsung ke pelanggan dan mengirim informasi serta biaya barang dari *retailer*/pelanggan ke manufaktur. (Utami & Cahyani, 2016) dalam penelitiannya menjelaskan bahwa sebagian besar *retail* menggunakan jasa distributor untuk memasok barang ke toko mereka dan sisanya sekitar 21% lainnya menggunakan jasa perusahaan manufaktur langsung sebagai pemasok barang yang akan dijual. Sebagian besar pedagang *retail* berskala kecil hanya

memiliki akses langsung dengan distributor, namun tidak memiliki akses untuk dapat berinteraksi secara langsung dengan perusahaan manufaktur sehingga pedagang *retail* dan distributor memiliki hubungan saling ketergantungan antara pihak satu dengan yang lain.

Sebagian pemasok biasanya memberikan fasilitas hutang dagang terhadap *retailer* berupa pembelian kredit dan pembelian konsinyasi. Pembelian kredit berarti penjual menerima bayaran dari pihak *retail* sesuai jangka waktu yang telah disepakati. Sedangkan pembelian konsinyasi merupakan bentuk kerja sama yang mana pemasok menitipkan barang ke pihak *retail* untuk dijual di tokonya, pihak *retail* membayar sesuai jumlah barang yang terjual dalam jangka waktu tertentu yang telah disepakati, sementara barang yang tidak terjual dapat ditarik kembali oleh pihak penyedia. Kesepakatan ini mengurangi risiko kerugian bagi pihak *retail* karena produk yang tidak laku terjual dapat dikembalikan ke penyedia produk (Utami & Cahyani, 2016).

2.3 Warehouse

Distributor dalam menjalankan kegiatan distribusi, memiliki elemen penting yang tidak bisa ditinggalkan yaitu *distribution center* atau yang sering disebut sebagai gudang (*warehouse*). *Warehouse* merupakan suatu bangunan atau ruang yang memiliki fungsi utama sebagai tempat penyimpanan bahan-bahan mentah (*raw material*), barang setengah jadi (*intermediate goods*), maupun produk yang telah jadi (*finish goods*) (Yusuf & Nuryanti, 2018). Didalam *warehouse* terdapat tiga aktivitas utama diantaranya yaitu proses penerimaan barang, proses penyimpanan barang, dan proses pendistribusian barang. *Warehouse* juga memiliki fungsi lain untuk memberikan informasi kepada manajemen tentang status, kondisi, dan disposisi barang-barang yang sedang disimpan (Ismail & Awaludin, 2017).

Warehouse merupakan suatu tempat persediaan barang yang menjadi salah satu elemen terpenting dalam kelangsungan perusahaan dagang. Persediaan barang dagang merupakan aset terbesar bagi perusahaan yang mana biasanya memiliki jenis yang beraneka ragam, hal ini merupakan karakteristik dari perusahaan dagang eceran. Di sisi lain, pengendalian barang yang tidak berjalan dengan baik menyebabkan berbagai macam kerugian. Dalam penelitian terkait pengendalian internal persediaan yang dilakukan pada suatu bisnis eceran, mencatat bahwa terjadi penumpukan barang dengan angka yang cukup tinggi, yang mana penyebabnya ialah pemesanan barang yang berlebihan (tidak sesuai) ke bagian pemasok, sementara kebutuhan konsumen tidak dapat diprediksi. Hal ini mengakibatkan risiko yang tidak diinginkan, diantaranya yaitu kerusakan, kadaluarsa, atau barang ketinggalan zaman sehingga tidak diminati lagi oleh konsumen (Maulina & Rahmi, 2016).

Penyediaan barang dengan jumlah besar atau *over stock* akan menimbulkan berbagai macam risiko serta perlu perusahaan menyediakan dana tambahan pemesanan kembali barang selanjutnya. Hal ini biasanya disebabkan karena pemesanan barang yang berlebihan untuk mengantisipasi akan kekurangan stok barang. Sementara itu ketika barang terlalu sedikit atau *out of stock* menyebabkan terhambatnya proses penjualan. Serta pembelian barang yang sedikit juga meningkatkan frekuensi pembelian ke pemasok yang tidak efektif dan meningkatkan biaya pemesanan (Zain et al., 2015). Permintaan pasar yang berbeda-beda dan cenderung tidak dapat diprediksi menjadi penyebab timbulnya risiko bagi distributor penyedia barang. Saat ini pasar lebih cepat mengalami perubahan yang dapat dilihat dari ragam produk yang ada dalam pasaran. Hal ini menuntut para distributor lebih detail dalam perencanaan strategi untuk meningkatkan penjualan setiap harinya (Hayati, 2014).

Beberapa hal yang perlu dilakukan oleh pihak distributor adalah melakukan analisis dan manajemen pasokan untuk menentukan strategi tepat agar dapat menentukan jumlah dan lokasi yang tepat barang yang akan didistribusikan untuk meminimalkan biaya persediaan dan meningkatkan keuntungan. Sehingga perlunya pengetahuan mengenai informasi terkait dengan penjualan produknya, termasuk diantaranya yaitu, mengenai sisa barang belum terjual yang telah di distribusikan kepada pedagang *retail*. Serta mengetahui berapa jumlah barangnya terjual dalam kurun waktu tertentu (Yosefa et al., 2015).

Maka dari itu analisis serta pengelolaan persediaan barang pada bagian gudang sangatlah penting agar dapat mengetahui kebijakan apa saja yang sesuai untuk dilakukan kedepannya (Maulina & Rahmi, 2016).

2.4 Supply Chain Management (SCM)

Supply Chain Management (SCM) merupakan serangkaian pendekatan yang diterapkan untuk mengintegrasikan antara pemasok, pengusaha, gudang (*warehouse*), dan toko atau tempat penjual akhir secara efisien sehingga barang diproduksi dan didistribusikan pada jumlah yang tepat, ke lokasi yang tepat, dan pada waktu yang tepat, untuk memuaskan kebutuhan pelanggan. SCM adalah suatu koordinasi yang membentuk mata rantai antara tiga unsur yaitu barang, informasi, dan arus keuangan antara perusahaan yang saling berkaitan (Simchi-Levi et al., 2004).

Tujuan utama dari SCM adalah untuk meningkatkan efektivitas dan efisiensi mulai dari pemasok, manufaktur, *warehouse* dan toko. Dengan melakukan koordinasi yang baik antara pihak terkait sehingga meminimalkan terjadinya kerugian. Pertukaran informasi antara toko *retail*, distributor dan perusahaan merupakan suatu cara yang dapat diterapkan untuk menghindari terjadinya *Bullwhip effect* atau perubahan permintaan dari *customer* akibat informasi yang tidak terdistribusi dengan baik yang akan berdampak mengacaukan proses bisnis. Sasaran dalam *supply chain* adalah memaksimalkan keseluruhan nilai (*value*) *supply chain* yang diciptakan (Hayati, 2014).

Supply chain value: Final Product Value – Total Supply Chain Cost

SC Value = Price of Widget – Cost of all Phases in SC

SC Value = Profit of SC

Menurut Simchi Levi dalam SCM ada tiga macam *strategic alliance* diantaranya, 3PL (*Third Party Logistics*) yang merupakan strategi yang melibatkan pihak ketiga untuk melakukan *outsourcing* suatu layanan logistik dalam SCM, *Distributor Integration* (DI) yang merupakan strategi integrasi distributor, dan *Retailer Pemasok Partnership* (RSP) yaitu strategi kolaborasi antara pemasok dan *retail* dimana dalam hubungan ini diperlukan kerjasama dan koordinasi yang baik antara kedua belah pihak untuk mendapatkan keuntungan bagi keduanya (Yosefa et al., 2015). Dengan sistem informasi yang baik memungkinkan RSP untuk dapat meningkatkan pengetahuan bagi kedua belah pihak. Dengan menggunakan teknologi pengambilan data dari pihak *retail* dan pertukaran data elektronik, RSP memungkinkan untuk memberikan respon yang cepat, berkelanjutan, sehingga dapat memberi manfaat baik bagi pihak distributor maupun *retailer*. (James S & Craig A, 2013).

2.5 Vendor Managed Inventory (VMI)

Vendor Managed Inventory (VMI) merupakan salah satu metode dalam *supply chain* yang mengutamakan kerja sama antara pihak pemasok dan *retailer*. Dalam sistem ini, pihak pemasok memonitor dan bertanggung jawab atas persediaan yang ada pada *retailer*. Dalam

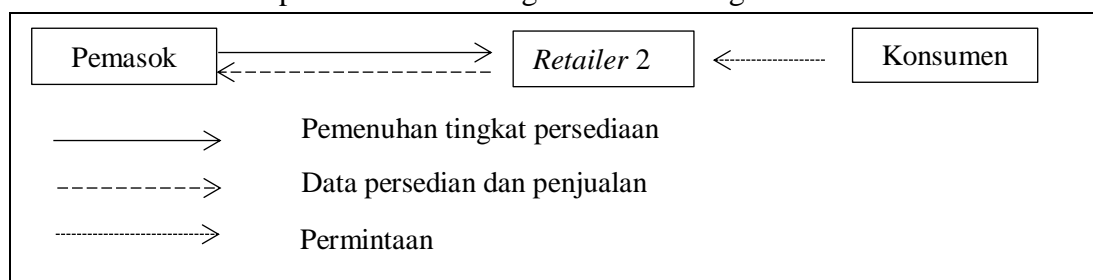
model VMI, pihak *retailer* tidak lagi memutuskan apa, kapan, dan bagaimana barang akan dibeli dari pemasok. Pihak *retailer* hanya memberikan informasi tentang permintaan pelanggan dan sisa persediaan serta informasi mengenai persediaan minimum dan maksimum yang diharapkan. Informasi ini berguna bagi pihak pemasok untuk memutuskan waktu dan jumlah penambahan persediaan (Anna, 2016). Dalam model VMI, pemasok memiliki wewenang untuk menentukan jumlah order yang akan dikirimkan ke pihak *retailer* berdasarkan informasi data penjualan dan tingkat persediaan yang telah ditetapkan. (Yosefa et al., 2015).

Simchi Levi dkk. Menjelaskan ada dua langkah utama dalam mengimplementasikan VMI, diantaranya :

1. Negosiasi ketentuan kontrak perjanjian yang mencakup kepemilikan, pembayaran, persyaratan kredit, waktu pengiriman, tanggung jawab pemesanan serta tingkat pelayanan dan ketentuan persediaan jika diperlukan.
2. Melakukan eksekusi dalam hal pengembangan sistem informasi yang terintegrasi serta mudah diakses untuk distributor (pemasok) dan *retailer*, mengembangkan teknik peramalan serta *decision support tools* untuk membantu dalam mengkoordinasikan kebijakan yang akan diambil.

VMI merupakan strategi *supply chain* yang memerlukan waktu *replenishment* singkat dengan frekuensi dan pengiriman yang tepat waktu sehingga dapat mengurangi biaya persediaan. Model VMI juga dapat memperbaiki pelayanan terhadap konsumen sebab pemasok dapat memenuhi permintaan *retailer* tepat pada waktunya. Dalam rantai pasokan tradisional (non VMI), pemasok tidak dapat memperoleh informasi permintaan konsumen secara langsung tetapi melalui kebijakan pemesanan pembeli. Sebaliknya, dalam rantai pasokan dengan VMI, pemasok dapat memperoleh informasi permintaan konsumen secara langsung oleh sistem informasinya. Sehingga pemasok dapat memiliki informasi dari kombinasi dua biaya persediaan, yaitu pengaturan pesanan dan biaya penyimpanan (Anna, 2016).

Secara sederhana pemodelan VMI digambarkan sebagai berikut:

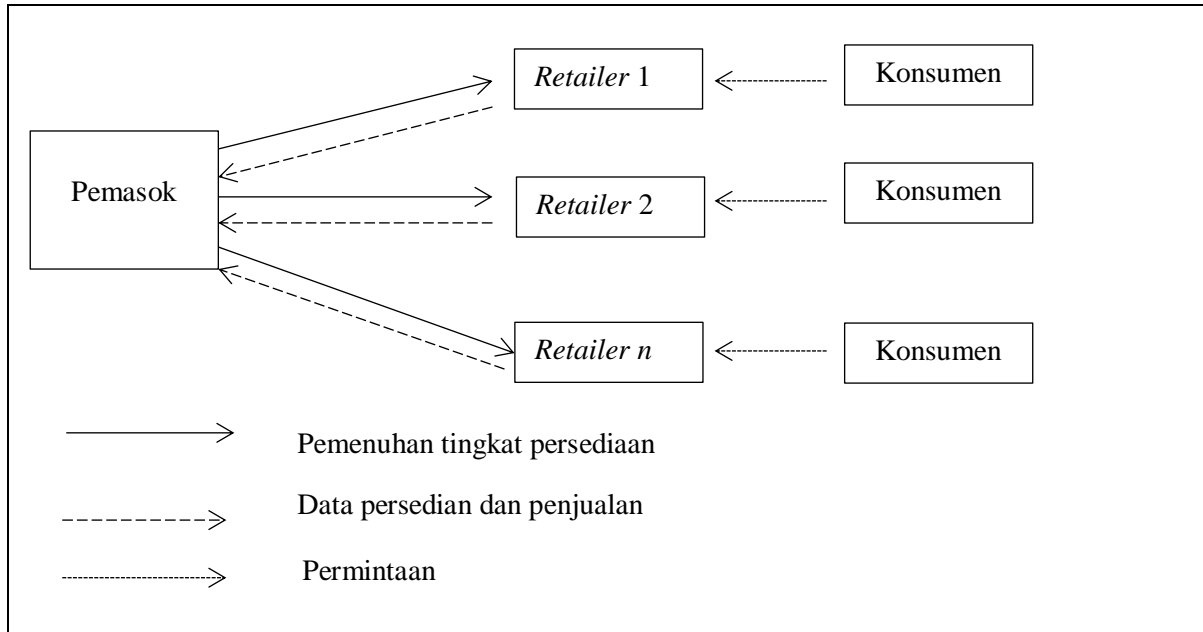


Gambar 2.1 Pemodelan VMI (Aritonang et al., 2017)

Informasi permintaan konsumen akhir yang diterima oleh *retailer* akan dibagikan kepada pemasok yang berperan sebagai *vendor*. Selain itu pemasok juga menerima informasi terkait persediaan yang dimiliki oleh *retailer*. Berdasarkan informasi tersebut pihak pemasok dapat mengirimkan barang kepada *retailer* untuk memenuhi kebutuhan *retailer* tersebut (Aritonang et al., 2017).

Pada penelitian yang dilakukan oleh (Yosefa et al., 2015). telah mengembangkan model VMI dengan satu pemasok banyak *retailer*. Pada model pemasok dengan banyak *retailer* didapatkan bahwa permintaan pada pemasok bersifat *lumpy* dikarenakan pemasok memiliki tingkat permintaan yang berbeda-beda sehingga frekuensi pemesanan barang kepada pihak pemasok juga mengalami perbedaan antara suatu *retailer* dengan *retailer* yang lain, sehingga digunakan model optimasi yang dirancang untuk permasalahan VMI dimana

pemasok adalah pengambil keputusan bagi kedua belah pihak (pemasok dan *retailer*).



Gambar 2.2 Pemodelan VMI Multi *Retailer* (Yosefa et al., 2015)

Namun model ini memiliki kelemahan dimana model ini tidak dapat digunakan untuk *retailer* dengan jumlah tujuh atau lebih dengan periode waktu pengiriman 12 bulan karena besarnya komputasi memori yang digunakan.

Penelitian lain oleh (Karma, 2018) pendekatan VMI berbasis *cloud computing* pada industri farmasi yang berbasis *web* dengan menggunakan *database* MySQL menghasilkan kolaborasi dan koordinasi yang baik antara distributor farmasi dan *retailer*. Selain itu, pendekatan ini dapat mengatasi masalah redundansi data, gap informasi, pemrosesan yang lama serta mempermudah akses informasi dan data. Namun sistem ini hanya dapat diterapkan oleh dua peranan rantai pasok yaitu *single retailer* dan *single* pemasok yang bertugas menyuplai barang farmasi. Sementara penerapan untuk multi *retailer* perlu dilakukan kembali analisis dan pendesainan yang lebih sesuai, baik dari sisi integrasi maupun pengiriman data.

2.6 Point Of Sale

Untuk memudahkan dokumentasi dan pengiriman data penjualan maka dibutuhkan suatu sistem yang dapat secara otomatis melakukan pendataan barang terutama barang yang keluar dan masuk. Hal ini dapat diatasi dengan diterapkannya aplikasi kasir atau *Point of Sales*. *Point of Sales* merupakan sebuah sistem elektronik atau *software* yang digunakan untuk memfasilitasi bisnis antara *merchant* atau retail dan konsumen menjadi lebih baik dan digunakan untuk melengkapi transaksi, serta dokumentasi penjualan untuk membuat proses penjualan menjadi praktis dan cepat (Bahri & Suhada, 2016). *Point of Sale* merupakan kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi sistem ini banyak dipakai para perusahaan yang berjalan dibidang *retail*, karena sistem memudahkan perusahaan dalam menghitung seluruh data yang berkaitan dengan transaksi sebuah perusahaan. Beberapa fungsi yang dapat dilakukan melakukan pendataan setiap transaksi dengan lengkap dan detail, melakukan cek persediaan barang, memberikan laporan penjualan secara *real-time* serta mempersingkat proses transaksi (Nuha, 2017).

2.7 Algoritma K-Means

K-means merupakan salah satu metode *clustering* yang digunakan untuk mengategorikan objek menjadi kelompok-kelompok objek serupa yang merupakan pengelompokan data non-hirarki berdasarkan nilai rata-rata objek dalam suatu kelompok yang membagi data menjadi dua atau lebih bentuk kelompok dimana data yang memiliki karakteristik sama dimasukkan ke dalam satu kelompok yang sama dan sebaliknya data yang memiliki karakteristik berbeda dikelompokkan ke dalam kelompok lain. Tujuannya untuk meminimalkan fungsi objektif yang diatur dalam proses pengelompokan dan memaksimalkan variasi antar kelompok (Putra & Wadisman, 2018). Secara umum proses *clustering* menggunakan algoritma *K-Means* adalah sebagai berikut (Silalahi, 2018):

Penentuan titik *centroid* atau titik pusat pada setiap kelompok yang diambil dari nilai rata-rata atau *means* semua nilai data pada setiap fitur nya. dengan persamaan sebagai berikut:

$$C_i = \frac{1}{p} \sum_{j=1}^p x_j \dots \dots \dots (2.1)$$

Keterangan

C = Cluster

p = Jumlah data pada suatu kelompok

i = Fitur ke-i dalam sebuah kelompok

Selanjutnya menghitung jarak setiap data dengan *centroid* menggunakan rumus *Euclidean Distance*.

$$D_{ik} = \sqrt{\sum_{j=1}^m |x_{ij} - c_{kj}|^2} \dots \dots \dots (2.2)$$

Keterangan

D = Titik dokumen

k = Data kriteria

Setelah diketahui jarak setiap data terhadap *centroid* maka data dialokasikan ke *centroid* terdekat (dengan jarak terpendek) dan dikelompokkan ke setiap *cluster* menggunakan persamaan berikut:

$$\min \sum_{k=1}^k D_{ik} \dots \dots \dots (2.3)$$

Selanjutnya adalah menghitung nilai *centroid* atau pusat *cluster* baru menggunakan persamaan berikut

$$C_{kj} = \frac{1}{p} \sum_{i=1}^p x_{ij} \dots \dots \dots (2.4)$$

Jika masih ada data yang berpindah *cluster* atau ada perubahan nilai *centroid* diatas nilai ambang/*Threshold* yang ditentukan maka Langkah (2.2) hingga (2.4) akan diulangi. Prinsip utama pengelompokan atau *clustering* dengan algoritma *K-means* adalah meminimalkan nilai jarak setiap data dengan titik pusat *cluster* yang disebut *centroid*.

2.7.1 Contoh Kalkulasi Algoritma K-means

Kalkulasi dengan metode *K-Means* diimplementasikan menggunakan contoh data *sample* penjualan seperti pada table 2.1 yang mana data akan dikategorikan menjadi dua *cluster* yaitu barang yang laris dan kurang laris.

Tabel 2.1 Tabel Data Sample (Mandala et al., 2018)

No	Item	Stok (x)	Terjual (y)
1	FRESTEA Lemon	45	25
2	MIZONE Passion Fruit	39	21
3	Biscuit GOOD TIME Cheese	65	10
4	Minute Maid PULPY ORANGE	62	15
5	AQUA 600 ml	32	26
6	GERY Salut Chocolate	37	23
7	Wafer NISSIN Chocolate	56	10

Langkah pertama adalah penentuan jumlah *cluster* (k). dalam contoh kasus ini penentuan *cluster* dipilih secara *random* yaitu sebanyak dua *cluster*. Dimana *cluster* 1 adalah barang yang laris dan *cluster* 2 adalah barang yang kurang laris.

Tabel 2.2 Titik Awal Cluster (Mandala et al., 2018)

Titik Pusat Awal	Stok (x)	Terjual (y)
<i>Cluster 1</i>	30	26
<i>Cluster 2</i>	55	13

Selanjutnya adalah penentuan titik pusat atau *centroid* serta jarak terdekat dengan menggunakan rumus perhitungan *Euclidean Distance* sebagai berikut:

Jarak data ke-1 dengan pusat *cluster*:

$$\begin{aligned}
 C(1) &= \sqrt{(45 - 30)^2 + (25 - 26)^2} \\
 &= \sqrt{225 + 1} \\
 &= \sqrt{226} \\
 &= 15,03
 \end{aligned}$$

$$\begin{aligned}
 C(2) &= \sqrt{(45 - 55)^2 + (25 - 13)^2} \\
 &= \sqrt{100 + 144} \\
 &= \sqrt{244} \\
 &= 15,62
 \end{aligned}$$

Hasil yang diperoleh dari perhitungan diatas merupakan jarak data ke-1 dengan titik pusat awal *cluster*. Dimana jarak data ke-1 dengan pusat *cluster* 1 adalah 15,03 sementara jarak dengan *cluster* 2 adalah 15,62. Begitu pula dengan data selanjutnya dihitung menggunakan rumus yang sama. Setelah semua data dihitung maka akan diperoleh jarak masing-masing data dengan *cluster* 1 (C1) dan *cluster* 2 (C2) sebagai berikut:

Tabel 2.3 Tabel Jarak Item (M) ke C1 dan C2 (Mandala et al., 2018)

	M1	M2	M3	M4	M5	M6	M7
C1	15,03	10,30	38,48	33,84	2,00	7,62	30,53
C2	15,62	17,89	10,44	7,28	26,42	20,59	3,16

Langkah selanjutnya adalah membandingkan jarak dari masing-masing item dengan C1 dan C2 yang mana nilai yang lebih kecil berarti jarak yang lebih dekat. Kemudian dikelompokkan sesuai dengan jarak terdekat misalnya M1 memiliki jarak yang lebih dekat dengan C1 maka M1 adalah anggota C1 dan seterusnya.

Tabel 2.4 Tabel Kelompok Data dengan Masing-masing Cluster (Mandala et al., 2018)

	Anggota
C1	M1, M2, M5, M6
C2	M3, M4, M6

Setelah anggota dari tiap-tiap *cluster* diketahui, maka langkah selanjutnya ditentukan pusat *cluster* atau *centroid* baru dengan perhitungan sebagai berikut.

C1 Baru =

$$\frac{M1x + M2x + M5x + M6x}{4} ; \frac{M1y + M2y + M5y + M6y}{4}$$

$$\frac{45 + 39 + 32 + 37}{4} ; \frac{25 + 21 + 26 + 23}{4}$$

C1 baru = 38,25; 23,75

C2 Baru =

$$\frac{M3x + M4x + M7x}{3} ; \frac{M3y + M4y + M7y}{3}$$

$$\frac{65 + 62 + 56}{3} ; \frac{10 + 15 + 10}{3}$$

C1 baru = 61,00; 11,57

Langkah selanjutnya menghitung kembali jarak masing-masing *item* (M) dengan C1 dan C2 yang baru menggunakan rumus *Euclidean Distance* sama seperti sebelumnya. Kemudian bandingkan jarak antara M dengan C1 dan C2 yang baru dan kelompokkan seperti Langkah sebelumnya apabila terjadi perubahan kelompok atau *cluster* maka iterasi akan terus dilakukan hingga tidak terjadi perubahan pada data kelompok atau tidak ada perubahan nilai *centroid* diatas nilai ambang/*Threshold* yang ditentukan (Mandala et al., 2018).

2.7.2 Penelitian Terkait

Dalam penelitian yang dilakukan oleh (Negara, Iis Setiawan Mangku; et al., 2022) mengenai analisis *cluster* data transaksi penjualan selama pandemi Covid-19 dengan menggunakan algoritma *K-Means*, yang dilakukan dengan mengelompokkan data berdasarkan kemiripan karakteristiknya terhadap 278480 data transaksi dengan dibagi menjadi tiga kelompok atau *cluster* yaitu penjualan terbanyak, penjualan sedang, dan penjualan rendah menghasilkan akurasi dengan *confusius matrix* sebanyak 87%. Penelitian lain dilakukan oleh (Herianto; et al., 2021) terkait analisis tingkat penjualan yang membandingkan algoritma *K-Nearest Neighbor* (KNN) dan algoritma *K-Means* dengan melakukan analisis *clustering* dengan menggunakan data yang sama dan diperoleh hasil bahwa algoritma *K-Means* memiliki tingkat akurasi yang lebih tinggi yaitu 78,37% dibanding algoritma KNN dengan akurasi sebanyak 76,77%.

Penelitian terkait penggunaan algoritma *K-Mean* untuk pengelompokan data logistik di Timur Tengah terbukti menghasilkan pengetahuan yang lebih mendalam karena hasil pengelompokannya yang relevan dan sesuai sehingga diusulkan menjadi model yang paling cocok untuk pengelompokan (Qabbaah, Hamzah ; et al., 2019). Penerapan algoritma *K-Means* dalam pengelompokan penjualan diketahui dapat membantu perencanaan persediaan, mengurangi penumpukan stok barang, serta perencanaan strategi penjualan. Selain itu, algoritma ini juga menangkap dinamika penjualan *retail* dalam jangka pendek dengan informasi histori dengan jumlah terbatas. (Jabat & Murdani, 2019; Lingxian, You ; et al., 2019; Imron, Mohammad; etal., 2020).

2.8 Fuzzy

Logika *Fuzzy* ditemukan oleh prof. Lotfu A. Zedeh pada tahun 1965. Logika ini merupakan bentuk generalisasi dari logika klasik yang mana hanya memiliki dua nilai keanggotaan yaitu 1 dan 0. Sementara dalam logika *Fuzzy* nilai keanggotaan berkisar antara 0 dan 1 atau yang dapat diartikan dari yang sepenuhnya benar hingga yang sepenuhnya salah. Konsep ini berbeda dengan teori himpunan klasik (*crip*) yang bergantung pada dua logika untuk menentukan keanggotaan suatu objek (Wibawa, I Made Bagus Wiradivka Laksa ; et al., 2018).

Salah satu jenis metode dalam logika *Fuzzy* adalah *Fuzzy Tsukamoto* yang merupakan perluasan dari penalaran monoton dimana setiap konsekuen pada setiap aturan *if-then* dipresentasikan dengan suatu himpunan *Fuzzy* dengan fungsi keanggotaan yang monoton dengan hasil *output* yang diberikan secara tegas berdasarkan α -predikat dan hasil akhirnya diperoleh menggunakan nilai rata-rata bobot (Novianti et al., 2018). Secara umum proses perhitungan algoritma *Fuzzy Tsukamoto* adalah sebagai berikut (Ayuningtias et al., 2017).

Menentukan fungsi implikasi. Metode *Fuzzy Tsukamoto* menggunakan fungsi implikasi MIN untuk mendapatkan nilai α -predikat tiap-tiap *rule* ($\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$). Masing-masing nilai α -predikat digunakan untuk menghitung hasil inferensi secara tegas (*crisp*) masing-masing *rule* ($z_1, z_2, z_3, \dots, z_n$).

Proses *defuzzyfikasi* pada metode *Fuzzy Tsukamoto* menggunakan metode rata-rata (*average*) dengan rumus berikut:

$$z = \frac{\sum_{i=1}^n a_i z_i}{\sum_{i=1}^n a_i} \dots \dots \dots (2.5)$$

2.8.1 Contoh Kalkulasi Algoritma Fuzzy Tsukamoto

Kalkulasi dengan algoritma *Fuzzy Tsukamoto* diimplementasikan dengan menggunakan contoh data penjualan susu Dancow pada bulan November (Adithama, Dewi, & Hariyadi, 2020), diketahui:

- a. Penjualan terbanyak dalam satu hari adalah 8 kardus
- b. Penjualan terkecil dalam satu hari adalah 0 kardus
- c. Persediaan terbanyak dalam satu hari adalah 18 kardus
- d. Persediaan terkecil dalam satu hari adalah 3 kardus
- e. Pembelian terbanyak adalah 25 kardus
- f. Pembelian terkecil adalah 0 kardus

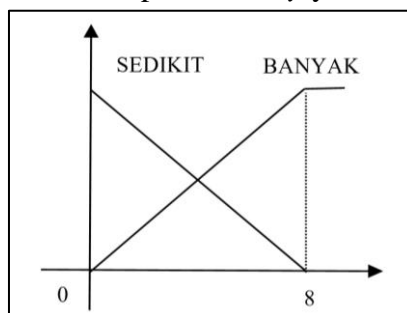
Jika diketahui persediaan sekarang adalah 15 kardus dan penjualan rata-rata adalah 3 kardus perhari maka jumlah rekomendasi pembelian susu Dancow pada bulan Desember dapat dihitung menggunakan kalkulasi sebagai berikut.

Langkah pertama adalah dengan melakukan pembentukan basis pengetahuan atau *rule* sebagai berikut:

- [R1] Jika penjualan banyak dan persediaan turun, maka pembelian banyak.
- [R2] Jika penjualan banyak dan persediaan naik, maka pembelian sedikit.
- [R3] Jika penjualan sedikit dan persediaan turun, maka pembelian sedikit.
- [R4] Jika penjualan sedikit dan persediaan naik, maka pembelian sedikit.

Kemudian melakukan pemodelan fungsi keanggotaan variable *Fuzzy* yang meliputi penjualan, persediaan, dan pembelian sebagai berikut.

- a. Penjualan, terdiri atas dua himpunan *Fuzzy* yaitu BANYAK dan SEDIKIT



Gambar 2.3 Fungsi Keanggotaan Variable Penjualan (Adithama et al., 2020)

Fungsi keanggotaan variable penjualan:

$$\mu_{\text{PenjualanSedikit}}(X) = \begin{cases} 1 & x \leq 0 \\ \frac{8-x}{8-0} & 0 < x < 8 \\ 0 & x \geq 8 \end{cases}$$

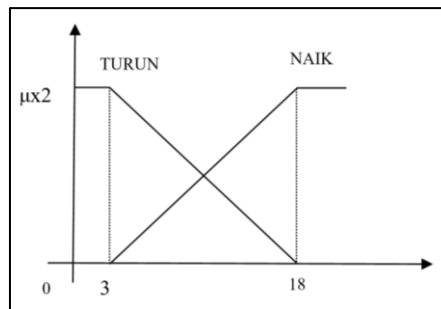
$$\mu_{\text{PenjualanBanyak}}(X) = \begin{cases} 1 & x \leq 0 \\ \frac{x-0}{8-0} & 0 < x < 8 \\ 0 & x \geq 8 \end{cases}$$

Nilai Keanggotaan:

$$\mu_{\text{PenjualanSedikit}}(3) = \frac{8-3}{8-0} = 0.625$$

$$\mu_{\text{PenjualanBanyak}}(3) = \frac{3-0}{8-0} = 0.375$$

b. Persediaan, terdiri atas dua himpunan *Fuzzy* yaitu TURUN dan NAIK



Gambar 2.4 Fungsi Keanggotaan Variable Persediaan (Adithama et al., 2020)

Fungsi keanggotaan variable persediaan:

$$\mu_{\text{PersediaanTurun}}(X) = \begin{cases} 1 & x \leq 3 \\ \frac{18-x}{18-3} & 3 < x < 18 \\ 0 & x \geq 18 \end{cases}$$

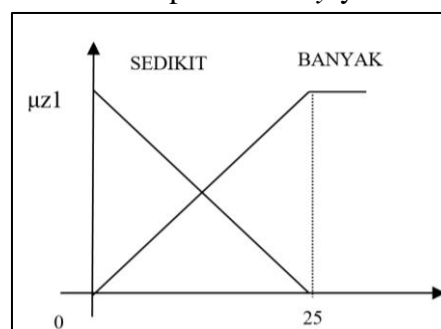
$$\mu_{\text{PersediaanNaik}}(X) = \begin{cases} 0 & x \leq 3 \\ \frac{x-3}{18-3} & 3 < x < 18 \\ 1 & x \geq 18 \end{cases}$$

Nilai Keanggotaan:

$$\mu_{\text{PersediaanTurun}}(15) = \frac{18-15}{18-3} = 0.2$$

$$\mu_{\text{PersediaanNaik}}(15) = \frac{15-3}{18-3} = 0.8$$

c. Pembelian, terdiri atas dua himpunan *Fuzzy* yaitu BANYAK dan SEDIKIT



Gambar 2.5 Fungsi Keanggotaan Variable Pembelian (Adithama et al., 2020)

Fungsi keanggotaan variable persediaan:

$$\mu_{\text{PembelianSedikit}}(z) = \begin{cases} 1 & z \leq 0 \\ \frac{25-z}{25-0} & 0 < z < 25 \\ 0 & z \geq 25 \end{cases}$$

$$\mu_{\text{PembelianBanyak}}(z) = \begin{cases} 0 & z \leq 0 \\ \frac{z-3}{25-3} & 0 < z < 25 \\ 1 & z \geq 25 \end{cases}$$

Selanjutnya proses infensi menggunakan fungsi implikasi MIN untuk menghitung keluaran hasil secara tegas masing-masing *rule*

[R1] Jika penjualan banyak dan persediaan turun, maka pembelian banyak

$$\min(0.375, 0.2) = 0.2 \Rightarrow 5$$

[R2] Jika penjualan banyak dan persediaan naik, maka pembelian sedikit.

$$\min(0.375, 0.8) = 0.375 \Rightarrow 15.625$$

[R3] Jika penjualan sedikit dan persediaan turun, maka pembelian sedikit

$$\min(0.625, 0.2) = 0.2 \Rightarrow 20$$

[R4] Jika penjualan sedikit dan persediaan naik, maka pembelian sedikit

$$\min(0.625, 0.8) = 0.625 \Rightarrow 9.375$$

Kemudian dilakukan proses *deFuzzyfikasi* menggunakan nilai rata-rata bobot sebagai berikut:

$$\frac{\alpha_{\text{predikat1}} \cdot Z1 + \alpha_{\text{predikat2}} \cdot Z2 + \alpha_{\text{predikat3}} \cdot Z3 + \alpha_{\text{predikat4}} \cdot Z4}{\alpha_{\text{predikat2}} + \alpha_{\text{predikat3}} + \alpha_{\text{predikat4}}}$$

$$\frac{(0,2.5) + (0,625.9,375) + (0,375.15,625) + (0,2.20)}{(0,2 + 0,625 + 0,375 + 0,2)} = 11,90121$$

Maka hasil rekomendasi pada bulan desember yang diperoleh adalah 11,90121 kardus yang dibulatkan menjadi 12 kardus perhari.

2.8.2 Penelitian Terkait

Pada penelitian yang dilakukan oleh (Novianti et al., 2018) terkait prediksi pemesanan bahan baku menggunakan metode *Fuzzy Tsukamoto* diketahui lebih optimal dalam menghasilkan prediksi jumlah pemesanan, sehingga dapat meminimalkan terjadinya kekurangan maupun kelebihan barang di gudang. Penelitian lain juga dilakukan oleh (Beu & Husna, 2019) terkait prediksi jumlah produksi dengan menggunakan algoritma *Fuzzy Tsukamoto* menghasilkan akurasi sebanyak 70% yang diketahui dapat meminimalisir biaya produksi.

Perbandingan penelitian juga telah dilakukan oleh (Agus, Fahrul ; et al., 2016) antara metode *Tsukamoto* dan *Sugeno* untuk optimasi pengadaan barang yang dilakukan dengan membandingkan rata-rata kesalahan relatif pada hasil perhitungan pengadaan barang yang dilakukan pada suatu perusahaan yang sama dan diperoleh nilai rata-rata kesalahan metode *Tsukamoto* lebih rendah yaitu 39,67% dibandingkan dengan metode *Sugeno* dengan kesalahan sebesar 50,39%.

2.8.3 Pengujian MAPE

Pengujian *Mean Absolute Percentage Error* atau MAPE merupakan salah satu metode pengujian yang digunakan untuk menghitung kesalahan prediksi *forces error* dengan cara membandingkan hasil perkiraan dengan kenyataan yang terjadi. Metode MAPE menghitung rata-rata *diferensiasi absolute* antara nilai hasil peramalan dengan data aktual. Hasil perhitungan ini akan dinyatakan dalam bentuk persentase nilai aktual. MAPE mengindikasikan seberapa besar nilai kesalahan atau ketidaksesuaian nilai peramalan yang dibandingkan nilai nyata (Sinaga & Irwanti, 2018). Berikut merupakan rumus yang digunakan untuk mencari nilai MAPE:

$$MAPE = \frac{100}{n} \sum \frac{|At - Ft|}{At} \dots\dots\dots(2.5)$$

Dimana:

n = jumlah periode data

At = nilai aktual pada data t

Ft = nilai prediksi pada data t

Hasil dari perhitungan nilai MAPE yang diperoleh dapat menunjukkan kemampuan mengenai teknik atau algoritma yang digunakan. Semakin rendah nilai MAPE maka algoritma atau teknik yang digunakan memiliki kemampuan yang semakin baik dalam melakukan peramalan. Metode MAPE memiliki kriteria untuk menentukan bahwa suatu teknik peramalan bernilai baik atau buruk (Puspitasari, 2019). Berikut merupakan tabel kriteria nilai MAPE dapat dilihat pada tabel 2.5 (Puspitasari et al., 2019).

Tabel 2.5 Tabel Kriteria Nilai MAPE (Puspitasari, 2019)

Nilai MAPE	Kriteria
<10%	Sangat Baik
10-20%	Baik
20-50%	Layak
>50%	Buruk

Diketahui bahwa jika nilai MAPE bernilai kurang dari 10% maka dapat dikatakan baik, jika berada antara 10-20% maka dikatakan baik, 20-50% dikatakan layak dan jika nilai lebih dari 50% maka dikatakan buruk

2.9 Penelitian Terkait

Tabel 2.5 menjelaskan mengenai penelitian-penelitian yang telah dilakukan sebelumnya terkait dengan algoritma, metode maupun model perancangan yang akan di susun.

Tabel 2.5 Rangkuman Penelitian Terkait

No.	Penulis	Judul Penelitian	Intisari
1.	Yosefa et al., 2015 Jurnal Rekayasa Sistem Industri	Perancangan Model VMI (<i>Vendor Managed Inventory</i>) dengan Satu Pemasok dan Banyak <i>Retailer</i> yang Meminimasi Ongkos Total Rantai Pasok	SCM Model VMI dengan satu pemasok dan banyak <i>retailer</i> mampu mengurangi biaya rantai pasokan dengan melakukan analisis dan koordinasi antara distributor dengan pedagang <i>retail</i> . Solusi yang dihasilkan lebih baik dibandingkan oleh <i>Q System</i> maupun <i>Wagner ithin Algorithm</i> .
2.	Karma 2018	Perancangan Sistem Informasi Manajemen pada Industri Farmasi dengan Pendekatan <i>Vendor Managed Inventory</i> (VMI) Berbasis <i>Cloud Computing</i>	Menghasilkan kolaborasi dan koordinasi yang baik antara distributor farmasi dan <i>retailer</i> selain itu pendekatan ini dapat mengatasi masalah redudansi data, gap informasi, pemrosesan yang lama serta mempermudah akses informasi dan data.
3.	Negara, Iis Setiawan Mangku; et al., 2022 (JOINTECS) Journal of Information	Analisa <i>Cluster</i> Data Transaksi Penjualan Minimarket Selama Pandemi Covid-19 dengan <i>Algoritma K-means</i>	Penelitian ini dilakukan dengan mengelompokkan data menjadi tiga <i>cluster</i> menggunakan <i>algoritma K-Means</i> yang menghasilkan akurasi dengan <i>confusius matrix</i> sebanyak 87%.

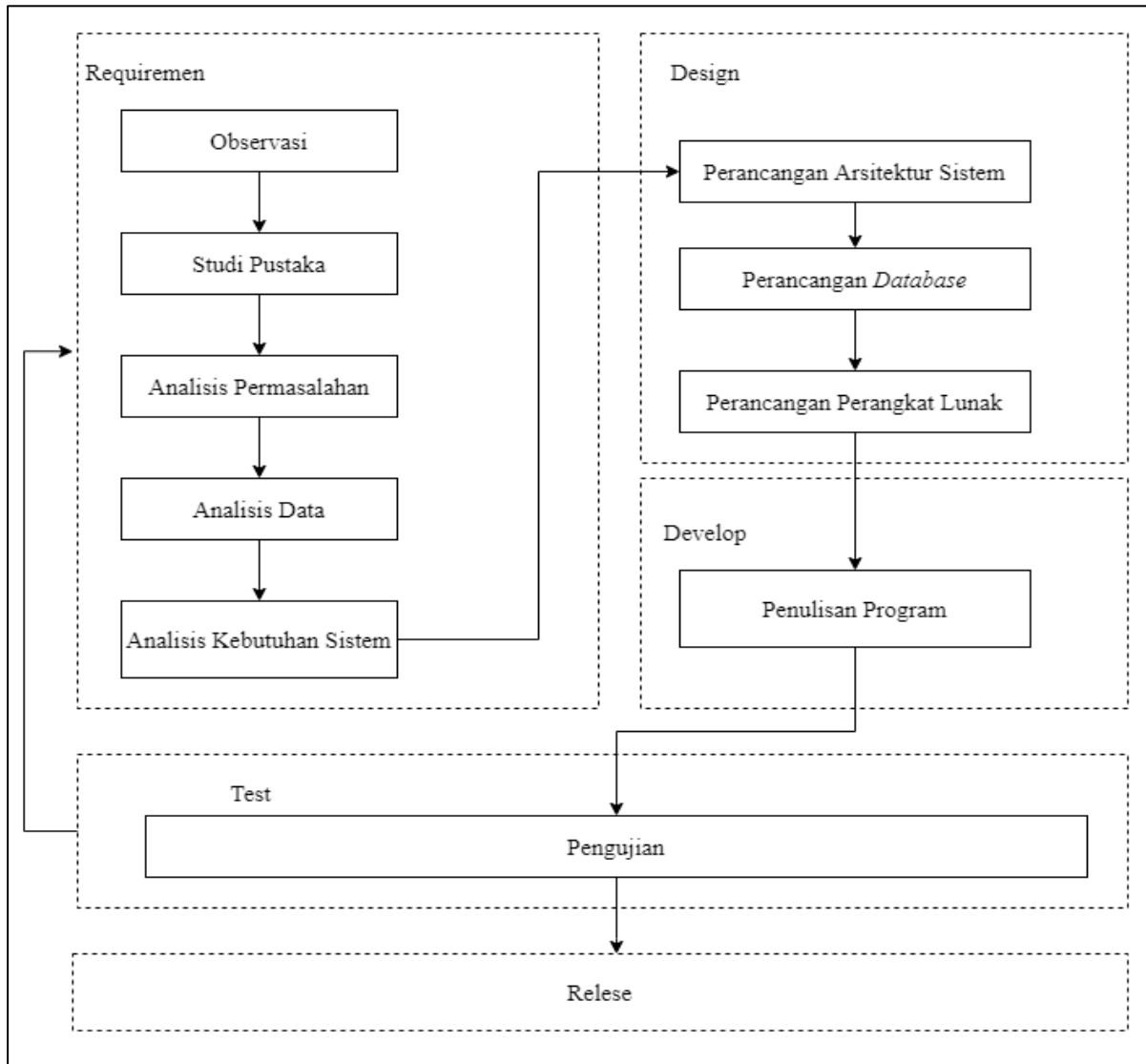
Tabel 2.6 Lanjutan Rangkuman Penelitian Terkait

	Technology and Computer Science		
4.	Herianto; at al., 2021 Jurnal Sains dan Teknologi	Analisa Tingkat Penjualan Produk Menggunakan <i>K-Nearest Neighbor (K-NN)</i> Dan <i>K-Means</i> (Studi Kasus Perusahaan Kayu Elang Perkasa)	Pada penelitian ini diterapkan perbandingan kedua metode dengan melakukan analisis <i>clustering</i> dengan menggunakan data penjualan yang sama dan diperoleh hasil bahwa tingkat akurasi <i>k-means</i> lebih tinggi.
5.	Beu & Husna, 2019 Jurnal Nasional cosPhi	Metode <i>Fuzzy Tsukamoto</i> Untuk Memprediksi Jumlah Produksi Kue Pia	Penelitian ini menerapkan <i>algoritma Fuzzy Tsukamoto</i> untuk prediksi jumlah produksi kue pia berdasarkan jumlah penjualan dan permintaan periode sebelumnya dan menghasilkan akurasi sebanyak 70% yang diketahui dapat meminimalisir biaya produksi.
6.	Agus, Fahrul; et al., 2016 International Journal of Computing and Informatics (IJCANDI)	<i>Comparative Analysis of Tsukamoto and Sugeno Fuzzy Methods For Procurement Optimization</i>	Pada penelitian ini dilakukan perbandingan antara metode <i>Tsukamoto</i> dan <i>Sugeno</i> untuk optimasi pengadaan barang yang dengan membandingkan rata-rata kesalahan relatif pada hasil perhitungan pengadaan barang yang dilakukan pada suatu perusahaan yang sama dan diperoleh nilai rata-rata kesalahan metode <i>Tsukamoto</i> lebih rendah yaitu 39,67% dibandingkan dengan metode <i>Sugeno</i> dengan kesalahan sebesar 50,39%.
7.	Zalina, Debby; 2021 Tesis Universitas Brawijaya	Perancangan Sistem Informasi Untuk Mengintegrasikan Informasi Antara Produsen Dengan Pemasok Dan Distributor	Penelitian ini dilakukan dengan melakukan perancangan terintegrasi dari setiap elemen yang terkait dengan mengutamakan kegiatan kolaborasi dan koordinasi sehingga dapat digunakan untuk mengelola suatu proses bisnis secara lebih efektif dan efisien.

BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini dilakukan dengan menggunakan metodologi penelitian kuantitatif, menggunakan metodologi pengembangan sistem *Agile Development*. Metode ini bersifat adaptif sehingga pada pengerjaanya dapat ditinjau dan dilakukan revisi serta pengembangan lebih lanjut terhadap suatu sistem informasi tanpa adanya batasan-batasan tertentu ketika ingin menambahkan ide baru di masa depan. (Hutauruk & Pakpahan, 2021) Metode ini menyediakan beberapa tahapan seperti pada Gambar 3.1.



Gambar 3.1 Kerangka Kerja Pengembangan Sistem

3.2 Requirement

Requirement atau pengumpulan kebutuhan merupakan tahapan yang bertujuan untuk mengetahui dan memahami berbagai macam kebutuhan suatu sistem serta untuk mendapatkan data yang lengkap terkait kebutuhan sistem maupun kebutuhan pengguna akan sistem yang akan dikembangkan (Hutauruk & Pakpahan, 2021). Dalam penelitian ini proses pengumpulan kebutuhan dibagi menjadi lima tahapan diantaranya studi pustaka, observasi, analisis permasalahan, analisis data, serta analisis kebutuhan.

3.2.1 Studi Pustaka

Pada tahap ini dilakukan studi terkait jurnal, penelitian dan buku yang berhubungan dengan SCM, VMI, algoritma *clustering* dan pendukung keputusan yang diterapkan dalam berbagai studi kasus. Dilakukan studi mengenai perbandingan metode-metode yang akan digunakan melalui jurnal dan penelitian yang pernah dilakukan sebelumnya sebagai bahan pertimbangan untuk menentukan metode yang akan digunakan. Serta mempelajari mengenai teori-teori pemrograman melalui materi kuliah, buku, jurnal maupun literatur lain yang dapat dijadikan referensi.

3.2.2 Observasi

Proses observasi dilakukan dengan cara pengamatan langsung serta wawancara terhadap objek yang akan diteliti. Proses ini bertujuan untuk mendapatkan informasi lengkap serta gambaran terkait penelitian yang akan dilakukan (Mahendra & Yanto, 2018). Pada penelitian ini observasi dilakukan pada sejumlah *retailer* kecil yang berada di kecamatan Selo, kabupaten Boyolali dengan dilakukannya pengamatan, wawancara terkait proses transaksi, proses pendataan produk, proses pengadaan barang mulai dari penentuan jenis barang, penentuan harga, jangka waktu pengadaan barang hingga proses kredit maupun pembayaran yang dilakukan.

Selain itu juga dilakukan wawancara terkait teknologi atau sistem informasi yang telah digunakan yang mana 8 dari 10 *retailer* masih menggunakan cara manual dan dua diantaranya telah menggunakan sistem POS atau *Point of Sale*. Dari 10 *retailer* yang diteliti menyatakan respon positif mengenai penggunaan sistem informasi penjualan seperti sistem yang akan dikembangkan pada penelitian ini.

Selain melakukan pengamatan dan wawancara dalam proses ini juga dilakukan pengambilan data yang dibutuhkan untuk penelitian diantaranya data penjualan, data produk terkait harga, stok dan sebagainya juga data pengadaan barang dalam jangka waktu tiga minggu.

Observasi juga dilakukan pada salah satu distributor dengan konsep *warehouse* di daerah Yogyakarta. Yang mana distributor ini bertugas sebagai pengepul berbagai macam barang atau produk dari berbagai macam perusahaan atau pabrik yang kemudian didistribusikan kepada toko-toko *retail* tradisional. Proses ini dilakukan dengan wawancara dan pengamatan langsung mengenai proses transaksi dan pengiriman barang.

3.2.3 Analisis Permasalahan

Di Indonesia *retail* atau pedagang eceran memiliki kontribusi yang sangat besar dalam bidang perdagangan yaitu 19-31% terhadap Pendapatan Domestik Regional Bruto (PDRB) tiap daerahnya. Peningkatan pertumbuhan *retail* modern sangat pesat sebesar 13% yang mana berbanding terbalik dengan *retail* tradisional yang tidak mengalami peningkatan (Biro Perekonomian DKI Jakarta, 2015). Meningkatnya keberadaan *retail* modern berpengaruh signifikan terhadap kinerja *retail* tradisional yang diwakili proksi pendekatan jumlah pedagang, atau dapat dikatakan bahwa meningkatnya *retail* modern menjadi salah satu faktor penyebab menurunnya jumlah *retail* tradisional hingga *retail* tradisional perlahan mulai ditinggalkan (Hikmawati & Nuryakin, 2018).

Pada hasil observasi yang dilakukan terhadap pedagang *retail* di daerah Selo, Boyolali diketahui persaingan yang cukup ketat antara pedagang *retail* kecil atau perseorangan dengan *retail* modern yang dimiliki oleh perusahaan besar dengan konsep *franchise* atau waralaba

seperti *Alfamart*, *Indomaret* dan sebagainya. Dikuasainya bisnis *retail* modern oleh suatu perusahaan tertentu mengakibatkan persaingan dan monopoli yang menimbulkan dampak negatif bagi pengusaha lokal yang berada di daerah tersebut (Jumaidi, Jalaludin, & Ahyar, 2019).

Selain itu, masalah lain juga terjadi dikarenakan penumpukan barang yang tidak terjual, tidak terdatanya barang dengan baik, tidak adanya informasi yang jelas mengenai data barang yang tersedia, serta tidak adanya dokumentasi mengenai data penjualan. Hal ini juga menimbulkan dampak negatif bagi distributor yang mana sulitnya melakukan analisis terhadap barang yang mereka distribusikan juga penumpukan barang yang tidak terjual pada suatu *retail* tertentu mengakibatkan melambatnya aliran penjualan barang dan tidak jarang barang akan kadaluarsa sebelum sampai ketangan konsumen hal ini mengakibatkan kerugian karena distributor harus menarik kembali barang yang telah didistribusikan sehingga menjadi terbuang. Selain itu, proses penjualan atau aliran barang yang lambat juga dapat memicu menurunnya minat konsumen karena barang yang telah ketinggalan jaman.

Oleh karena itu, dalam penelitian ini akan dilakukan pengembangan sistem informasi yang terintegrasi antara distributor dan pedagang *retail* dengan konsep VMI yang mana pihak pemasok atau distributor memonitor dan bertanggung jawab atas persediaan yang ada pada *retail*. Dalam model VMI, pihak *retail* tidak lagi memutuskan apa, kapan, dan bagaimana barang akan dibeli dari distributor hanya memberikan informasi tentang permintaan pelanggan dan sisa persediaan serta informasi mengenai persediaan minimum dan maksimum yang diharapkan. Informasi ini berguna bagi pihak distributor untuk memutuskan waktu dan jumlah penambahan persediaan (Anna, 2016).

Proses ini memerlukan *database* terpusat yang dapat diakses oleh kedua belah pihak dengan perijinan yang berbeda sesuai dengan peranan masing-masing. Selain itu, proses analisis juga diperlukan untuk mengetahui mana barang yang laris, maupun kurang laris, serta rekomendasi jumlah pengadaan barang yang akan dilakukan oleh distributor. Penerapan sistem informasi *web* terpusat yang terintegrasi memudahkan akses secara terpusat bagi kedua belah pihak yang akan membantu terdistribusinya data secara *real-time*.

3.2.4 Analisis Data

Data yang diperoleh dalam penelitian ini merupakan data primer yang diperoleh langsung dari objek penelitian melalui pencatatan manual terkait data produk, data penyetokan, data pemasok, data *retailer*, serta data transaksi. Dengan rincian sebagai berikut.

Tabel 3.1 Tabel Data Primer

NO	Data (Tabel)	Field
1.	Produk	<ol style="list-style-type: none"> 1. Kode produk 2. Nama Produk 3. Harga 4. Stok 5. Tanggal Kadaluarsa
2.	Penyetokan	<ol style="list-style-type: none"> 1. Kode Stok 2. Pemasok 3. Kode Produk 4. Qnt 5. Harga 6. Tanggal

Tabel 3.2 Lanjutan Tabel Data Primer

No	Data (Tabel)	Field
3.	Pemasok	1. Nama pemasok 2. Alamat 3. No Telephone
4.	<i>Retailer</i>	1. Nama pemilik 2. Nama toko 3. No Telephone 4. Alamat
5.	Transaksi	1. Kode produk 2. Qnt 3. Tanggal transaksi 4. Total harga

3.2.5 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem digunakan untuk mengetahui perangkat yang dibutuhkan untuk dalam pengerjaan sistem yang ingin diterapkan. Dalam penelitian ini melibatkan beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*) diantaranya:

Perangkat keras yang memiliki spesifikasi:

1. *Processor* AMD Athlon Silver 3050U
2. RAM 8 GB
3. SSD Memory 250 GB
4. Layar dengan resolusi 1925 × 1080 pixel
5. *Mouse* dan *keyboard*
6. Koneksi internet

Perangkat lunak yang digunakan adalah sebagai berikut:

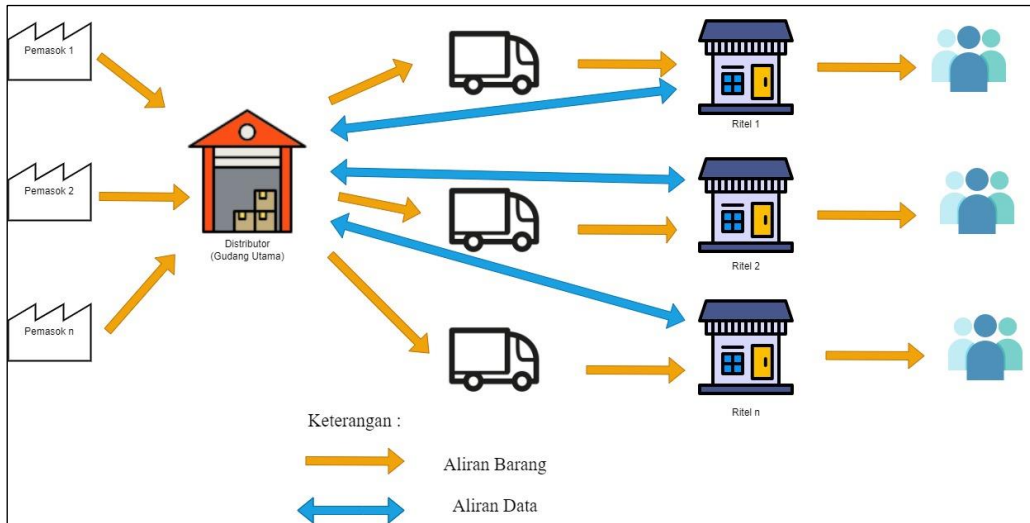
1. Sistem Operasi *Microsoft Windows* 10 Pro
2. XAMPP Control Panel v3.2.4
3. Sublime Text 3
4. Bahasa PHP

3.3 Proses Desain

Proses desain adalah proses perancangan dari sistem yang akan dibangun sesuai dengan data yang telah diperoleh secara mendetail dan terperinci mulai dari tahapan perancangan arsitektur sistem, perancangan *database*, hingga perancangan perangkat lunak dengan tujuan memberikan gambaran mengenai apa yang akan dikerjakan.

3.3.1 Perancangan Arsitektur

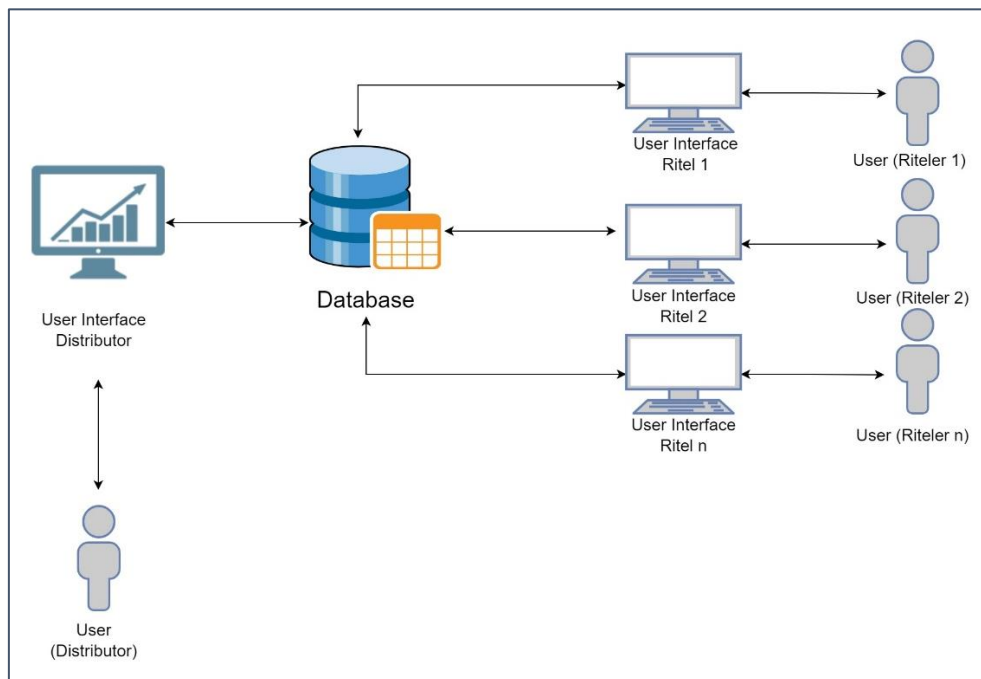
Sistem yang akan dikembangkan menerapkan model VMI dengan satu pemasok banyak *retailer*. Yang mana pihak pemasok atau distributor ini memiliki peranan sebagai gudang pusat yang menghimpun barang dari multi-manufaktur atau pemasok dan mendistribusikan kepada pihak *retail*. Pihak distributor memiliki wewenang untuk pengambilan keputusan bagi kedua belah pihak. Keputusan ini berupa jumlah order atau stok yang ditetapkan kepada masing-masing *retail* dengan mempertimbangkan rekomendasi melalui data penjualan yang dikirimkan oleh masing-masing pihak *retail*. Ilustrasi perancangan model VMI satu pemasok banyak *retailer* yang akan diterapkan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur Sistem Model VMI Satu Pemasok Multi Retailer

Sistem ini melibatkan dua macam *user* yaitu pihak distributor dan pihak *retail*. Pihak distributor memiliki perijinan untuk mengakses data *retail* yang telah ikut tergabung. Dengan tujuan untuk melakukan analisis penjualan untuk menentukan kebijakan penyetokan barang pada periode selanjutnya. *Database* terpusat digunakan untuk memudahkan pemeliharaan dan pembaharuan serta untuk menghindari duplikasi dan mempermudah untuk menjaga integrasi data yang disimpan pada satu lokasi. *Database* ini dapat diakses baik oleh distributor maupun pihak *retail* sesuai dengan perijinan masing-masing.

Pihak *retail* hanya dapat mengakses data produk dari tokonya masing-masing untuk memudahkan proses transaksi. Sementara pihak distributor dapat mengakses seluruh data termasuk data transaksi pada setiap *retail* untuk keperluan analisis data. Data transaksi ini yang kemudian akan diolah menggunakan algoritma *K-Means* untuk menentukan *clustering* atau kategori barang yang laris cukup laris dan kurang laris. Selanjutnya data diolah menggunakan algoritma *Fuzzy Tsukamoto* untuk menentukan rekomendasi jumlah stok yang akan dikirimkan untuk periode selanjutnya. Ilustrasi sistem dapat dilihat pada Gambar 3.3.



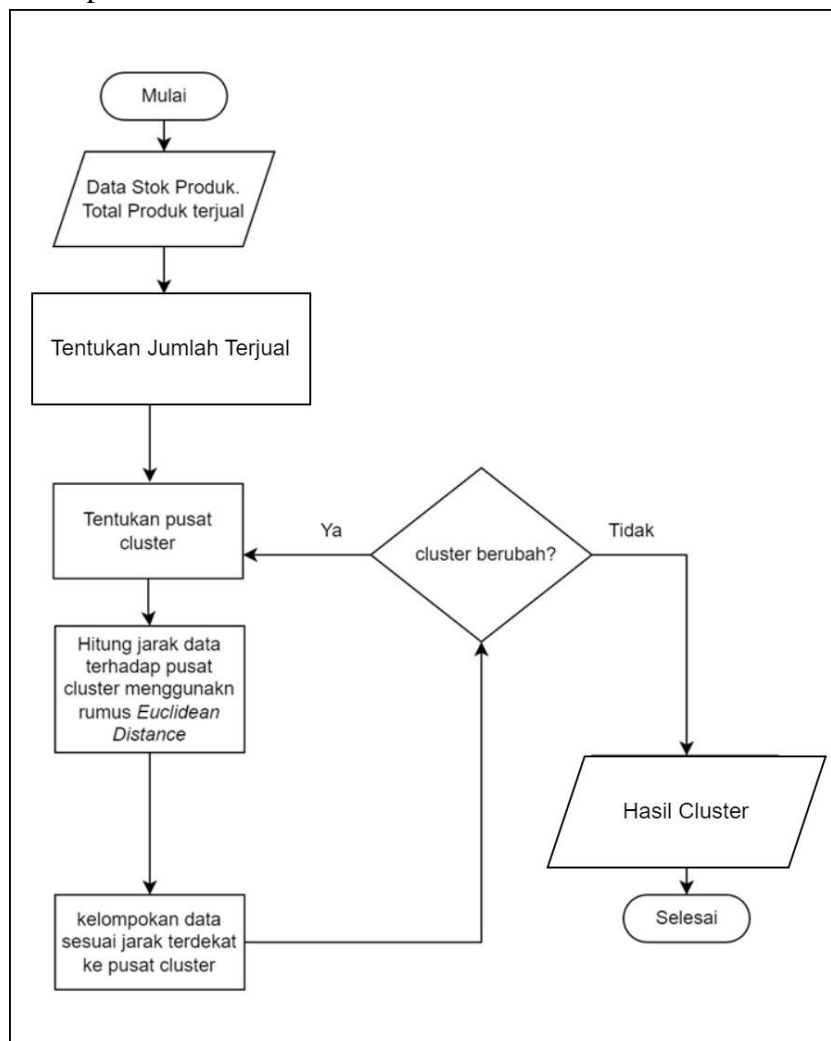
Gambar 3.3 Arsitektur Sistem

3.3.1.1 Flowchart

Flowchart menggambarkan urutan proses atau langkah-langkah secara mendetail yang mewakili ilustrasi penyelesaian suatu masalah dalam suatu program. Berikut merupakan *flowchart* dari algoritma yang digunakan untuk perhitungan atau proses penentuan kategori dan rekomendasi pengiriman barang.

1. Proses Penentuan Kelompok Cluster Produk

Proses penentuan kelompok cluster produk dilakukan dengan menggunakan algoritma *K-Means* dengan membaginya menjadi tiga *cluster* C1 yang berarti laris C2 cukup laris dan C3 kurang laris. Berikut merupakan *flowchart* pembagian kelompok cluster dengan algoritma *K-Means* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Penentuan Kelompok Cluster dengan Algoritma *K-Means*

Gambar 3.4 merupakan *flowchart* pembagian kelompok *cluster* produk dengan menggunakan algoritma *K-Means*. Langkah pertama adalah dengan melakukan pengambilan data penjualan, data yang akan digunakan dalam algoritma ini adalah data stok masuk dan data produk terjual. Data ini diambil dari data transaksi tiap-tiap toko yang akan dikelompokkan berdasarkan kode produk dalam kurun waktu satu minggu. Hal ini berguna untuk mengelompokkan data suatu produk yang sama pada toko yang berbeda. Berikut merupakan contoh data yang akan digunakan untuk kalkulasi.

Tabel 3.3 Tabel Data Penjualan Produk Aqua

No	Nama Toko	StokIn (x)	Terjual (y)
1	Toko A	192	124
2	Toko B	144	28
3	Toko C	96	68
4	Took D	144	46
5	Toko E	144	82
6	Toko F	48	42
7	Toko G	96	47
8	Toko H	96	31

Langkah pertama adalah penentuan jumlah *cluster* (k). dalam contoh kasus ini penentuan *cluster* dipilih sebanyak tiga *cluster*. Dimana *cluster* 1 adalah barang yang laris, *cluster* 2 cukup laris dan *cluster* 3 adalah barang yang kurang laris. *Cluster*1 dipilih secara acak dari data yang telah tersedia. Berikut Tabel 3.4 merupakan tabel *cluster* awal yang telah ditentukan.

Tabel 3.4 Titik Awal Cluster

Titik Pusat Awal	StokIn(x)	Terjual (y)
<i>Cluster 1</i>	48	42
<i>Cluster 2</i>	144	82
<i>Cluster 3</i>	144	28

Selanjutnya adalah penentuan titik pusat atau *centroid* serta jarak terdekat dengan menggunakan rumus perhitungan *Euclidean Distance* sebagai berikut:

Jarak data ke-1 dengan pusat *cluster*:

$$\begin{aligned}
 C(1) &= \sqrt{(192 - 48)^2 + (124 - 42)^2} \\
 &= \sqrt{20.736 + 6.724} \\
 &= \sqrt{27.460} \\
 &= 165,71
 \end{aligned}$$

$$\begin{aligned}
 C(2) &= \sqrt{(192 - 144)^2 + (124 - 82)^2} \\
 &= \sqrt{2.304 + 1.476} \\
 &= \sqrt{3.780} \\
 &= 61,48
 \end{aligned}$$

$$\begin{aligned}
 C(3) &= \sqrt{(192 - 144)^2 + (144 - 28)^2} \\
 &= \sqrt{2.304 + 13.456} \\
 &= \sqrt{15760} \\
 &= 125,53
 \end{aligned}$$

Hasil yang diperoleh dari perhitungan di atas merupakan jarak data ke-1 dengan titik pusat awal *cluster*. Dimana jarak data ke-1 dengan pusat *cluster* 1 adalah 165,71. Jarak *cluster* 2 adalah 78,41 sementara jarak dengan *cluster* 3 adalah 12.553. Begitu pula dengan data selanjutnya dihitung menggunakan rumus yang sama. Setelah semua data dihitung maka akan diperoleh jarak masing-masing data dengan *cluster* 1 (C1), *cluster* 2 (C2) dan *cluster* 3 (C3) dapat dilihat pada Tabel 3.5

Tabel 3.5 Tabel Jarak Item (M) ke C1 dan C2

	M1	M2	M3	M4	M5	M6	M7	M8
C1	165,71	96,5	54,6	96	104	0	48,2	49,2
C2	61,48	54	50	36	0	104	59,4	70
C3	125,53	0	62,5	18	54	97,0	51,6	48,1

Langkah selanjutnya adalah membandingkan jarak dari masing-masing *item* dengan C1, C2 dan C3 yang mana nilai yang lebih kecil berarti jarak yang lebih dekat. Kemudian dikelompokkan sesuai dengan jarak terdekat misalnya M1 memiliki jarak yang lebih dekat dengan C1 maka M1 adalah anggota C1 dan seterusnya. Berikut hasil pengelompokan data dapat dilihat pada Tabel 3.6.

Tabel 3.6 Tabel Kelompok Data dengan Masing-masing Cluster

	Anggota
C1	M6, M7
C2	M1, M3, M5
C3	M2, M4, M8

Setelah anggota dari tiap-tiap *cluster* diketahui maka langkah selanjutnya ditentukan pusat *cluster* atau *centroid* baru dengan perhitungan sebagai berikut.

C1 Baru:

$$\frac{M6x + M7x}{2}; \frac{M6y + M7y}{2}$$

$$\frac{48 + 96}{2}; \frac{42 + 47}{2} = 72; 44,5$$

C2 Baru:

$$\frac{M1x + M3x + M5x}{3}; \frac{M1y + M3y + M5y}{3}$$

$$\frac{192 + 96 + 144}{3}; \frac{124 + 68 + 82}{3} = 144; 91,33$$

C3 Baru:

$$\frac{M2x + M4x + M8x}{3}; \frac{M2y + M4y + M8y}{3}$$

$$\frac{144 + 144 + 96}{3}; \frac{28 + 46 + 31}{3} = 128; 35$$

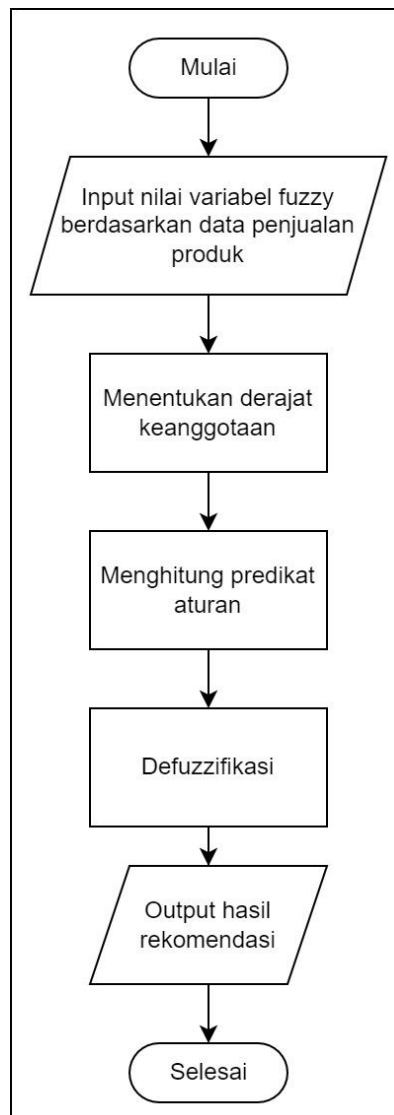
Langkah selanjutnya adalah menghitung kembali jarak masing-masing *item* (M) dengan C1, C2 dan C3 yang baru menggunakan rumus *Euclidean Distance* sama seperti sebelumnya. Kemudian bandingkan jarak antara M dengan C1, C2 dan C3 yang baru dan kelompokkan seperti langkah sebelumnya. Apabila terjadi perubahan kelompok atau *cluster* maka iterasi akan terus dilakukan hingga tidak terjadi perubahan pada data kelompok atau tidak ada perubahan nilai *centroid*. Jika sudah tidak ada perubahan maka iterasi selesai dan menghasilkan output berupa hasil pengelompokan *cluster* produk. Berikut merupakan hasil akhir yang diperoleh dapat dilihat pada Tabel 3.7.

Tabel 3.7 Tabel Data Penjualan Produk Aqua

No	Nama Toko	StokIn (x)	Terjual (y)	Cluster
1	Toko A	192	124	C2
2	Toko B	144	28	C3
3	Toko C	96	68	C1
4	Took D	144	46	C3
5	Toko E	144	82	C2
6	Toko F	48	42	C1
7	Toko G	96	47	C1
8	Toko H	96	31	C

2. Proses Penentuan Rekomendasi Penyetokan Produk

Proses penentuan rekomendasi penyetokan produk dilakukan dengan menggunakan algoritma *Fuzzy Tsukamoto*. Berikut merupakan *flowchart* yang algoritma yang diterapkan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Rekomendasi Penyetokan Produk dengan Algoritma *Fuzzy Tsukamoto*

Gambar 3.5 merupakan *flowchart* penentuan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto*. Data yang akan digunakan dalam contoh kalkulasi ini adalah data penjualan produk gula pasir 1 kg dengan kode produk 10020031 pada tanggal 01 Januari 2022 hingga 05 Februari 2022 dengan stok awal sebesar 38 buah. Berikut detail data dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Tabel Data Penjualan Produk Gula Pasir

No	Tanggal Mingguan	Jumlah Restok	Jumlah Transaksi	Jumlah Stok
1	01/01/2022	46	74	10
2	08/01/2022	40	42	8
3	15/01/2022	30	32	6
4	22/01/2022	40	35	11
5	29/01/2022	60	55	16
6	05/02/2022	74	68	22

Langkah pertama adalah dengan mengambil nilai masukan produk yang sesuai dengan variabel *fuzzy* yang akan di gunakan, diantaranya:

- Penjualan terbanyak dalam satu minggu
- Penjualan terkecil dalam satu minggu
- Persediaan terbanyak dalam satu minggu
- Persediaan terkecil dalam satu minggu
- Pembelian terbanyak
- Pembelian terkecil
- Persediaan saat ini
- Penjualan rata-rata per minggu

Data dari variabel ini akan ditampung dalam *view table Fuzzy* dengan atribut penjualan, stok (diambil dari nilai restok – penjualan), restok dan tanggal yang dikelompokan berdasarkan waktu transaksi perhari. Dalam bab ini akan digunakan contoh perhitungan untuk penjualan salah satu produk sebagai berikut:

Tabel 3.9 Tabel Variabel *Fuzzy* Data Penjualan Produk Gula Pasir

Kode Produk	Min Penjualan	Max Penjualan	Min Persediaan	Max Persediaan	Min Restok	Max Restok	stok	Rata-rata Penjualan
100203	32	74	6	38	30	74	22	51

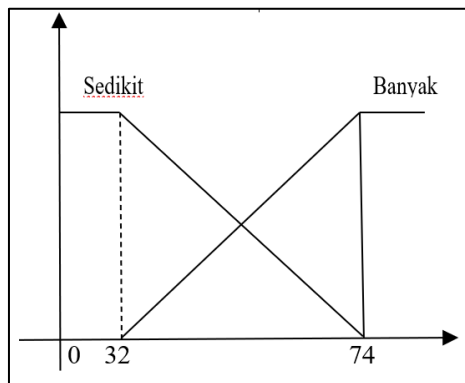
Langkah pertama adalah dengan melakukan pembentukan basis pengetahuan atau *rule*.

Rule yang ditetapkan adalah sebagai berikut:

- [R1] Jika penjualan banyak dan persediaan turun, maka pembelian banyak.
- [R2] Jika penjualan banyak dan persediaan naik, maka pembelian banyak.
- [R3] Jika penjualan sedikit dan persediaan turun, maka pembelian sedikit.
- [R4] Jika penjualan sedikit dan persediaan naik, maka pembelian sedikit.

Kemudian melakukan pemodelan fungsi keanggotaan variable *Fuzzy* yang meliputi penjualan, persediaan, dan pembelian sebagai berikut.

- Penjualan, terdiri atas dua himpunan *Fuzzy* yaitu BANYAK dan SEDIKIT yang dapat dilihat pada ilustrasi Gambar 3.6.



Gambar 3.6 Fungsi Keanggotaan Variable Penjualan

Fungsi keanggotaan variable penjualan:

$$\mu_{\text{PenjualanSedikit}}(X) = \begin{cases} 1 & x \leq 32 \\ \frac{74-x}{74-32} & 32 < x < 74 \\ 0 & x \geq 74 \end{cases}$$

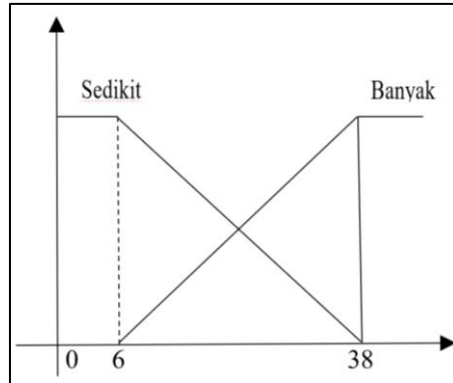
$$\mu_{\text{PenjualanBanyak}}(X) = \begin{cases} 0 & x \leq 32 \\ \frac{x-32}{74-32} & 32 < x < 74 \\ 1 & x \geq 74 \end{cases}$$

Nilai Keanggotaan:

$$\mu_{\text{PenjualanSedikit}}(51) = \frac{74 - 51}{74 - 32} = 0.5476$$

$$\mu_{\text{PenjualanBanyak}}(51) = \frac{51 - 32}{74 - 32} = 0.4524$$

- b. Persediaan, terdiri atas dua himpunan *Fuzzy* yaitu TURUN dan NAIK yang dapat dilihat pada ilustrasi Gambar 3.7.



Gambar 3.7 Fungsi Keanggotaan Variable Persediaan

Fungsi keanggotaan variable persediaan:

$$\mu_{\text{PersediaanTurun}}(X) = \begin{cases} 1 & x \leq 6 \\ \frac{38-x}{38-6} & 6 < x < 38 \\ 0 & x \geq 38 \end{cases}$$

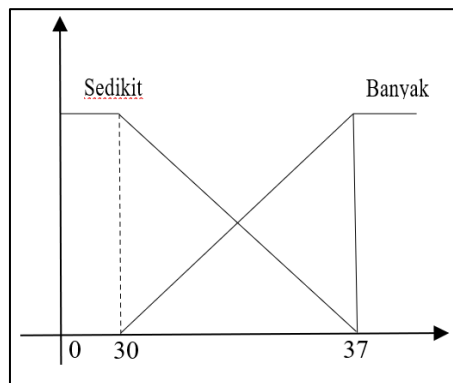
$$\mu_{\text{PersediaanNaik}}(X) = \begin{cases} 0 & x \leq 6 \\ \frac{x-6}{38-6} & 6 < x < 38 \\ 1 & x \geq 38 \end{cases}$$

Nilai Keanggotaan:

$$\mu_{\text{PersediaanTurun}}(22) = \frac{38 - 22}{38 - 6} = 0.5$$

$$\mu_{\text{PersediaanNaik}}(22) = \frac{22 - 6}{38 - 6} = 0.5$$

- c. Pembelian, terdiri atas dua himpunan *Fuzzy* yaitu BANYAK dan SEDIKIT yang dapat dilihat pada ilustrasi Gambar 3.8.



Gambar 3.8 Fungsi Keanggotaan Variable Pembelian

Fungsi keanggotaan variable persediaan:

$$\mu_{\text{PembelianSedikit}}(z) = \begin{cases} 1 & z \leq 30 \\ \frac{74-z}{74-30} & 30 < z < 74 \\ 30 & z \geq 74 \end{cases}$$

$$\mu_{\text{PembelianBanyak}}(z) = \begin{cases} 0 & z \leq 30 \\ \frac{z-30}{74-30} & 30 < z < 74 \\ 1 & z \geq 74 \end{cases}$$

Selanjutnya proses infensi menggunakan fungsi implikasi MIN untuk menghitung keluaran hasil secara tegas masing-masing *rule*

[R1] Jika penjualan banyak dan persediaan turun, maka pembelian banyak

$$\min(0.452, 0.5) = 0.452 \Rightarrow 49.90$$

[R2] Jika penjualan banyak dan persediaan naik, maka pembelian banyak.

$$\min(0.452, 0.5) = 0.452 \Rightarrow 49.90$$

[R3] Jika penjualan sedikit dan persediaan turun, maka pembelian sedikit

$$\min(0.548, 0.5) = 0.5 \Rightarrow 52$$

[R4] Jika penjualan sedikit dan persediaan naik, maka pembelian sedikit

$$\min(0.548, 0.5) = 0.5 \Rightarrow 52$$

Kemudian dilakukan proses *defuzzyfikasi* menggunakan nilai rata-rata bobot sebagai berikut :

$$\frac{\alpha_{\text{predikat1}} \cdot Z1 + \alpha_{\text{predikat2}} \cdot Z2 + \alpha_{\text{predikat3}} \cdot Z3 + \alpha_{\text{predikat4}} \cdot Z4}{\alpha_{\text{predikat2}} + \alpha_{\text{predikat3}} + \alpha_{\text{predikat4}}}$$

$$\frac{(0,452 \cdot 49,90) + (0,452 \cdot 49,90) + (0,5 \cdot 52) + (0,5 \cdot 52)}{(0,452 + 0,452 + 0,5 + 0,5)} = 51,025$$

Maka hasil rekomendasi pada minggu selanjutnya yang diperoleh adalah 51,025 produk yang dibulatkan menjadi 51 buah.

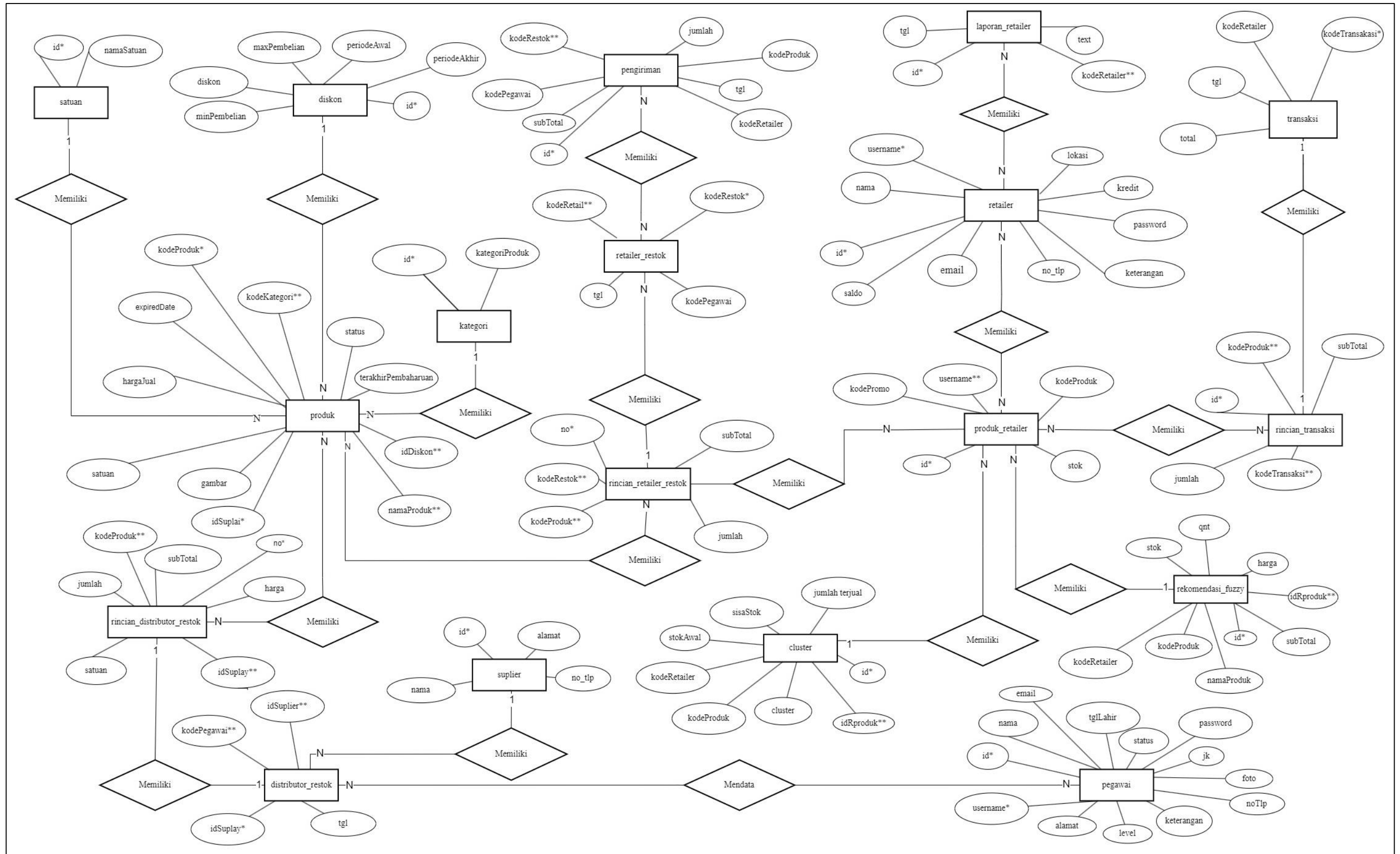
3.3.2 Perancangan Basis Data

Pada bagian ini akan dijelaskan mengenai perancangan basis data yang meliputi perancangan *Entity Relationship Diagram* (ERD) untuk memodelkan struktur data dan hubungan antar data dan mengimplementasikan basis data secara logika dengan *database*, serta perancangan struktur tabel untuk menyusun basis data yang akan diterapkan ke dalam sistem.

3.3.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) atau diagram hubungan entitas adalah suatu model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpanan data. ERD digunakan untuk menjelaskan hubungan antara data dalam *database* berdasarkan persepsi yang terdiri dari objek-objek dasar yang memiliki hubungan relasi antara objek-objek tersebut. Sementara Model data sendiri terdiri dari model hubungan entitas atau objek yang dapat didefinisikan dalam lingkungan pemakai serta model relasional. Sehingga ERD dapat membantu dalam menjawab persoalan tentang data yang diperlukan dan bagaimana data tersebut saling berhubungan.

Pada penelitian ini dirancang ERD yang menjelaskan mengenai hubungan antar penyimpanan data yang terdiri dari entitas, atribut, himpunan relasi serta hubungan antara entitas dengan relasi yang dijelaskan dengan kardinalitas relasi *many to many*, *many to one* atau *one to many*. Dalam ERD ini terdapat 18 entitas yang saling berelasi diantaranya entitas *retailer*, *produk_retailer*, *transaksi*, *rincian_transaksi*, *retailer_restok*, *rincian_retailer_restok*, *pegawai*, *produk*, *satuan*, *kategori*, *diskon*, *supplier*, *distributor_restok*, *rincian_distributor_restok*, *pengiriman*, *laporan_retailer*, *cluster* dan *rekomendasi_fuzzy*. Berikut perancangan ERD dapat dilihat pada ilustrasi Gambar 3.9.



Gambar 3.9 Entity Relationship Diagram (ERD)

1. Entitas *retailer* memiliki tujuh atribut dengan *username* sebagai *primary key*. Entitas ini memiliki relasi *many to many* dengan entitas *retailer_produk* yang dihubungkan dengan atribut *username* pada entitas *retailer* sebagai *primary key* dan atribut *username* pada entitas *retailer_produk* sebagai *foreign key*. Entitas *retailer* juga berelasi dengan entitas *laporan_retailer* dengan kardinalitas relasi *many to one* dengan atribut *username* pada entitas *retailer* sebagai *primary key* dan atribut *kodeRetail* pada entitas *laporaRetailer* sebagai *foreign key*.
2. Entitas *produk_retailer* memiliki lima atribut dengan *id* sebagai *primary key*. Entitas ini memiliki relasi dengan entitas *retailer* seperti yang telah dijelaskan pada *point 1* dan juga entitas *rincian_transaksi* yang memiliki kardinalitas relasi *many to many* dengan atribut *kodeProduk* pada entitas *retailer_produk* sebagai *primary key* dan atribut *kodeProduk* pada entitas *rincian_transaksi* sebagai *foreign key*. Entitas *produk_retailer* juga berelasi dengan entitas *retailer_rincian_Restok* dengan kardinalitas relasi *many to many* dengan atribut *kodeProduk* pada entitas *retailer_produk* sebagai *primary key* dan atribut *kodeProduk* pada entitas *retailer_rincian_restok* sebagai *foreign key*. entitas ini juga berelasi dengan entitas *rekomendasi_fuzzy* dan entitas *cluster* dengan kardinalitas relasi *many to one*.
3. Entitas *rincian_transaksi* memiliki lima atribut dengan *id* sebagai *primary key*. Entitas ini berhubungan dengan entitas *produk_retailer* seperti yang telah dijelaskan pada *point 2* dan juga entitas *transaksi* yang memiliki kardinalitas relasi *one to one* dengan atribut *kodeTransaksi* pada entitas *retailer_transaksi* sebagai *primary key* dan atribut *kodeTransaksi* pada entitas *rincian_transaksi* sebagai *foreign key*.
4. Entitas *transaksi* memiliki lima atribut dengan *id* sebagai *primary key*. Entitas ini memiliki relasi dengan entitas *rincian_transaksi* seperti yang telah dijelaskan pada *point 2*.
5. Entitas *retailer_rincian_restok* memiliki lima atribut dengan *no* sebagai *primary key*. Entitas ini berhubungan dengan entitas *produk_retailer* seperti yang telah dijelaskan pada *point 2* dan juga entitas *produk* yang memiliki kardinalitas relasi *many to many* dengan atribut *kodeProduk* pada entitas *produk* sebagai *primary key* dan atribut *kodeProduk* pada entitas *retailer_rincian_restok* sebagai *foreign key*. Entitas ini juga berelasi dengan entitas *retailer_restok* dengan kardinalitas relasi *one to one*. Atribut *kodeRestok* dalam entitas *retailer_restok* berperan sebagai *primary key* sementara atribut *kodeRestok* pada entitas *retailer_rincian_restok* sebagai *foreign key*.
6. Entitas *retailer_restok* memiliki empat atribut dengan *kodeRestok* sebagai *primary key*. Entitas ini memiliki relasi dengan entitas *retailer_rincian_restok* seperti yang telah dijelaskan pada *point 5* dan juga entitas *pengiriman* dengan kardinalitas relasi *many to many*. Atribut *kodeRestok* dalam entitas *retailer_restok* berperan sebagai *foreign key* sementara atribut *id* pada entitas *pengiriman* sebagai *primary key*.
7. Entitas *pengiriman* memiliki delapan atribut dengan *id* sebagai *primary key*. Entitas ini berhubungan dengan entitas *retailer_restok* seperti yang telah dijelaskan pada *point 6*.
8. Entitas *produk* memiliki yang memiliki sepuluh atribut dengan *kodeProduk* sebagai *primary key*. Entitas ini berhubungan dengan entitas *retailer_rincian_restok* seperti yang telah dijelaskan pada *point 5* dan juga entitas *kategori* yang memiliki kardinalitas relasi *many to one* dengan atribut *kodeKategori* pada entitas *produk* sebagai *foreign key* dan atribut *id* pada entitas *kategori* sebagai *primary key*. Entitas *produk* juga memiliki relasi dengan entitas *satuan* yang memiliki kardinalitas relasi *many to one* dengan atribut *kodeSatuan* pada entitas

produk sebagai *foreign key* dan atribut id pada entitas kategori sebagai *primary key*. Entitas *produk* juga memiliki relasi dengan entitas diskon yang memiliki kardinalitas relasi *many to one* dengan atribut kodeDiskon pada entitas produk sebagai *foreign key* dan atribut id pada entitas kategori sebagai *primary key*. Selain itu entitas produk memiliki relasi dengan entitas rincian_distributor_restok yang memiliki kardinalitas relasi *many to many* dengan atribut kodeProduk pada entitas produk sebagai *primary key* dan atribut kodeProduk pada entitas retailer_rincian_restok sebagai *foreign key*.

9. Entitas kategori memiliki dua atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas produk seperti yang telah dijelaskan pada *point 8*.
10. Entitas satuan memiliki dua atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas produk seperti yang telah dijelaskan pada *point 8*.
11. Entitas diskon memiliki enam atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas produk seperti yang telah dijelaskan pada *point 8*.
12. Entitas suplier memiliki empat atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas distributor_restok dengan kardinalitas relasi *one to many* yang mana atribut id pada entitas suplier berperan sebagai *primary key* dan atribut idPemasok pada entitas distributor_restok sebagai *foreign key*.
13. Entitas distributor_restok memiliki lima atribut dengan idSuplay sebagai *primary key*. Entitas ini berhubungan dengan entitas suplier seperti yang telah dijelaskan pada *point 12* dan juga berelasi *one to one* dengan entitas rincian_distributor_restok yang mana atribut idSuplay pada entitas distributor_restok berperan sebagai *primary key* dan atribut idSuplay pada entitas rincian_distributor_restok sebagai *foreign key*.
14. Entitas rincian_distributor_restok memiliki tujuh atribut dengan no sebagai *primary key*. Entitas ini berhubungan dengan entitas produk seperti yang telah dijelaskan pada *point 8* serta entitas suplai seperti yang telah dijelaskan pada *point 13*.
15. Entitas laporan_retailer memiliki empat atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas retailer seperti yang telah dijelaskan pada *point 1*.
16. Entitas rekomendasi_fuzzy memiliki sembilan atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas produkRetailer seperti yang telah dijelaskan pada *point 1*.
17. Entitas *cluster* memiliki delapan atribut dengan id sebagai *primary key*. Entitas ini berhubungan dengan entitas produkRetailer seperti yang telah dijelaskan pada *point 1*.
18. Entitas pegawai memiliki sebelas atribut dengan *username* sebagai *primary key*. Entitas ini memiliki relasi *one to many*.

3.3.2.2 Perancangan Struktur Tabel

Perancangan struktur tabel diperlukan untuk menyusun basis data yang akan diterapkan ke dalam sistem. Beberapa tabel yang akan diterapkan kedalam *database* sistem diantaranya sebagai berikut.

a. Tabel produk

Tabel produk merupakan tabel yang digunakan untuk menyimpan data produk milik distributor atau gudang utama. Tabel ini memiliki 16 *field* diantaranya id, kodeProduk, namaProduk, kodeKategori, stok, status, hargaBeli, hargaDistribusi, harga, satuan, idDiskon, idStok, experiedDate, idPemasok, gambar dan terakhir pembaharuan dengan field id sebagai *primary key*. Berikut tabel produk dapat dilihat pada Tabel 3.10.

Tabel 3.10 Tabel produk

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	int (11)	Not Null, Primary Key	
kodeProduk	varchar (13)		
namaProduk	varchar (50)		
kodeKategori	char (3)	Foreign key	
stok	int (11)		
status	int (11)		
hargaBeli	int (11)		
hargaDistribusi	int (11)		
harga	int (11)		
satuan	int (11)	Foreign key	
idDiskon	int (11)	Foreign key	
idStok	int (11)	Foreign key	
experiedDate	date		
idPemasok	int (11)	Foreign key	
gambar	varchar (300)		
terakhirPembaharuan	date		

b. Tabel satuan

Tabel satuan digunakan untuk menyimpan data satuan produk. Tabel ini memiliki dua *field* yaitu *field* id dengan tipe data *integer* dan namaSatuan dengan tipe data *varchar*. dengan *field* id sebagai *primary key*. Berikut tabel satuan dapat dilihat pada Tabel 3.11.

Tabel 3.11 Tabel satuan

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	int (11)	Not Null, Primary Key	Berisi kode satuan
namaSatuan	varchar (50)		

c. Tabel kategori

Tabel kategori digunakan untuk menyimpan data kategori produk. Tabel ini memiliki dua *field* yaitu *field* id dengan tipe data *integer* dan *field* kategoriProduk yang merupakan nama kategori dari produk dengan tipe data *varchar*. *Field* id pada tabel ini berperan sebagai *primary key*. Berikut tabel kategori dapat dilihat pada Tabel 3.12.

Tabel 3.12 Tabel kategori

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	int (11)	Not Null, Primary Key	
kategoriProduk	varchar (30)		

d. Tabel diskon

Tabel diskon memiliki enam *field* diantaranya id, diskon, periodeAwal untuk menyimpan data tanggal awal diskon, periodeAkhir untuk menyimpan data tanggal akhir berlakunya diskon, minPembelian dan maxPembelian. *field* id dalam tabel ini berperan sebagai *primary key*. Berikut tabel diskon dapat dilihat pada Tabel 3.13.

Tabel 3.13 Tabel diskon

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	int (11)	Not Null, Primary Key	
diskon	float		
periodeAwal	date		
periodeAkhir	date		
minPembelian	int (11)		
maxPembelian	int (11)		

e. Tabel supplier

Tabel supplier digunakan untuk menyimpan rincian data pemasok yang mengirim barang ke distributor. Tabel ini memiliki empat *field* yaitu id dengan tipe data *integer*, nama dengan tipe data *varchar*, alamat dengan tipe data *varchar*, dan tlp dengan tipe data *varchar*. *Field* id pada tabel ini berperan sebagai *primary key*. Berikut tabel supplier dapat dilihat pada Tabel 3.14.

Tabel 3.14 Tabel supplier

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
nama	<i>varchar</i> (30)		
alamat	<i>varchar</i> (200)		
tlp	<i>varchar</i> (15)		

f. Tabel distributor_restok

Tabel distributor_restok digunakan untuk menyimpan rincian data *suplay* yaitu data pemesanan barang dari pihak distributor ke pemasok. Table ini memiliki empat *field* diantaranya id, idPemasok, tgl yang berisi data tanggal barang diterima, username yang berisi *username* karyawan yang memasukan data distributor_restok. *field* id pada tabel ini berperan sebagai *primary key*. Berikut tabel distributor_restok dapat dilihat pada Tabel 3.15.

Tabel 3.15 Tabel distributor_restok

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
idPemasok	<i>int</i> (11)	<i>Foreign key</i>	
tgl	<i>date</i>		
username	<i>int</i> (11)		<i>username</i> karyawan

g. Tabel rincian_distributor_restok

Table rincian_distributor_restok digunakan untuk menyimpan rincian dari data Distributor_restok. Tabel ini memiliki enam *field* diantaranya no, idSuplay, kodeProduk, harga, jumlah, dan subtotal. Pada tabel ini *field* no sebagai *primary key*. Berikut tabel Rincian_distributor_restok dapat dilihat pada Tabel 3.16.

Tabel 3.16 Tabel rincianDetailSuplay

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
no	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
idSuplay	<i>int</i> (11)	<i>Foreign key</i>	
kodeProduk	<i>varchar</i> (13)	<i>Foreign key</i>	
harga	<i>int</i> (11)		
jumlah	<i>int</i> (11)		
subtotal	<i>int</i> (11)		

h. Tabel retailer_restok

Tabel retailer_restok digunakan untuk menyimpan data restok yaitu data pengiriman barang dari pihak distributor ke pihak *retailer*. Tabel ini memiliki empat *field* yaitu kodeRestok, kodeRetailer, tgl dan kodePegawai. *Field* kodeRestok pada tabel ini sebagai *primary key*. Berikut tabel Restok dapat dilihat pada Tabel 3.17.

Tabel 3.17 Tabel retailer_restok

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
kodeRestok	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeRetail	<i>varchar</i> (50)	<i>Foreign key</i>	Berisi <i>username</i> retailer
tgl	<i>date</i>		
kodePegawai	<i>varchar</i> (50)	<i>Foreign key</i>	Berisi <i>username</i> pegawai

i. Tabel rincian_retailer_restok

Tabel rincian_retailer_restok digunakan untuk menyimpan rincian detail dari data restok. Tabel ini memiliki lima *field* yaitu no, kodeRestok, kodeProduk, jumlah dan subtotal. *Field* no dalam tabel ini sebagai *primary key*. Berikut tabel rincian_retailer_restok dapat dilihat pada Tabel 3.18.

Tabel 3.18 Tabel rincian_retailer_restok

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
no	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeRestok	<i>int</i> (11)	<i>Foreign key</i>	
kodeProduk	<i>varchar</i> (30)	<i>Foreign key</i>	
jumlah	<i>int</i> (11)		
subtotal	<i>int</i> (11)		

j. Tabel produk_retailer

Tabel produk_retailer produk digunakan untuk menyimpan data produk yang dimiliki oleh tiap-tiap *retailer*. Tabel ini memiliki lima *field* diantaranya id, kodeRetailer, kodeProduk, stok dan kodePromo. Pada tabel ini *field id* sebagai *primary key*. Berikut tabel produk_retailer produk dapat dilihat pada Tabel 3.19.

Tabel 3.19 Tabel produk_retailer

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeRetailer	<i>varchar</i> (50)	<i>Foreign key</i>	Berisi <i>username retailer</i>
kodeProduk	<i>varchar</i> (100)	<i>Foreign key</i>	
stok	<i>int</i> (11)		
kodePromo	<i>int</i> (11)		

k. Tabel transaksi

Tabel transaksi digunakan untuk menyimpan data transaksi penjualan produk yang dilakukan oleh tiap-tiap *retailer*. Tabel ini memiliki empat *field* diantaranya kodeTransaksi, kodeRetailer, total, dan tanggal. Pada tabel ini *field* kodeTransaksi sebagai *primary key*. Berikut tabel *retailer* transaksi dapat dilihat pada Tabel 3.20.

Tabel 3.20 Tabel transaksi

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
kodeTransaksi	<i>int</i> (100)	<i>Not Null, Primary Key</i>	
kodeRetailer	<i>varchar</i> (50)	<i>Foreign key</i>	
total	<i>int</i> (11)		
tanggal	<i>date</i>		

l. Tabel rincian_transaksi

Tabel rincian_transaksi digunakan untuk menyimpan rincian detail dari data transaksi penjualan produk milik *retailer*. Tabel ini memiliki lima *field* diantaranya id, kodeTransaksi, kodeProduk, jumlah, dan subtotal. Pada tabel ini *field id* sebagai *primary key*. Berikut tabel rincian_transaksi dapat dilihat pada Tabel 3.21.

Tabel 3.21 Tabel rincian_transaksi

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeTransaksi	<i>int</i> (11)	<i>Foreign key</i>	
kodeProduk	<i>varchar</i> (100)	<i>Foreign key</i>	
jumlah	<i>int</i> (11)		
subtotal	<i>int</i> (11)		

m. Tabel pegawai

Tabel pegawai digunakan untuk menyimpan rincian data pegawai milik distributor. Tabel ini memiliki 11 *field* diantaranya id, username, password, nama, jk, usia, level, status, alamat, noTlp, foto, dan keterangan. Pada tabel ini *field* username sebagai *primary key*. Berikut tabel pegawai dapat dilihat pada Tabel 3.22.

Tabel 3.22 Tabel pegawai

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i>	<i>Not Null, Primary Key</i>	
username	<i>varchar</i> (50)		
Password	<i>varchar</i> (500)		
nama	<i>varchar</i> (30)		
jk	<i>varchar</i> (10)		
usia	<i>int</i> (3)		
level	<i>int</i> (1)		
status	<i>int</i> (1)		
alamat	<i>varchar</i> (200)		
noTlp	<i>varchar</i> (15)		
foto	<i>varchar</i> (100)		
keterangan	<i>varchar</i> (500)		

n. Tabel retailer

Tabel retailer digunakan untuk menyimpan rincian data *retailer* yang tergabung. Tabel ini memiliki sembilan *field* diantaranya id, username, password, nama, lokasi, noTlp, saldo, kredit, status, dan email. Pada tabel ini *field* username berperan sebagai *primary key*. Berikut tabel retailer dapat dilihat pada Tabel 3.23.

Tabel 3.23 Tabel retailer

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i>	<i>Not Null, Primary Key</i>	
username	<i>varchar</i> (50)		
password	<i>varchar</i> (500)		
nama	<i>varchar</i> (30)		
lokasi	<i>varchar</i> (300)		
noTlp	<i>varchar</i> (15)		
saldo	<i>int</i> (11)		
kredit	<i>int</i> (11)		
status	<i>int</i> (1)		
email	<i>varchar</i> (100)		

o. Tabel laporan_retailer

Tabel laporan_retailer digunakan untuk menyimpan data pelaporan yang dikirimkan *retailer* kepada distributor. Tabel ini memiliki empat *field* dan *field* id sebagai *primary key*. Berikut tabel laporan_retailer dapat dilihat pada Tabel 3.24.

Tabel 3.24 Tabel laporan_retailer

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeRetailer	<i>varchar</i> (30)		
Text	<i>text</i>		
tgl	<i>date</i>		

p. Tabel pengiriman

Tabel pengiriman digunakan untuk menyimpan data pengiriman dari pihak distributor kepada pihak *retailer*. Tabel ini memiliki delapan *field* diantaranya id, kodeRestok, kodeRetailer, kodeProduk, kodePegawai, jumlah, subtotal, dan tglPengiriman. Pada tabel ini

field id sebagai *primary key*. Berikut tabel pengiriman dapat dilihat pada Tabel 3.25.

Tabel 3. 25 Tabel pengiriman

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
kodeRestok	<i>varchar</i> (30)	<i>Foreign key</i>	
kodeRetailer	<i>varchar</i> (30)		
kodeProduk	<i>varchar</i> (30)		
kodePegawai	<i>varchar</i> (30)		
jumlah	<i>int</i> (11)		
subTotal	<i>int</i> (11)		
tglPengiriman	<i>date</i>		

q. Tabel rekomendasi_fuzzy

Tabel rekomendasi_fuzzy digunakan untuk menyimpan data rekomendasi pengiriman produk dari hasil perhitungan algoritma *Fuzzy Tsukamoto* dari pihak distributor kepada pihak *retailer*. pada tabel ini terdapat sembilan *field* diantaranya *id* sebagai *primary key*, *idRproduk*, *kodeRetailer*, *kodeProduk*, *namaProduk*, *stok*, *qnt*, *harga*, dan *subTotal*. Berikut tabel rekomendasi_fuzzy dapat dilihat pada Tabel 3.26.

Tabel 3.26 Tabel rekomendasi_fuzzy

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
idRproduk	<i>varchar</i> (30)	<i>Foreign key</i>	
kodeRetailer	<i>varchar</i> (30)		
kodeProduk	<i>varchar</i> (30)		
namaProduk	<i>varchar</i> (30)		
stok	<i>int</i> (11)		
qnt	<i>int</i> (11)		
harga	<i>int</i> (11)		
subTotal	<i>int</i> (11)		

r. Tabel cluster

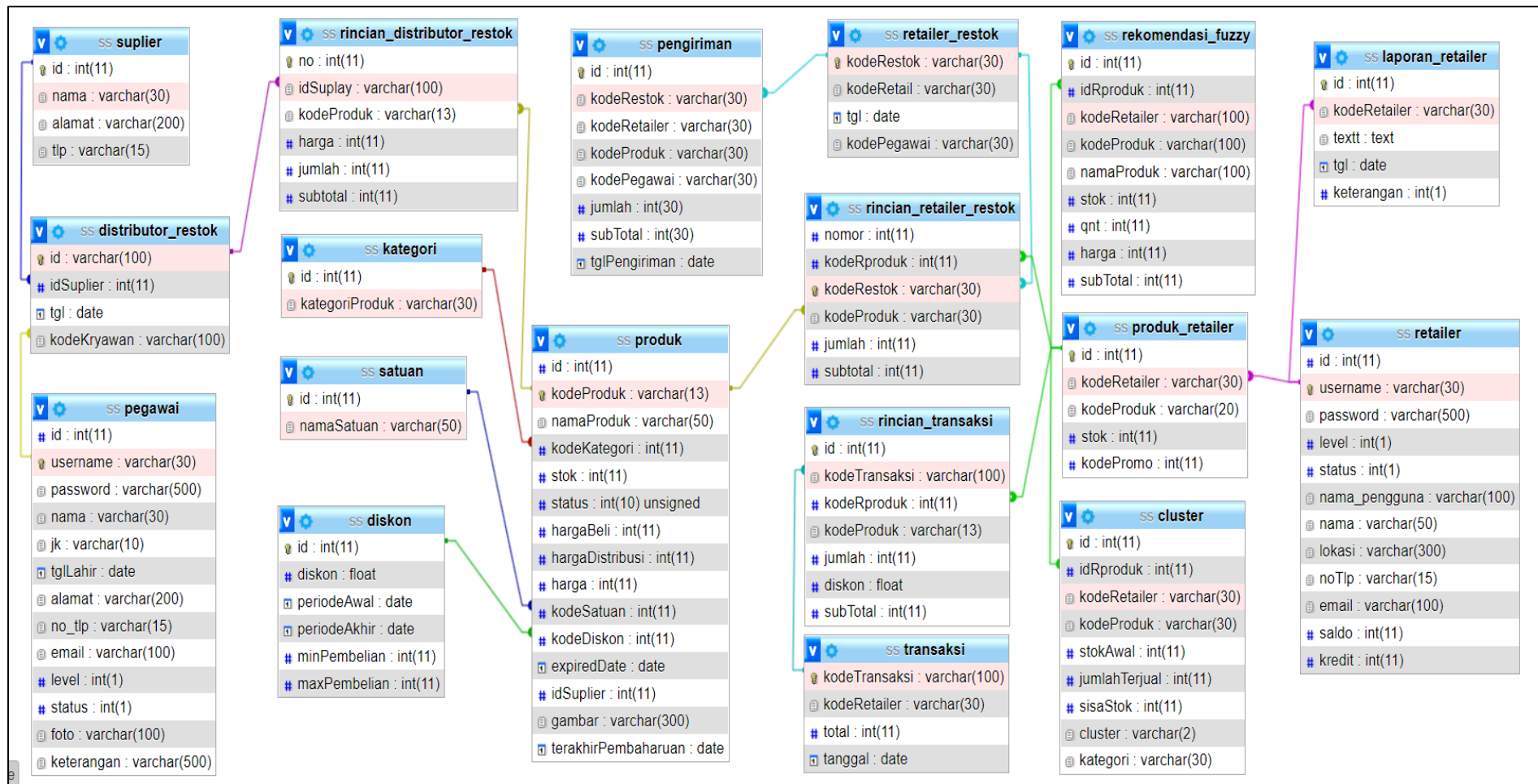
Tabel cluster digunakan untuk menyimpan data pengelompokan produk milik semua *retailer* dari hasil perhitungan *kmeans*. pada tabel ini terdapat delapan *field* diantaranya *id* sebagai *primary key*, *idRproduk*, *kodeRetailer*, *kodeProduk*, *stokAwal*, *jumlahTerjual*, *sisastok* dan *cluster*. Berikut tabel *kmeans* dapat dilihat pada Tabel 3.27.

Tabel 3.27 Tabel cluster

<i>Field</i>	<i>Type Data</i>	<i>Constraint</i>	Keterangan
id	<i>int</i> (11)	<i>Not Null, Primary Key</i>	
idRproduk	<i>varchar</i> (30)	<i>Foreign key</i>	
kodeRetailer	<i>varchar</i> (30)		
kodeProduk	<i>varchar</i> (30)		
stokAwal	<i>int</i> (11)		
JumlahTerjual	<i>int</i> (11)		
sisastok	<i>int</i> (11)		
cluster	<i>int</i> (11)		

3.3.2.3 Relasi Antar Tabel (RAT)

Relasi antar tabel merupakan relasi yang menggambarkan hubungan antar tabel dalam *database* pada sistem yang akan dibangun. Relasi ini dihubungkan menggunakan atribut yang berperan sebagai *primary key* dan *foreign key* pada tiap-tiap tabel yang terhubung. Berikut RAT dapat dilihat pada Gambar 3.10.



Gambar 3.10 Relasi Antar Tabel

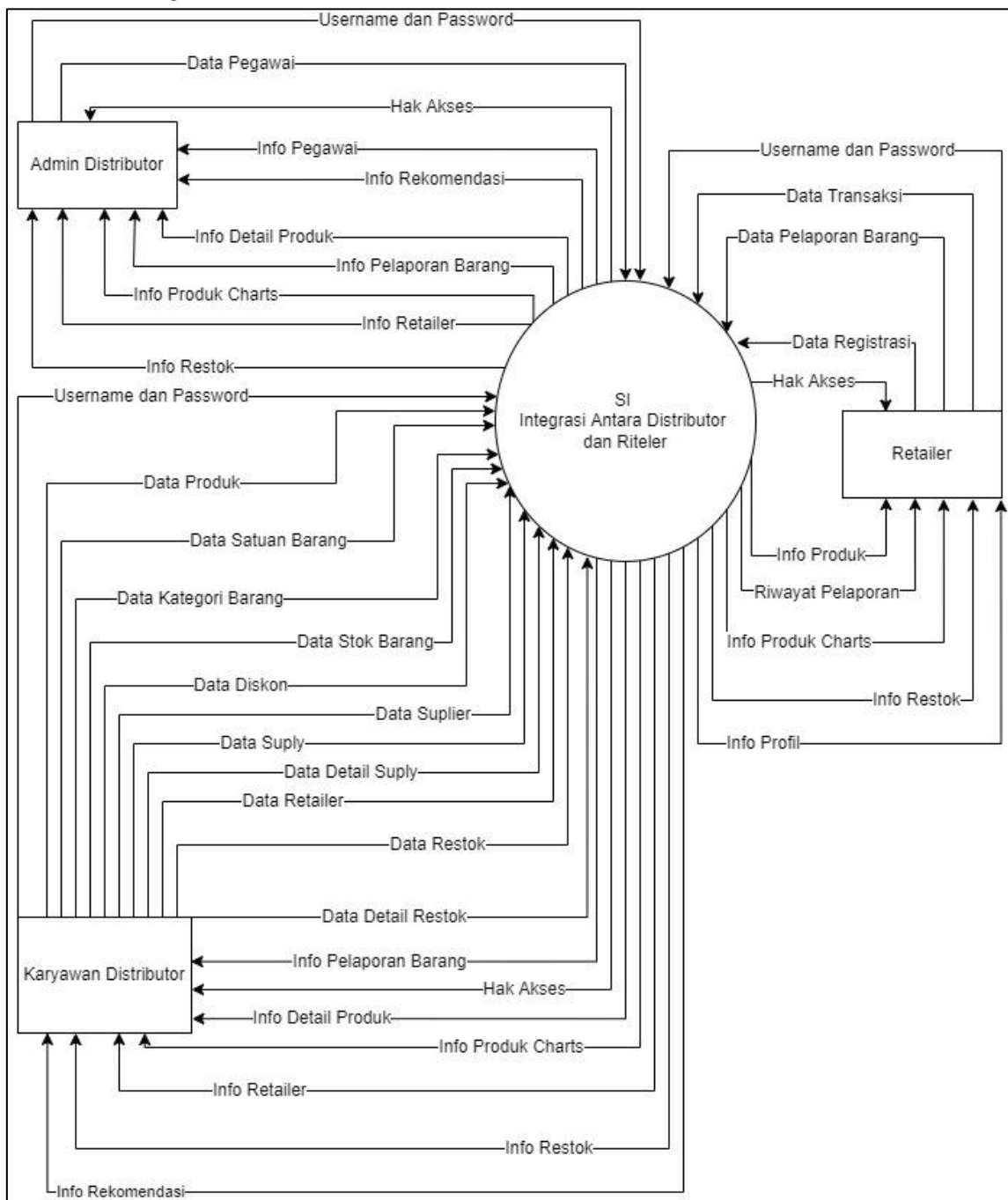
3.3.3 Perancangan Perangkat Lunak

Pada bagian ini akan dijelaskan mengenai perancangan perangkat lunak berupa perancangan *Data Flow Diagram* (DFD) dan perancangan *interface* sistem yang akan dibangun.

3.3.3.1 Data Flow Diagram (DFD)

Data Flow Diagram atau DFD menggambarkan aliran data dari suatu sistem yang akan dibangun. DFD ini berfungsi untuk membantu memahami sistem secara terstruktur dan jelas. Pada bagian ini akan dijelaskan pemodelan DFD dari *context diagram*, DFD level 0 dan DFD level 1. Perancangan *context diagram* dapat dilihat pada ilustrasi Gambar 3.11.

1. Context Diagram



Gambar 3.11 Context Diagram

Context diagram menggambarkan hubungan sistem dengan entitas luar yang memanfaatkan sistem. Pada bagian ini digambarkan aliran data dan informasi yang terjadi pada sistem yang akan dibangun dengan entitas yang menggunakan sistem.

Pada *context diagram* ini memiliki satu proses utama yaitu SI Integrasi antara Distributor dan *Retailer*. Dan memiliki tiga entitas eksternal yaitu admin distributor, karyawan distributor dan admin. Entitas eksternal ini menggambarkan pengguna yang memanfaatkan sistem yang akan dikembangkan.

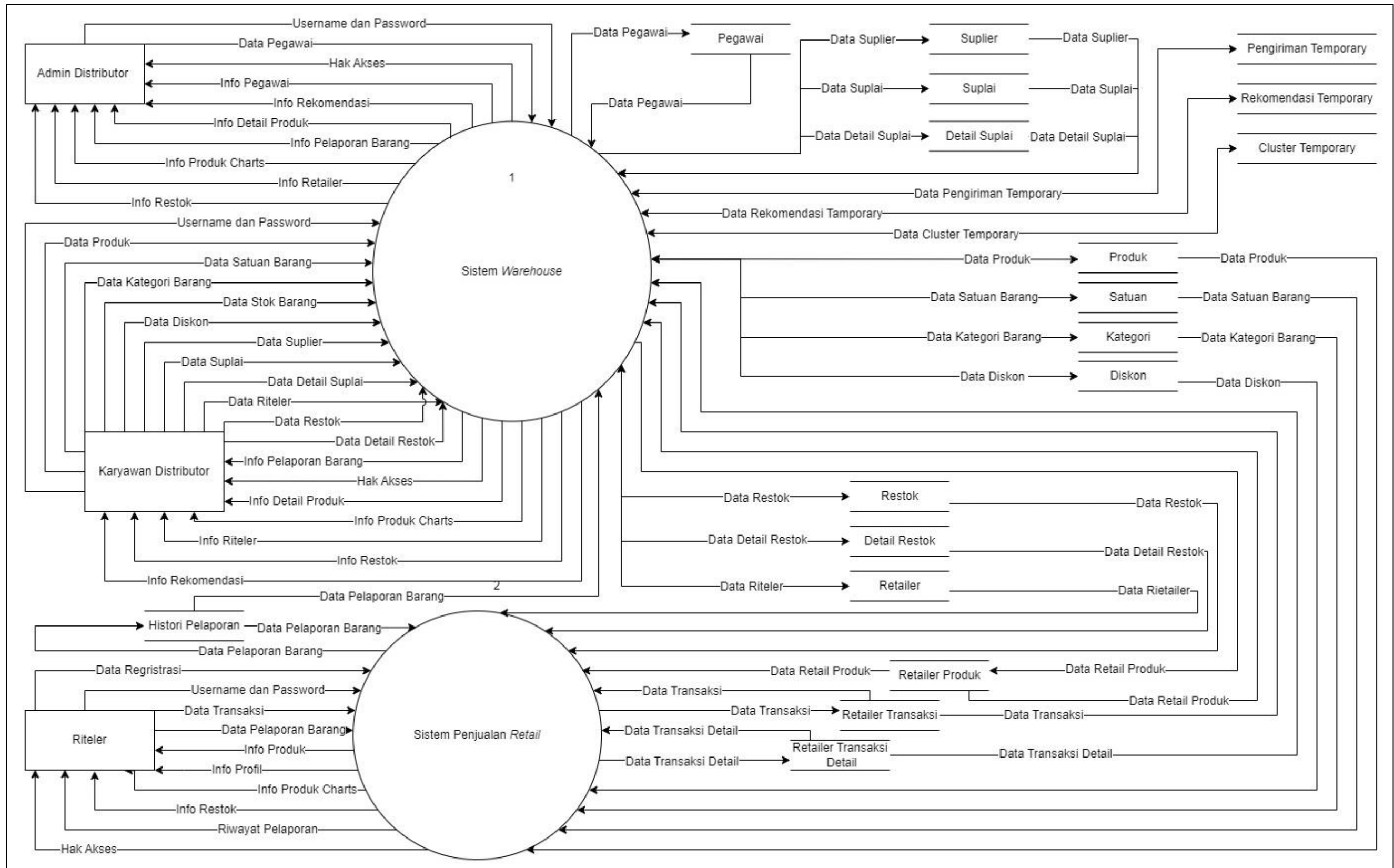
Entitas admin distributor bertugas mengelola data pegawai distributor dengan memberikan inputan data pegawai milik distributor ke sistem. Entitas admin distributor memiliki perizinan untuk mengakses semua proses yang dihasilkan sistem dengan melakukan pengiriman data *username* dan *password* yang kemudian diproses oleh sistem dan akan menghasilkan hak akses milik *admin*. Dari hak akses ini *admin* akan dapat menerima aliran data berupa info pegawai, info detail produk, info rekomendasi, info pelaporan barang, info produk *charts*, info *retailer*, serta info restok.

Selanjutnya entitas karyawan distributor, entitas ini bertugas untuk mengelola data milik distributor dengan melakukan pengiriman *username* dan *password* kemudian entitas karyawan distributor akan mendapatkan akses ke sistem sebagai karyawan. Entitas ini bertugas untuk mengelola berbagai data diantaranya data produk, data stuan barang, data stok barang, data kategori barang, data diskon, serta data pemasok yang kemudian diproses oleh sistem dan menghasilkan info detail produk. Entitas ini juga bertugas untuk memberikan masukan data suplai yaitu data pemesanan barang kepihak pemasok berupa data suplai dan data detail suplai yang kemudian akan diproses oleh sistem menjadi info suplai yang selanjutnya dapat diakses oleh pihak karyawan. Kemudian entitas ini juga bertugas untuk memberikan masukan ke sistem berupa data restok yang merupakan data pengiriman barang kepada pihak *retailer* yang berupa data restok dan data detail restok. Data ini akan diproses oleh sistem dan menghasilkan info restok yang kemudian dapat diakses oleh entitas karyawan distributor. Dari data-data yang sebelumnya dikirimkan oleh entitas karyawan distributor dan di proses oleh sistem juga akan menghasilkan info produk *charts*.

Selanjutnya entitas *retailer* yang berinteraksi dengan sistem dengan memberikan informasi atau data berupa data transaksi penjualan dari pihak *retailer*, data ini akan diproses menjadi info produk *charts* yang akan dikirimkan kembali ke entitas *retailer*, dan diproses menjadi info rekomendasi yang dikirimkan ke entitas karyawan distributor dan entitas admin distributor. Sebelumnya entitas ini perlu mengirimkan data registrasi yang akan diproses menjadi info profil, dengan data ini entitas *retailer* dapat melakukan *Login* dengan memberikan informasi *username* dan *password* dan diproses menjadi hak akses *retailer*. Hak akses ini akan mengizinkan entitas *retailer* untuk mendapatkan info produk, info restok, info produk *charts* serta info restok. Selain itu entitas ini juga dapat mengirimkan data atau informasi pelaporan barang yang kemudian akan diproses menjadi info riwayat pelaporan yang akan dikirimkan ke entitas yang lain.

2. DFD Level 1

DFD level 1 menggambarkan proses utama yang terjadi pada sistem. Bagian ini merupakan pemecahan dari level sebelumnya dimana menjadi proses-proses yang lebih rinci. Pada penelitian ini DFD level 1 dua proses utama yaitu proses Sistem Distributor dan proses sistem Penjualan *Retail*. Proses Sistem Distributor akan berkaitan dengan entitas admin distributor dan karyawan distributor sementara proses Sistem Penjualan *Retail* berkaitan dengan entitas *retailer*. Perancangan DFD level 1 dapat dilihat pada ilustrasi Gambar 3.12.



Gambar 3.12 DFD level 1

Gambar 3.7 merupakan gambar perancangan *data flow diagram* level 1. Dalam bagian ini memiliki dua proses utama yang merupakan hasil pemecahan dari proses sebelumnya yaitu, proses Sistem *Warehouse* dan proses Sistem Penjualan *Retail* dan terdapat 18 data *store*. Untuk jumlah entitas eksternal dan aliran data dari entitas eksternal ke proses sama seperti pada *context diagram* pada bagian sebelumnya.

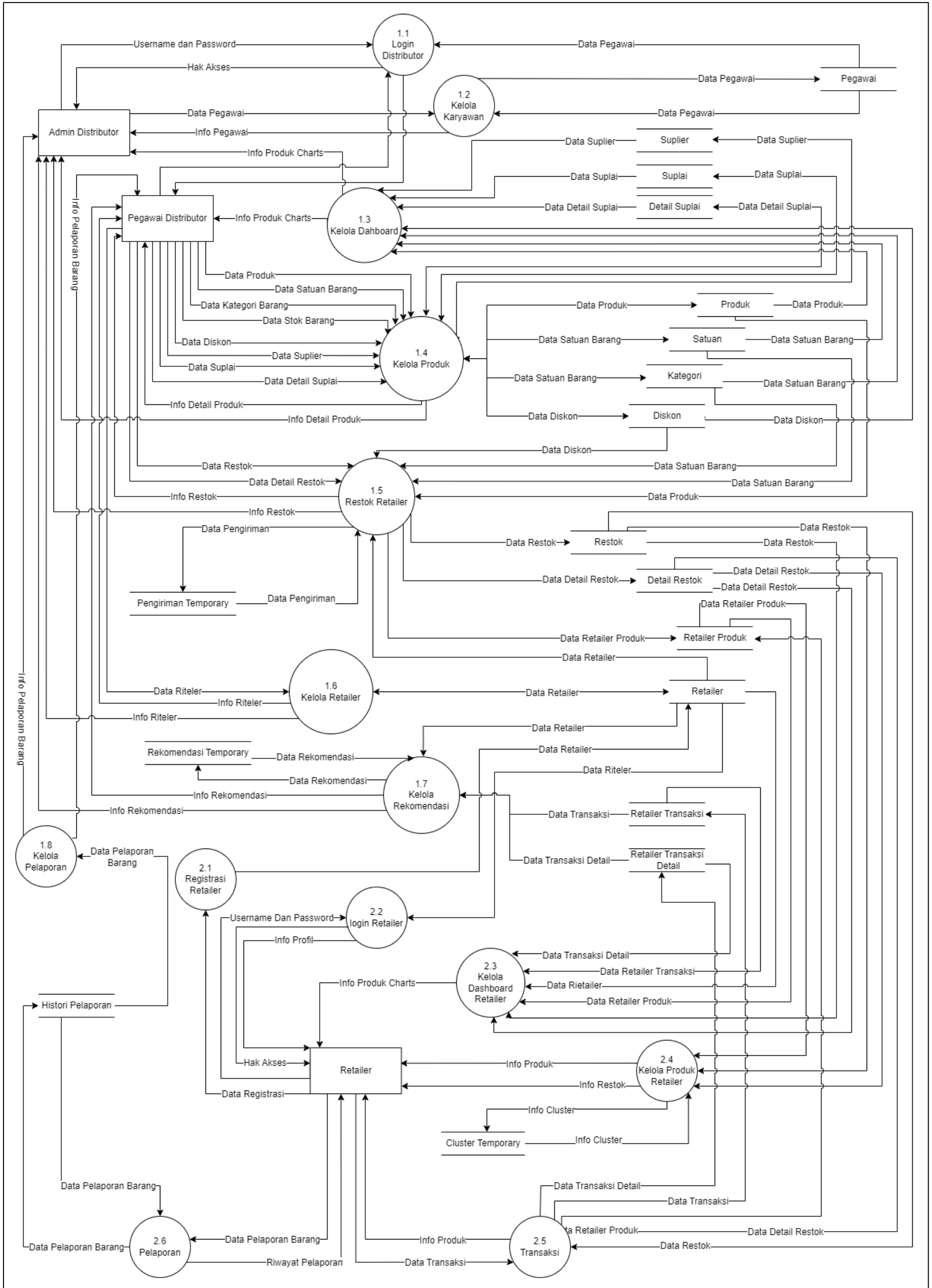
Proses yang pertama yaitu proses Sistem distributor berkaitan dengan entitas admin distributor dan karyawan distributor. Pada entitas admin distributor bertugas memberikan data pegawai ke sistem yang kemudian akan diproses dan dilanjutkan atau dikirim ke data *store* pegawai untuk disimpan dan akan dikirimkan kembali ke sistem jika diperlukan. Sebelumnya entitas admin distributor perlu memberikan data *username* dan *password* yang akan diproses dan dicocokkan dengan data yang telah disimpan oleh data *store* pegawai jika sesuai maka entitas admin distributor akan memperoleh hak akses untuk mendapatkan berbagai info sama seperti yang telah di jelaskan pada penjelasan *context diagram*.

Selanjutnya entitas karyawan distributor juga perlu melakukan *Login* dengan cara memberikan data *username* dan *password* yang kemudian diolah untuk dicocokkan dengan data *store* jika sesuai maka entitas karyawan distributor dapat memperoleh hak aksesnya. Diantaranya adalah melakukan memberikan masukan berupa data produk yang akan di proses dan di pisahkan lalu dikirimkan ke data *store* produk. Dan juga data-data lain seperti data kategori barang akan disimpan ke data *store* kategori, data pemasok akan disimpan ke data *store* pemasok dan sebagainya. Data-data yang telah disimpan di data *store* akan dikirimkan kembali ke proses untuk di olah menjadi informasi-informasi yang nantinya akan ditampilkan atau dikirimkan ke entitas karyawan distributor dan juga entitas admin distributor. Diantaranya info detail produk, info restok, info *retailer*, serta info produk *charts*.

Proses yang kedua yaitu proses Sistem Penjualan *Retail* akan bertugas untuk melakukan pemrosesan terkait data dari entitas *retailer*. Pada bagian pertama entitas *retailer* perlu memberikan data *username* dan *password* yang akan diproses dan dicocokkan ke data *store* *retailer* jika sesuai maka entitas *retailer* akan memperoleh hak akses. Sama seperti pada bagian *context diagram* entitas ini akan memberikan masukan data transaksi ke proses Sistem Penjualan *Retail* yang akan diproses dan dikirimkan untuk disimpan ke data *store* transaksi dan data *store* detail transaksi yang akan dikirimkan kembali dalam bentuk yang sama yaitu data detail transaksi ke proses Sistem Penjualan *Retail* ketika diperlukan kembali untuk dilakukan pemrosesan menjadi informasi terkait informasi penjualan dan informasi rekomendasi.

3. DFD Level 2

DFD Level 2 merupakan lanjutan dari DFD level 1 yang merupakan dekomposisi untuk menjelaskan sub proses yang lebih rinci didalamnya. Pada bagian ini proses yang sebelumnya dua dipecah dan dirincikan kembali menjadi 14 proses yaitu 8 proses merupakan sub proses dari proses 1 Sistem Distributor dan 6 proses merupakan sub proses dari proses Sistem Penjualan *Retail*. Dengan jumlah entitas dan data *store* yang sama. Berikut perancangan DFD level 2 dapat dilihat pada ilustrasi Gambar 3.13.



Gambar 3.13 DFD level 2

Proses 1.1 adalah *Login* distributor, proses ini berfungsi untuk mengelola *Login* milik sistem distributor. Entitas admin distributor dan karyawan distributor perlu memberikan masukan data berupa *username* dan *password* yang kemudian diproses dan dikirimkan ke data *store* untuk dicocokkan dengan data yang ada di data *store* jika sesuai maka proses *Login* distributor akan mengirimkan hak akses kepada entitas tersebut.

Proses 1.2 adalah kelola karyawan. Proses ini bertugas untuk mengelola data karyawan yang dilakukan oleh entitas admin distributor dengan cara memberikan masukan berupa data karyawan ke sistem yang akan diterima proses dan dikirimkan ke data *store* pegawai berupa data pegawai. Data ini dapat dipanggil kembali oleh sistem jika dibutuhkan dan diolah menjadi info pegawai yang ditampilkan kepada entitas admin distributor.

Proses 1.3 kelola *dashboard*. Proses ini berfungsi untuk menampilkan halaman *dashboard* yang berisi tampilan informasi dan grafik sekilas terkait data yang ada. Proses ini mendapat masukan dari data *store* produk, satuan, kategori, stok, diskon, pemasok, suplai dan detail suplai. Masukan dari data *store* ini kemudian diproses menjadi tampilan yang ada dalam *dashboard* atau informasi *dashboard* yang kemudian akan dikirimkan menjadi tampilan *interface* kepada entitas admin distributor dan karyawan distributor.

Proses 1.4 merupakan proses kelola produk. Proses ini berfungsi untuk melakukan pengelolaan data produk yang mana entitas karyawan distributor memberikan masukan berupa data produk, satuan, kategori, stok, diskon, pemasok, suplai dan detail suplai yang kemudian diproses untuk disimpan ke data *store* produk yang dapat dipanggil kembali ketika dibutuhkan dan diproses menjadi info produk yang akan dikirimkan ke entitas karyawan distributor dan admin distributor dalam bentuk *interface* info produk.

Proses 1.5 restok *retailer* merupakan proses yang bertugas untuk mengelola data restok yang mana entitas karyawan distributor dapat memasukkan data restok barang yang dikirimkan kepada pihak *retail*, yang kemudian data ini akan diproses untuk disimpan ke data *store* sehingga data ini dapat dipanggil kembali dan diproses menjadi informasi restok yang dapat dikirimkan kepada entitas karyawan distributor dan admin distributor dalam bentuk *interface* yang menampilkan informasi data restok.

Proses 1.6 merupakan proses kelola *retailer* yang merupakan proses yang bertugas untuk mengelola data *retailer*. Masukan dari proses ini merupakan data dari data *store retailer* berupa data diri milik *retailer* yang diperoleh dari proses 2.1 registrasi *retailer* melalui entitas *retailer*, selain itu proses ini juga mendapat masukan dari entitas karyawan distributor berupa data *retailer* ataupun perubahan data yang akan disimpan ke data *store retailer*. Hasil dari proses ini berupa info *retailer* yang dikirimkan ke entitas admin distributor dan karyawan distributor dan juga info profil yang akan ditampilkan kepada *interface retailer*.

Proses 1.7 kelola rekomendasi. Proses ini merupakan proses yang berfungsi untuk mengelola rekomendasi jumlah pengiriman restok dari distributor ke *retailer*. Dalam proses ini data diperoleh dari data *store* produk dan transaksi yang diperoleh dari proses transaksi. Proses rekomendasi ini dilakukan dengan menggunakan perhitungan menggunakan algoritma *Fuzzy tsukimoto* yang diterapkan pada sistem. Hasilnya merupakan jumlah rekomendasi barang yang akan ditampilkan ke entitas distributor *retailer* dan admin distributor.

Proses 1.8 kelola pelaporan. Masukan dari proses ini berasal dari data *store* histori pelaporan yang diperoleh dari proses 2.6 pelaporan dengan data terkait pelaporan barang yang dikirimkan oleh entitas *retail*. Keluaran dari proses ini merupakan info pelaporan yang akan

ditampilkan kepada entitas karyawan distributor dan entitas admin distributor juga dapat mengaksesnya.

Proses 2.1 registrasi *retailer*. Proses ini bertugas mengelola registrasi dari *retailer* yang ingin bergabung dengan cara pihak entitas *retailer* memberikan data registrasi ke sistem dan akan diproses untuk didimpan ke data *store retailer* dan akan dikirimkan kembali ke proses kelola *retailer*.

Proses 2.2 *Login retailer*. Proses ini berguna untuk mengelola *Login* milik sistem *retailer*. Entitas distributor *retailer* perlu memberikan masukan data berupa *username* dan *password* yang kemudian diproses dan dikirimkan ke data *store* untuk dicocokkan dengan data yang ada di data *store* jika sesuai maka proses *Login retailer* akan mengirimkan hak akses kepada entitas *retailer*.

Proses 2.3 kelola dashboard *retailer*. Proses ini berfungsi untuk menampilkan halaman *dashboard* yang berisi tampilan info dan grafik sekilas terkait data yang ada. Proses ini mendapat masukan dari data *store* produk *retailer*, transaksi, detail transaksi, restok dan detail restok. Masukan dari data *store* ini kemudian diproses menjadi tampilan yang ada dalam dashboard atau info dashboard yang kemudian akan dikirimkan menjadi tampilan *interface* kepada entitas *retailer*.

Proses 2.4 kelola produk *retailer* merupakan proses yang berperan untuk mengelola produk *retailer* proses ini mendapat masukan dari data *store retailer* produk, *retailer* restok, dan detail *retailer* restok. Yang kemudian diproses menjadi info produk *retailer* yang ditampilkan ke pada entitas *retailer*.

Proses 2.5 transaksi yang merupakan proses pengelolaan data transaksi yang dimasukan oleh entitas *retailer* terkait data penjualan produk data ini akan diproses dan dikirimkan ke data *store* transaksi dan detail transaksi kemudian dikirimkan ke proses rekomendasi.

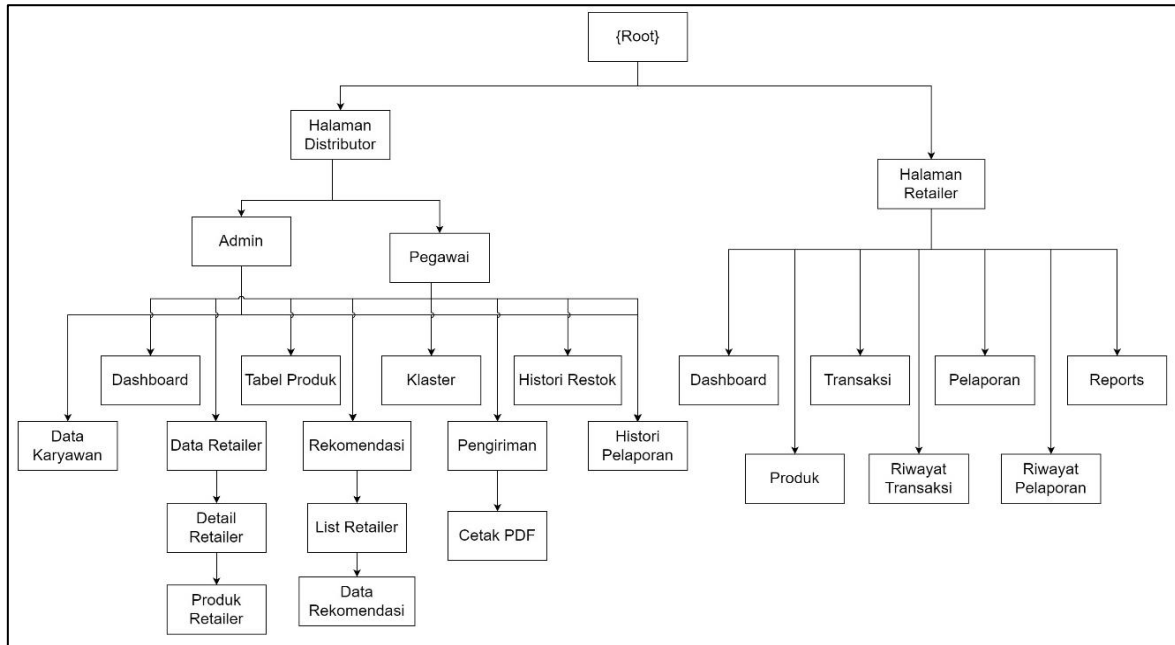
Proses 2.6 pelaporan yaitu proses yang menangani data pelaporan yang diinginkan oleh entitas *retailer* terkait pelaporan produk, seperti kehilangan, kesalahan dan sebagainya yang kemudian diolah dan simpan di data *store* histori pelaporan. Data ini dapat dikirimkan kembali ke proses pelaporan untuk kemudian diproses menjadi histori pelaporan yang akan ditampilkan ke entitas *retailer*. Selain itu data ini juga dikirimkan ke proses 1.8 kelola pelaporan.

3.3.3.2 Perancangan Interface

Pada bagian ini akan dijelaskan mengenai perancangan *interface* yang akan diterapkan pada sistem yang akan dikembangkan. Dalam penelitian ini *interface* akan dibagi menjadi dua bagian yaitu halaman *interface* untuk distributor dan halaman *interface* untuk *retailer*. Berikut merupakan tampilan *interface* yang akan dikembangkan.

1. Perancangan Struktur Menu

Perancangan Struktur Menu adalah bentuk umum dari suatu rancangan sistem atau aplikasi untuk memudahkan pengguna dalam memahami bagian-bagian sistem yang akan dikembangkan. Berikut merupakan rancangan struktur menu dari sitem yang akan dikembangkan dapat dilihat pada ilustrasi Gambar 3.14.



Gambar 3.14 Perancangan Struktur Menu

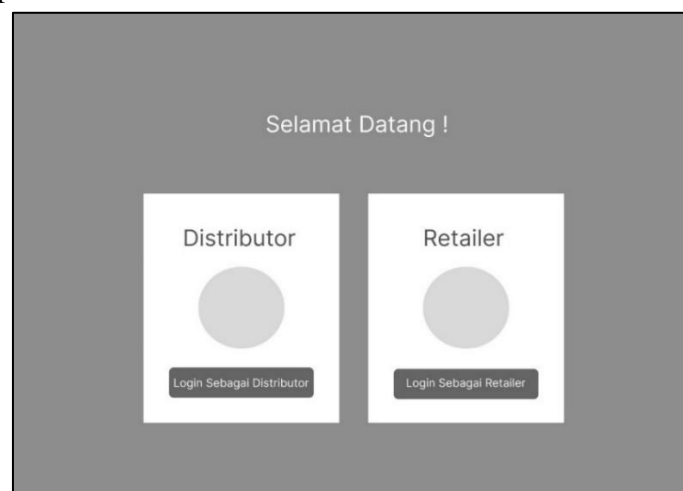
Gambar 3.14 merupakan perancangan struktur menu dari sistem yang akan dikembangkan. Terdapat dua menu utama yaitu menu pada halaman distributor dan menu pada halaman *retailer*. Pada pada distributor menu dibagi menjadi dua halaman yaitu halaman admin dan halaman pegawai. Pada halaman admin terdapat menu data karyawan, *dashboard*, tabel produk, katalog, data *retailer*, rekomendasi, kluster histori restok dan halaman histori pelaporan. Pada halaman pegawai terdapat menu sama seperti halaman admin, namun tidak memiliki menu data karyawan. Menu yang terdapat pada halaman *retailer* meliputi menu *dashboard*, produk, transaksi, riwayat transaksi, pelaporan, riwayat pelaporan, dan *reports*.

2. Perancangan Antarmuka

Pada perancangan antarmuka akan dijelaskan mengenai desain tampilan yang akan diterapkan pada sistem yang akan dikembangkan.

a. Halaman Awal

Halaman awal merupakan halaman yang pertama kali ditampilkan saat program pertama kali dijalankan. Berikut merupakan gambar perancangan halaman awal atau halaman *index* dapat dilihat pada Gambar 3.15.

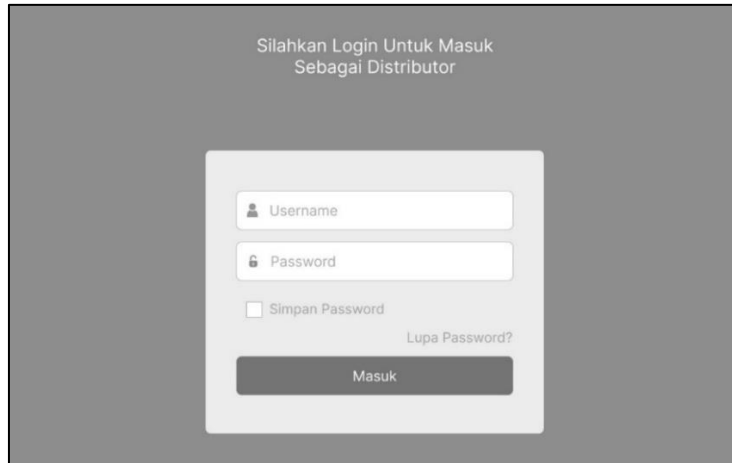


Gambar 3.15 Halaman Awal

Gambar 3.15 merupakan gambar perancangan halaman awal atau *index*. Pada halaman ini terdapat dua *button*. *Button* pertama adalah *login* sebagai distributor yang berfungsi untuk membuka halaman *login* bagi distributor dan *button* kedua yaitu *login* sebagai *retailer* yang berfungsi untuk membuka halaman *login* milik *retailer*.

b. Halaman *Login* Distributor

Halaman *Login* distributor merupakan halaman yang akan ditampilkan setelah pengguna memilih untuk masuk sebagai distributor. Berikut perancangan halaman *Login* distributor dapat dilihat pada Gambar 3.16.

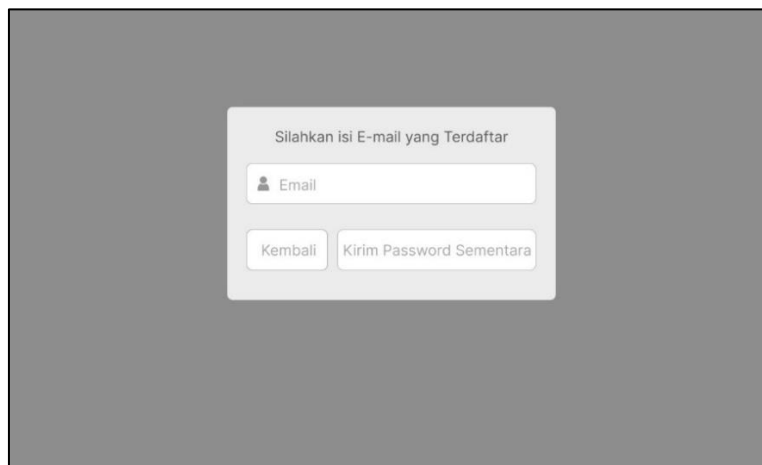


Gambar 3.16 Halaman *Login* Distributor

Gambar 3.16 merupakan gambar perancangan halaman *Login* distributor. Pada halaman ini terdapat *form input* dengan dua *field* untuk mengisi data yaitu *username* untuk memasukan *username* pengguna dan *password* untuk memasukan kata sandi. Kemudian, terdapat *button* masuk untuk proses *Login*. Selain itu ada *button* “lupa password?” apabila pengguna lupa dengan kata sandi.

c. Halaman Lupa *Password*

Halaman lupa *password* merupakan halaman yang akan ditampilkan apabila pengguna memilih *button* “lupa password?” ketika lupa kata sandi. Berikut perancangan halaman lupa kata sandi dapat dilihat pada gambar 3.17.

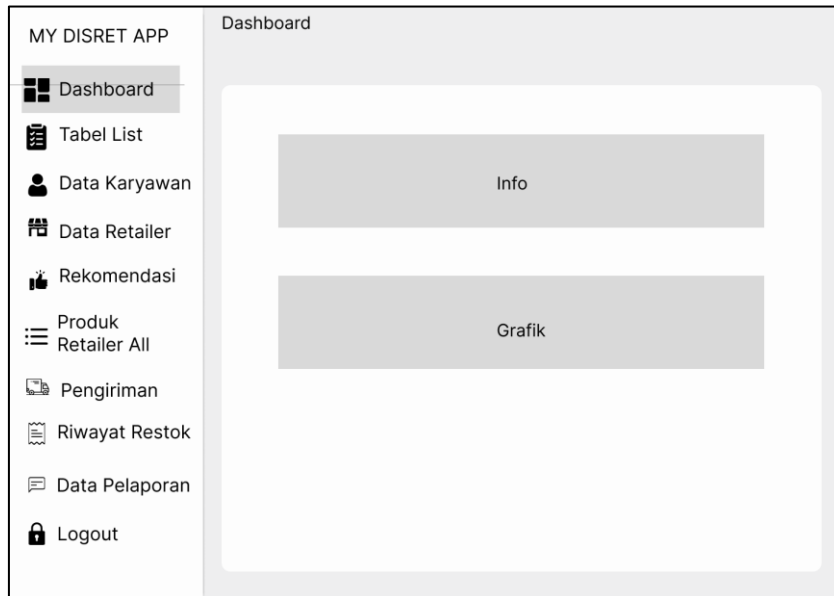


Gambar 3.17 Halaman Lupa *Password*

Gambar 3.17 merupakan perancangan pemodelan halaman lupa *password*. Pada halaman ini terdapat *form* untuk memasukan *e-mail* dan *button* kirim untuk mengirim data *password* sementara ke *email* yang telah terdaftar.

d. Halaman *Dashboard* Distributor

Halaman *dashboard* distributor merupakan halaman yang ditampilkan pertama kali setelah proses *login* berhasil. Berikut perancangan tampilan halaman *dashboard* distributor dapat dilihat pada Gambar 3.18.

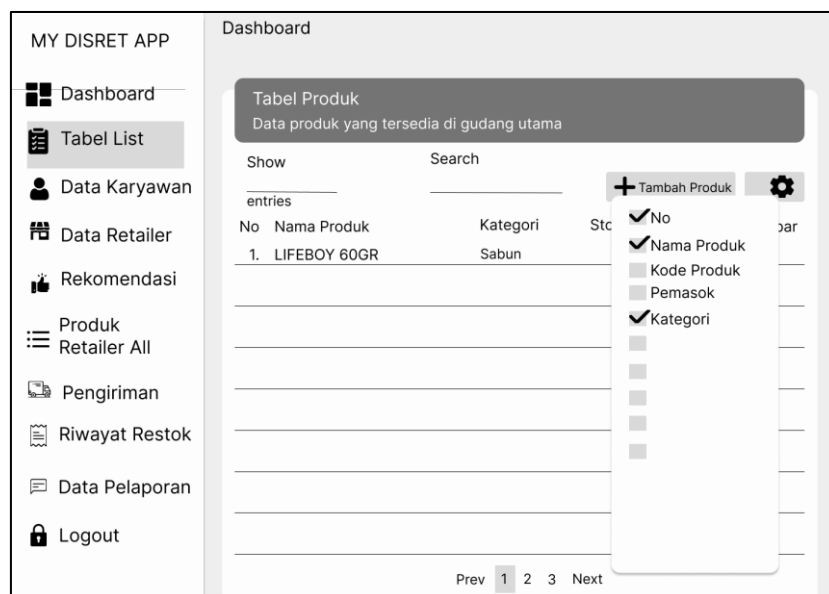


Gambar 3.18 Halaman *Dashboard*

Gambar 3.18 merupakan perancangan pemodelan halaman *dashboard* distributor. Pada halaman ini terdapat beberapa tampilan informasi singkat mengenai data distributor pada bagian info serta pada bagian grafik akan ditampilkan grafik penjualan milik distributor selama kurun waktu tertentu. Pada halaman ini juga telah ditampilkan panel *dashboard* yang berisi beberapa menu diantaranya menu *dashboard*, menu *table list*, menu data karyawan menu data *retailer*, menu rekomendasi, menu produk *retailer all*, dan menu *logout*.

e. Halaman *Tabel List*

Halaman *tabel list* merupakan halaman yang akan digunakan untuk menampilkan data produk milik distributor. Berikut perancangan halaman *tabel list* dapat dilihat pada Gambar 3.19.

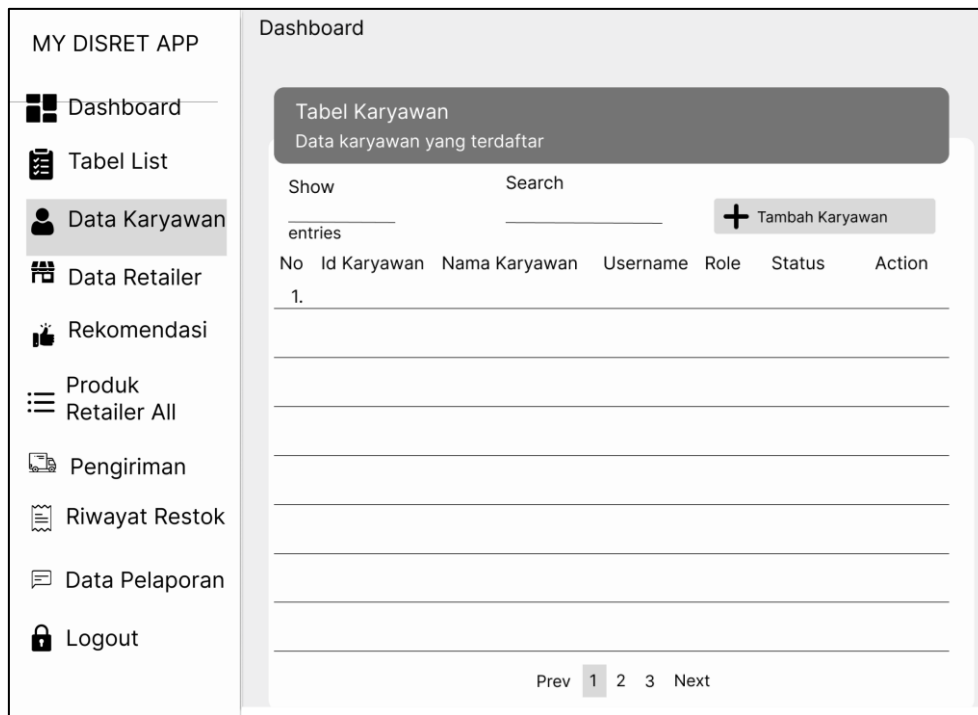


Gambar 3.19 Halaman *Tabel List*

Gambar 3.19 merupakan gambar perancangan pembuatan halaman tabel *list* yang akan digunakan untuk menampilkan data produk yang ada pada *database*. Pada halaman ini akan ditampilkan tabel yang berisi info produk dengan beberapa *field* diantaranya no, nama produk, kategori stok status dan sebagainya. pada halaman ini juga akan ditambahkan beberapa fungsi untuk melakukan pencarian, memilih berapa jumlah data yang akan ditampilkan serta memilih kolom mana saja yang akan ditampilkan. kemudian, terdapat juga *button* tambah produk yang berguna untuk menambahkan data produk

f. Halaman Data Karyawan

Halaman karyawan merupakan halaman yang akan digunakan untuk menampilkan data karyawan milik distributor. Berikut perancangan halaman data karyawan dapat dilihat pada Gambar 3.20

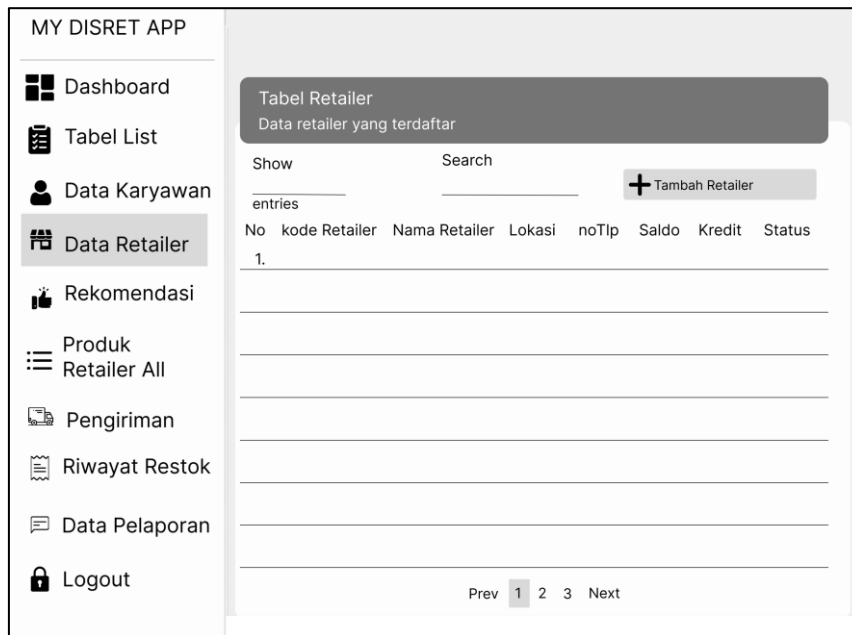


Gambar 3.20 Halaman Data Karyawan

Gambar 3.20 merupakan perancangan halaman data karyawan. pada bagian ini terdapat tabel karyawan yang akan digunakan untuk menampilkan data karyawan yang ada pada *database*. Tabel pada halaman ini memiliki beberapa *field* diantaranya no, id karyawan, username, *role*, status serta *action* untuk melakukan beberapa perintah seperti hapus dan *edit*. pada halaman ini juga akan ditambahkan beberapa fungsi untuk melakukan pencarian, memilih berapa jumlah data yang akan ditampilkan serta memilih kolom mana saja yang akan ditampilkan. Kemudian, terdapat juga *button* tambah karyawan yang berguna untuk menambahkan data karyawan.

g. Halaman Data *Retailer*

Halaman data *retailer* merupakan halaman yang akan digunakan untuk menampilkan data *retailer* yang telah bergabung. Berikut perancangan halaman data *retailer* dapat dilihat pada Gambar 3.21.

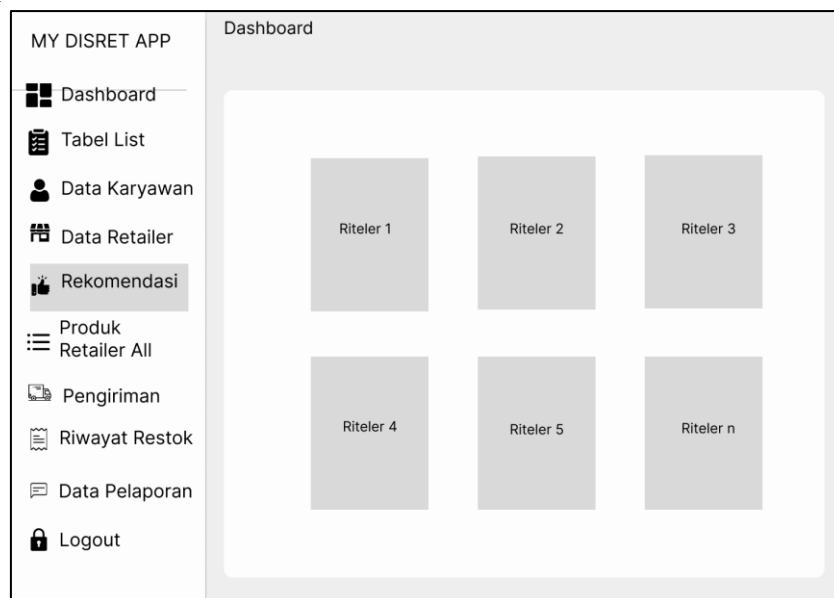


Gambar 3.21 Halaman Data Retailer

Gambar 3.21 merupakan perancangan halaman data *retailer*. pada bagian ini terdapat table *retailer* yang akan digunakan untuk menampilkan data *retailer* yang ada pada *database*. Tabel pada halaman ini memiliki beberapa *field* diantaranya no, kode *retailer*, nama *retailer*, lokasi dan sebagainya. Pada halaman ini juga akan ditambahkan beberapa fungsi untuk melakukan pencarian, memilih berapa jumlah data yang akan ditampilkan serta terdapat juga *button* tambah *retailer* yang berguna untuk menambahkan data *retailer*.

h. Halaman Rekomendasi

Halaman rekomendasi merupakan halaman yang akan digunakan untuk menampilkan data rekomendasi pengiriman kepada distributor. Berikut perancangan halaman rekomendasi dapat dilihat pada Gambar 3.22.

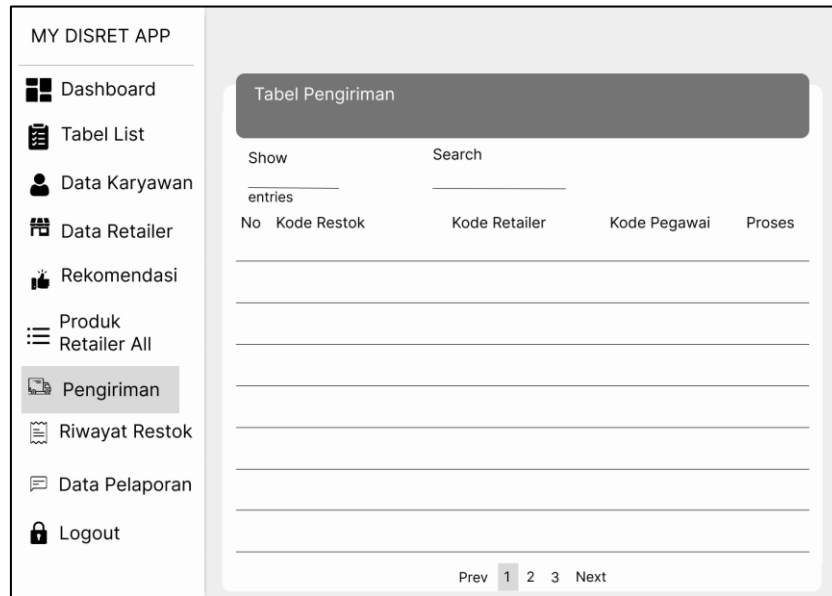


Gambar 3.22 Halaman Rekomendasi

Gambar 3.22 Merupakan perancangan halaman rekomendasi. Terdapat list data *retailer* yang akan ditampilkan dalam bentuk kolom. Setiap kolom akan ditambahkan *button* yang berguna untuk menampilkan halaman rekomendasi pada tiap-tiap *retailer*.

i. Hakaman Pengiriman

Halaman pengiriman merupakan halaman yang berguna untuk menampilkan informasi data pengiriman yang sedang berlangsung. Berikut merupakan perancangan halaman pengiriman barang. Berikut perancangan halaman pengiriman dapat dilihat pada Gambar 3.23.

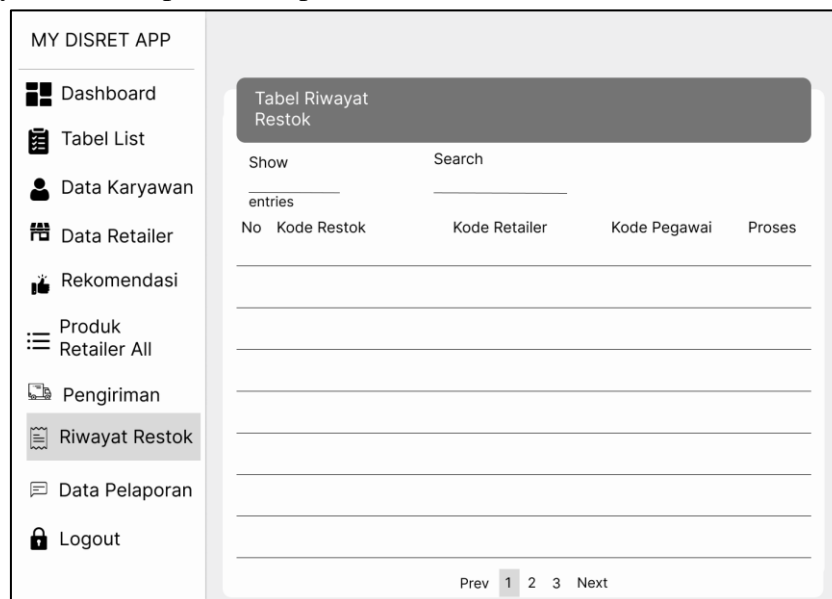


Gambar 3.23 Halaman Pengiriman

Gambar 3.23 Merupakan perancangan halaman pengiriman yang akan menampilkan tabel data pengiriman barang yang sedang berproses. Pada halaman ini data akan ditampilkan dalam bentuk tabel dengan beberapa *field* diantaranya no, kode restok, kode *retailer*, kode pegawai dan proses yang akan diisi oleh informasi status apakah pengiriman sedang diproses, sudah selesai atau dibatalkan. Selain itu pada halaman ini juga ditambahkan fungsi pencarian dan pilihan jumlah data yang ingin ditampilkan.

j. Hama Riwayat Restok

Halaman riwayat restok merupakan halaman yang akan menampilkan informasi riwayat restok yang dilakukan pihak distributor kepada pihak *retailer*. Berikut perancangan halaman riwayat restok dapat dilihat pada Gambar 3.24.

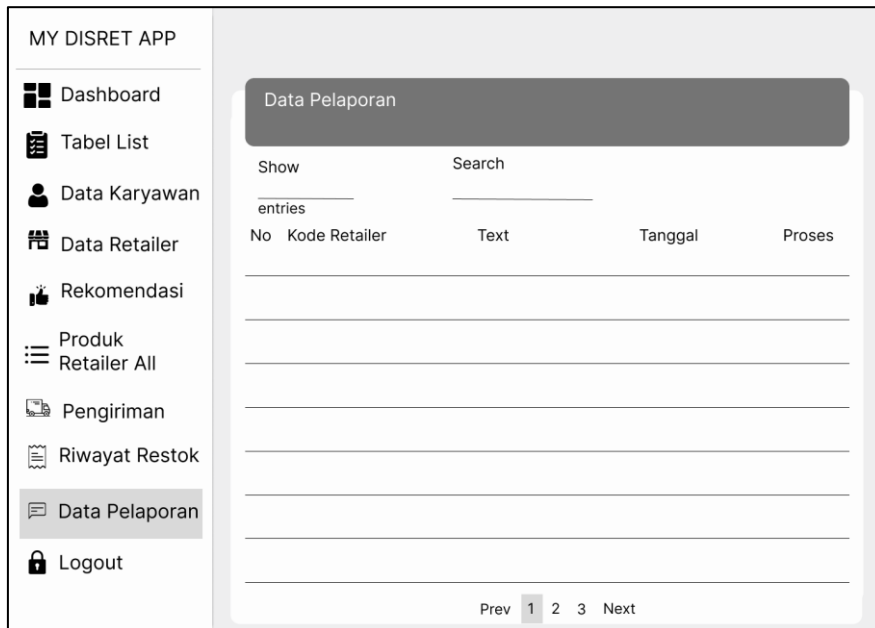


Gambar 3.24 Halaman Riwayat Restok

Gambar 3.24 Merupakan perancangan halaman riwayat restok yang didalamnya terdapat tabel yang berisi data informasi riwayat restok yang telah selesai dilakukan. Tabel pada halaman ini memiliki beberapa kolom diantaranya no, kode restok, kode *retailer*, kode pegawai dan proses, juga akan ditambahkan fungsi pencarian dan pilihan jumlah data yang ingin ditampilkan.

k. Halaman Pelaporan

Halaman pelaporan merupakan halaman yang akan menampilkan informasi data pelaporan yang telah dikirimkan pihak *retailer* kepada pihak distributor. Berikut perancangan halaman pelaporan dapat dilihat pada Gambar 3.25.

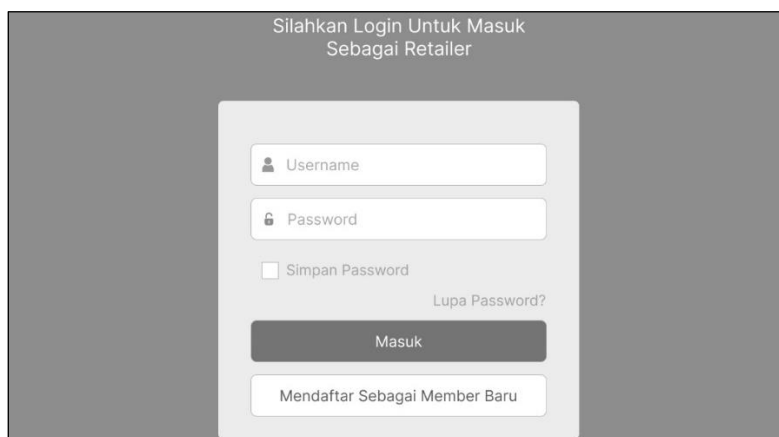


Gambar 3.25 Data Pelaporan

Gambar 3.25 Merupakan perancangan halaman data pelaporan yang akan menampilkan tabel data pelaporan yang terkirim ke distributor. Tabel pada halaman ini memiliki beberapa kolom diantaranya no, kode *retailer*, *text*, tanggal dan proses yang digunakan apabila data pelaporan telah ditangani oleh pihak distributor.

1. Halaman *Login Retailer*

Halaman *login retailer* merupakan halaman yang akan ditampilkan setelah pengguna memilih untuk masuk sebagai *retailer*. Berikut perancangan halaman *login retailer* dapat dilihat pada Gambar 3.26.

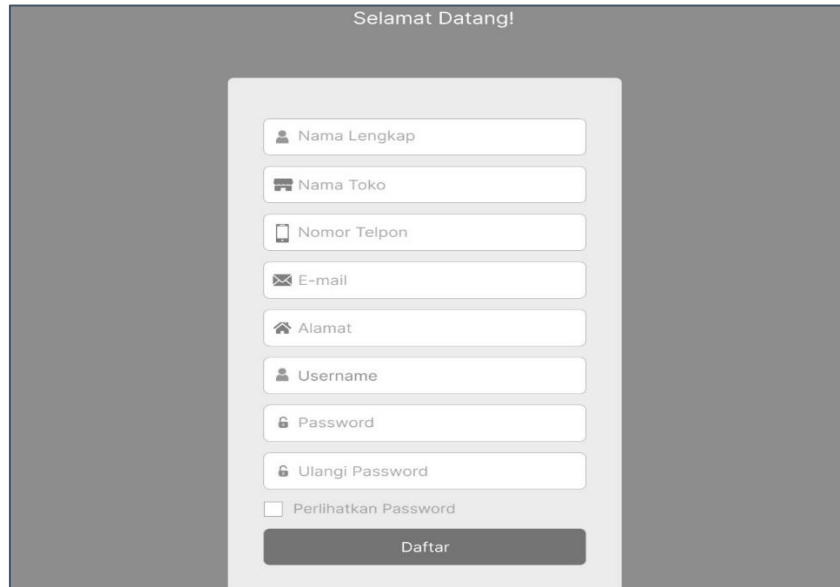


Gambar 3.26 Halaman *Login Retailer*

Gambar 3.30 merupakan gambar perancangan halaman *Login*. Terdapat *form* untuk memasukkan data *username* dan *password*. Selain itu terdapat *button* “lupa password?”, *button* masuk dan *button* mendaftar sebagai member baru.

m. Halaman Registrasi *Retailer*

Halaman registrasi *retailer* merupakan halaman yang digunakan untuk melakukan registrasi bagi *retailer* atau pengguna yang ingin bergabung. Berikut perancangan halaman registrasi untuk *retailer* dapat dilihat pada Gambar 3.27.

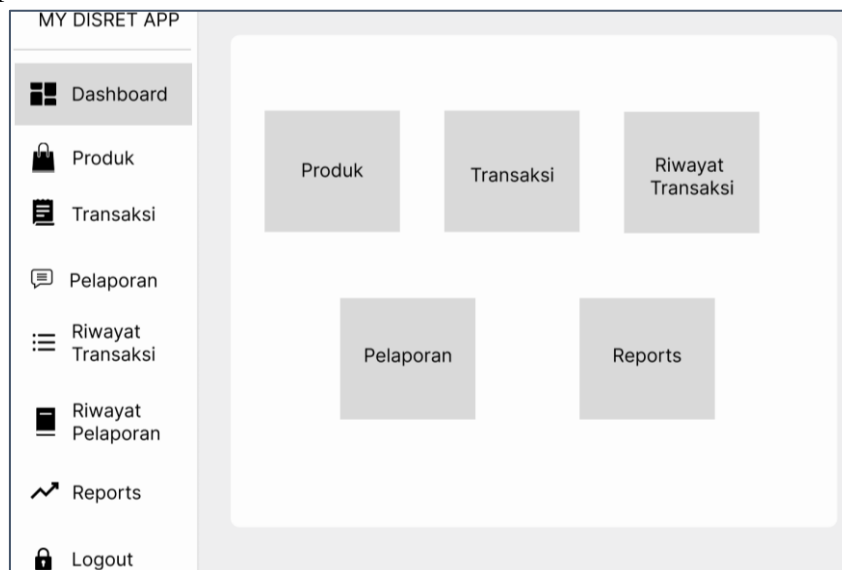


Gambar 3.27 Halaman Registrasi *Retailer*

Gambar 3.27 merupakan perancangan halaman registrasi. Pada halaman ini terdapat *form* untuk memasukkan data *retailer* seperti nama pengguna, nama toko, nomor telepon, *e-mail*, alamat, *username*, *password*, ulangi *password* dan terdapat *button* daftar untuk melakukan proses pendaftaran.

n. Halaman *Dashboard Retailer*

Halaman *dashboard retailer* merupakan halaman yang pertama ditampilkan setelah proses *login retailer* berhasil dilakukan. Berikut perancangan halaman *dashboard retailer* dapat dilihat pada Gambar 3.28.

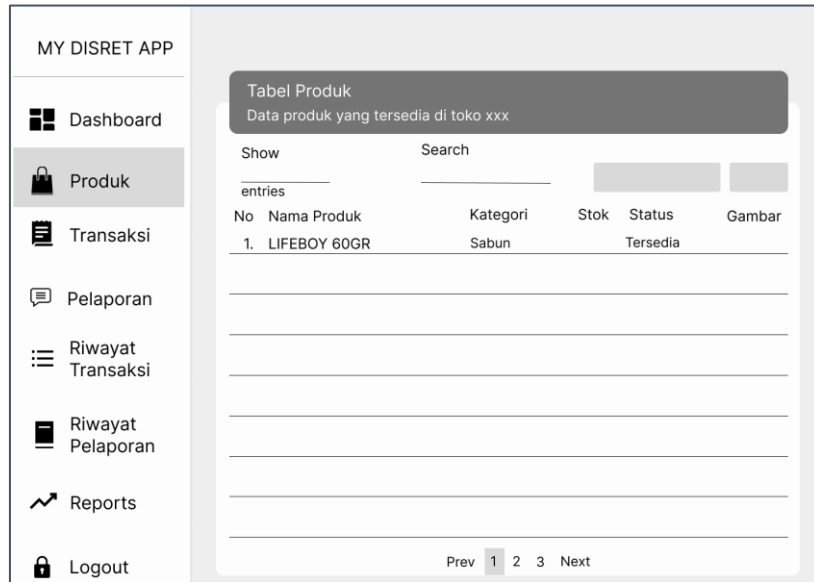


Gambar 3.28 Halaman *Dashboard Retailer*

Gambar 3.28 merupakan gambar perancangan halaman *dashboard retailer*. pada halaman ini terdapat beberapa panel menu diantaranya *dashboard*, produk, transaksi, pelaporan, riwayat transaksi, riwayat pelaporan, *reports*, dan halaman *logout*.

o. Halaman Produk *Retailer*

Halaman produk *retailer* merupakan halaman yang digunakan untuk menampilkan data produk milik *retailer*. Berikut perancangan halaman produk *retailer* dapat dilihat pada Gambar 3.29.

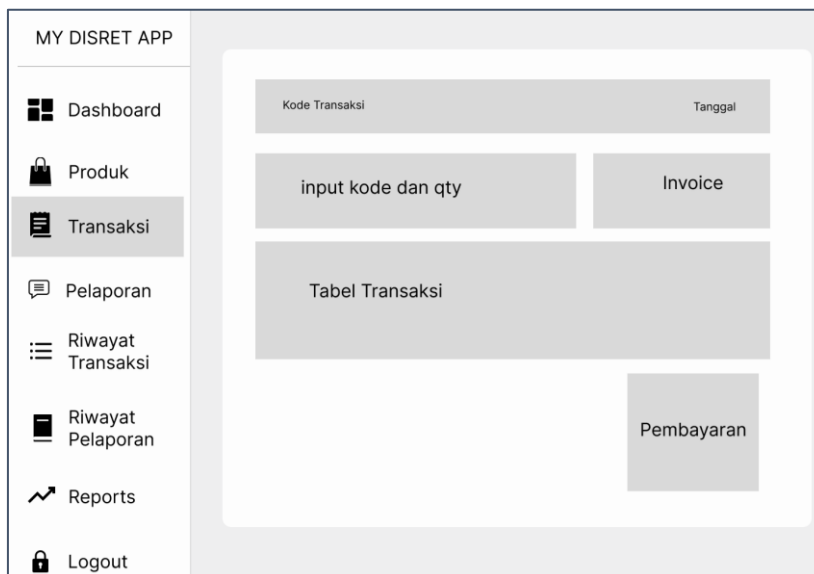


Gambar 3.29 Halaman Produk *Retailer*

Gambar 3.29 merupakan perancangan halaman produk *retailer*. Pada halaman ini terdapat tabel untuk menampilkan data produk milik *retailer*. Tabel pada halaman ini memiliki beberapa kolom diantaranya no, nama produk, kategori stok dan sebagainya. ditambahkan juga fungsi untuk melakukan pencarian, memilih berapa jumlah data yang akan ditampilkan serta memilih kolom mana saja yang akan ditampilkan.

p. Halaman Transaksi *Retailer*

Halaman transaksi *retailer* merupakan halaman untuk melakukan proses transaksi bagi *retailer*. Berikut perancangan halaman transaksi milik *retailer* dapat dilihat pada Gambar 3.30

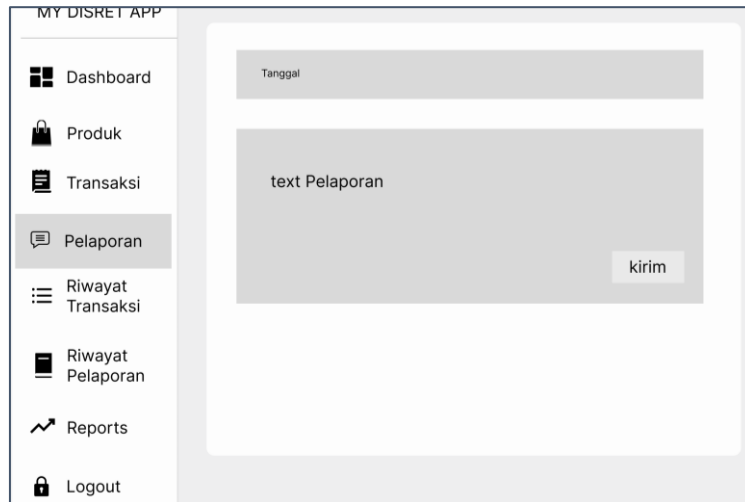


Gambar 3.30 Halaman Transaksi *Retailer*

Gambar 3.30 merupakan gambar perancangan halaman transaksi milik *retailer* yang terdapat *form* yang berisi *field* kode transaksi dan tanggal untuk memasukan tanggal kapan transaksi dilakukan. Pada halaman ini juga terdapat *form* yang berisi *field input* kode dan jumlah produk, serta *form invoice*, *form* tabel transaksi untuk melakukan proses transaksi dan *form* pembayaran transaksi yang berisi informasi harga total dan pembayaran.

q. Halaman Pelaporan *Retailer*

Halaman pelaporan *retailer* merupakan halaman yang akan digunakan untuk melakukan proses pelaporan oleh pihak *retailer* kepada pihak distributor. Berikut perancangan halaman pelaporan *retailer* dapat dilihat pada Gambar 3.31.

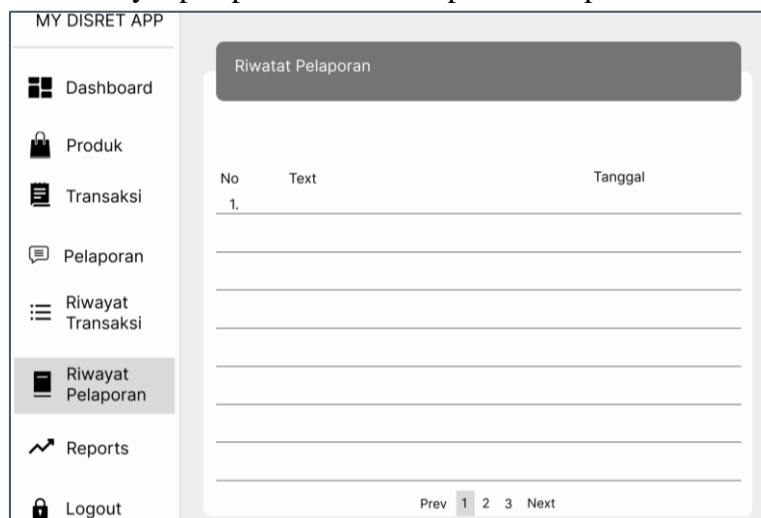


Gambar 3.31 Halaman Pelaporan *Retailer*

Gambar 3.31 merupakan perancangan halaman pelaporan data produk milik *retailer*. pada halaman ini terdapat *form* tanggal dan text untuk memasukan informasi pelaporan serta terdapat *button* kirim untuk mengirim data pelaporan yang akan disampaikan kepada pihak distributor.

r. Halaman Riwayat Pelaporan *Retailer*

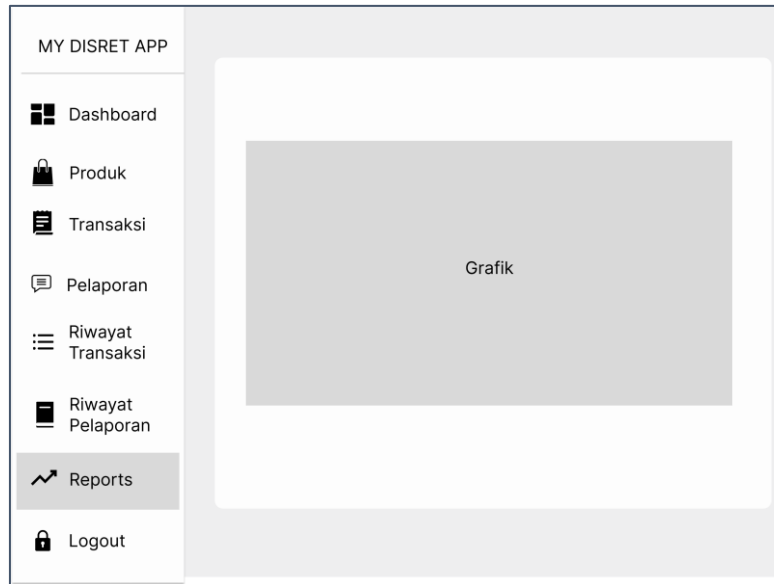
Halaman riwayat pelaporan merupakan halaman yang digunakan untuk menampilkan data riwayat pelaporan yang telah dilakukan oleh *retailer*. Pada halaman ini berisi tabel dengan beberapa kolom diantaranya no, *text* yang berisi data pelaporan dan tanggal. Berikut perancangan halaman riwayat pelaporan *retailer* dapat dilihat pada Gambar 3.32.



Gambar 3.32 Halaman Riwayat Pelaporan *Retailer*

s. Halaman Reports *Retailer*

Halaman *reports retailer* merupakan halaman yang berisi informasi *reports* singkat terkait penjualan milik *retailer*. Berikut perancangan halaman *reports retailer* dapat dilihat pada Gambar 3.33.



Gambar 3.33 Halaman Reports *Retailer*

Gambar 3.33 merupakan perancangan halaman *reports* milik *retailer*. Pada halaman ini akan ditampilkan beberapa info singkat terkait data total penjualan bersih, total dari harga pokok penjualan, total laba kotor serta kuantitas stok yang tersedia. Pada halaman ini juga ditampilkan grafik penjualan perminggu selama beberapa minggu terakhir.

3.4 Perancangan Pengujian

Tahapan ini menjelaskan mengenai pengujian terkait sistem yang akan dikembangkan. Yang mana akan dibagi menjadi dua macam pengujian diantaranya pengujian terkait fungsional aplikasi, dan pengujian terkait algoritma yang digunakan.

3.4.1 Perancangan Pengujian Fungsional Aplikasi

Perancangan pengujian dari aplikasi yang diterapkan menggunakan metode *black box testing*. Pengujian ini berfungsi untuk mengetahui ada tidaknya kesalahan pada program yang dikembangkan. Berikut merupakan tabel perancangan pengujian fungsional pada aplikasi yang dikembangkan.

Tabel 3.28 Tabel Rancangan Pengujian Fungsional Aplikasi

No.	Nama Halaman	Item Uji	Berhasil	Gagal

3.4.2 Perancangan Pengujian Algoritma

Pengujian algoritma *Fuzzy Tsukamoto* untuk menentukan rekomendasi penjualan produk dilakukan dengan pengujian *Mean Absolute Percentage Error* (MAPE) dan *Percentage Error* (PE) dengan membandingkan hasil rekomendasi restok atau produk dengan penjualan pada minggu berikutnya atau nilai asli. Berikut merupakan table perancangan yang disusun.

Tabel 3.29 Tabel Rancangan Pengujian Algoritma *Fuzzy Tsukamoto*

No.	Kode Produk	Nama Produk	Hasil Rekomendasi	Penjualan Aktual	Absolute	PE

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

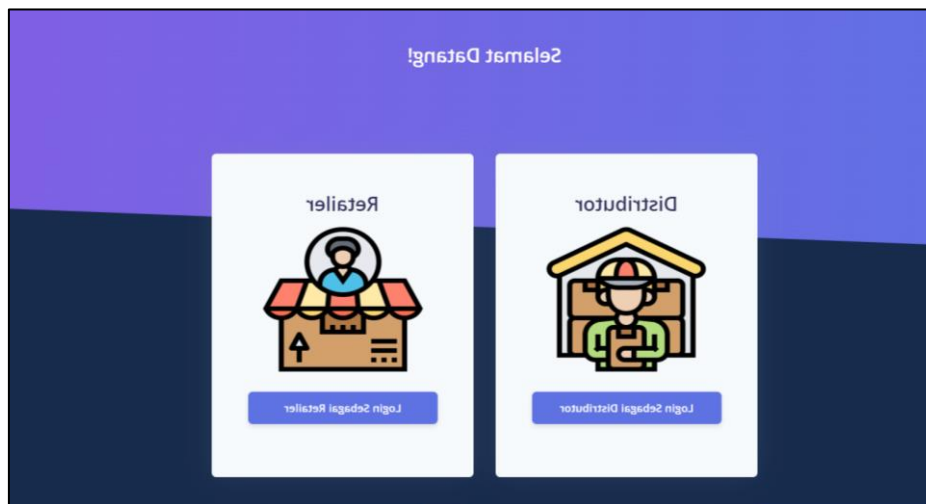
Pada bagian ini diuraikan pembahasan mengenai hasil implementasi dari perancangan sistem manajemen rantai pasok terintegrasi yang telah di susun pada bab III. Pembahasan yang di lakukan diantaranya terkait implementasi program, pengujian program dengan *black box testing* serta pengujian metode yang diterapkan menggunakan metode MAPE.

4.1.1 Hasil Perancangan Sistem

Pada bagian ini dijelaskan mengenai hasil implementasi dari perancangan yang telah disusun berupa tampilan *user interface* serta *pseudocode* yang digunakan untuk menyusunnya.

1. Halaman Utama

Halaman utama merupakan halaman yang ditampilkan ketika sistem pertama kali dijalankan. Pada halaman ini terdapat dua sub menu diantaranya menu *Login* untuk distributor dan menu *Login* untuk *retailer*. Tampilan halaman utama ditunjukkan pada Gambar 4.1.

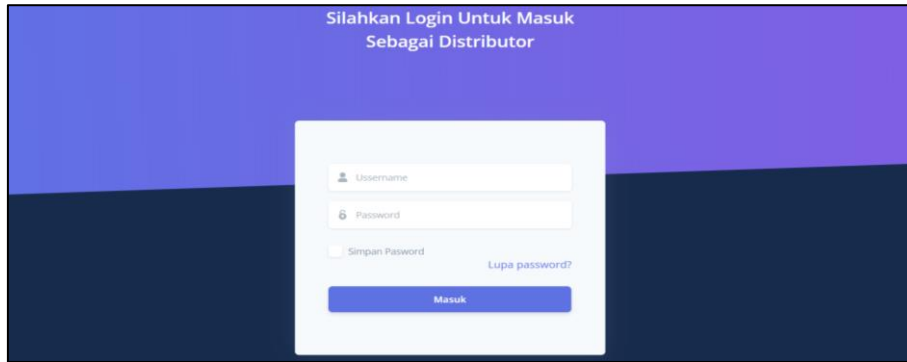


Gambar 4.1 Tampilan Halaman Utama

Pada Gambar 4.1 terdapat 2 *button* yaitu *button Login* sebagai distributor dan *button Login* sebagai *Retailer*. Pada *button Login* sebagai Distributor akan membawa kehalaman *Login* milik distributor dan sementara halaman *Login* sebagai *Retailer* akan membawa ke halaman *Login* milik *retailer*.

2. Halaman Login

Halaman *Login* merupakan halaman yang digunakan untuk masuk kehalaman sistem. Pengguna dapat melakukan *Login* dengan memasukan *username* dan *password* yang telah terdaftar. Pada sistem ini terdapat dua halaman *Login* yaitu halaman untuk distributor yang mana halaman ini hanya dapat digunakan untuk *Login* dengan data karyawan distributor aktif yang telah terdaftar dan juga halaman untuk *retailer* yang hanya dapat digunakan untuk *Login* dengan data *retailer* aktif yang telah terdaftar. Dalam halaman ini terdapat fitur lupa *password* yang akan membawa ketampilan *form* untuk mengisi email yang telah terdaftar. Tampilan halaman *Login* ditunjukkan pada Gambar 4.2.



Gambar 4.2 Tampilan Halaman Login

Proses *login* karyawan merupakan modul program yang digunakan untuk pemrosesan *login* dengan cara mencocokkan data yang dimasukkan dengan data pada *database* dengan beberapa ketentuan yang ditetapkan. Proses ini dimulai dengan membaca data *username* dan *password* yang telah dimasukkan kemudian jika data tersebut sesuai dengan data yang terdapat pada *database* serta data status adalah aktif maka akan berlanjut keproses selanjutnya yaitu, jika level adalah karyawan maka akan masuk ke halaman karyawan, dan jika level adalah admin maka akan masuk kehalaman admin. Namun jika data yang dimasukkan tidak sesuai maka akan terdapat pemberitahuan *login* gagal dan akan kembali ke halaman *login*. Berikut adalah modul program dari proses *login* karyawan.

Algoritma 1: karyawan Login process

```

Input: login data
Output: login information, privileges interface
Method:
Get: username, password, status, season level
if username = true and password = true and level = true and status = active
    if level = admin
        print admin interface
    endif
    if level = karyawan
        print karyawan interface
    endif
else
    print "Login Gagal"
    print login interface
endif

```

Modul Program 4.1 Pseudocode Proses Login Karyawan

Proses *login retailer* merupakan modul yang digunakan untuk pemrosesan *login* bagi *retailer* dengan membaca *username* dan *password* kemudian mencocokkan dengan data yang ada dalam *database*. Jika data sesuai, level sama dengan *retailer* dan status adalah aktif maka akan ditampilkan halaman *retailer*. Namun, jika data yang dimasukkan tidak sesuai maka akan terdapat pemberitahuan *login* gagal dan akan kembali ke halaman *login*. Berikut adalah modul program dari proses *login retailer*.

Algoritma 2: Retailer Login Process

```

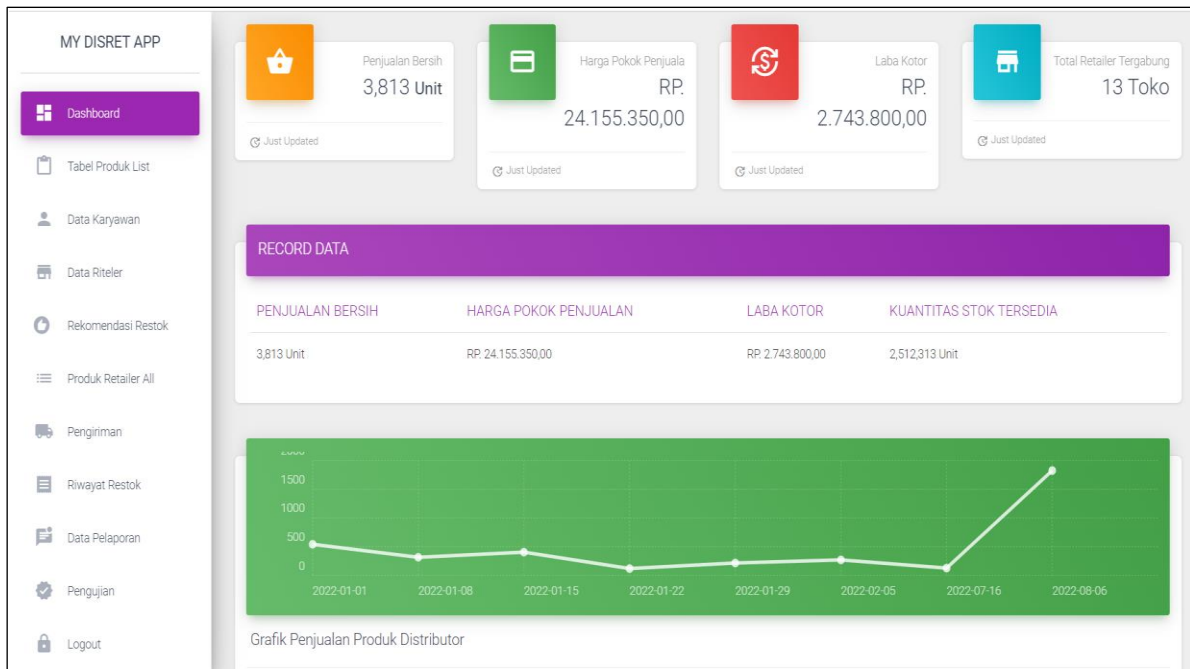
Input: login data
Output: login information, privileges interface
Method:
Get: username, password, status, season level
if username=true and password=true and level = retailer and status = active
    print Retailer interface
else
    print "Login Gagal"
    print login interface
endif

```

Modul Program 4.2 Pseudo Code Proses Login Retailer

3. Halaman *Dashboard* Distributor

Halaman *dashboard* distributor merupakan halaman yang ditampilkan pertama kali setelah proses *login* berhasil. Pada halaman ini terdapat beberapa tampilan informasi singkat mengenai data distributor. Diantaranya adalah jumlah penjualan bersih, jumlah pemasukan kotor, laba kotor serta jumlah *retailer* yang telah bergabung. Pada halaman ini juga ditampilkan grafik penjualan produk selama delapan minggu terakhir. Tampilan halaman *dashboard* distributor dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan Halaman *Dashboard* Distributor

Tampilan *dashboard* distributor merupakan modul program yang digunakan untuk menampilkan data informasi distributor dengan membaca data produk, data transaksi, dan data restok pada *database* kemudian melakukan penjumlahan total masing-masing jumlah penjualan, harga, keuntungan serta banyak *retailer* dengan melakukan perulangan *for* yang dimulai dari $x=1$ dimana selama nilai x belum mencapai nomor maka akan dilakukan penjumlahan data, setelah perulangan selesai data tersebut ditampilkan dalam bentuk kolom, table serta grafik seperti gambar 4.3. Berikut modul program dari proses untuk menampilkan informasi data *dashboard* distributor.

Algoritma 3: View Distributor Dashboard

Output: summary distributor information

Method:

Get: product, transaction, restoc

Number x, number, sold, cost, profit, retailer

```

For x= 1 to number step 1 do
    sumSold=sumSold+sold
    sumCost=sumCost+cost
    sumProfit=sumProfit+profit
    sumRetailer=sumRetailer+retailer
endfor

print sumSold
print sumCost
print sumProfit
print sumRetailer
print sumSold graph
    
```

Modul Program 4.3 Pseudo Code Tampilan Informasi Data *Dashboard*

4. Halaman Tabel Produk *List*

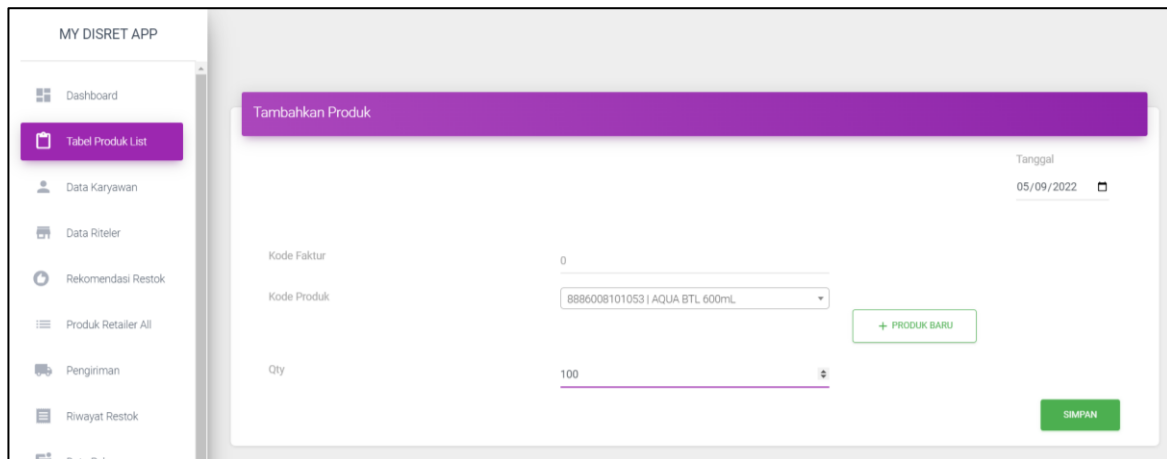
Halaman table produk *list* merupakan halaman yang digunakan untuk menampilkan daftar produk milik distributor. Pada bagian ini data yang ditampilkan berbentuk tabel dengan 16 kolom diantaranya kolom no, nama produk, kode produk, kategori, harga beli yang merupakan harga produk dibeli dari pihak pemasok, harga distribusi yaitu harga produk ketika dijual ke pihak *retailer*, serta harga jual *retailer* yaitu harga produk dari pihak *retailer* yang dijual ke konsumen tingkat akhir. Dalam halaman ini juga terdapat beberapa fungsi tambahan diantaranya fungsi *show* yang digunakan untuk memilih berapa jumlah produk yang ingin ditampilkan, fungsi *search* yang berguna untuk melakukan pencarian produk yang ingin ditampilkan, fungsi pengaturan yang digunakan untuk mengatur kolom apa saja yang ingin ditampilkan, kemudian ada *button* tambah produk yang berguna untuk menambahkan data produk. Berikut halaman table produk *list* dapat dilihat pada Gambar 4.4.

No	Nama Produk	Kode Produk	Pemasok	Kategori	Harga Beli	Harga Distribusi	Harga Jual	Satuan	Stok	Diskon	Status	Tanggal Kadaluarsa	Terakhir diperbaharu	Gambar	Hapus
1.	GULA PASIR 1KG	10020031	PT DELTA GUNA UTAMA - UNILEVER	gula	11000	12500	14000	PCS	100032		Tersedia	30 November -0001	2022-09-05		HAPUS
2.	GARAM MEJA CAP DAUN 250G	29731		bumbu	4000	4500	5000	PCS	100048		Tersedia	30 November -0001	2022-09-05		HAPUS
3.	PANTENE 10ML	4902430563871		shampoo	850	900	1000	PCS	100060		Tersedia	30 November -0001	2022-07-09		HAPUS
4.	AQUA BTL 600mL	8886008101053		minuman	2000	2350	3000	PCS	100119		Tersedia	30 November -0001	2022-09-05		HAPUS
5.	INDOMIE AYAM	89686010015		mie instan	2000	2400	2800	PCS	100164		Tersedia	30 November -0001	2022-07-09		HAPUS

Gambar 4.4 Tampilan Halaman *Dashboard* Distributor

5. Halaman Tambah Produk

Halaman tambah produk merupakan halaman yang berguna untuk menambahkan data produk. Pada halaman ini terdapat *form* pengisian yang terdiri dari beberapa *field* diantaranya *field* tanggal yang berguna untuk memasukan tanggal penambahan produk, tanggal ini akan otomatis mengambil tanggal saat ini, namun dapat dilakukan pengeditan jika ingin memasukan tanggal lainnya, selanjutnya terdapat *field* kode faktur untuk memasukan kode faktur yang di dapat dari pihak pemasok, lalu *field* kode produk dengan fungsi pencarian untuk memilih produk mana yang akan dilakukan penambahan data, serta *field* qnt untuk memasukan berapa jumlah data yang ingin ditambahkan. Dalam halaman ini hanya dapat dilakukan penambahan atau pengeditan jumlah kuantitas produk sementara untuk menambahkan jenis produk baru terdapat *button* produk baru. *Button* ini akan membawa ke halaman *input* produk. Berikut gambar untuk tampilan halaman tambah produk dapat dilihat pada Gambar 4.5.



Gambar 4.5 Tampilan Halaman Tambah Produk

Tambah jumlah produk merupakan modul yang berguna untuk menambahkan data kuantitas produk. Proses dalam modul ini dimulai dengan membaca data yang telah dimasukan. Pada bagian *field* kode produk dilakukan pembacaan data yang dimasukan. Jika data yang dimasukan sesuai dengan data nama produk atau data kode produk maka data nama dan kode produk akan ditampilkan namu jika tidak sesuai maka akan tampil info pemberitahuan *no results found*. Kemudian perintah yang dikerjakan ketika *button* simpan diklik adalah pengkoreksian *field* yaitu jika *field* tidak sama dengan *null* atau tidak kosong maka akan dilakukan proses *input* data restok produk distributor kemudian dilakukan pencarian atau pencocokan data menggunakan fungsi *while* yaitu selama nomor data produk kurang atau sama dengan jumlah data produk dilakukan perintah *if* atau jika kode produk yang dibaca sama dengan kode produk yang dimasukan maka *update* jumlah kuantitas produk dengan menambahkan kuantitas produk awal dengan nilai kuantitas yang dimasukan. Berikut merupakan modul program dari proses penambahan jumlah data produk.

Algoritma 4: Add Product Quantity

Input: product code, product name, qty, date, product, facture code

Method:

Get: product code, product name, qty, date, product, facture code

While product data number <= count product data

If product code = true or product name = true

Print product code

Print product name

Else

Print "no results found"

End if

End while

If facture code not null and product code not null date not null

Input: facture code, date

Read product code

While product data number <= count product data

If product code = true

Update: quantity =quantity + qty

End If

End while

Else

Print "please fill out this field"

Modul Program 4.4 Pseudo Code Tambah Jumlah Data Produk

6. Halaman *Input* Produk

Halaman *input* produk merupakan halaman yang berguna untuk menambahkan data produk baru ke *database*. Pada halaman ini ditampilkan *form* pengisian data produk dengan beberapa *field* pengisian diantaranya kode produk, nama produk, harga beli, harga distributor,

harga jual *retailer*, serta beberapa fungsi *dropdown* untuk melakukan pencarian, menampilkan serta menyimpan data kedalam fungsi *field* diantaranya *dropdown* kategori, satuan, diskon, dan *supplier*. Pada halaman ini terdapat juga beberapa *button* untuk menambahkan data ketika data yang diinginkan belum terdapat pada *database* diantaranya *button* kategori baru untuk menambah data kategori baru, *button* satuan baru untuk menambahkan data satuan, *button* diskon baru untuk menambahkan data diskon, serta *button supplier* baru untuk menambahkan data pemasok. Pada akhir *form* terdapat *button* simpan yang akan membawa pada proses penyimpanan data. Berikut tampilan *input* produk dapat dilihat pada Gambar 4.6.

Gambar 4.6 Tampilan Halaman *Input* Produk

Input produk merupakan modul yang digunakan untuk menambahkan data produk pada *database*. Proses pada modul ini dimulai dengan membaca data masukan berupa data kode produk, nama produk, kategori, tanggal, gambar, status, harga beli, harga jual, harga jual *retailer*, satuan, diskon dan pemasok. Kemudian dikerjakan fungsi *if* yaitu jika data kode produk tidak kosong maka akan dilakukan proses selanjutnya namun jika koseong makan akan ditampilkan pemebritahuan untuk mengisi *field* kode produk. Sementara proses yang dikerjakan jika yaitu perulangan *while* dimana selama data nomor produk kurang dari atau sama dengan cumlah seluruh data produk dikerjakan perintah untuk mengambil data kode produk. Selanjutnya, dikerjakan fungsi *if* yaitu jika kode produk yang didapatkan dari proses sebelumnya tidak sama dengan *null* maka akan di tampilkan pemberitahuan bahwa data kode produk telah ada dalam *database* namun jika tidak maka dikerjakan proses *input* data kode produk, nama produk, kategori, tanggal, gambar, status, harga beli, harga jual, harga jual

retailer, satuan, diskon dan pemasok. Berikut merupakan modul program dari proses *input* data produk.

Algoritma 5: Input Product

```

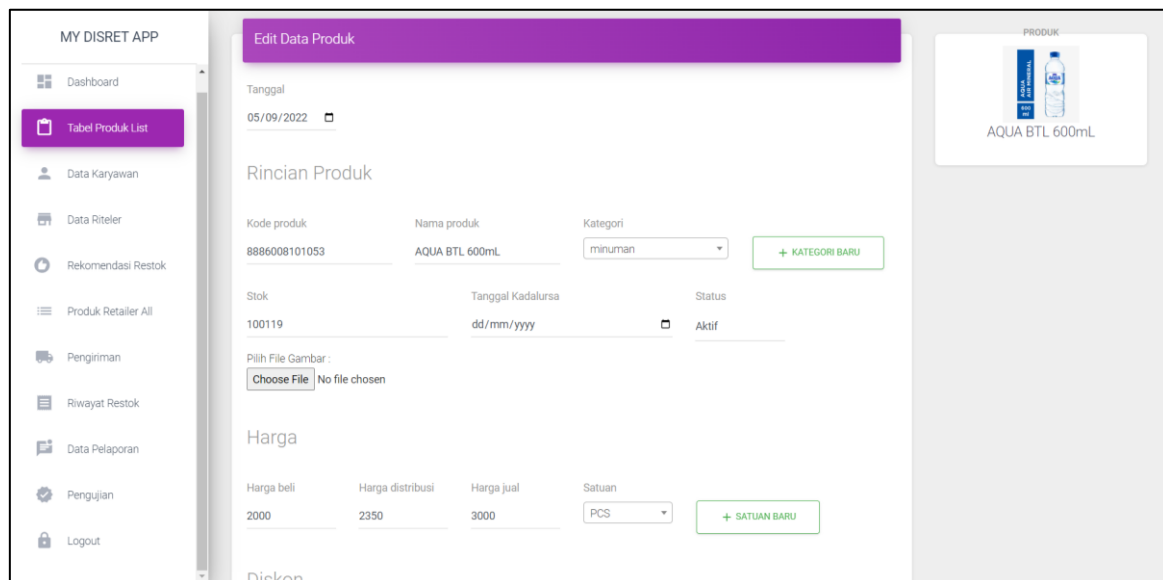
Input: product code, product name, category code, date, image file, status
code, purchase price, sale price, retail sale price, unit code, discount
code, supplier code
Method:
Get: product code, product name, category code, date, image file, status
code, purchase price, sale price, retail sale price, unit code,
discount code, supplier code
If product code not null
  While product data number <= count product data
    Get product code
  End while
  If product code = null
    Input product code, product name, category code, date, image file,
status code, purchase price, sale price, retail sale price, unit
code, discount code, supplier code
  Else
    Print "kode produk telah ada dalam databse"
  Endif
Else
  Print "please fill out this field"
Endif

```

Modul Program 4.5 Pseudo Code Input Data Produk

7. Halaman *Edit* Data Produk

Halaman edit data produk berfungsi untuk melakukan pengeditan data produk yang telah tersimpan dalam *database*. Pada bagian ini terdapat *form* dengan *field* yang sama dengan penjelasan pada nomor 6 namun pada halaman ini *field* telah berisi *default* sesuai dengan id yang berupa kode produk yang telah dibawa dari halaman sebelumnya, isi *fields* dalam *form* ini dapat diubah sesuai oleh pengguna. Pada halaman ini juga terdapat *button* simpan untuk menyimpan perubahan data. Berikut tampilan halaman *edit* data produk dapat dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan Halaman *Edit* Produk

Edit produk merupakan modul yang digunakan untuk melakukan perubahan data produk pada *database*. Proses pada modul ini dimulai dengan membaca data masukan berupa id yaitu kode produk yang akan dicocokkan ke *database* jika data sesuai maka data dengan kode produk yang dimasukkan akan ditampilkan. Data dalam halaman ini dapat diubah kemudian

disimpan. Pada proses penyimpanan, akan dilakukan pengambilan atau pembacaan data yang telah dimasukan berupa data kode produk, nama produk, kategori, tanggal, gambar, status, harga beli, harga jual, harga jual *retailer*, satuan, diskon dan pemasok. Kemudian dikerjakan fungsi *if* yaitu jika data kode produk tidak kosong maka akan dilakukan proses selanjutnya namun jika kosong maka akan ditampilkan pemberitahuan untuk mengisi *field* kode produk. Sementara proses yang dikerjakan yaitu perulangan *while* dimana selama data nomor produk kurang dari atau sama dengan jumlah seluruh data produk dikerjakan proses pengambilan data kode produk, data ini akan di cocokan menggunakan fungsi *if* yaitu jika kode produk yang didapatkan sesuai dengan kode produk yang di masukan maka data dengan kode produk tersebut akan di *update*. Berikut merupakan modul program dari proses *edit* data produk.

Algoritma 6: Edit Product

```

Output: product code, product name, category code, date, image file, status
           code, purchase price, sale price, retail sale price, unit code,
           discount code, supplier code
Input : product data edit
Method:
Get: product code
If product code = true
Output: product code, product name, category code, date, image file, status
           code, purchase price, sale price, retail sale price, unit code,
           discount code, supplier code

Get: product code, product name, category code, date, image file, status
           code, purchase price, sale price, retail sale price, unit code,
           discount code, supplier code
If product code not null

    While product data number <= count product data
        Get product code AS cp
    End while
        If cp=product code
            Update product code, product name, category code, date,
            image file, status code, purchase price, sale price, retail sale
            price, unit code, discount code, supplier code
        Else
            Print "kode produk tidak sesuai"
        End if
    Else
        Print "please fill out this field"
    Endif
Endif

```

Modul Program 4.6 Pseudo Code Edit Data Produk

8. Halaman Data Karyawan

Halaman data karyawan merupakan halaman yang berfungsi untuk menampilkan data karyawan. Dalam halaman ini terdapat table dengan beberapa kolom yaitu no, nama, *username*, jenis kelamin, alamat, tlp, email, *role* yaitu peranan yang dibedakan menjadi dua peran diantaranya peran admin dan karyawan, status yang dibagi menjadi status aktif dan status non aktif, foto serta hapus yang berguna untuk melakukan fungsi hapus data. Dalam table ini data dengan status aktif akan berwarna hitam sementara data dengan status non aktif akan berwarna merah. Dalam halaman ini juga memiliki fungsi pencarian dan pilihan jumlah data yang ingin ditampilkan, kemudian terdapat *button* tambah data karyawan untuk menambahkan data karyawan baru. Berikut tampilan halaman data karyawan dapat dilihat pada Gambar 4.8.

No	Nama Karyawan	Username	Jenis Kelamin	Tanggal Lahir	Alamat	Tlp	e-mail	Role	Status	Foto	Hapus
1.	admin	admin	Laki-laki	1999-02-10	boyolali	085646104333	admin@gmail.com	Admin	Aktif		HAPUS
2.	admin2	admin2	Laki-laki	2000-01-01	slaman	081646123643	admin2@gmail.com	Admin	Non Aktif		HAPUS
3.	pegawai	pegawai	Laki-laki	1995-02-06	yogyakarta	087656432108	pegawai@gmail.com	Pegawai	Aktif		HAPUS
4.	rena	pegawai2	Perempuan	1993-07-07	yogyakarta	085432819111	rena@gmail.com	Pegawai	Aktif		HAPUS

Gambar 4.8 Tampilan Halaman Tabel Karyawan

9. Halaman Tabel *Retailer*

Halaman tabel *retailer* merupakan halaman yang berguna untuk menampilkan data *retailer* yang terdaftar. Tabel ini memiliki 11 kolom diantaranya no, nama *retailer*, nama produk, *username* yang digunakan oleh *retailer*, lokasi, noTlp, email, saldo, kredit, status yang terdiri dari status aktif dan non aktif, ketika status akun aktif maka akan berwarna hijau dan berwarna merah jika setatus non-aktif serta kolom hapus data. Pada halaman ini juga terdapat fungsi pencarian dan fungsi untuk memilih jumlah data yang ingin ditampilkan, serta terdapat *button* tambah *retailer*. Pada setiap baris tabel memiliki fungsi referensi hiperteks atau *herf* yang ketika di klik akan membawa ke halaman detail produk yang dipilih. Berikut tampilan halaman tabel *retailer* dapat dilihat pada Gambar 4.9.

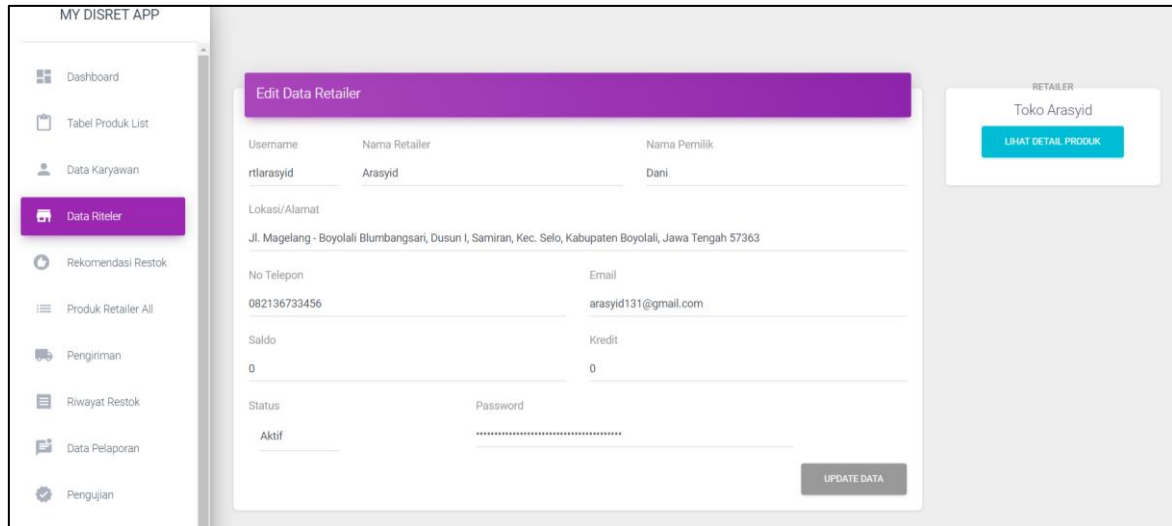
No	Nama Retailer	Nama Pemilik	Username	Lokasi	No Tlp	Email	Saldo	Kredit	Status	Hapus Data
1.	AA	Ning	rtlaa	Jl. Magelang - Boyolali Blumbangsari, Dusun IV, Samiran, Kec. Selo, Kabupaten Boyolali, Jawa Tengah 57363	085641223980	ningrum789@gmail.com	0	0	Aktif	HAPUS
2.	Adhila	Harmi	rtladhila	Jl. Muntian-Boyolali, Dusun V, Samiran, Kec. Selo, Kabupaten Boyolali, Jawa Tengah 57363	85647502327	adhila67@gmail.com	0	0	Aktif	HAPUS
3.	Arahma	Rahma Adinda	rtlarahma	semarang	085646323122	rahma@gmail.com	0	0	Non Aktif	HAPUS
4.	Arasyid	Dani	rtlarasyid	Jl. Magelang - Boyolali Blumbangsari, Dusun I, Samiran, Kec. Selo, Kabupaten Boyolali, Jawa Tengah 57363	082136733456	arasyid131@gmail.com	0	0	Aktif	HAPUS
5.	Endang	Endang	rtlendang	Jalan Boyolali-Magelang Blumbangsari Samiran Selo, Dusun IV, Samiran, Kec. Selo, Kabupaten Boyolali, Jawa Tengah 57363	087836002456	endang44434@gmail.com	0	0	Aktif	HAPUS
6.	Kami	Kami	rtkami	Dusun II, Samiran, Kec. Selo, Kabupaten Boyolali, Jawa Tengah 57363	87836747227	kami56@gmail.com	0	0	Aktif	HAPUS

Gambar 4.9 Tampilan Halaman Tabel Retailer

10. Halaman *Edit Data Retailer*

Halaman *edit data retailer* merupakan halaman yang berguna untuk melakukan perubahan pada data *retailer*. Pada halaman ini akan ditampilkan data *retailer* yang telah dipilih pada halaman sebelumnya. Data *retailer* yang ditampilkan adalah *default* dari data yang telah tersimpan. Pada halaman ini terdapat *form* yang terdiri dari 10 *field* diantaranya *username*,

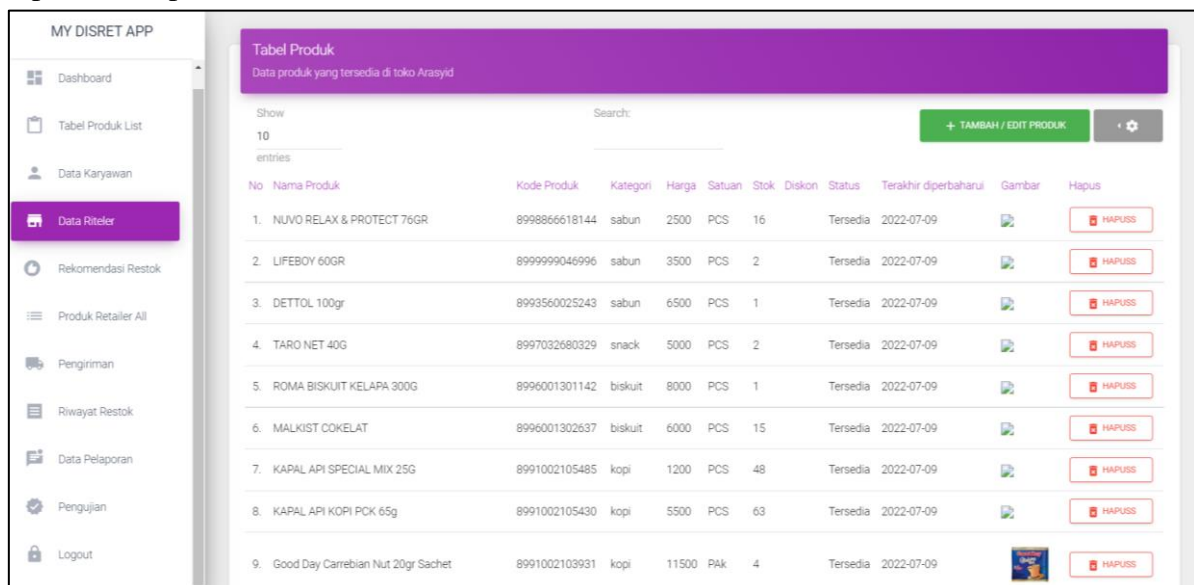
nama *retailer*, nama pemilik, lokasi/alamat, no telepon, email, status serta *password*. Pada bagian akhir *form* terdapat *button update* data yang berguna untuk menyimpan perubahan data. Pada halaman ini juga terdapat *button* lihat data produk pada bagian samping *form*. *Button* ini berfungsi untuk membuka halaman detail produk yang dimiliki *retailer* pada halaman *edit*. Tampilan halaman *edit* data *retailer* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Tampilan Halaman *Edit Data Retailer*

11. Halaman Tabel Produk *Retailer*

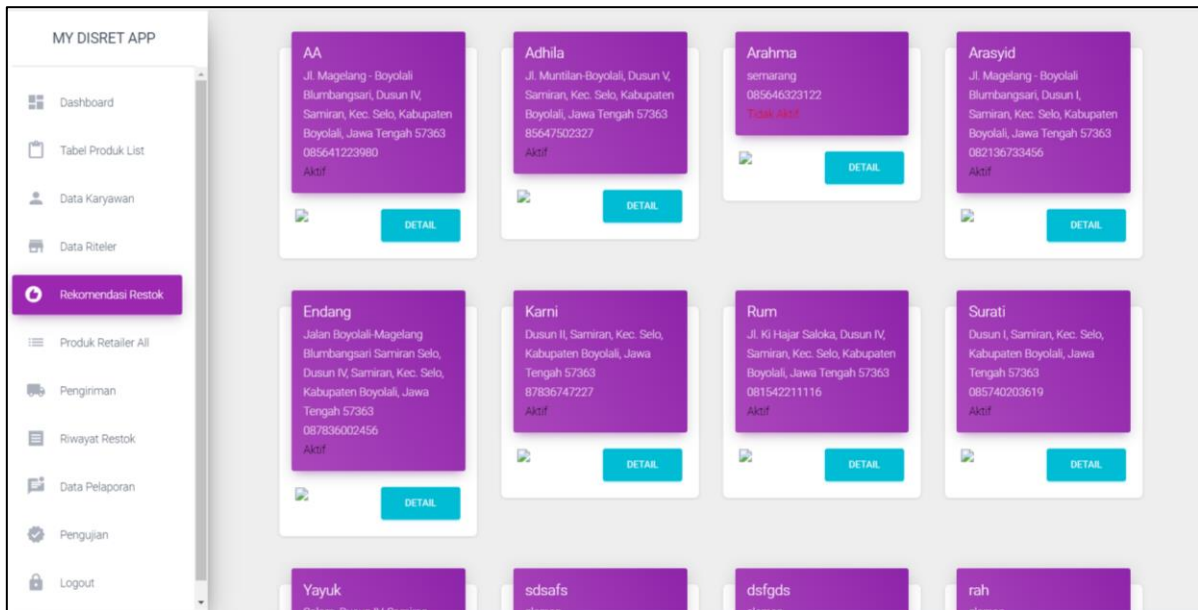
Halaman tabel produk *retailer* merupakan halaman yang berguna untuk menampilkan data produk milik *retailer* sesuai dengan *username retailer* yang telah dipilih pada halaman sebelumnya. Tabel pada halaman ini memiliki 12 kolom diantaranya kolom no, nama produk, kode produk, kategori, harga, satuan, stok, diskon, status, terakhir diperbaharui, gambar, dan hapus. Pada halaman ini juga memiliki beberapa fungsi diantaranya fungsi pencarian, fungsi pilihan untuk menampilkan berapa jumlah data yang ingin ditampilkan, serta fungsi pengaturan untuk memilih kolom apa saja yang ingin ditampilkan. Pada halaman ini terdapat *button* untuk melakukan tambah atau ubah data untuk melakukan proses pengubahan data. Data yang dapat diubah diantaranya data jumlah stok dan data diskon. Berikut tampilan halaman tabel produk dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tampilan Halaman Tabel Produk *Retailer*

12. Halaman Rekomendasi Restok

Halama rekomendasi merupakan halaman yang berguna untuk menampilkan rekomendasi bagi distributor untuk menentukan jumlah pengiriman barang. Namun, sebelum menuju pada halaman data rekomendasi terlebih dahulu ditampilkan halaman untuk memilih *retailer* mana yang akan ditampilkan. Pada halaman ini menampilkan *list retailer* yang telah terdaftar. Info yang ditampilkan berupa nama *retailer*, lokasi, nomor telepon serta status aktif atau non-aktif. Pada setiap kotaknya terdapat *button* detail untuk menampilkan halaman data rekomendasi dari *retailer* yang dipilih. Berikut tampilan halaman rekomendasi restok dapat dilihat pada Gambar 4.12.



Gambar 4.12 Tampilan Halaman Rekomendasi Restok

13. Halaman Rekomendasi Distribusi Produk

Halaman rekomendasi distribusi produk merupakan halaman yang berguna untuk menampilkan data rekomendasi jumlah barang yang akan didistribusikan kepada *retailer* yang telah dipilih pada halaman sebelumnya. Pada halaman ini terdapat beberapa informasi yang ditampilkan diantaranya nomor faktur yang telah otomatis dibuat oleh sistem, tanggal faktur yaitu tanggal waktu sekarang, serta tabel yang menampilkan data rekomendasi. Tabel ini terdiri dari beberapa *field* diantaranya no, kode item, nama item, stok saat ini yang menampilkan jumlah stok yang dimiliki oleh *retailer*, qnt yaitu jumlah rekomendasi jumlah produk yang akan dikirim, harga satuan, subtotal serta fungsi *edit* yang di dalamnya terdiri dari *button edit* qnt atau jumlah produk yang akan dikirim serta *button* hapus. Pada bagian akhir tabel terdapat baris jumlah total yang merupakan info total harga barang yang akan dikirimkan. Dalam halaman ini juga terdapat *button* tambah item yang berguna untuk menambahkan data produk baru yang belum ada dalam list. Pada bagian terakhir terdapat *button* setuju yang berguna untuk menyimpan data untuk pengiriman yang ketika telah disetujui. Berikut tampilan halaman data rekomendasi distribusi produk dapat dilihat pada Gambar 4.13.

Rekomendasi Distribusi Produk							
Arasyid							
Kode. Faktur : dis/rtlarasyid/05/09/2022/001							
Tgl. Faktur : 05/09/22							
							+ TAMBAH ITEM
No	Kode Item	Nama Item	Stok Saat Ini	Qnt	Harga Satuan + PPN	Sub Total	Edit
1.	10020031	GULA PASIR 1KG	17	51	Rp. 12.500,00	Rp. 637.500,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
2.	8886008101053	AQUA BTL 600mL	21	56	Rp. 2.350,00	Rp. 131.600,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
3.	89686010015	INDOMIE AYAM BAWANG	4	35	Rp. 2.400,00	Rp. 84.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
4.	8991002103931	Good Day Caribbean Nut 20gr Sachet	4	4	Rp. 10.500,00	Rp. 42.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
5.	8993296101112	CAKRA KEMBAR 1KG EKONOMI	0	5	Rp. 10.000,00	Rp. 50.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
6.	8993296301116	BOGASARI SEGITIGA BIRU 1KG	8	9	Rp. 10.000,00	Rp. 90.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
7.	8993560025243	DETTOL 100gr	1	2	Rp. 5.750,00	Rp. 11.500,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
8.	8995177101112	GULAKU PCK 1KG	4	5	Rp. 12.500,00	Rp. 62.500,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
9.	8996001301142	ROMA BISKUIT KELAPA 300G	1	6	Rp. 7.200,00	Rp. 43.200,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
10.	8996001600146	TEH PUCUK HARUM 350ML	2	17	Rp. 2.500,00	Rp. 42.500,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
11.	8997032680329	TARO NET 40G	2	23	Rp. 4.200,00	Rp. 96.600,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
12.	8998866200318	SEDAP MIE AYAM BAWANG	10	35	Rp. 2.300,00	Rp. 80.500,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
13.	899886610070	SD KLINI deterjen 275 gram	2	7	Rp. 5.000,00	Rp. 35.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
14.	8999999046996	LIFEBY 60GR	2	6	Rp. 3.000,00	Rp. 18.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
15.	9311931024036	INDOCAFE COFFEE MIX 3 IN 1PCK 10X20g	-2	4	Rp. 12.000,00	Rp. 48.000,00	<input type="button" value="EDIT QNT"/> <input type="button" value="HAPUS"/>
Jumlah Total						Rp. 1.472.900,00	<input type="button" value="SETUJUI"/>

Gambar 4.13 Tampilan Halaman Rekomendasi Distribusi Produk

Halaman rekomendasi menampilkan rekomendasi jumlah barang yang akan didistribusikan. Untuk menampilkan jumlah rekomendasi digunakan algoritma *Fuzzy Tsukamoto* untuk menghitung jumlah rekomendasi optimal. Perhitungan dengan algoritma ini dimulai dengan mendapatkan id *retailer* yang akan didistribusikan. Id ini didapatkan ketika pengguna memilih *retailer* pada halaman sebelumnya. Kemudian id yang didapatkan digunakan untuk proses *select* data *retailer* dengan id tersebut. Setelah data *retailer* didapatkan maka akan dilanjutkan ke proses perhitungan *Fuzzy*. Pada proses perhitungan *Fuzzy* yang pertama dilakukan adalah mendapatkan nilai masukan yang akan digunakan sebagai variabel *Fuzzy*. Nilai ini diantaranya minimal restok, maksimal restok, minimal transaksi, maksimal transaksi, minimal stok, maksimal stok, rata-rata transaksi serta rata-rata stok. Yang disimpan pada variabel *array* v_{Min} , v_{Max} dan *null*.

Proses selanjutnya adalah pembentukan basis pengetahuan atau *rule* dengan membuat empat variabel awal untuk menyimpan nilai diantaranya *high* yang akan menyimpan nilai 1, *low* menyimpan nilai 0, *and* menyimpan nilai 0 dan *or* menyimpan nilai 1. Selanjutnya pembuatan *rule* disimpan dalam empat variabel lain yaitu *transaction*, *oprator*, *stock* dan *restock* dalam bentuk *array* ke-0 sampai 3. *array* ke-0 adalah *rule* pertama yaitu [R1] *IF transaction is High and stock is Low THAN restock is High*. *Rule* kedua yaitu [R2] *IF transaction is High and stock is High THAN restock is High*. *Rule* ketiga yaitu [R3] *IF transaction is Low and stock is Low THAN restock is Low*. Dan *rule* keempat yaitu [R4] *IF*

transaction is Low and stock is High THAN restock is Low.

Proses berikutnya adalah penentuan fungsi keanggotaan dan menghitung fungsi keanggotaan dengan menggunakan perulangan *for* untuk membaca data yang sebelumnya telah disimpan dalam *array* $vMin$, $vMax$ dan nil . Nilai hasil dari perhitungan ini akan disimpan dalam *array* $K[0][0]$ untuk transaksi sedikit atau *low*, $K[0][1]$ untuk transaksi banyak atau *high*, $K[1][0]$ untuk stok sedikit dan $K[1][1]$ untuk stok banyak.

Pertama untuk menentukan $K[0][0]$ atau transaksi sedikit adalah jika nilai rata-rata transaksi kurang dari atau sama dengan nilai minimal transaksi maka nilai yang disimpan dalam variabel $K[0][0]$ akan bernilai 1. Jika nilai rata-rata transaksi lebih dari atau sama dengan nilai minimal transaksi dan nilai rata-rata transaksi kurang dari atau sama dengan nilai maksimal transaksi maka nilai $K[0][0]$ adalah $(maksimal\ transaksi - rata-rata\ transaksi) / (maksimal\ transaksi - minimal\ transaksi)$. Jika nilai rata-rata transaksi lebih besar atau sama dengan nilai maksimal transaksi maka nilai $K[0][0]$ adalah 0.

Selanjutnya untuk menentukan nilai $K[0][1]$ atau transaksi banyak yaitu jika nilai rata-rata transaksi kurang dari atau sama dengan nilai minimal transaksi maka nilai yang disimpan dalam variabel $K[0][1]$ akan bernilai 0. Jika nilai rata-rata transaksi lebih dari atau sama dengan nilai minimal transaksi dan nilai rata-rata transaksi kurang dari atau sama dengan nilai maksimal transaksi maka nilai $K[0][1]$ adalah $(rata-rata\ transaksi - minimal\ transaksi) / (maksimal\ transaksi - minimal\ transaksi)$. Jika nilai rata-rata transaksi lebih besar atau sama dengan nilai maksimal transaksi maka nilai $K[0][1]$ adalah 1.

kemudian untuk menentukan $K[1][0]$ atau stok sedikit adalah jika nilai rata-rata stok kurang dari atau sama dengan nilai minimal stok maka nilai yang disimpan dalam variabel $K[1][0]$ akan bernilai 1. Jika nilai rata-rata stok lebih dari atau sama dengan nilai minimal stok dan nilai rata-rata stok kurang dari atau sama dengan nilai maksimal stok maka nilai $K[1][0]$ adalah $(maksimal\ stok - rata-rata\ stok) / (maksimal\ stok - minimal\ stok)$. Jika nilai rata-rata stok lebih besar atau sama dengan nilai maksimal stok maka nilai $K[1][0]$ adalah 0.

Selanjutnya untuk menentukan nilai $K[1][1]$ atau stok banyak yaitu jika nilai rata-rata stok kurang dari atau sama dengan nilai minimal stok maka nilai yang disimpan dalam variabel $K[1][1]$ akan bernilai 0. Jika nilai rata-rata stok lebih dari atau sama dengan nilai minimal stok dan nilai rata-rata stok kurang dari atau sama dengan nilai maksimal stok maka nilai $K[1][1]$ adalah $(rata-rata\ stok - minimal\ stok) / (maksimal\ stok - minimal\ stok)$. Jika nilai rata-rata stok lebih besar atau sama dengan nilai maksimal stok maka nilai $K[1][1]$ adalah 1.

Bagian berikutnya adalah menghitung predikat aturan dengan fungsi implikasi minimum. Predikat aturan atau *rule* yang akan dihitung sesuai dengan *rule* yang sebelumnya telah ditentukan yaitu *rule* 1 atau [R1], *rule* 2 atau [R2], *rule* 3 atau [R3] dan *rule* 4 atau [R4]. Pada proses ini digunakan fungsi *for* dan *switch case* untuk membaca data *array* yang telah di definisikan pada bagian sebelumnya.

Untuk *case* 0 yaitu *case* yang digunakan untuk operator *and* adalah `If $K[0][transaction[i]] >= K[1][stock[i]]$` pada perulangan *for* pertama nilai I adalah 0 maka dapat dikatakan bahwa variabel $transaction[0]$ yaitu *high* yang menyimpan nilai 1 dan variabel $stock[0]$ yaitu *low* yang menyimpan nilai 0 maka `If $K[0][1] >= K[1][0]$` yaitu $K[0][1]$ adalah transaksi banyak lebih dari atau sama dengan $K[1][0]$ stok sedikit maka *rule* yang disimpan dalam variabel $rule[i]$ adalah $K[1][stock[i]]$ yaitu $K[1][0]$ atau nilai dari stok sedikit. Kemudian jika tidak maka yang disimpan adalah $K[0][transaction[0]]$ atau nilai dari transaksi banyak. Lalu

untuk *case 1* yaitu *case* yang digunakan untuk operator *or* adalah `If K[0][transaction[i]] <= K[1][stock[i]]` yaitu `K[0][1]` adalah transaksi banyak kurang dari atau sama dengan `K[1][0]` stok sedikit maka *rule* yang disimpan dalam variabel `rule[i]` adalah `K[1][stock[i]]` yaitu `K[1][0]` atau nilai dari stok sedikit. Kemudian jika tidak maka yang disimpan adalah `K[0][transaction[0]` atau nilai dari transaksi banyak.

Proses selanjutnya adalah penentuan nilai *Z* dengan menggunakan fungsi *switch case*. *Case 0* adalah *case* untuk restok sedikit yaitu nilai yang akan di simpan pada variabel `nilZ[i]` adalah nilai maksimal stok – (nilai *rule* ke-*i* * nilai maksimal stok - nilai minimal stok). Untuk *case 1* yaitu *case* untuk restok banyak adalah *rule* ke-*i* * (nilai maksimal stok – nilai minimal stok) + nilai minimal stok. Fungsi ini akan diulang hingga nilai *i* dalam fungsi *for* bernilai 3.

Kemudian bagian terakhir adalah proses *defuzzyfikasi* yaitu dengan menjumlahkan perkalian antara nilai alfa predikat atau *rule* dengan nilai *z* kemudian dibagi dengan penjumlahan nilai alfa predikat atau *rule*. Hasil dari perhitungan ini akan dibulatkan dengan fungsi *round* dan menghasilkan jumlah rekomendasi. Berikut merupakan modul program dari proses rekomendasi dengan algoritma *Fuzzy Tsukamoto* data produk.

Algoritma 7: Fuzzy Tsukamoto

```

Input: retailer product data, retailer data
Output: Recommendation value
Method:
Get: id
total = 0
jumRule = 0
Get retailer data
While data = retailer data
vMin = array[data -> min transaction, data -> min stock, data-> min Restock]
vMax = array[data -> max transaction, data -> max stock, data-> max Restock]
nil = array[data -> avg transaction, data-> stock]
high = 1
low = 0
and = 0
or= 1
transaction[0] = high
oprator[0]= and
stock[0] = low
restok[0]= high
transaction[1] = high
oprator[1]= and
stock[1] = high
restok[1]= high
transaction[2] = low
oprator[2]= and
stock[2] = low
restok[2]= low
transaction[3] = low
oprator[3]= and
stock[3] = high
restok[3]= low
For i=0 to i<=2
    If vMin[i]= null
        exit
    End if
    If vMax[i]= null
        exit
    End if
End for
For i=0 to i<=1
    If nil[i]= null
        Exit
End if

```

Modul Program 4.7 Pseudo Code Rekomendasi dengan Algoritma Fuzzy Tsukamoto

Algoritma 8: Lanjutan Fuzzy Tsukamoto

```
End for
For i=0 to i<=1
  If nil[i]<=vMin[i]
    K[i][0] = 1
  End if
  Else if nil[i]>=vMin[i] && nil[i]<=vMax[i]
    K[i][0] = (vMax[i] - nil[i]) / (vMax[i]-vMin[i])
  End if
  Else if nil[i]>=vMax[i]
    $K[$i][0] = 0;
  End if
  If nil[i]<=vMin[i]
    K[i][1] = 0;
  End if
  Else if nil[i]>=vMin[i] && nil[i]<=vMax[i]
    K[i][1] = (nil[i]-vMin[i]) / (vMax[i]-vMin[i])
  End if
  Else if nil[i]>=vMax[i]
    K[i][1] = 1
  End if
End for
c=0; jumz=0
For i=0 to i<=3
  Switch operator[i]
  Case 0:
    If K[0][transaction[i]] >= K[1][stock[i]]
      rule[i] = K[1][stock[i]]
    End if
    Else
      rule[i] = K[0][transaction[i]]
    Break
  Case 1:
    If K[0][transaction[i]] <= K[1][stock[$i]]
      rule[i] = K[1][stock[i]]
    End if
    Else
      rule[i] = K[0][transaction[i]]
    Break
  End switch
  jumRule += rule[i]
  Switch restok[i]
  Case 0:
    nilZ[i] = (vMax[2] - (rule[i] * (vMax[2]-vMin[2])))
  Break
  Case 1:
    nilZ[i] = ((rule[i] * (vMax[2] - vMin[2])) + vMin[2])
  Break
  JumZ = (JumZ + (rule[i] * nilZ[i]))
  End switch
End for
Print round (JumZ/jumRule)
```

Modul Program 4.8 Lanjutan Pseudo Code Rekomendasi dengan Algoritma Fuzzy Tsukamoto

14. Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto

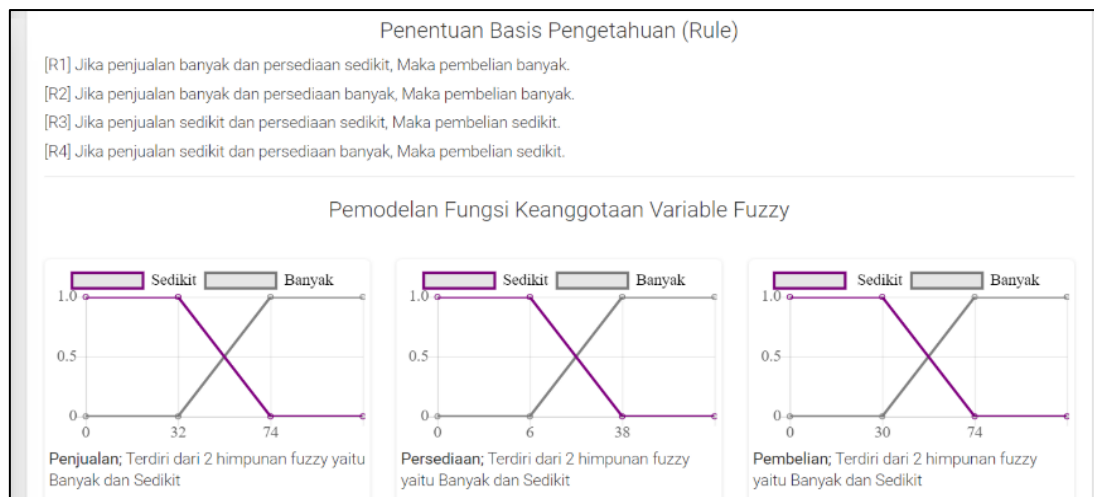
Halaman detail perhitungan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto* merupakan halaman yang berfungsi untuk menjelaskan detail perhitungan algoritma yang digunakan untuk rekomendasi. Pada bagian pertama adalah menampilkan tabel yang berisi data transaksi yang telah dikelompokkan perminggu. Data dalam tabel ini merupakan data *sample* yang diambil dari bulan Januari sampai Februari 2022. Tabel ini memiliki lima kolom diantaranya kolom no, tanggal yang telah dikelompokkan perminggu, jumlah restok, jumlah transaksi dan jumlah stok. Tabel berikutnya adalah tabel untuk menampilkan nilai dari variabel yang akan digunakan untuk proses perhitungan diantaranya min restok, max restok, min transaksi, max transaksi, min stok, max stok, rata-rata transaksi dan stok saat ini. Berikut

merupakan tampilan detail perhitungan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto* dapat dilihat pada Gambar 4.14.

Detail Perhitungan Rekomendasi dengan Metode Fuzzy									
Kode Produk : 10020031 Toko rtlarasyid									
No	Tanggal (per-Minggu)	Jumlah Restok	Jumlah Transaksi	Jumlah Stok					
1.	2022-01-01	46	74	10					
2.	2022-01-08	40	42	8					
3.	2022-01-15	30	32	6					
4.	2022-01-22	40	35	11					
5.	2022-01-29	60	55	16					
6.	2022-02-05	74	68	22					
Kode Retailer	Kode Produk	Min Restok	Max Restok	Min Transaksi	Max Transaksi	Min Stok	Max Stok	Rata-rata Transaksi	Stok saat ini
rtlarasyid	10020031	30	74	32	74	6	38	51	22

Gambar 4.14 Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma *Fuzzy Tsukamoto*

Bagian berikutnya adalah penentuan basis *rule* yang terdiri dari empat *rule* diantaranya R1 Jika penjualan banyak dan persediaan sedikit maka pembelian banyak, R2 Jika penjualan banyak dan persediaan banyak maka pembelian banyak, R3 Jika penjualan sedikit dan persediaan sedikit maka pembelian sedikit dan R4 Jika penjualan sedikit dan persediaan banyak maka pembelian sedikit. Setelah penentuan basis pengetahuan, maka ditampilkan grafik pemodelan fungsi keanggotaan variabel *Fuzzy*. Grafik ini berfungsi menampilkan informasi nilai dengan dua himpunan yaitu banyak dan sedikit dari variabel yang akan digunakan. Nilai sedikit berarti turun dan nilai banyak berarti naik. Berikut merupakan lanjutan tampilan detail perhitungan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto* dapat dilihat pada Gambar 4.15.



Gambar 4.15 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma *Fuzzy Tsukamoto*

Bagian selanjutnya menampilkan rumus untuk menghitung fungsi keanggotaan yang terbagi menjadi μ penjualan sedikit(x), μ penjualan banyak(x), μ persediaan sedikit(x), μ persediaan banyak(x), μ pembelian sedikit(z) dan μ pembelian banyak(z). Selanjutnya ditampilkan hasil dari proses perhitungan fungsi keanggotaan menjadi penjualan sedikit(x), penjualan banyak(x), persediaan sedikit(x) dan persediaan banyak(x) dengan nilai x yang

sesuai dengan nilai data yang ditampilkan. x penjualan adalah nilai rata-rata transaksi sementara x persediaan adalah nilai rata-rata stok. Nilai ini akan dimasukkan rumus yang sesuai misalnya untuk penjualan sedikit (nilai rata-rata transaksi) maka akan digunakan salah satu dari rumus dalam μ penjualan sedikit(x), yaitu jika nilai rata-rata transaksi kurang dari nilai minimal transaksi maka hasilnya adalah satu, namun jika nilai rata-rata transaksi berada diantara nilai minimal dan maksimal transaksi maka akan digunakan rumus (maksimal transaksi-nilai rata-rata transaksi) / (nilai maksimal transaksi-nilai minimal transaksi). Dan jika nilai rata-rata transaksi lebih dari atau sama dengan nilai maksimal transaksi maka hasilnya adalah 0. Begitu juga dengan penjualan banyak(x), persediaan sedikit(x) dan persediaan banyak(x) yang akan dihitung menggunakan rumus pada μ masing-masing. Berikut merupakan lanjutan tampilan detail perhitungan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto* dapat dilihat pada Gambar 4.16.

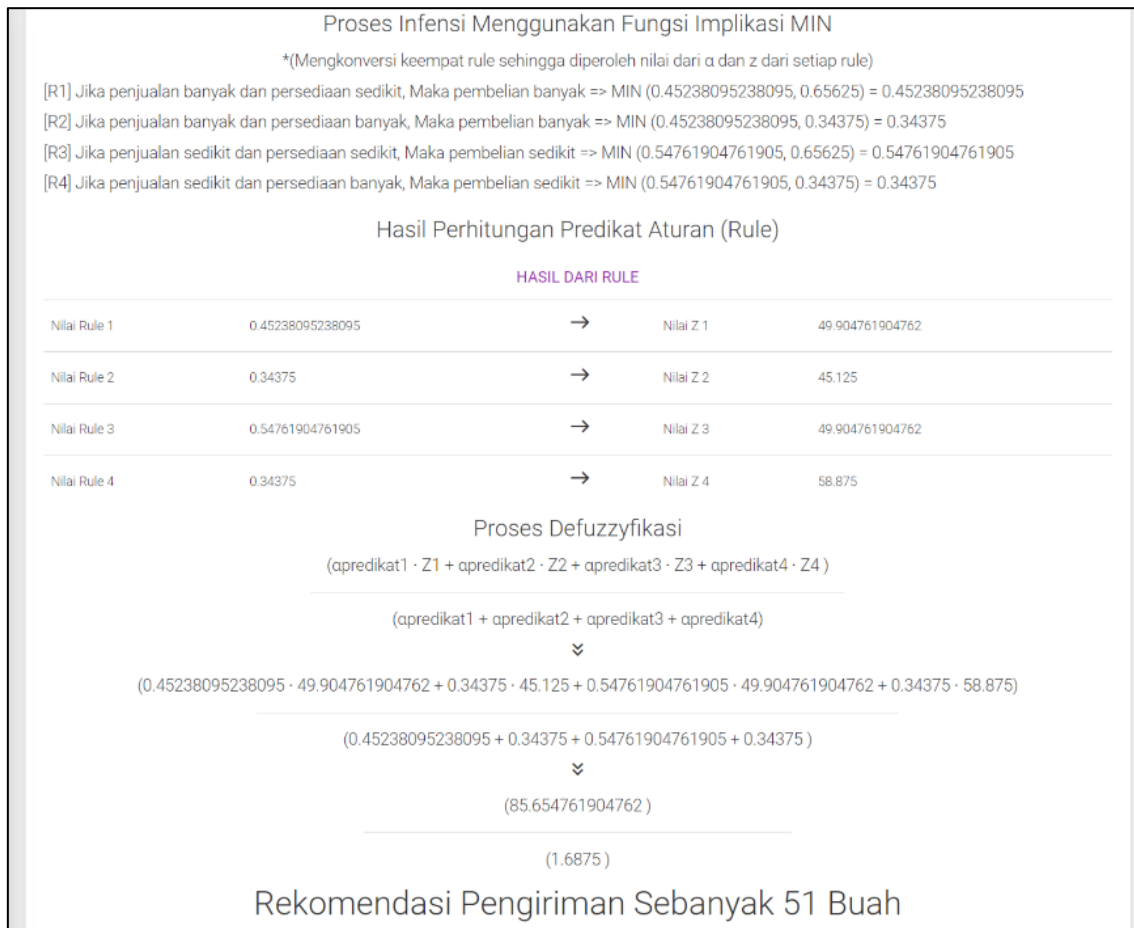
Fungsi Keanggotaan Variable Fuzzy (Penjualan dan Persediaan)			
$\mu_{\text{PenjualanSedikit}}(X)$:	$\mu_{\text{PenjualanBanyak}}(X)$:	$\mu_{\text{PersediaanSedikit}}(X)$:	$\mu_{\text{PersediaanBanyak}}(X)$:
$x \leq 32 \rightarrow 1$	$x \leq 32 \rightarrow 1$	$x \leq 6 \rightarrow 1$	$x \leq 6 \rightarrow 0$
$32 < x < 74 \rightarrow (74 - x)/(74 - 32)$	$32 < x < 74 \rightarrow (x - 32)/(74 - 32)$	$6 < x < 38 \rightarrow (38 - x)/(38 - 6)$	$6 < x < 38 \rightarrow (x - 6)/(38 - 6)$
$x \geq 74 \rightarrow 0$	$x \geq 74 \rightarrow 0$	$x \geq 38 \rightarrow 0$	$x \geq 38 \rightarrow 1$
Fungsi Keanggotaan Variable Fuzzy (Pembelian)			
$\mu_{\text{PembelianSedikit}}(z)$:		$\mu_{\text{PembelianBanyak}}(z)$:	
$z \leq 30 \rightarrow 1$		$z \leq 30 \rightarrow 1$	
$30 < z < 74 \rightarrow (74 - z)/(74 - 30)$		$30 < z < 74 \rightarrow (z - 30)/(74 - 30)$	
$z \geq 74 \rightarrow 0$		$z \geq 74 \rightarrow 0$	
Hasil Perhitungan Fungsi Keanggotaan			
Penjualan Sedikit(51)	Penjualan Banyak(51)	Persediaan Sedikit(22)	Persediaan Banyak(22)
0.54761904761905	0.45238095238095	0.5	0.5

Gambar 4.16 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma *Fuzzy Tsukamoto*

Setelah hasil fungsi keanggotaan didapatkan maka akan ditampilkan proses infensi menggunakan fungsi implikasi MIN. Proses ini adalah proses mengkonfensi keempat *rule* yang diperoleh sebelumnya sehingga diperoleh nilai dari setiap *rule* yang digunakan dengan mengambil nilai minimum dari nilai keanggotaan pada setiap *rule*.

Bagian berikutnya setelah nilai dari setiap *rule* didapatkan adalah dengan melakukan perhitungan predikat aturan untuk mengetahui nilai z . rumus yang digunakan dalam bagian ini adalah μ pembelian sedikit(z) dan μ pembelian banyak(z). penentuan pembelian banyak dan pembelian sedikit berdasarkan *rule* yang telah ditentukan. Misalnya adalah *rule* dengan pembelian sedikit maka menggunakan μ pembelian sedikit(z) yaitu nilai *rule* = (z – nilai minimal restok) / (nilai maksimal restok – nilai minimal restok) maka nilai z = nilai *rule* (nilai maksimal restok – nilai minimal restok) + nilai minimal restok.

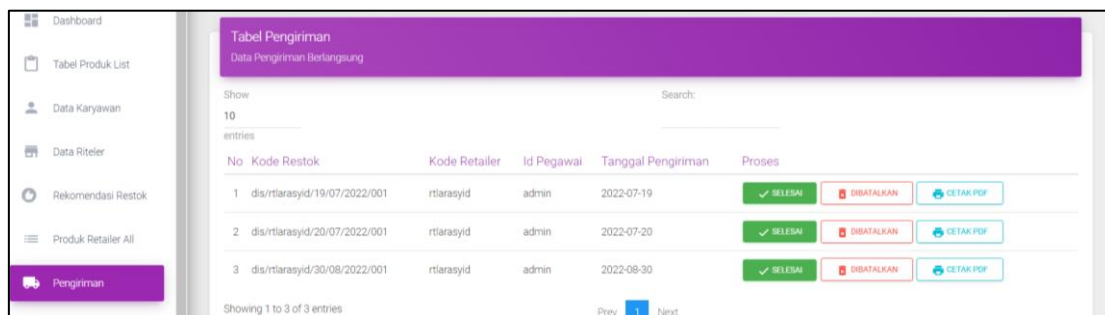
Setelah nilai *rule* dan nilai z didapatkan maka proses selanjutnya adalah proses *defuzzyfikasi*. Peroses ini dilakukan dengan menjumlahkan semua perkalian *alfa* predikat atau nilai *rule* dengan nilai z yang dibagi dengan hasil jumlah nilai *alfa* predikat. Hasil dari perhitungan ini merupakan jumlah rekomendasi pengiriman barang. Berikut merupakan lanjutan tampilan detail perhitungan rekomendasi dengan menggunakan algoritma *Fuzzy Tsukamoto* dapat dilihat pada Gambar 4.17.



Gambar 4.17 Lanjutan Tampilan Halaman Detail Perhitungan Rekomendasi dengan Algoritma Fuzzy Tsukamoto

15. Halaman Tabel Pengiriman

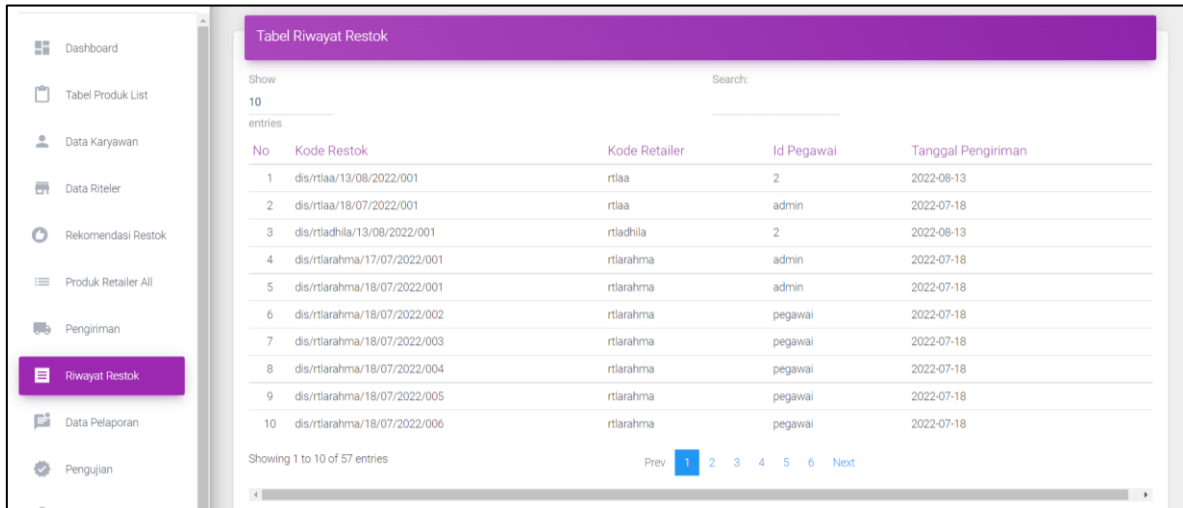
Halaman pengiriman merupakan halaman yang berguna untuk menampilkan data pengiriman pengiriman yang sedang berlangsung. Pada halaman ini terdapat tabel yang memiliki beberapa kolom diantaranya no, kode restok, kode *retailer* id pegawai, tanggal pengiriman dan proses. Pada bagian proses terdapat tiga menu *button* yaitu *button* selesai adalah *button* yang digunakan apabila proses pengiriman telah selesai sehingga data akan disimpan pada riwayat pengiriman, sementara liat data yang berada dalam tabel ini akan dihapus, selanjutnya adalah *button* dibatalkan yaitu *button* yang berguna untuk membatalkan pengiriman sehingga list data dalam tabel ini akan dihapus tanpa disimpan dalam Riwayat pengiriman, yang terakhir adalah *button* cetak pdf yang berguna untuk menampilkan halaman cetak data pengiriman. Berikut merupakan tampilan halaman tabel pengiriman dapat dilihat pada Gambar 4.18.



Gambar 4.18 Tampilan Halaman Tabel Pengiriman

16. Halaman Riwayat Restok

Halaman Riwayat restok menampilkan data riwayat restok dari barang yang telah selesai dikirim oleh distributor ke pihak *retailer*. Pada halaman ini terdapat tabel yang memiliki beberapa kolom diantaranya nomor, kode restok, kode *retailer*, id pegawai dan id pengirim. Pada halaman ini juga terdapat fungsi pencarian dan juga pilihan untuk memilih jumlah data yang ingin ditampilkan. Berikut merupakan tampilan halaman tabel riwayat restok dapat dilihat pada Gambar 4.19.

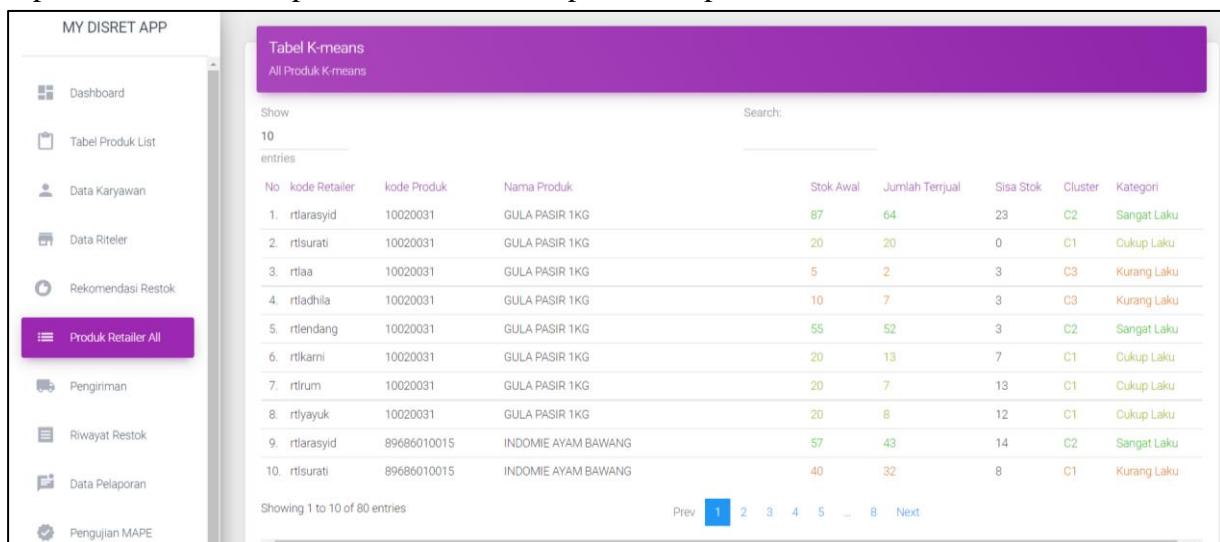


No	Kode Restok	Kode Retailer	Id Pegawai	Tanggal Pengiriman
1	dis/rtlaa/13/08/2022/001	rtlaa	2	2022-08-13
2	dis/rtlaa/18/07/2022/001	rtlaa	admin	2022-07-18
3	dis/rtadhila/13/08/2022/001	rtadhila	2	2022-08-13
4	dis/rtlarahma/17/07/2022/001	rtlarahma	admin	2022-07-18
5	dis/rtlarahma/18/07/2022/001	rtlarahma	admin	2022-07-18
6	dis/rtlarahma/18/07/2022/002	rtlarahma	pegawai	2022-07-18
7	dis/rtlarahma/18/07/2022/003	rtlarahma	pegawai	2022-07-18
8	dis/rtlarahma/18/07/2022/004	rtlarahma	pegawai	2022-07-18
9	dis/rtlarahma/18/07/2022/005	rtlarahma	pegawai	2022-07-18
10	dis/rtlarahma/18/07/2022/006	rtlarahma	pegawai	2022-07-18

Gambar 4.19 Tampilan Halaman Riwayat Restok

17. Halaman Produk *Retailer All*

Halaman produk *retailer all* menampilkan semua data produk yang berada di semua *retailer* yang telah bergabung. Pada halaman ini data produk dikelompokkan dengan menjadi tiga *cluster* berdasarkan tingkat penjualan produk yang dibedakan menjadi tiga warna. Pada halaman ini terdapat tabel dengan beberapa kolom diantaranya no, kode *retailer*, kode produk, nama produk, stok awal, jumlah terjual minggu lalu, sisa stok dan kelompok *cluster*. Berikut merupakan tampilan halaman tabel produk *retailer all* dapat dilihat pada Gambar 4.20.



No	kode Retailer	kode Produk	Nama Produk	Stok Awal	Jumlah Terjual	Sisa Stok	Cluster	Kategori
1.	rtlarasyid	10020031	GULA PASIR 1KG	87	64	23	C2	Sangat Laku
2.	rtlsurati	10020031	GULA PASIR 1KG	20	20	0	C1	Cukup Laku
3.	rtlaa	10020031	GULA PASIR 1KG	5	2	3	C3	Kurang Laku
4.	rtadhila	10020031	GULA PASIR 1KG	10	7	3	C3	Kurang Laku
5.	rtlendam	10020031	GULA PASIR 1KG	55	52	3	C2	Sangat Laku
6.	rtkarni	10020031	GULA PASIR 1KG	20	13	7	C1	Cukup Laku
7.	rtfirum	10020031	GULA PASIR 1KG	20	7	13	C1	Cukup Laku
8.	rtiyayuk	10020031	GULA PASIR 1KG	20	8	12	C1	Cukup Laku
9.	rtlarasyid	89686010015	INDOMIE AYAM BAWANG	57	43	14	C2	Sangat Laku
10.	rtlsurati	89686010015	INDOMIE AYAM BAWANG	40	32	8	C1	Kurang Laku

Gambar 4.20 Tampilan Halaman Produk *Retailer All*

Pembagian *cluster* pada halaman produk *retailer all* dilakukan dengan menggunakan algoritma *k-means*. Perhitungan dengan algoritma ini dimulai dengan mendapatkan transaksi dan stok sebanyak 3 data secara acak yang akan digunakan sebagai titik pusat atau *centroid*

awal yaitu *centroid 1*, *centroid 2* dan *centroid 3* serta dilakukan pendeklarasian beberapa variabel diantaranya *cluster* awal dengan nilai 1, status dengan nilai *false* serta *loop* dengan nilai 0. Bagian selanjutnya adalah melakukan perulangan *while* atau selama nilai status bernilai *false* maka di deklarasikan *x* dengan nilai 0 dan didapatkan data dari tabel *temp_kmeans* dimana kode produk sama dengan *value id*.

Selanjutnya, dilakukan kembali perulangan *while* yaitu selama data yang didapatkan bernilai benar maka akan dikerjakan perintah diantaranya mendeklarasikan variabel *result cluster 1*, *result cluster 2* dan *result cluster 3* dengan nilai 0, lalu dilakukan proses perhitungan dengan rumus $\sqrt{(\text{transaksi} - \text{centroid } 1 [\text{loop}][0])^2 + (\text{stok} - \text{centroid } 1 [\text{loop}][1])^2}$ yang disimpan pada variabel *result cluster 1*, $\sqrt{(\text{transaksi} - \text{centroid } 2 [\text{loop}][0])^2 + (\text{stok} - \text{centroid } 2 [\text{loop}][1])^2}$ yang disimpan pada variabel *result cluster 2*, dan $\sqrt{(\text{transaksi} - \text{centroid } 3 [\text{loop}][0])^2 + (\text{stok} - \text{centroid } 3 [\text{loop}][1])^2}$ yang disimpan pada variabel *result cluster 3*. Kemudian, dikerjakan fungsi *if* yaitu jika nilai *result cluster 1* kurang dari nilai *result cluster 2* dan nilai *result cluster 1* kurang dari nilai *result cluster 3* maka variabel **Final cluster [x]** akan diisi dengan C1 lalu data *cluster* pada tabel *temp_kmeans* diubah menjadi C1, jika nilai *result cluster 2* kurang dari nilai *result cluster 1* dan nilai *result cluster 2* kurang dari nilai *result cluster 3* maka variabel **Final cluster [x]** akan diisi dengan C2 lalu data *cluster* pada tabel *temp_kmeans* diubah menjadi C2 dan jika bukan variabel **Final cluster [x]** akan diisi dengan C3 lalu data *cluster* pada tabel *temp_kmeans* diubah menjadi C3. Selanjutnya nilai variabel *x* ditambah dengan 1, proses ini akan diulang hingga data yang didapatkan bernilai salah.

Perintah selanjutnya adalah dengan menambahkan nilai variabel *loop* dengan nilai 1 lalu mendapatkan data rata-rata transaksi dari tabel *temp_kmeans* dimana nilai *cluster* sama dengan C1 yang akan disimpan pada variabel *centroid 1 [loop][0]* lalu data rata-rata stok dari tabel *temp_kmeans* dimana nilai *cluster* sama dengan C1 akan disimpan pada variabel *centroid 1 [loop][1]* begitu pula pada data dengan *cluster C2* akan disimpan pada *centroid 2* dan data dengan *cluster C3* akan disimpan pada *centroid 3*. Kemudian nilai dari variabel status sama dengan *true*. Selanjutnya dikerjakan perintah *for* dari *i* sama dengan 0 hingga nilai *i* sudah tidak kurang dari nilai variabel jumlah *value* maka dikerjakan perintah *if* yaitu jika nilai *cluster* awal tidak sama dengan *final cluster* maka status bernilai *false*. Kemudian jika status bernilai salah maka *cluster* awal akan diisi dengan nilai *final cluster*. Iterasi ini akan dilakukan hingga nilai *cluster* awal sama dengan nilai *final cluster*. Berikut merupakan modul program dari algoritma *K-Means*.

Algoritma 9: K-Means

Input: retailer product data, retailer transaction data, retailer data
Output: cluster value
Method:
Get: Random (transaction, stock) limit 3, value id
count list = **Get count**(list) **From** temp_kmeans **Where** product code = value id
For i=0 to i<count list
 First cluster [i]=1
End For
centroid 1[0] = array (transaction [0], stock [0])
centroid 2[0] = array (transaction [1], stock [1])
centroid 3[0] = array (transaction [2], stock [2])
status = **false**
loop = 0
While (status == **false**)

Modul Program 4.9 Pseudo Code Algoritma K-Means

Algoritma 10: Lanjutan K-Means

```
X=0
Get: data from temp_kmeans Where product code = value id
While (data is true)
  result cluster 1 = 0
  result cluster 2 = 0
  result cluster 1 = square root (pow (transaction - centroid 1[loop] [0],2)
  + (pow (stock - centroid 2 [loop][1],2))
  result cluster 2 = square root (pow (transaction - centroid 2[loop] [0],2)
  + (pow (stock - centroid 2 [loop][1],2))
  result cluster 3 = square root (pow (transaction - centroid 3[loop] [0],2)
  + (pow (stock - centroid 2 [loop][1],2))
  If result cluster 1 < result cluster 2 And result cluster 1 < result
  cluster 3 Then
    Final cluster [x] = C1
    Update temp_kmeans database Set cluster = C1 Where id = value id
  End If
  Else If result cluster 2 < result cluster 1 And result cluster 2 < result
  cluster 3 Then
    Final cluster [x] = C2
    Update temp_kmeans database Set cluster = C2 Where id = value id
  End If
  Else
    Final cluster [x] = C3
    Update temp_kmeans database Set cluster = C3 Where id = value id
  End If
  X = x + 1
End While
loop = loop + 1
centroid 1 [loop][0] = Get average (transaction) From temp_kmeans Where
cluster = C1
centroid 1 [loop][1] = Get average (stock) From temp_kmeans Where cluster =
C1
centroid 2 [loop][0] = Get average (transaction) From temp_kmeans Where
cluster = C2
centroid 2 [loop][1] = Get average (stock) From temp_kmeans Where cluster =
C2
centroid 3 [loop][0] = Get average (transaction) From temp_kmeans Where
cluster = C3
centroid 3 [loop][1] = Get average (stock) From temp_kmeans Where cluster =
C3
status = true
For i=0 to i<count list
  If (first cluster [i] != final cluster [i]) Then
    Status = false
  End If
End For
If status = false Then
  first cluster = final cluster
End If
```

Modul Program 4.10 Lanjutan Pseudo Code Algoritma K-Means

18. Halaman Detail Perhitungan Algoritma K-Means

Halaman detail perhitungan algoritma *K-Means* merupakan halaman yang berfungsi untuk menjelaskan detail perhitungan algoritma yang digunakan untuk pengelompokan data penjualan. Pada bagian pertama adalah menampilkan tabel yang berisi data jumlah transaksi atau penjualan dan stok suatu produk pada masing-masing *retailer*. Data dalam tabel ini merupakan data *sample* yang diambil dari bulan Agustus 2022.

Pada tabel ke-2 merupakan tabel yang berisi data titik pusat *cluster* atau *centroid* awal. Yaitu *cluster 1*, *cluster 2* dan *cluster 3*. Data ini diperoleh dari data pada tabel sebelumnya dengan mengambil tiga baris data secara acak. Kemudian setelah didapatkan data nilai *centroid* awal maka dilakukan perhitungan dengan rumus *euclidean distance* untuk menentukan jarak data ke *centroid* yang hasilnya ditampilkan pada tabel ke-3. Selanjutnya adalah membandingkan

jarak masing-masing data dengan nilai setiap *cluster* dan diambil data dengan jarak terdekat maka akan dikelompokkan ke *cluster* tersebut yang mana hasilnya dapat dilihat pada tabel setelahnya. Berikut tampilan algoritma *k-means* dapat dilihat pada Gambar 4.21.

Tabel K-means
10020031 || GULA PASIR 1KG

No	kode Retailer	Nama Toko	Stok Awal	Jumlah Terjual
1.	rtiarasyid	Arasyid	87	64
2.	rtlsurati	Surati	20	20
3.	rtlaa	AA	5	2
4.	rtiadhila	Adhila	10	7
5.	rtiendang	Endang	55	52
6.	rtikarni	Karni	20	13
7.	rtirum	Rum	20	7
8.	rtlyayuk	Yayuk	20	8

Titik Pust Cluster (centroid)

Titik Pusat Awal	Stok (x)	Terjual (y)
Cluster 1	20	7
Cluster 2	20	13
Cluster 3	10	7

Hasil Perhitungan Euclidean Distance (Jarak Data dengan Pusat Cluster) *Iterasi Ke- 1

	M1	M2	M3	M4	M5	M6	M7	M8
Cluster 1	87.965902484997	13	15.811388300842	10	57.008771254957	6	0	1
Cluster 2	84.202137740084	7	18.601075237738	11.661903789691	52.402290026296	0	6	5
Cluster 3	95.80187889598	16.401219466857	7.0710678118655	0	63.639610306789	11.661903789691	10	10.049875621121

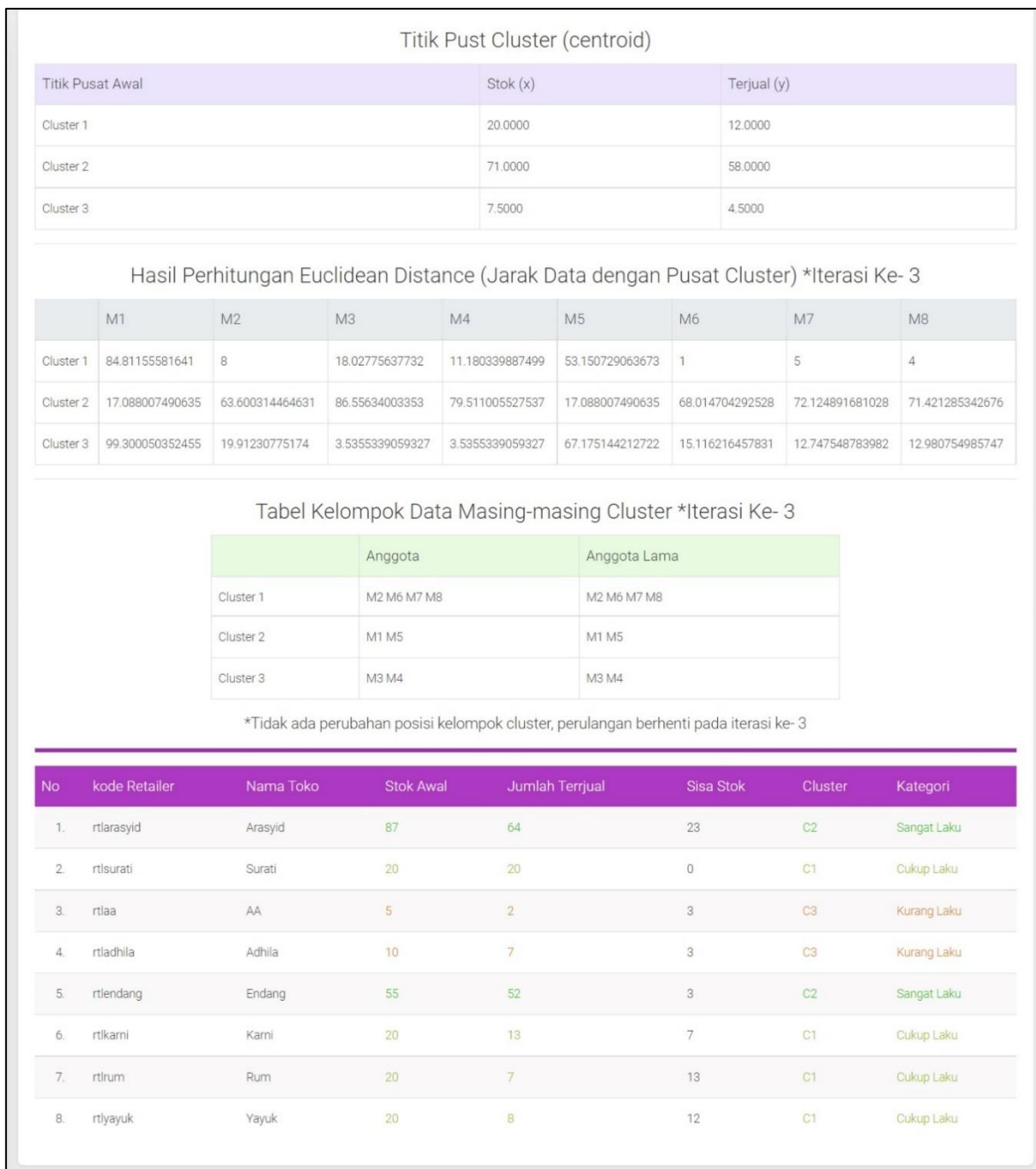
Tabel Kelompok Data Masing-masing Cluster *Iterasi Ke- 1

	Anggota	Anggota Lama
Cluster 1	M7 M8	
Cluster 2	M1 M2 M5 M6	
Cluster 3	M3 M4	

Gambar 4.21 Tampilan Halaman Detail Perhitungan Algoritma *K-Means*

Proses selanjutnya adalah membandingkan anggota tiap-tiap *cluster* yang didapat dengan anggota *cluster* lama jika terdapat perbedaan atau perubahan anggota maka diulangi kembali proses sebelumnya yaitu penentuan *centroid* baru dengan menghitung nilai rata-rata data pada setiap *cluster* data yang didapatkan akan menjadi *centroid* baru selanjutnya dihitung kembali jarak setiap data dengan *centroid* baru menggunakan rumus *euclidean distance* dan dicari jarak terdekat hingga tidak ada lagi perubahan anggota *cluster* dan proses iterasi akan selesai dan diperoleh data akhir pengelompokan tiap-tiap *cluster*. Setelah data didapatkan maka data *cluster* diurutkan dari yang paling besar sebagai data dengan kategori sangat laku hingga

data yang paling kecil akan dikategorikan menjadi kurang laku. Berikut lanjutan tampilan algoritma *k-means* dapat dilihat pada Gambar 4.22.



Gambar 4.22 Lanjutan Tampilan Halaman Detail Perhitungan Algoritma *K-Means*

19. Halaman Data Pelaporan

Halaman tabel pelaporan merupakan halaman yang berguna untuk menampilkan data pelaporan yang dikirimkan pihak *retailer* kepada pihak distributor. Pada halaman ini terdapat tabel yang memiliki beberapa kolom diantaranya no, kode *retailer*, teks pelaporan, tanggal keterangan dan juga proses. Tabel ini menampilkan list data pelaporan baik yang sudah diproses maupun yang belum diproses. Data yang telah diproses akan berwarna hitam dengan keterangan selesai sementara data yang belum diproses berwarna hijau dengan keterangan belum diproses. *Button* selesai digunakan apabila pelaporan telah diproses oleh pihak distributor, jika *button* selesai di klik maka keterangan pelaporan akan berubah menjadi selesai. Berikut merupakan tampilan halaman data pelaporan dapat dilihat pada Gambar 4.23.

No	Kode Retailer	Text	Tanggal	Keterangan	Proses
1	rtladhila	kehilangan produk KAPAL API SPECIAL MIX 25G dengan kode 8991002105485 satu buah	2022-01-12	Selesai	✓ SELESAI
2	rtlyayuk	Produk NUVO hilang 1pcs	2022-07-02	Selesai	✓ SELESAI
3	rtlyayuk	coba lapor	2022-07-02	Selesai	✓ SELESAI
4	rtlyayuk	coba	2022-07-07	Selesai	✓ SELESAI
5	rtlyayuk	barang hilang dengan kode 1132	2022-07-16	Dalam Proses	✓ SELESAI

Gambar 4.23 Tampilan Halaman Data Pelaporan

20. Halaman Pengujian MAPE

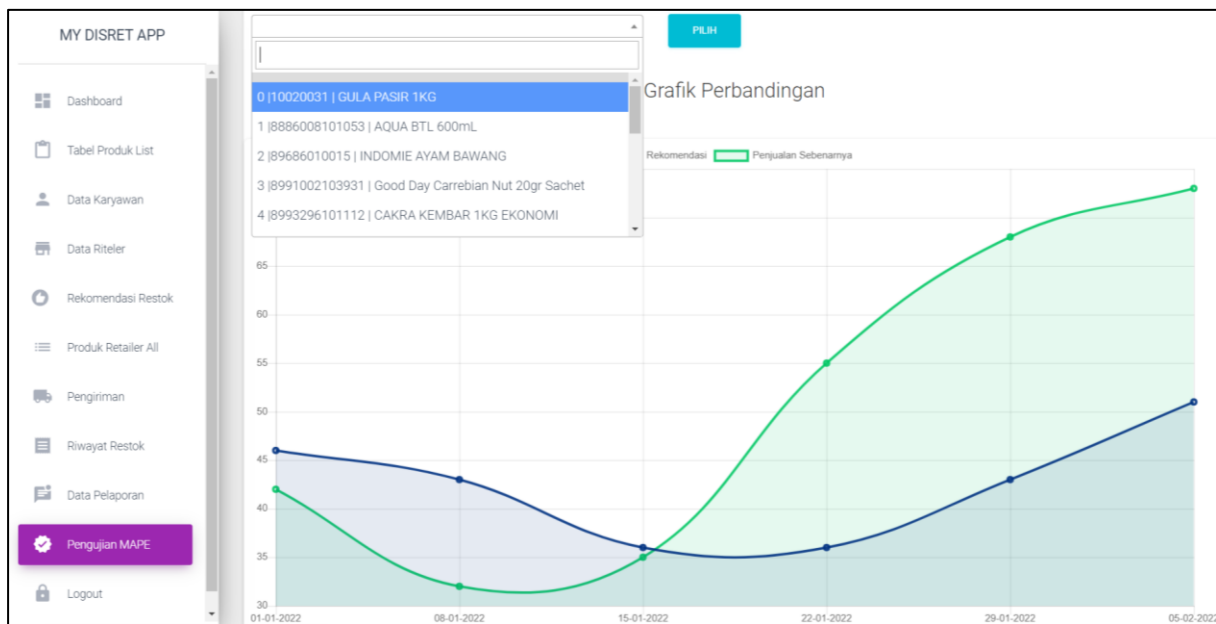
Halaman pengujian MAPE adalah halaman yang berfungsi untuk menampilkan hasil pengujian dari hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE. Pada halaman ini terdapat tabel pengujian yang menampilkan nilai MAPE pada setiap data rekomendasi dari minggu pertama hingga minggu keenam. Tabel pada halaman ini memiliki beberapa kolom diantaranya no, tanggal perminggu, kode produk, nama produk, penjualan minggu depan yaitu penjualan asli minggu selanjutnya, stok tersisa, rekomendasi yaitu nilai rekomendasi yang dihasilkan dengan algoritma *Fuzzy Tsukamoto*, serta persentase PE yang merupakan nilai presentase kesalahan nilai rekomen dasi dibandingkan dengan nilai asli penjualan minggu selanjutnya. Berikut merupakan tampilan halaman pengujian MAPE dapat dilihat pada Gambar 4.24.

No	Tgl Per-Minggu	Kode Produk	Nama Produk	Penjualan Minggu Depan	Stok Tersisa	Rekomendasi	Presentase PE (%)
1.	2022-01-01	10020031	GULA PASIR 1KG	42	10	46	43.75
2.	2022-01-01	8886008101053	AQUA BTL 600mL	89	66	120	421.73913043478
3.	2022-01-01	89686010015	INDOMIE AYAM BAWANG	56	63	100	9900
4.	2022-01-01	8991002103931	Good Day Carrebian Nut 20gr Sachet	6	14	10	900
5.	2022-01-01	8993296101112	CAKRA KEMBAR 1KG EKONOMI	4	10	15	1400
6.	2022-01-01	8993296301116	BOGASARI SEGITIGA BIRU 1KG	9	11	12	1100
7.	2022-01-01	8993560025243	DETTOL 100gr	2	1	5	400
8.	2022-01-01	8995177101112	GULAKU PCK 1KG	6	4	0	100
9.	2022-01-01	8996001301142	ROMA BISKUIT KELAPA 300G	1	1	5	400
10.	2022-01-01	8996001600146	TEH PUCUK HARUM 350ML	31	42	48	4700
11.	2022-01-01	8997032680329	TARO NET 40G	10	51	60	5900
12.	2022-01-01	8998866200318	SEDAP MIE AYAM BAWANG	41	37	80	1900

Gambar 4.24 Tampilan Halaman Pengujian MAPE

Pada halaman pengujian MAPE juga ditampilkan grafik perbandingan hasil rekomendasi dengan data asli. Grafik ditampilkan berdasarkan data produk yang dipilih dengan memasukan kode produk atau nama produk. grafik ini memberikan informasi perbandingan data berdasarkan kurun waktu yang telah dikelompokan perminggu. Garis warna biru merupakan nilai rekomendasi yang dihasilkan dari algoritma *Fuzzy Tsukamoto*, garis dengan warna hijau merupakan nilai jumlah penjualan sebenarnya dikurangi stok tersisa pada minggu berikutnya yang digunakan sebagai pembanding dengan nilai rekomendasi pada perhitungan

metode MAPE. Berikut merupakan lanjutan tampilan halaman pengujian MAPE dapat dilihat pada Gambar 4.25.



Gambar 4.25 Lanjutan Tampilan Halaman Pengujian MAPE

Metode MAPE atau *mean absolute percentage error* merupakan pengujian yang digunakan untuk mengetahui nilai *error* hasil suatu metode dengan nilai yang sebenarnya. Pada penelitian ini pengujian MAPE digunakan untuk menguji hasil rekomendasi dengan nilai sebenarnya, nilai yang digunakan untuk menguji adalah nilai penjualan sebenarnya dikurangi nilai stok tersisa pada minggu sebelumnya. Data yang digunakan dalam pengujian ini meliputi data jumlah penjualan asli minggu selanjutnya, stok terkini, dan data rekomendasi. Untuk memperoleh nilai yang akan digunakan sebagai pembandingan dalam perhitungan MAPE akan dilakukan perhitungan dengan mengurangi jumlah penjualan minggu depan dengan stok terkini nilai ini akan disimpan pada variabel *actual value* atau nilai aktual. Kemudian dicari nilai PE atau *percentage error* dengan rumus $((\text{actual value} - \text{recommendation}) / \text{actual value}) * 100$ yaitu nilai aktual dikurangi dengan nilai rekomendasi dibagi nilai rekomendasi kemudian dikali 100. Setelah mendapatkan nilai PE maka dicari nilai MAPE dengan cara menjumlahkan seluruh nilai PE dibagi dengan jumlah data PE maka akan diperoleh presentase nilai MAPE. Berikut merupakan modul program dari metode pengujian MAPE data produk.

Algoritma 9: MAPE Testing

Input: recommendation value

Output: MAPE value

Method:

Get: next week sales, current stock, recommendation

Actual value = next week sales - current stock

PE = $((\text{actual value} - \text{recommendation}) / \text{actual value}) * 100$

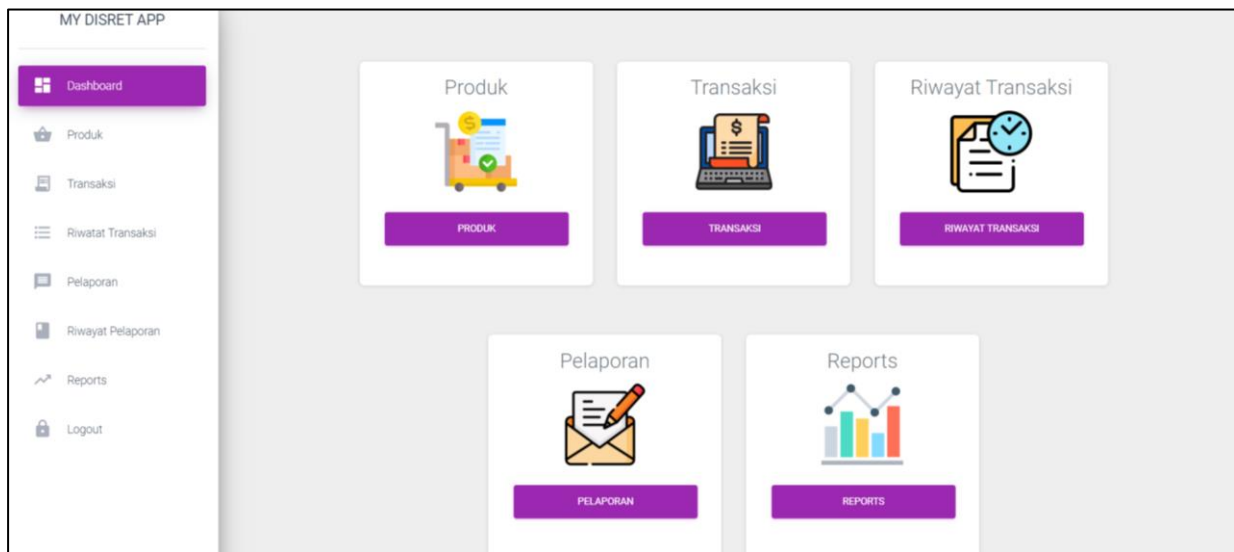
MAPE = $\text{SUM}(\text{PE}) / \text{COUNT}(\text{PE})$

Modul Program 4.11 Pengujian MAPE

21. Halaman *Retailer Dashboard*

Halaman *retailer dashboard* adalah halam utama yang pertama ditampilkan ketika *retailer* berhasil *login*. Pada halaman ini terdapat beberapa menu utama diantaranya menu produk untuk menampilkan data produk, menu transaksi untuk melakukan proses transaksi, menu riwayat transaksi, menu pelaporan, dan menu repts. Halaman ini hanya dapat diakses

oleh *retailer* yang telah terdaftar, data produk dan penjualan yang ditampilkan pada halaman ini sesuai dengan data yang dimiliki masing-masing *retailer* berdasarkan data *username* yang melakukan *login*. Berikut merupakan tampilan halaman *retailer dashboard* dapat dilihat pada Gambar 4.26.



Gambar 4.26 Tampilan Halaman *Retailer Dashboard*

22. Halaman *Retailer Produk*

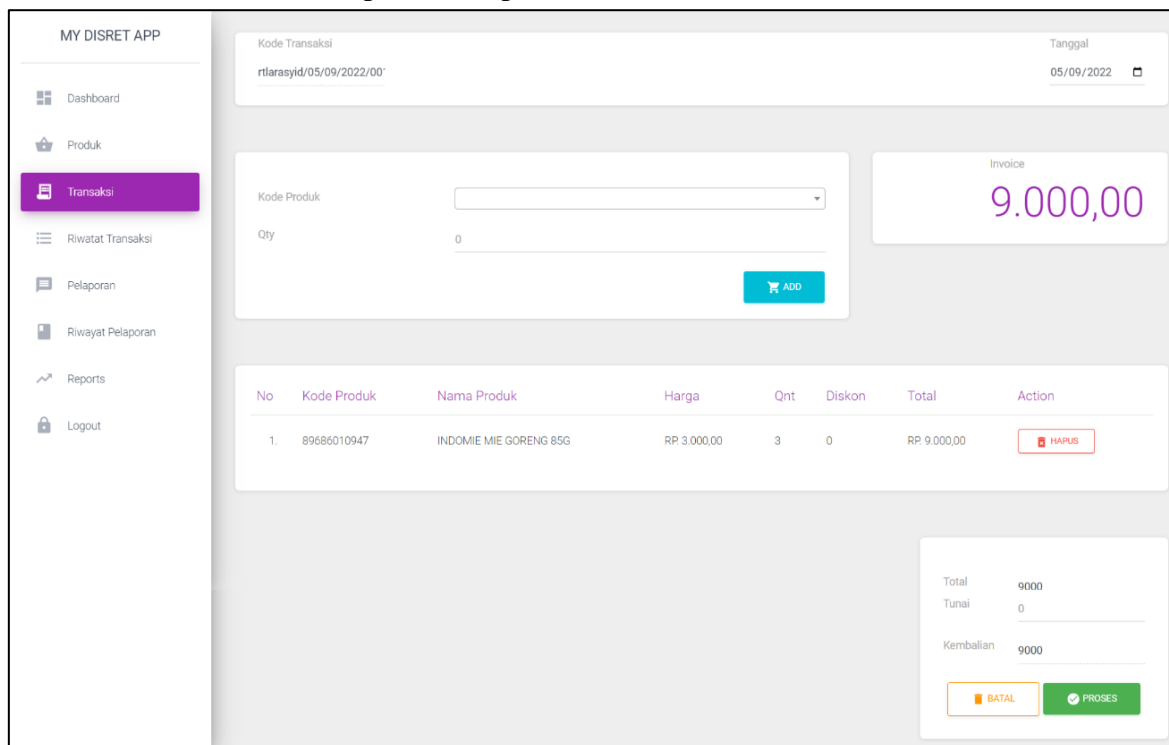
Halaman *retailer produk* merupakan halaman yang digunakan untuk menampilkan daftar produk yang dimiliki *retailer* dalam bentuk tabel. Tabel pada halaman ini memiliki beberapa kolom diantaranya no, nama produk, kode produk, kategori, harga, satuan, stok, diskon, periode awal diskon, periode akhir diskon, minimal pembelian, maksimal pembelian, status, tanggal kadaluarsa, terakhir diperbaharui dan gambar. Pada halaman ini terdapat fungsi pencarian dan pilihan jumlah data yang ingin ditampilkan serta pengaturan untuk memilih kolom mana yang ingin ditampilkan. tidak seperti pada halaman distributor, pada halaman *retailer* tidak dapat melakukan perubahan pada data produk, *retailer* hanya dapat melihat informasi data produk. Berikut tampilan halaman *retailer produk* dapat dilihat pada Gambar 4.27.

No	Nama Produk	Kode Produk	Kategori	Harga	Satuan	Stok	Diskon	Periode Awal diskon	Periode Akhir diskon	Minimal Pembelian (diskon)	Maksimal Pembelian (diskon)	Status	Tanggal Kadaluarsa	Terakhir diperbaharui	Gambar
1.	NUVO RELAX & PROTECT 76GR	8998866618144	sabun	2500	PCS	16						Tersedia	30 November -0001	2022-07-09	
2.	LIFEBOY 60GR	8999999046996	sabun	3500	PCS	2						Tersedia	30 November -0001	2022-07-09	
3.	DETTOL 100gr	8993560025243	sabun	6500	PCS	1						Tersedia	30 November -0001	2022-07-09	
4.	TARO NET 40G	8997032680329	snack	5000	PCS	2						Tersedia	30 November -0001	2022-07-09	
5.	ROMA BISKUIT KELAPA 300G	8996001301142	biskuit	8000	PCS	1						Tersedia	30 November -0001	2022-07-09	

Gambar 4.27 Tampilan Halaman *Retailer Produk*

23. Halaman *Retailer* Transaksi

Halaman *retailer* transaksi merupakan halaman untuk melakukan proses transaksi penjualan dari pihak *retailer* kepada konsumen akhir pada halam ini terdapat kode transaksi yang telah otomatis dibuat oleh sistem dengan mengambil *username* dari masing-masing *retailer* kode berikutnya adalah tanggal waktu transaksi lalu kode terakhir adalah kode nomor penjualan yang otomatis bertambah pada transaksi berikutnya. Terdapat juga tanggal transaksi yang otomatis ter-*sating* dengan waktu saat ini namun tanggal ini dapat diubah oleh pengguna. Halaman selanjutnya adalah kolom untuk melakukan transaksi dengan cara memasukkan kode produk yang dapat dilakukan dengan pencarian kode produk ataupun nama produk, serta memasukkan jumlah kuantitas produk yang dijual. Terdapat *button add* yang berguna untuk menyimpan sementara data transaksi. Data yang telah di tambahkan atau *add* akan muncul ke tabel sementara yang berada dibawah kolom transaksi. Tabel ini memiliki beberapa kolom diantaranya no, kode produk, nama produk, harga, qnt atau jumlah kuantitas produk, diskon, total, dan fungsi hapus yang berguna untuk membatalkan list pesanan. Pada halaman ini juga terdapat *price* sementara untuk menampilkan jumlah harga sementara pada saat data barang dimasukan. Pada bagian akhir terdapat kolom pembayaran yang menampilkan total harga dan juga *field* untuk mengisi uang tunai serta secara otomatis akan menampilkan jumlah kembalian. Pada bagian ini terdapat dua *button* yaitu batal untuk membatalkan transaksi dan menghapus seluruh data transaksi yang sedang berlangsung serta *button* proses untuk memproses transaksi. Ketika transaksi telah selesai di proses maka data transaksi akan tersimpan. Berikut tampilan halaman *retailer* transaksi dapat dilihat pada Gambar 4.28.



Gambar 4.28 Tampilan Halaman *Retailer* Transaksi

24. Halaman *Retailer* Riwayat Transaksi

Halaman *retailer* riwayat transaksi merupakan halaman yang digunakan untuk menampilkan data riwayat transaksi milik *retailer*. Pada halaman ini menampilkan tabel yang memiliki beberapa kolom diantaranya kolom no, kode transaksi, kode produk, nama produk,

satuan, harga, qnt, diskon, subtotal dan tanggal transaksi. Pada halaman ini terdapat fungsi untuk melakukan pencarian dan memilih berapa jumlah data yang ingin ditampilkan. Berikut tampilan halaman *retailer* riwayat transaksi dapat dilihat pada Gambar 4.29.

The screenshot shows a sidebar menu on the left with options: Dashboard, Produk, Transaksi, Riwayat Transaksi (highlighted), Pelaporan, Riwayat Pelaporan, Reports, and Logout. The main content area is titled 'Tabel Produk' and contains a table of transactions.

No	Kode Transaksi	Kode Produk	Nama Produk	Satuan	Harga	Qty	Diskon	Subtotal	Tanggal
1.	rtiarasyid/02/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	14	0	196000	2022-01-02
2.	rtiarasyid/03/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	6	0	84000	2022-01-03
3.	rtiarasyid/04/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	4	0	56000	2022-01-04
4.	rtiarasyid/05/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	14	0	196000	2022-01-05
5.	rtiarasyid/06/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	17	0	238000	2022-01-06
6.	rtiarasyid/07/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	11	0	154000	2022-01-07
7.	rtiarasyid/08/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	8	0	112000	2022-01-08
8.	rtiarasyid/09/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	3	0	42000	2022-01-09
9.	rtiarasyid/10/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	1	0	14000	2022-01-10
10.	rtiarasyid/11/01/2022/001	10020031	GULA PASIR 1KG	PCS	14000	9	0	126000	2022-01-11

Showing 1 to 10 of 439 entries. Navigation: Prev 1 2 3 4 5 ... 44 Next

Gambar 4.29 Tampilan Halaman *Retailer* Riwayat Transaksi

25. Halaman *Retailer* Pelaporan

Halaman *retailer* pelaporan merupakan halaman yang berfungsi untuk melakukan pelaporan terkait perubahan barang seperti kehilangan barang, kerusakan barang, barang kadaluarsa dan lain sebagainya oleh pihak *retailer* kepada pihak distributor. Pada halaman ini terdapat *form* dengan beberapa *field* diantaranya tanggal yang berguna untuk memasukan tanggal kejadian dan *text* yang digunakan untuk mengisi keterangan pelaporan barang. Bagian terakhir terdapat *button* kirim yang berguna untuk mengirimkan data pelaporan kepada pihak distributor. Berikut tampilan halaman *retailer* pelaporan dapat dilihat pada Gambar 4.30.

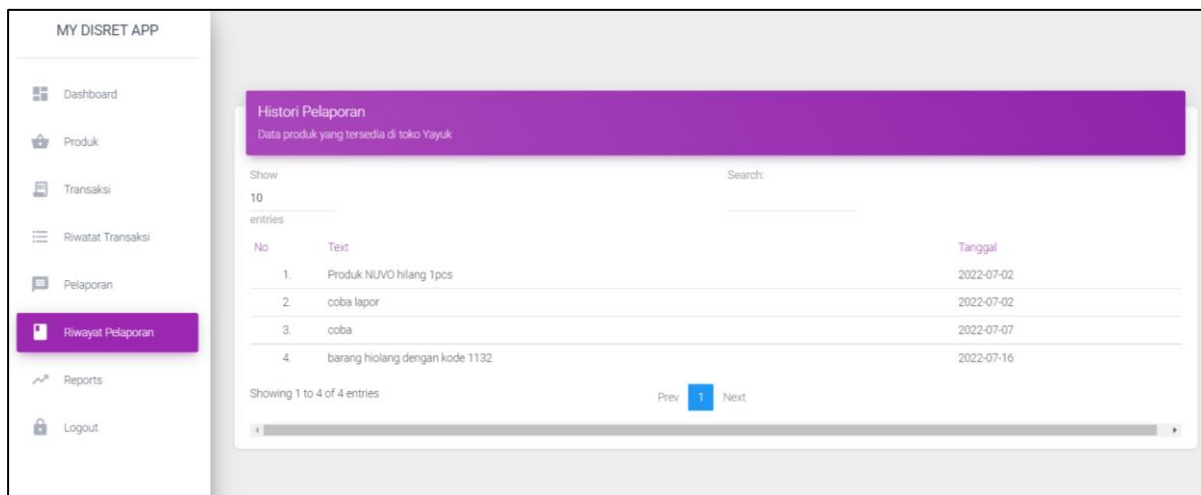
The screenshot shows the same sidebar menu as in Gambar 4.29, with 'Pelaporan' highlighted. The main content area is titled 'Pelaporan Barang' and contains a form with a 'Tanggal' field set to '05/09/2022' and a 'Text' field. A blue 'KIRIM' button is located at the bottom right of the form.

Gambar 4.30 Tampilan Halaman *Retailer* Pelaporan

26. Halaman *Retailer* Riwayat Pelaporan

Halaman *retailer* riwayat pelaporan merupakan halaman yang berguna untuk menampilkan riwayat pelaporan yang telah dilakukan oleh *retailer*. Pada halaman ini terdapat tabel yang berisi list riwayat pelaporan dengan beberapa kolom diantaranya adalah no, text, dan tanggal. Pada halaman ini juga memiliki fungsi pencarian dan pilihan untuk menentukan

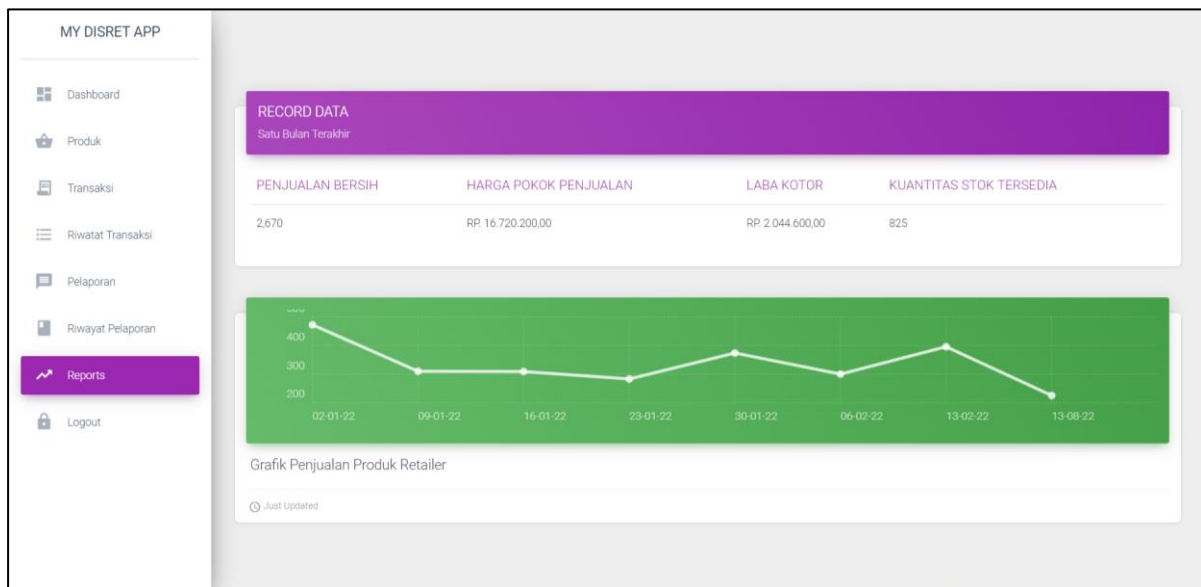
jumlah data yang ingin ditampilkan. Berikut tampilan halaman *retailer* riwayat pelaporan dapat dilihat pada Gambar 4.31.



Gambar 4.31 Tampilan Halaman *Retailer* Riwayat Pelaporan

27. Halaman *Retailer Reports*

Halaman *retailer reports* merupakan halaman yang menampilkan *reports* singkat terkait penjualan milik *retailer*. Pada halaman ini ditampilkan data total penjualan bersih, total dari harga pokok penjualan, total laba kotor serta kuantitas stok yang tersedia. Pada halaman ini juga ditampilkan grafik penjualan perminggu selama beberapa minggu terakhir. Berikut tampilan dari halaman *retailer reports* dapat dilihat pada Gambar 4.32.



Gambar 4.32 Tampilan Halaman *Retailer Reports*

4.2 Pengujian

Pengujian merupakan proses yang dilakukan untuk mengetahui apakah hasil implementasi suatu sistem dan algoritma dapat berjalan dengan optimal serta mengetahui apakah suatu sistem telah dapat memenuhi kebutuhan pengguna. Pada bagian ini pengujian dibagi menjadi dua tahapan yaitu pengujian fungsional sistem yang dilakukan dengan menggunakan metode *black box testing* yang berfokus pada persyaratan *interface* dan fungsional sistem yang kedua yaitu pengujian algoritma rekomendasi dengan menggunakan metode pengujian MAPE untuk mengetahui presentase *error* pada algoritma yang digunakan dengan membandingkan nilai hasil rekomendasi dengan nilai aktual.

4.2.1 Pengujian Fungsional Sistem

Pengujian ini berfungsi untuk mengetahui tingkat kelayakan sistem dari segi fungsional. Pengujian dilakukan oleh mahasiswa Informatika UPN “Veteran” Yogyakarta yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel Responden Pengujian Aplikasi

No	Nama	Pekerjaan	NIM
1	Tirta Buana Choir	Mahasiswa Informatika	123150039
2	Theresia Liana Sinaga	Mahasiswa Informatika	123150006
3	Mifta Diah Novitasari	Mahasiswa Informatika	123150073
4	Novita Indriani	Mahasiswa Informatika	123150115
5	Lolita Tambunan	Mahasiswa Informatika	123150003
6	Fitriana Kusumastuti	Mahasiswa Informatika	123150070
7	Linda Khoirul Inayah	Mahasiswa Informatika	123150122
8	Tita Rindu Astuti	Mahasiswa Informatika	123150129
9	Aisyah Nur Atiko	Mahasiswa Informatika	123150111
10	Anna Farida	Mahasiswa Informatika	123150082

Pengujian fungsional sistem bertujuan untuk mengetahui apakah fungsi-fungsi dalam sistem telah bekerja dengan sesuai. Data rincian pengujian fungsional dilampirkan pada Lampiran 1. Hasil dari pengujian fungsional sistem dapat dilihat pada tabel 4.2.

Tabel 4.2 Tabel Pengujian Aplikasi

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	10	0
2	Menampilkan halaman <i>login</i> distributor	10	0
3	Proses <i>login</i> distributor admin	10	0
4	Proses <i>login</i> distributor karyawan	10	0
5	Menampilkan halaman <i>dashoard</i>	10	0
6	Menampilkan halaman tabel produk <i>list</i>	10	0
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	10	0
8	Menampilkan halaman tambah kuantitas produk	10	0
9	Proses tambah kuantitas produk	10	0
10	Proses tambah produk baru	10	0
11	Proses <i>edit</i> produk	10	0
12	Proses hapus produk	10	0
13	Menampilkan halaman tabel karyawan	10	0
14	Menampilkan halaman detail data karyawan	10	0
15	Proses tambah, <i>edit</i> , hapus data karyawan	10	0
16	Menampilkan halaman data <i>retailer</i>	10	0
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	10	0
18	Menampilkan halaman detail data <i>retailer</i>	10	0
19	Menampilkan halaman edit produk <i>retailer</i>	10	0
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	10	0
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	10	0
22	Menampilkan halaman rekomendasi	10	0
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	10	0
24	Menampilkan detail perhitungan rekomendasi	10	0
25	Menampilkan halaman produk <i>retailer all</i>	10	0
26	Menampilkan halaman pengiriman	10	0
27	Menjalankan proses selesai, dibatalkan dan cetak pdf	10	0
28	Menampilkan halaman riwayat restok	10	0
29	Menampilkan halaman data pelaporan	10	0

Tabel 4.3 Lanjutan Tabel Pengujian Aplikasi

No	Item Uji	Penilaian	
		Berhasil	Gagal
30	Menampilkan halaman pengujian MAPE	10	0
31	Proses <i>logout</i>	10	0
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	10	0
33	Proses <i>login</i> retailer	10	0
34	Menampilkan halaman <i>dashboard</i>	10	0
35	Menampilkan halaman tabel produk	10	0
36	Menampilkan halaman transaksi	10	0
37	Menjalankan proses transaksi	10	0
38	Menampilkan halaman riwayat transaksi	10	0
39	Menampilkan halaman pelaporan	10	0
40	Menjalankan fungsi pelaporan	10	0
41	Menampilkan halaman riwayat pelaporan	10	0
42	Menampilkan halaman <i>reports</i>	10	0
43	Proses <i>logout</i>	10	0
Jumlah tiap penilaian		430	0

Untuk menghitung nilai persentase masing-masing pilihan jawaban digunakan rumus sebagai berikut:

$$\text{Persentase} = \frac{\text{Jumlah tiap Penilaian}}{\text{Total Penilaian}} \times 100\%$$

Hasil persentase dari penilaian sistem yang dilakukan oleh responden yang memberi penilaian dapat dilihat dengan perhitungan sebagai berikut:

$$\text{Berhasil} = \frac{430}{10 \times 43} \times 100\% = 100\%$$

$$\text{Gagal} = \frac{0}{0 \times 43} \times 100\% = 0\%$$

Berdasarkan hasil pengujian fungsional sistem diperoleh nilai pengujian berhasil sebesar 100%. Yang dibuktikan dengan kuisioner yang di isi oleh pengguna yang mencoba menjalankan sistem. Kuisioner selengkapnya dapat dilihat pada lampiran 1.

4.2.2 Pengujian MAPE

Pengujian MAPE digunakan untuk menguji tingkat kesalahan hasil rekomendasi yang dihasilkan yang dilakukan dengan menggunakan algoritma *Fuzzy Tsukamoto*. Pengujian ini dilakukan dengan membandingkan data hasil rekomendasi dengan data aktual penjualan pada minggu berikutnya. Pada pengujian MAPE memiliki empat kriteria data hasil uji diantaranya data dengan tingkat kesalahan kurang dari 10% berarti sangat baik, data dengan kesalahan antara 10-20 % berarti baik, data dengan tingkat kesalahan 20-50% berarti layak, sementara tingkat kesalahan lebih dari 50% maka dikatakan buruk (Puspitasari et al., 2019). Berikut merupakan rumus yang digunakan untuk mencari nilai MAPE:

$$PE = \frac{|At - Ft|}{At} \times 100$$

$$MAPE = \sum \frac{PE}{n}$$

Keterangan:

n = jumlah periode data

At = nilai aktual pada data t

Ft = nilai prediksi pada data

Data yang akan diuji merupakan data sampel yang diambil dari data transaksi dari toko

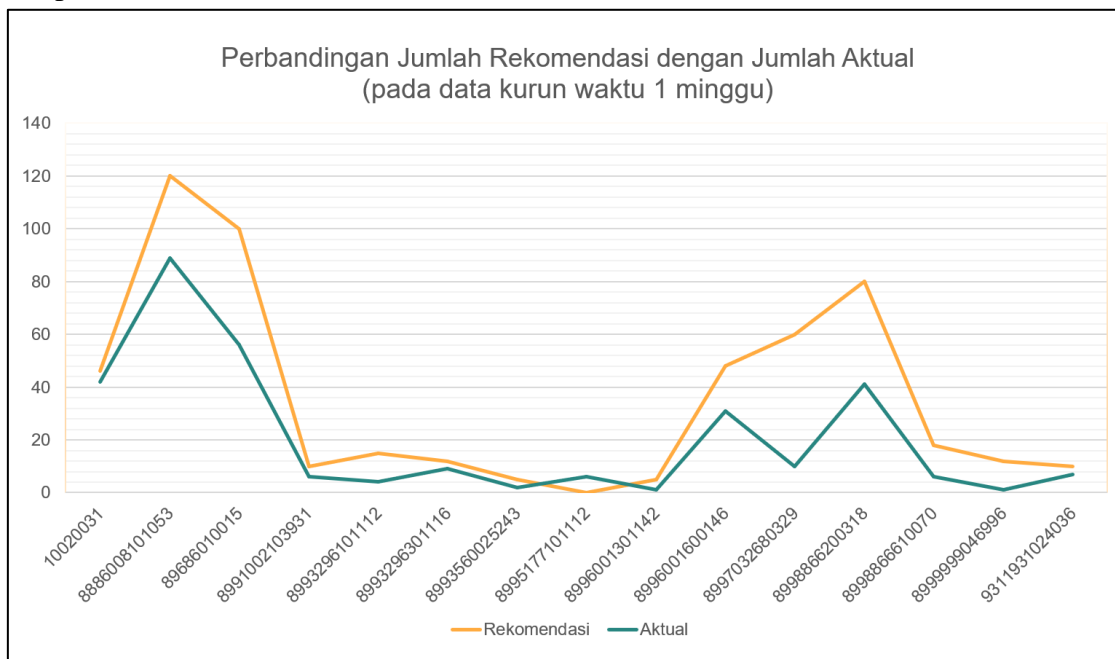
Arasyid pada bulan Januari hingga Februari 2022. Berikut data produk yang digunakan sebagai sampel dalam pengujian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Tabel Data Sample Produk

No	Kode Produk	Nama Produk
1	10020031	Gula Pasir 1kg
2	8886008101053	AQUA BTL 600ml
3	89686010015	Indomie Ayam Bawang
4	8991002103931	Good Day Carrebian Nut 20gr Sachet
5	8993296101112	Cakra Kembar 1kg Ekonomi
6	8993296301116	Bogasari Segitiga Biru 1kg
7	8993560025243	DETTOL 100gr
8	8995177101112	Gulaku Pck 1kg
9	8996001301142	Roma Biskuit Kelapa 300g
10	8996001600146	Teh Pucuk Harum 350ml
11	8997032680329	Taro Net 40g
12	8998866200318	Sedap Mie Ayam Bawang
13	8998866610070	SO KLIN Deterjen 275 Gram
14	8999999046996	Lifeboy 60gr
15	9311931024036	INDOCAFE COFFEE MIX 3 IN 1PCK 10x20g

Tabel 4.4 merupakan tabel yang menampilkan data produk yang akan di uji. Terdapat 15 macam produk yang diambil secara acak. Dari 15 produk ini dilakukan pendataan jumlah penjualan yang dirangkum setiap minggunya yang kemudian akan dibandingkan dengan data rekomendasi yang telah dihitung dengan menggunakan algoritma *Fuzzy Tsukamoto*.

Pengujian pertama dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada satu minggu pertama yang dibandingkan dengan data penjualan asli pada minggu ke-2. Berikut grafik perbandingan nilai hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan nilai penjualan asli pada minggu berikutnya dapat dilihat pada Gambar 4.33.



Gambar 4.33 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Satu Minggu

Gambar 4.33 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama satu minggu. Garis berwarna oranye menunjukan nilai rekomendasi sementara garis berwarna hijau tua menunjukan nilai aktual penjualan pada minggu berikutnya. Misalnya produk dengan kode 10020031 memiliki nilai hasil rekomendasi 46 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 42 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 120 buah dengan nilai aktual yang lebih sedikit yaitu 89 buah dan seterusnya.

Kemudian data yang telah didapat dihitung dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu satu minggu dapat dilihat pada Tabel 4.5.

Tabel 4.5 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Satu Minggu

No	Kode Produk	Data hasil kalkulasi <i>Fuzzy Tsukamoto</i>	Data Aktual	$ At - Ft $	PE%
1	10020031	46	42	4	9,523%
2	8886008101053	120	89	31	34,831%
3	89686010015	100	56	44	78,571%
4	8991002103931	10	6	4	66,667%
5	8993296101112	15	4	11	275%
6	8993296301116	12	9	3	33.333%
7	8993560025243	5	2	3	150%
8	8995177101112	0	6	6	100%
9	8996001301142	5	1	4	400%
10	8996001600146	48	31	17	54,838%
11	8997032680329	60	10	50	500%
12	8998866200318	80	41	39	95,121%
13	8998866610070	18	6	12	200%
14	8999999046996	12	1	11	1100%
15	9311931024036	10	7	3	42,857%
Jumlah PE%					3140,744%

Berdasarkan tabel 4.5 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu satu minggu adalah sebagai berikut:

$$MAPE = \frac{3140,744\%}{15} = 209,382\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu satu minggu didapatkan nilai rata-rata kesalahan sebesar 209,382%. Hal ini menunjukan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil buruk untuk pengujian data sampel dalam kurun waktu satu minggu berdasarkan kriteria yang ditunjukkan pada tabel 2.5.

Pengujian kedua dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada kurun waktu dua minggu yang dibandingkan dengan data penjualan asli pada minggu ke-3. Berikut grafik perbandingan nilai hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan nilai penjualan asli pada minggu berikutnya dapat dilihat pada Gambar 4.34.



Gambar 4.34 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Dua Minggu

Gambar 4.34 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama dua minggu. Garis berwarna oranye menunjukan nilai rekomendasi sementara garis berwarna hijau tua menunjukan nilai aktual penjualan pada minggu berikutnya. Diketahui produk dengan kode 10020031 memiliki nilai hasil rekomendasi 43 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 32 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 97 buah dengan nilai aktual yang lebih tinggi yaitu 108 buah dan seterusnya.

Selanjutnya dilakukan perhitungan dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu dua minggu dapat dilihat pada Tabel 4.6.

Tabel 4.6 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Dua Minggu

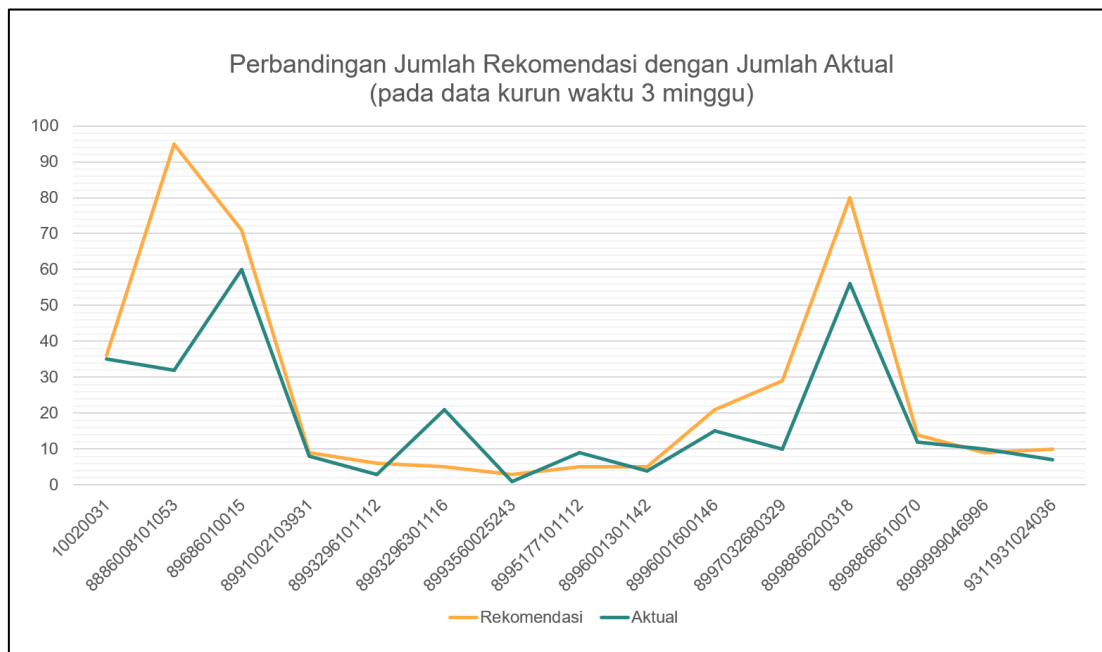
No	Kode Produk	Data hasil kalkulasi Fuzzy Tsukamoto	Data Aktual	$ At - Ft $	PE%
1	10020031	43	32	11	34,375%
2	8886008101053	97	108	11	10,185%
3	89686010015	76	67	9	13,432%
4	8991002103931	8	10	2	20%
5	8993296101112	8	7	1	14,285%
6	8993296301116	6	9	3	33,333%
7	8993560025243	7	1	6	600%
8	8995177101112	6	6	0	0%
9	8996001301142	8	4	4	150%
10	8996001600146	25	10	15	433,333%
11	8997032680329	32	6	26	116,216%
12	8998866200318	80	37	43	300%
13	8998866610070	16	4	12	300%
14	8999999046996	12	3	9	1100%
15	9311931024036	10	6	4	66,667%
Jumlah PE%					2191,828%

Berdasarkan tabel 4.6 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu tiga minggu adalah sebagai berikut:

$$\text{MAPE} = \frac{2192,828\%}{15} = 146,121\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu tiga minggu didapatkan nilai rata-rata kesalahan sebesar 146,121%. Hal ini menunjukkan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil buruk untuk pengujian data sampel dalam kurun waktu tiga minggu berdasarkan kriteria yang ditunjukkan pada tabel 2.5.

Pengujian ketiga dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada kurun waktu tiga minggu yang dibandingkan dengan data penjualan asli pada minggu ke-4. Berikut grafik perbandingan nilai hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan nilai penjualan asli pada minggu berikutnya dapat dilihat pada Gambar 4.35.



Gambar 4.35 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Tiga Minggu

Gambar 4.35 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama tiga minggu. Diketahui produk dengan kode 10020031 memiliki nilai hasil rekomendasi 36 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 35 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 95 buah dengan nilai aktual yang jauh lebih rendah yaitu 32 buah dan seterusnya.

Selanjutnya dilakukan perhitungan dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dengan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu tiga minggu dapat dilihat pada Tabel 4.7.

Tabel 4.7 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Tiga Minggu

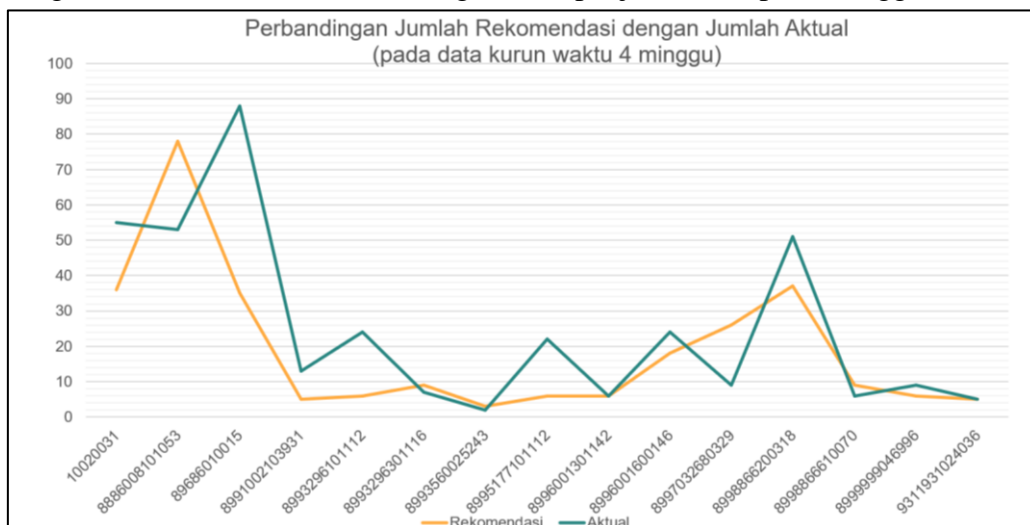
No	Kode Produk	Data hasil kalkulasi <i>Fuzzy Tsukamoto</i>	Data Aktual	$ At - Ft $	PE%
1	10020031	36	35	1	2,857%
2	8886008101053	95	32	63	196,875%
3	89686010015	71	60	11	18,333%
4	8991002103931	9	8	1	12,5%
5	8993296101112	6	3	3	100%
6	8993296301116	5	21	16	76,190%
7	8993560025243	3	1	2	200%
8	8995177101112	5	9	4	44,444%
9	8996001301142	5	4	1	25%
10	8996001600146	21	15	6	40%
11	8997032680329	29	10	19	190%
12	8998866200318	80	56	24	42,857%
13	8998866610070	14	12	2	16,667%
14	8999999046996	9	10	1	10%
15	9311931024036	10	7	3	42,857%
Jumlah PE%					1018,581%

Berdasarkan tabel 4.7 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu tiga minggu adalah sebagai berikut:

$$MAPE = \frac{1018,581\%}{15} = 67,905\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu tiga minggu didapatkan nilai rata-rata kesalahan sebesar 67,905%. Hal ini menunjukan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil buruk untuk pengujian data sampel dalam kurun waktu tiga minggu berdasarkan kriteria yang ditunjukan pada tabel 2.5.

Pengujian keempat dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada kurun waktu empat minggu yang dibandingkan dengan data penjualan asli pada minggu ke-5. Berikut Gambar 4.36 merupakan grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan asli pada minggu berikutnya.



Gambar 4.36 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Empat Minggu

Gambar 4.36 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama empat minggu. Nilai rekomendasi diperoleh dari perhitungan data penjualan selama empat minggu dengan menggunakan algoritma *Fuzzy Tsukamoto*. Sementara nilai aktual adalah nilai penjualan aktual pada minggu ke-5. Diketahui produk dengan kode 10020031 memiliki nilai hasil rekomendasi 36 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 55 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 78 buah dengan nilai aktual yang lebih rendah yaitu 53 buah dan seterusnya.

Selanjutnya dilakukan perhitungan dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dengan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu tiga minggu dapat dilihat pada Tabel 4.8.

Tabel 4.8 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Empat Minggu

No	Kode Produk	Data hasil kalkulasi <i>Fuzzy Tsukamoto</i>	Data Aktual	$ At - Ft $	PE%
1	10020031	36	55	19	34,545%
2	8886008101053	78	53	25	47,169%
3	89686010015	35	88	53	60,227%
4	8991002103931	5	13	8	61,538%
5	8993296101112	6	24	18	75%
6	8993296301116	9	7	2	28,571%
7	8993560025243	3	2	1	50%
8	8995177101112	6	22	16	72,73%
9	8996001301142	6	6	0	0%
10	8996001600146	18	24	6	25%
11	8997032680329	26	9	17	188,889%
12	8998866200318	37	51	14	27,451%
13	8998866610070	9	6	3	50%
14	8999999046996	6	9	3	33,333%
15	9311931024036	5	5	0	0%
Jumlah PE%					754,453%

Berdasarkan tabel 4.8 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu empat minggu adalah sebagai berikut:

$$MAPE = \frac{754,453\%}{15} = 50,23\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu empat minggu didapatkan nilai rata-rata kesalahan sebesar 50,23%. Hal ini menunjukkan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil buruk untuk pengujian data sampel dalam kurun waktu empat minggu berdasarkan kriteria yang ditunjukkan pada tabel 2.5.

Pengujian kelima dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada kurun waktu lima minggu yang dibandingkan dengan data penjualan asli pada minggu ke-5. Berikut grafik perbandingan nilai hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan nilai penjualan asli pada minggu berikutnya dapat dilihat pada Gambar 4.37.



Gambar 4.37 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Lima Minggu

Gambar 4.37 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama lima minggu. Nilai rekomendasi diperoleh dari perhitungan data penjualan selama lima minggu dengan menggunakan algoritma *Fuzzy Tsukamoto*. Sementara nilai aktual adalah nilai penjualan aktual pada minggu ke-6. Diketahui produk dengan kode 10020031 memiliki nilai hasil rekomendasi 43 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 68 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 60 buah dengan nilai aktual yang lebih rendah yaitu 51 buah dan seterusnya.

Selanjutnya dilakukan perhitungan dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dengan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu tiga minggu dapat dilihat pada Tabel 4.9.

Tabel 4.9 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Lima Minggu

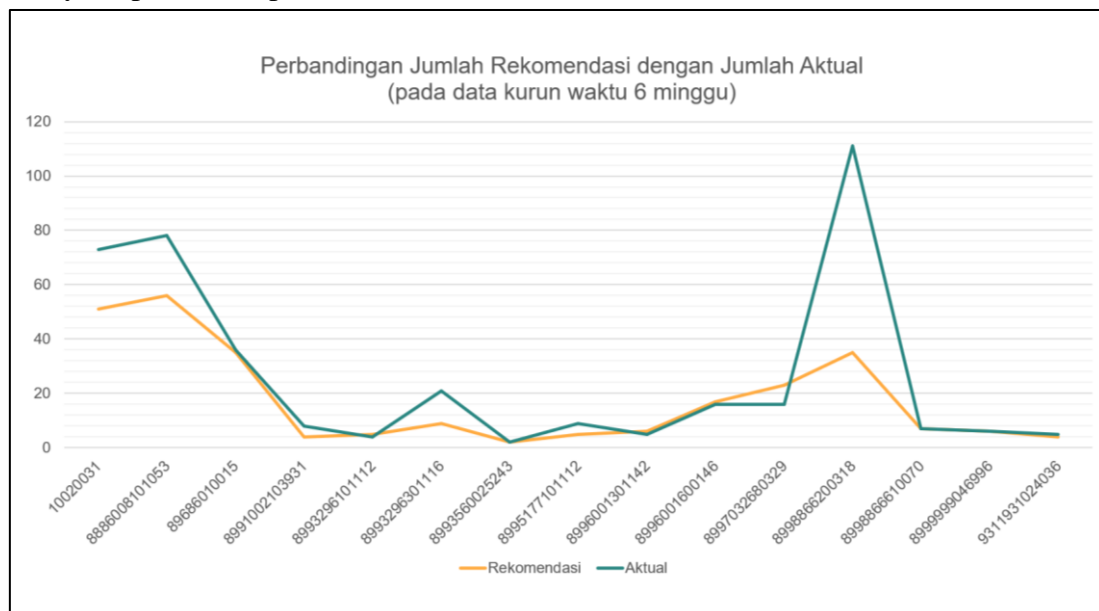
No	Kode Produk	Data hasil kalkulasi <i>Fuzzy Tsukamoto</i>	Data Aktual	$ At - Ft $	PE%
1	10020031	43	68	25	36,767%
2	8886008101053	60	51	9	17,647%
3	89686010015	44	58	14	24,138%
4	8991002103931	4	9	5	55,555%
5	8993296101112	6	7	1	14,285%
6	8993296301116	9	9	0	0%
7	8993560025243	3	3	0	0%
8	8995177101112	5	10	5	50%
9	8996001301142	5	5	0	0%
10	8996001600146	20	8	12	150%
11	8997032680329	25	14	11	78,571%
12	8998866200318	38	42	4	9,523%
13	8998866610070	7	5	2	40%
14	8999999046996	7	3	4	133,333%
15	9311931024036	4	8	4	50%
Jumlah PE%					659,819%

Berdasarkan tabel 4.9 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu lima minggu adalah sebagai berikut:

$$\text{MAPE} = \frac{659,819\%}{15} = 43,988\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu lima minggu didapatkan nilai rata-rata kesalahan sebesar 43,988%. Hal ini menunjukan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil layak untuk data sampel dalam kurun waktu lima minggu berdasarkan kriteria yang ditunjukkan pada tabel 2.5.

Pengujian keenam dilakukan dengan membandingkan rekomendasi restok yang dihitung menggunakan data penjualan pada kurun waktu enam minggu yang dibandingkan dengan data penjualan asli pada minggu ke-7. Berikut grafik perbandingan nilai hasil rekomendasi dengan algoritma *Fuzzy Tsukamoto* dengan nilai penjualan asli pada minggu berikutnya dapat dilihat pada Gambar 4.38.



Gambar 4.38 Grafik Perbandingan Nilai Rekomendasi dan Nilai Aktual pada Data dengan Kurun Waktu Enam Minggu

Gambar 4.38 merupakan gambar grafik perbandingan nilai hasil rekomendasi dengan nilai penjualan aktual pada data transaksi selama enam minggu. Nilai rekomendasi diperoleh dari perhitungan data penjualan selama enam minggu dengan menggunakan algoritma *Fuzzy Tsukamoto*. Sementara nilai aktual adalah nilai penjualan aktual pada minggu ke-7. Diketahui produk dengan kode 10020031 memiliki nilai hasil rekomendasi 51 buah dengan nilai aktual penjualan pada minggu berikutnya sebanyak 73 buah. Data berikutnya yaitu produk dengan kode 8886008101053 memiliki nilai rekomendasi sebanyak 56 buah dengan nilai aktual yang lebih rendah yaitu 78 buah dan seterusnya.

Selanjutnya dilakukan perhitungan dengan menggunakan metode pengujian MAPE yaitu dengan cara membandingkan nilai rekomendasi dengan nilai aktual kemudian dicari nilai selisih dan presentase PE. Berikut pengujian MAPE dengan data kurun waktu tiga minggu dapat dilihat pada Tabel 4.10.

Tabel 4.10 Tabel Pengujian MAPE pada Data dengan Kurun Waktu Lima Minggu

No	Kode Produk	Data hasil kalkulasi <i>Fuzzy Tsukamoto</i>	Data Aktual	$ At - Ft $	PE%
1	10020031	51	73	22	30,136%
2	8886008101053	56	78	22	28,205%
3	89686010015	35	36	1	2,778%
4	8991002103931	4	8	4	50%
5	8993296101112	5	4	1	25%
6	8993296301116	9	21	12	57,143%
7	8993560025243	2	2	0	0%
8	8995177101112	5	9	4	44,444%
9	8996001301142	6	5	1	20%
10	8996001600146	17	16	1	6,25%
11	8997032680329	23	16	7	43,75%
12	8998866200318	35	111	76	68,468%
13	8998866610070	7	7	0	0%
14	8999999046996	6	6	0	0%
15	9311931024036	4	5	1	20%
Jumlah PE%					396,175%

Berdasarkan tabel 4.10 maka persentase tingkat *error* algoritma *Fuzzy Tsukamoto* dengan menggunakan metode MAPE pada data transaksi dengan kurun waktu enam minggu adalah sebagai berikut:

$$MAPE = \frac{396,175\%}{15} = 26,411\%$$

Berdasarkan hasil pengujian yang dilakukan terhadap 15 data dengan perhitungan MAPE pada data transaksi dengan kurun waktu enam minggu didapatkan nilai rata-rata kesalahan sebesar 26,411%. Hal ini menunjukan bahwa algoritma *Fuzzy Tsukamoto* memiliki hasil layak untuk data sampel dalam kurun waktu enam minggu berdasarkan kriteria yang ditunjukkan pada tabel 2.5.

4.3 Pembahasan

Pada penelitian ini dilakukan untuk mengetahui bagaimana implementasi konsep manajemen rantai pasok dengan model VMI satu pemasok banyak *retailer* pada sebuah sistem yang mana distributor berperan sebagai gudang utama sekaligus sebagai vendor. Implementasi dalam penelitian ini dilakukan dengan pembuatan sistem berbasis web yang terbagi menjadi dua bagian *interface* yaitu *interface* untuk *retailer* dan *interface* untuk distributor. Pada *interface retailer* terdapat fungsi *point of sale* yang berguna untuk merekam data transaksi dan disimpan pada satu *database* pusat. Sementara *interface* distributor memiliki beberapa fungsi terkait pengadaan barang, pengiriman, serta menampilkan informasi pengelompokan dan rekomendasi yang diperoleh dari pengolahan data transaksi milik *retailer* yang telah disimpan pada *database* pusat. Proses pengelompokan penjualan produk dilakukan dengan algoritma *k-means* yang membagi data penjualan menjadi tiga *cluster* kemudian setiap *cluster* akan diurutkan dari nilai terendah hingga nilai tertinggi yang selanjutnya dikategorikan menjadi tiga kelompok data yaitu kurang laris, cukup laris dan sangat laris.

Penelitian ini juga dilakukan untuk mengetahui bagaimana implementasi algoritma *Fuzzy Tsukamoto* untuk menghitung nilai rekomendasi jumlah penjualan minggu berikutnya yang dilakukan dengan mengolah data produk dan data transaksi milik masing-masing *retailer*. Data ini diperoleh dengan melakukan observasi dan pencatatan secara langsung terhadap beberapa *retailer* tradisional di kecamatan Selo, kabupaten Boyolali. Data yang diperoleh

kemudahan diolah sesuai dengan variabel yang dibutuhkan untuk melakukan perhitungan dengan algoritma yang digunakan sehingga menghasilkan nilai rekomendasi penyetoran pada tiap-tiap *retailer*.

Pada pengujian fungsional sistem yang telah dilakukan dengan metode *black box* diperoleh nilai pengujian berhasil sebesar 100% yang berarti sistem telah bekerja dengan baik. Sementara pengujian algoritma dilakukan dengan metode MAPE untuk menguji tingkat kesalahan pada algoritma *Fuzzy Tsukamoto* dalam melakukan perekomendasi. Data yang digunakan untuk pengujian ini adalah data penjualan toko Arasyid selama bulan Januari 2022 hingga bulan februari 2022 yang dikelompokkan perminggu. Pengujian dilakukan dengan cara membandingkan nilai rekomendasi dengan nilai aktual pada data tiap minggunya dan diperoleh hasil yaitu pada proses rekomendasi dengan pengolahan data transaksi selama satu minggu menghasilkan nilai rekomendasi dengan tingkat kesalahan sebanyak 209,382%, kemudian pada pengolahan data selama dua minggu menghasilkan nilai rekomendasi dengan tingkat kesalahan sebanyak 146,121%, pada pengolahan data selama tiga minggu menghasilkan nilai rekomendasi dengan tingkat kesalahan sebanyak 67,905%, pada pengolahan data selama empat minggu menghasilkan nilai rekomendasi dengan tingkat kesalahan sebanyak 50,23% yang berarti algoritma *Fuzzy Tsukamoto* bekerja dengan buruk berdasarkan kriteria MAPE. Namun pada pengujian yang dilakukan dengan data transaksi selama lebih dari empat minggu menghasilkan hasil yang layak berdasarkan kriteria MAPE yaitu pada minggu ke lima dan minggu ke enam yang menghasilkan rata-rata kesalahan sebesar 43,988% dan 26,411%. Berdasarkan hasil tersebut, diketahui algoritma *Fuzzy Tsukamoto* bekerja dengan layak untuk melakukan rekomendasi pada data penjualan *retailer* Arasyid dalam kurun waktu lima minggu atau lebih, amun bekerja dengan buruk pada data transaksi kurang dari lima minggu.

BAB V

PENUTUP

Pada bab penutup membahas mengenai kesimpulan dan saran dari hasil penelitian tugas akhir ini.

5.1 Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan mengenai manajemen rantai pasok terintegrasi dengan konsep VMI menggunakan algoritma *Fuzzy Tsukamoto* dapat diambil kesimpulan bahwa:

1. Implementasi manajemen rantai pasok terintegrasi menggunakan model VMI dengan satu pemasok dan banyak *retailer* telah berhasil dilakukan. Sistem ini mampu mengintegrasikan data barang dan penjualan milik masing-masing *retailer* kepada pihak distributor yang berperan sebagai gudang utama. Pada pengujiannya yang dilakukan dengan metode *black box tasting* menghasilkan nilai 100% berhasil yang berarti sistem dapat bekerja dengan baik.
2. Algoritma *fuzzy tsukamoto* diketahui mampu bekerja dengan layak untuk melakukan rekomendasi jumlah restok oleh distributor kepada pihak *retailer* pada data penjualan *retailer* Arasyid dalam data kurun waktu lima minggu atau lebih, namun bekerja dengan buruk pada data transaksi kurang dari lima minggu. Hal ini dibuktikan dengan hasil pengujian MAPE yang telah dilakukan pada data transaksi selama lima minggu dan enam minggu yang masing-masing mendapatkan hasil perhitungan nilai rata-rata kesalahan sebesar 43,988% dan 26,411% yang masuk dalam kategori layak serta mendapatkan nilai rata-rata kesalahan lebih dari 50% pada pengujian data transaksi dengan kurun waktu kurang dari lima minggu yang masuk dalam kategori buruk berdasarkan kriteria MAPE.
3. Pada penelitian ini algoritma *K-Means* hanya berfungsi untuk mengelompokan tingkat penjualan produk menjadi tiga *cluster* yaitu laris, cukup laris dan kurang laris.

5.2 Saran

Berdasarkan pembahasan yang telah disampaikan pada bab sebelumnya, saran yang diberikan oleh penulis untuk penelitian selanjutnya agar memberikan pengembangan penelitian yang lebih baik yaitu:

1. Mengembangkan sistem yang telah dibuat menjadi lebih luas dengan mengintegrasikan sistem kepada pihak *supplier* dan pihak manufaktur.
2. Pada proses rekomendasi pengiriman sebaiknya mempertimbangkan juga biaya transportasi, jarak pengiriman serta biaya penyimpanan barang.
3. Menambahkan algoritma atau metode perhitungan lain untuk mendukung pengolahan data menjadi informasi lain yang dibutuhkan seperti penentuan promo, pengadaan barang dan lain sebagainya.

DAFTAR PUSTAKA

- Adithama, A. A., Dewi, F. K., & Hariyadi, E. (2020). Penerapan Algoritma Apriori dan Fuzzy Tsukamoto untuk Rekomendasi Jumlah Pembelian Barang dan untuk Rekomendasi Jumlah Pembelian Barang dan. *JUITA: Jurnal Informatika* e-ISSN: 2579-8901.
- Agus, Fahrul ; et al. (2016). Comparative Analysis of Tsukamoto and Sugeno Fuzzy Methods For Procurement Optimization. *International Journal of Computing and Informatics (IJCANDI)*.
- Anna, I. D. (2016). The Implementation Of Vendor Managed Inventory In The Supply Chain With Simple Probabilistic Inventory Model. *MATEC Web of Conferences* 58, 1-6.
- Beu, L., & Husna, A. (2019). Metode Fuzzy Tsukamoto Untuk Memprediksi Jumlah Produksi Kue Pia . *Jurnal Nasional cosPhi*, 46-49.
- Harefa, K. (2017). Penerapan Fuzzy Inference System Untuk Menentukan Jumlah Pembelian Produk Berdasarkan Data Persediaan Dan Penjualan. *Jurnal Informatika Universitas Pamulang*, 205-213.
- Hayati, E. N. (2014). Supply Chain Management (Scm) Dan Logistic Management. *Hayati*, 25-34.
- Hayati, E. N. (2014). Supply Chain Management (Scm) Dan Logistic Management. *Jurnal Dinamika Teknik*, 25-34.
- Herianto; at al. (2021). Analisa Tingkat Penjualan Produk Menggunakan K-Nearest Neighbor (K-Nn) Dan K-Means (Studi Kasus Perusahaan Kayu Elang Perkasa). *Jurnal Sains Dan Teknologi* , 8-18.
- Hutauruk, A. C., & Pakpahan, A. F. (2021). Perancangan Sistem Informasi Organisasi Kemahasiswaan Perancangan Sistem Informasi Organisasi Kemahasiswaan Menggunakan Metode Agile Development (Studi Kasus:Menggunakan Metode Agile Development (Studi Kasus:Menggunakan Metode Agile Development. *Cogito Smart Journal / VOL. 7 - NO.2*, 315-328.
- Imron, Mohammad; etal. (2020). Analysis of Data Mining Using K-Means Clustering Algorithm for Product Grouping. *Analysis of Data Mining Using K-Means Clustering Algorithm for Product*, 12-22.
- indrajat, r. e. (2016). *electronic business*. yogyakarta: preinexus.
- Jabat, J. T., & Murdani. (2019). Penerapan Data Mining Pada Penjualan Produk Retail Penerapan Data Mining Pada Penjualan Produk Retail. *Jurnal Pelita Informatika*, 26-32.
- James S, K., & Craig A, H. (2013). A conceptual model for the measurement of the expected value of a supply chain relationship. *Int. J. Logistics Economics and Globalisation*, 43-58.
- Jumaidi, L. T., Jalaludin, & Ahyar, M. (2019). Eksistensi Minimarket Terhadap Kelangsungan Usaha Toko Kelontong Dan Waserda. *Jurnal Magister Manajemen Unram*, 186-202.
- Karma. (2018). Perancangan Sistem Informasi Manajemen pada Industri Farmasi dengan Pendekatan Vendor Managed Inventory (VMI) Berbasis Cloud Computing. *Thesis*.
- Kosasih,, B. C., & Setiyawati, N. (2020). Sistem Pendukung Keputusan Penentuan Pemesanan Barang Menggunakan Logika Fuzzy Tsukamoto (Studi Kasus: Studio Foto Kencana). *Jurnal Algoritma, Logika dan Komputasi*, 215-222.
- Lestario, F. (2017). Dampak Pertumbuhan Bisnis Franchise Waralaba Minimarket Terhadap

- Perkembangan Kedai Tradisional Di Kota Binjai. *Jurnal Manajemen Tools*, 29-36.
- Lingxian, You ; et al. (2019). Online Retail Sales Prediction with Integrated Framework of K-mean and Neural Network. *Association for Computing Machinery*, 115-118.
- Mahendra, I., & Yanto, D. T. (2018). Agile Development Methods dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus : Bank BRI Unit Kolonel Sugiono). *Jurnal Teknologi dan Open Source*, 13-24.
- Mandala, E. P., Yanto, M., & Putri, D. E. (2018). Aplikasi Pengelompokan Penjualan Dengan Clustering Data Mining Pada Toko Retail Kota Padang. *PROSIDING SISFOTEK (Sistem Informasi dan Teknologi)*.
- Maulina, & Rahmi , A. N. (2016). Peranan Pengendalian Internal Persediaan Terhadap Pengelolaan Persediaan Barang Dagang Pada Toserba Berkah Baru Cibadak. *Jurnal Ilmiah Ilmu Ekonomi* , 37-54.
- Mohammadi, R., Bidabad, B., Nourasteh, T., & Sherafati, M. (2014). Credit Ranking of Bank Customers (An Integrated Model of RFM, FAHP and K-means) . *European Online Journal of Natural and Social Sciences 2014*, 563-571.
- Muningsih, E., & Kiswati, S. (2015). Penerapan Metode K-Means Untuk Clustering Produk Online Shop Dalam Penentuan Stok Barang. *Jurnal Bianglala Informatika* , 10-17.
- Negara, Iis Setiawan Mangku; et al. (2022). Analisa Cluster Data Transaksi Penjualan Minimarket Selama Pandemi Covid-19 dengan Algoritma K-means. (*JOINTECS*) *Journal of Information Technology and Computer Science*, 153-160.
- Novianti, Andi Gita ; et al. (2018). Penerapan Metode Fuzzy Tsukamoto Untuk Prediksi Penerapan Metode Fuzzy Tsukamoto Untuk Prediksi Penerapan Metode Fuzzy Tsukamoto Untuk Prediksi. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 611-618.
- Nuha , M. S. (2017). Implementasi Web Service Restfull Dalam Aplikasi Sistem Point Of Sale (Jsc Pos Sistem) Studi Kasus Toko Retail Jasa Sarana Computer Surabaya. *Tugas Akhir*.
- Nur, T. M. (2015). Analisis Perubahan Pendapatan Usaha Pedagang Eceran Sebelum Dan Sesudah Berdirinya Indomaret Dan Alfamart Di Kecamatan Tampan Pekanbaru. *Jom Fekon*, 1-10.
- Putra, R. R., & Wadisman, C. (2018). Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K-Means. *Journal of Information Technology and Computer Science (INTECOMS)*, 72-77.
- Qabbaah, Hamzah ; at al. (2019). Using K-Means Clustering and Data Visualization for Monetizing logistics Data. 1-6.
- Runyan, S. A. (2015). *Introduction to Agile Methods, 1st ed.* . New Jersey: earson Education, Inc.
- Septianingrum, A. W. (2021). *Pelaksanaan Perjanjian Waralaba (Franchise Agreement) Chic Lin Di Kota Jepara*.
- Silalahi, M. (2018). Analisa Clustering Menggunakan Algoritma K-means Terhadap Penjualan Produk Pada PT Batamas Niaga Jaya. *Computer Based Information System Journal*, 20-35.
- Silveira , P., & Marreiros , C. (2014). Shopper Marketing: A Literature Review. *International Review of Management and Marketing*, 4, 90-97.

- Sommerville, I. (2011). *Software Engineering (Rekayasa Perangkat Lunak)*.
- Triyawan, A., & Jayanti, K. F. (2018). Analisis Pengaruh Toko *Retail* Modern Terhadap Pendapatan Pedagang Tradisional dalam Perspektif Ekonomi Islam. *Jurnal Ekonomi Syariah*, 1-11.
- Triyawan, A., & Jayanti, K. F. (2018). Analisis Pengaruh Toko *Retail* Modern Terhadap Pendapatan Pedagang Tradisional dalam Perspektif Ekonomi Islam. *Jurnal Ekonomi Syariah*, 1-11.
- Wibawa, I Made Bagus Wiradivka Laksa ; et al. (2018). In this base model, in general there are three steps to In this base model, in general there are three steps to. *International Journal of Engineering and Emerging Technology*, 97-103.
- Wibawa, I. L., Satrio, M. R., & Ariyani, N. W. (2018). Recommendation System Cooperative Stock Goods Orders Using Fuzzy Tsukamoto. *International Journal of Engineering and Emerging Technology*, 97-103.
- Yosefa, Sitompul, C., & Alfian. (2015). Perancangan Model VMI (Vendor Managed Inventory) dengan Satu Pemasok dan Banyak *Retailer* yang Meminimasi Ongkos Total Rantai Pasok. *Jurnal Rekayasa Sistem Industri*, 88-96.
- Zain, N., Novianty, A., & Michrandi, S. (2015). Sistem Pendukung Keputusan Untuk Menentukan Besaran Order Menggunakan Algoritma Fuzzy Dalam Toko Pintar Tanpa Kasir. *e-proceeding of engineering*, 3598-3607.

LAMPIRAN

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Tirta Buana Choir
 NIM : 123150039
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Theresia Liana Sinaga
 NIM : 123150006
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Mifta Diah Novitasari
 NIM : 123150073
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Novita Indriani
 NIM : 123150115
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Lolita Tambunan
 NIM : 123150003
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Fitriana Kusumastuti
 NIM : 123150070
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Linda Khoirul Inayah
 NIM : 123150122
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Tita Rindu Astuti
 NIM : 123150129
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Aisyah Nur Atiko
 NIM : 123150111
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	

FORM PENGUJIAN TEST SISTEM SCM

NAMA : Anna Farida
 NIM : 123150082
 JURUSAN : Informatika

1. Pengujian fungsional sistem
 Berikan tanda (✓) pada kolom yang sesuai.

No	Item Uji	Penilaian	
		Berhasil	Gagal
Hak Akses Distributor			
1	Menampilkan halaman <i>home</i>	✓	
2	Menampilkan halaman <i>login</i> distributor	✓	
3	Proses <i>login</i> distributor admin	✓	
4	Proses <i>login</i> distributor karyawan	✓	
5	Menampilkan halaman <i>dashoard</i>	✓	
6	Menampilkan halaman tabel produk <i>list</i>	✓	
7	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>list</i>	✓	
8	Menampilkan halaman tambah kuantitas produk	✓	
9	Proses tambah kuantitas produk	✓	
10	Proses tambah produk baru	✓	
11	Proses <i>edit</i> produk	✓	
12	Proses hapus produk	✓	
13	Menampilkan halaman tabel karyawan	✓	
14	Menampilkan halaman detail data karyawan	✓	
15	Proses tambah, <i>edit</i> , hapus data karyawan	✓	
16	Menampilkan halaman data <i>retailer</i>	✓	
17	Proses tambah, <i>edit</i> , hapus data <i>retailer</i>	✓	
18	Menampilkan halaman detail data <i>retailer</i>	✓	
19	Menampilkan halaman edit produk <i>retailer</i>	✓	
20	Menjalankan fungsi <i>show, search, setting</i> halaman tabel produk <i>retailer</i>	✓	
21	Proses tambah, <i>edit</i> , hapus data tabel produk <i>retailer</i>	✓	
22	Menampilkan halaman rekomendasi	✓	
23	Proses tambah, <i>edit</i> , hapus data rekomendasi	✓	
24	Menampilkan detail perhitungan rekomendasi	✓	
25	Menampilkan halaman produk <i>retailer all</i>	✓	
26	Menampilkan halaman pengiriman	✓	

27	Menjalankan proses selesai, dibatalkan dan cetak pdf	✓	
28	Menampilkan halaman riwayat restok	✓	
29	Menampilkan halaman data pelaporan	✓	
30	Menampilkan halaman pengujian MAPE	✓	
31	Proses <i>logout</i>	✓	
Hak Akses Retailer			
32	Menampilkan halaman <i>login</i> distributor	✓	
33	Proses <i>login retailer</i>	✓	
34	Menampilkan halaman <i>dashoard</i>	✓	
35	Menampilkan halaman tabel produk	✓	
36	Menampilkan halaman transaksi	✓	
37	Menjalankan proses transaksi	✓	
38	Menampilkan halaman riwayat transaksi	✓	
39	Menampilkan halaman pelaporan	✓	
40	Menjalankan fungsi pelaporan	✓	
41	Menampilkan halaman riwayat pelaporan	✓	
42	Menampilkan halaman <i>reports</i>	✓	
43	Proses <i>logout</i>	✓	