



**SISTEM DISTRIBUSI SARI ROTI UNTUK OPERASIONAL MITRA
KERJA PT. NIPPON INDOSARI CORPINDO, TBK. BERBASIS WEB**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

**NAMA : FEBRI WINANDO PURBA
NPM : 1624370734
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

ABSTRAK

FEBRI WINANDO PURBA

Sistem Distribusi Sari Roti Untuk Operasional Mitra Kerja PT. Nippon
Indosari Corpindo, Tbk. Berbasis Web
2019

Pendistribusian Sari Roti membutuhkan pengambilan keputusan yang cepat dan tepat, dikarenakan umur roti yang sangat singkat, yaitu lima sampai enam hari. Cara kerja yang cepat adalah hal yang penting untuk mengumpulkan informasi agar dapat mendukung dalam pengambilan keputusan. Dengan dirancangnya sistem distribusi yang dapat digunakan ketika dilapangan oleh salesman, akan memberikan dukungan pengambilan keputusan, dengan kerja yang cepat dan data yang diperoleh adalah data yang *real time*, tidak perlu lagi menunggu salesman, kembali dari lapangan untuk menginput data transaksi. Metode penelitian yang digunakan untuk melakukan penelitian adalah Metode Penelitian Tindakan (*Action Research*). Dengan metode tersebut mitra dan pengembang dapat mendiskusikan dan merencanakan pengembangan yang tepat untuk perancangan sistem. Sistem dirancang dapat mendukung *user* atau salesman dapat menentukan pemesanan dengan memberikan data dan *suggest estimation*. Dengan menggunakan Metode *Simple Additive Weighing* (SAW) atribut keuntungan, memberikan saran secara kuantitas berdasarkan V_i (alternatif) yang memiliki nilai tertinggi. Data yang diperlukan adalah berdasarkan tanggal *droping* yang ditentukan oleh salesman. Sistem dengan tingkat *Decision Support System* (DSS) membantu salesman lebih cepat mengambil keputusan, dan dapat menghindari kerugian yang cukup besar.

Kata Kunci: *Action Research*, *Decision Support System*, Distribusi, *Droping*, Keputusan, *Real Time*, Roti, Salesman, *Simpel Additive Weighing*, Sistem.

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	viii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Pengumpulan Data	4
1.6 Tahap Perancangan	5
BAB II LANDASAN TEORI	
2.1 Distribusi	12
2.2 Konsep Dasar Sistem	14
2.2.1 Karakteristik Sistem	15
2.2.2 Klasifikasi Sistem	16
2.3 Sistem Informasi	17
2.4 Website	20
2.5 Hardware	20
2.6 Software	21
2.7 Database	21
2.8 Database Management System (DBMS)	24
2.8.1 Manfaat DBMS	24
2.9 Programming Language (Bahasa Pemrograman)	26
2.10 Flowchart (Diagram Alir)	28

2.11 Metode Simple Addictive Weighting (SAW)	29
--	----

BAB III ANALISIS MASALAH DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian	32
3.2 Analisis Permasalahan	32
3.3 Perancangan Sistem	33
3.3.1 Diagram Konteks Distribusi Mitra Kerja PT. NIC, Tbk.....	34
3.3.2 DFD Level Nol.....	35
3.3.3 Perancangan Database.....	37
3.3.4 Entity Relationship.....	43
3.3.5 Flowchart Sistem Distribusi Mitra Kerja PT. NIC, Tbk	45
3.4 Perancangan User Interface Sistem Distribusi	53
3.5 Perancangan Algoritma Metode SAW Dalam Sistem Distribusi	65

BAB IV IMPLEMENTASI PROGRAM

4.1 Implementasi Program	72
4.2 Spesifikasi Perangkat	72
4.2.1 Spesifikasi Hardware	72
4.2.2 Software	74
4.3 Tampilan Program.....	74
4.3.1 Tampilan Desktop	75
4.3.2 Tampilan Mobile.....	95

BAB V

5.1 Kesimpulan	103
5.2 Saran.....	103

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 1.6 Metode Waterfall.....	6
Gambar 2.1 Alur Distribusi.....	13
Gambar 2.2 Piramid Jenis Sistem	17
Gambar 2.3 Struktur Software Website	20
Gambar 2.6 Hirarki Data.....	22
Gambar 2.7 Perancangan DBMS	23
Gambar 2.8 Element Entiti Relationship	23
Gambar 2.9 Sekian	24
Gambar 2.10 Simbol Flowchart dan Fungsinya.....	29
Gambar 3.1 Diagram Konteks Distribusi Mitra Kerja PT. NIC, Tbk	34
Gambar 3.2 DFD Level Nol Sistem Distribusi Mitra Kerja PT. NIC, Tbk	36
Gambar 3.3 ER Tabel Database Sistem Distribusi Mitra Kerja PT. NIC, Tbk	44
Gambar 3.4 Flowchart Login	46
Gambar 3.5 Flowchart Estimasi SO.....	47
Gambar 3.6 Flowchart PO.....	48
Gambar 3.7 Flowchart Receiving	49
Gambar 3.8 Flowchart Alokasi Stock	50
Gambar 3.9 Flowchart Transaksi Salesman.....	52
Gambar 3.10 User Interface Login User	54
Gambar 3.11 User Interface Master Data List Outlet	55
Gambar 3.12 User Interface Master Data Item	56
Gambar 3.14 User Interface Account Receivable.....	57
Gambar 3.15 User Interface Check In Outlet.....	58
Gambar 3.16 User Interface Transaksi Outlet.....	58

Gambar 4.19 Halaman Insert Suggest pada App Android	101
Gambar 4.20 Halaman Insert Kuantitas Droping pada App Android	102
Gambar 4.21 Halaman Account Receivable pada App Android	104
Gambar 4.22 Halaman Support Activity pada App Android	105
Gambar 4.23 Halaman NOO pada App Android	107
Gambar 4.24 Halaman Stock Salesman	108

DAFTAR TABEL

	Halaman
Tabel 3.2.1 Tabel Daftar Distributor.....	33
Tabel 3.3.2 Tabel Data Master Karyawan Distributor.....	34
Tabel 3.2.3 Tabel Data Master Item.....	34
Tabel 3.2.4 Tabel Data Master Outlet.....	35
Tabel 3.2.5 Tabel Sales <i>Order</i>	36
Tabel 3.2.6 Tabel Estimasi <i>Order</i>	36
Tabel 3.2.7 Tabel <i>Receiving</i>	37
Tabel 2.3.8 Tabel Gudang Salesman.....	38
Tabel 3.2.9 Tabel Penjualan.....	38
Tabel 3.4 Daftar Kriteria (C_i).....	62
Tabel 3.4.1 Tabel <i>Rating History</i> Sales Berdasarkan Presenase Penjualan	63
Tabel 3.4.2 Rating F1 Berdasarkan Roti Habis Sebelum JWK	64
Tabel 3.4.4 Rating F2 Berdasarkan Roti Habis Setelah JWK dan Presentase Sales.....	64
Tabel 3.4.5 Rating F3 Berdasarkan Roti Habis Sebelum JWK	65
Tabel 3.4.6 Rating F3 Berdasarkan Roti tidak Habis saat JWK Dan Presentase Sales.....	65
Tabel 3.4.7 Tabel Rating Lokasi Outlet	66
Tabel 3.4.8 Rating Sublokasi Outlet Terhadap Waktu	68
Tabel 3.4.9 Tabel Rating Berdasarkan Jenis Outlet.....	69

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT. Nippon Indosari Corpindo, Tbk adalah perusahaan yang bergerak di bidang produksi dan distribusi makanan. Makanan yang diproduksi dan didistribusikan di pasar adalah Roti yang diberi label Sari Roti. Sari Roti didistribusikan membutuhkan cara kerja yang cepat dikarenakan masa waktu Sari Roti hanya lima dan enam hari dan terhitung keluar dari produksi sampai kepada konsumen. Oleh sebab itu, strategi bisnis juga dibutuhkan dengan cepat agar estimasi dan eksekusi di lapangan dapat memberikan hasil yang baik.

Distribusi PT. Nippon Indosari Corpindo, Tbk, ditangani oleh mitra distribusi yaitu Distributor dan agen. (Suryanto, 2016:4), menyatakan dengan adanya bantuan mitra, konsumen akan dengan mudah memperoleh produk atau jasa yang mereka butuhkan. Proses pemindahan suatu produk atau jasa inilah yang harus dikelola sesuai dengan visi dan misi perusahaan dalam suatu kondisi lingkungan tertentu dan sesuai dengan kebutuhan dari konsumen yang dinamakan manajemen distribusi (Walter, 1977). Jadi, dalam mengelola proses pemindahan ini diperlukan suatu pendekatan pada pengambilan keputusan (*decision oriented approach*) yang dimulai dari perencanaan (*planning*), pengorganisasian (*organization*), pengoperasian (*actualization*) dan pengendalian (*controlling*). Dalam pendistribusian Sariroti harus dilakukan dengan cepat dan tepat, dibutuhkan sistem

yang mampu mendukung *decision maker* dalam mengambil keputusan. Oleh sebab itu, PT. Nippon Indosari Corpindo, Tbk membutuhkan sistem yang *real time*, sehingga informasi yang diberikan oleh sistem adalah informasi yang terjadi pada saat itu juga di lapangan, sistem tersebut membuat *decition maker* memberikan keputusan yang mendekati kondisi. Sistem keputusan yang aka di terapkan akan menggunakan Metode Saw sebagai *suggest* kepada salesman yang bekerja dilapangan.

Kegiatan operasional distribusi, semakin lama periode pembayaran akan mematkan distribusi (Suryanto, 2016:15). Umumnya kisaran lama periode pembayaran 1 – 14 hari (FCMG); 21 – 90 hari (material bangunan, bahan kimia, alat-alat tulis kantor, peralatan listrik); dan lebih 90 hari (otomotif, properti dll.). Sistem yang di rancang juga harus bisa mendukung setiap operasional distributor sebagai mitra kerja PT. Nippon Indosari Corpindo, Tbk, baik operasional keuangan (*Account Receivable/AR* dan Kasir) dan gudang barang. Keterkaitan operasional tersebut sangat mempengaruhi kinerja perusahaan mitra kerja, oleh sebab itu sistem harus mampu memaksimalkan kerja setiap operasional perusahaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka dirumuskan beberapa rumusan masalah yang akan dibahas.

- a. Bagaimana merancang sebuah sistem yang dapat mendukung operasional distributor sebagai mitra kerja PT. Nippon Indosari Corpindo, Tbk?
- b. Bagaimana sistem dapat memberikan informasi *real time*?

- c. Bagaimana sistem dengan penerapan Metode SAW dapat mendukung setiap keputusan dalam melakukan estimasi order roti per *salesman*?

1.3 Tujuan dan Manfaat Penelitian

Sistem dirancang harus dilakukan dengan tahap penelitian, sehingga dapat mencapai tujuan dan manfaat dalam perancangan sistem distribusi tersebut.

Penelitian pada sistem distribusi ini bersifat *development* atau pengembangan pada sistem yang telah ada saat ini. Tujuan dari penelitiannya adalah sebagai berikut:

1. Sistem distribusi yang dirancang menjadi *tools* yang efektif untuk operasional distributor di tengah keterbatasan karyawan
2. Memberikan informasi yang *real time* kepada *user*
3. Menjadi sumber informasi terpercaya dalam pengambilan keputusan

Manfaat dari penelitian ini adalah bersifat praktis karena condong kepada pemecahan rumusan masalah di atas yang bersifat praktis. Berikut beberapa manfaat penelitian:

1. Meningkatkan kinerja operasional distributor
2. Informasi yang diterima cepat dan akurat
3. Mendukung *salesman* dalam melakukan estimasi order roti

1.4 Batasan Masalah

Penelitian dilakukan dengan tujuan di atas, mendukung operasional distributor dan memberikan informasi yang *real time* kepada user dan menjadi informasi terpercaya untuk pengambilan keputusan yaitu dengan memberikan

rekomendasi estimasi berdasarkan keputusan yang dilakukan dengan metode *Simple Addictive Weithing* (SAW). Sistem tidak akan menjadi pengambil keputusan, tetapi sebagai pendukung user untuk mengambil keputusan. Peran sebagai *Decision Maker* hanya akan diberikan kepada user, sehingga sistem yang dirancang tetap dalam rangka perancangan sistem *Decision Support System* (DSS).

Perancangan sistem tidak terhubung dengan sistem internal untuk operasional perusahaan PT. NIC, Tbk, yaitu *System Application and Processing* (SAP), dikarenakan sistem hanya dapat dikembangkan oleh *developer* yang bertugas diinternal perusahaan dan menyimpan data penting perusahaan. Oleh sebab itu, sistem dirancang sebatas penggunaan internal mitra kerja PT. NIC, Tbk.

1.5 Pengumpulan Data

Pengumpulan data dilakukan adalah untuk keperluan penelitian. Data dikumpulkan harus cukup valid untuk digunakan. Validasi data dapat ditingkatkan jika alat pengukur serta kualitas dari pengambil data sendiri cukup valid.

Metode pengumpulan data yang digunakan sebagai prosedur penelitian adalah pengumpulan data dengan wawancara dan pengumpulan data dengan observasi langsung. Wawancara adalah proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden dengan menggunakan alat yang dinamakan *interview guide* (panduan wawancara).

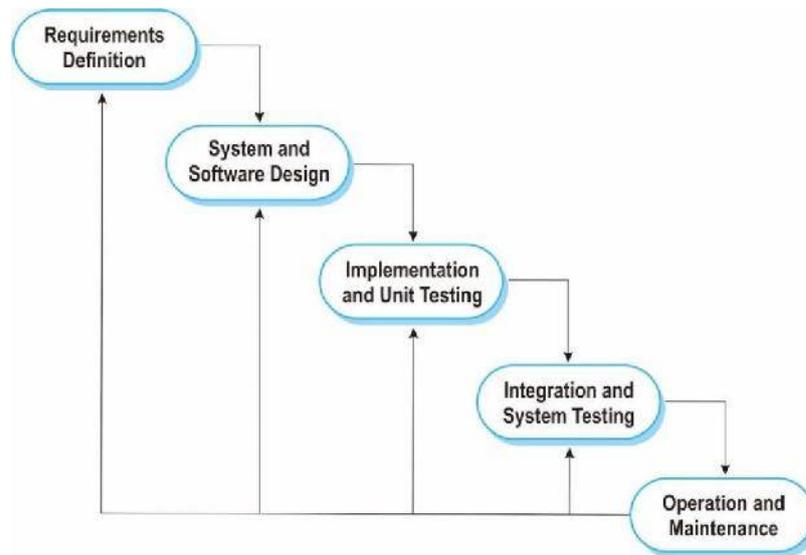
Metode pengumpulan data kedua yang digunakan pada penelitian ini adalah observasi langsung. Observasi langsung merupakan proses pengumpulan data yang

dilakukan dengan cara pengamatan langsung tanpa penggunaan menggunakan alat standar untuk mengumpulkan data. Pengamatan observasi langsung dapat digolongkan sebagai teknik pengumpulan data, jika pengamatan mempunyai kriteria sebagai berikut (Nazir, 2011: 175):

- a. Pengamatan digunakan untuk penelitian dan telah direncanakan secara sistematis
- b. Pengamatan harus berkaitan dengan tujuan penelitian yang telah direncanakan
- c. Pengamatan tersebut dicatat secara sistematis dan dihubungkan dengan proposisi umum dan bukan dipaparkan sebagai suatu set yang menarik perhatian saja
- d. Pengamatan dapat dicek dan dikontrol atas validasi dan reliabilitasnya.

1.6 Tahap Perancangan

Metode Perancangan atau pengembangan sistem yang diterapkan pada penelitian ini adalah dengan pengembangan Metode *Waterfall*. Metode *Waterfall* adalah metode yang menggunakan aktivitas proses yang mendasar dari perencanaan, pengembangan, validasi dan perkembangan dan mempunyai tahap proses yang berbeda dalam perencanaan kebutuhan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya. Metode *Waterfall* yang digunakan adalah metode versi Sommerville (2011) yaitu versi terbaru dari Metode *Waterfall* (Somerville, 2011:29).



Gambar 1.6 Metode *Waterfall*
Sumber (Somerville, 2011: 31)

Dalam Somerville, 2011: 31, Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut:

a. *Requirements Analysis and Definition*

Tahapan ini adalah tahapan untuk mengumpulkan informasi pelayanan dengan metode pengumpulan data yang telah dibahas pada subbab sebelumnya, batasan dan tujuan sistem dengan melakukan konsultasi terhadap pengguna sistem. Mereka menjelaskan dan menyajikan secara detail permintaan sistem.

b. *System and Software Design*

Tahapan ini merupakan proses pengalokasian persyaratan terhadap sistem *hardware* maupun *software* melalui merancang arsitektur sistem secara menyeluruh. *Software* dirancang melibatkan identifikasi dan deskripsi mendasar *software* sistem abstrak dan relasinya.

c. *Implementation and Unit Testing*

Pada tahapan ini adalah proses perancangan *software* ke dalam bentuk program-program atau unit-unit program. Setiap unit diuji dengan memverifikasi apakah setiap unit program sudah sesuai dengan fungsi masing-masing.

d. *Integration and System Testing*

Tahapan ini merupakan tahapan penyatuan unit-unit program atau pengintegrasian program-program dan pengujian terhadap sistem secara menyeluruh apakah kebutuhan perangkat lunak sudah terpenuhi. Setelah diuji, sistem *software* dikirim kepada *costummer*.

e. *Operation and Maintenance*

Biasanya (walaupun tidak selalu) tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara langsung. *Maintenance* dilakukan secara langsung dalam memperbaiki kesalahan yang tidak ditemukan dalam tahapan-tahapan sebelumnya, peningkatan penerapan dari unit-unit sistem dan peningkatan pelayanan kebutuhan sistem yang belum dimiliki.

BAB II

LANDASAN TEORI

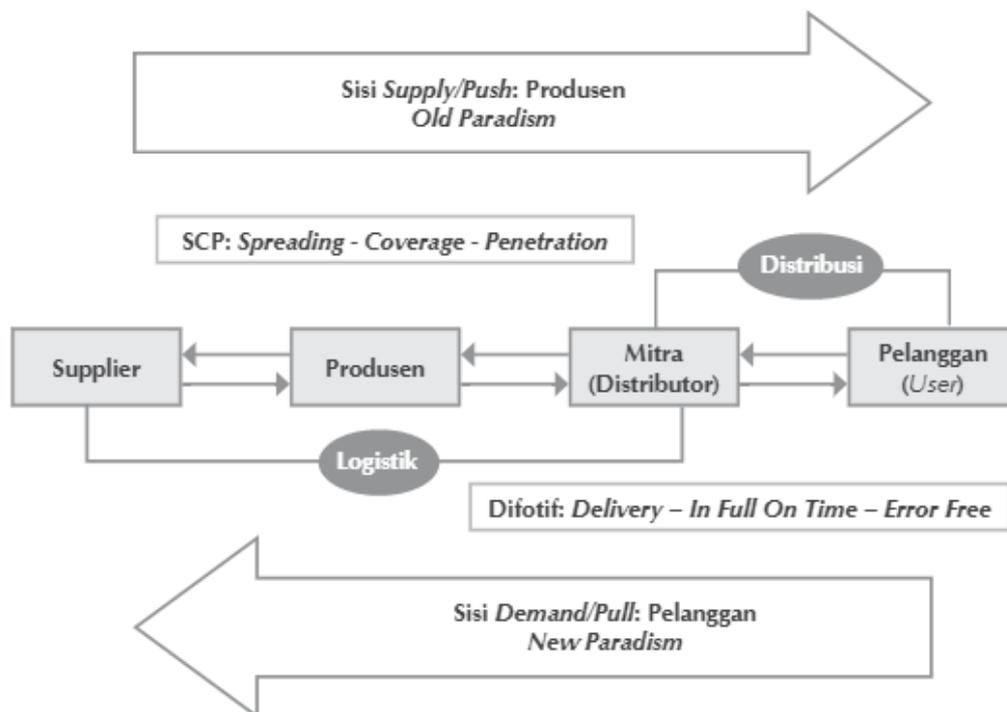
2.1 Distribusi

Dalam kegiatan manufaktur, distribusi memegang peranan penting dalam menjamin produk dipasarkan supaya produk tersebut tersedia secara merata di setiap wilayah (Suryato, 2016:1). Dapat dibayangkan apabila suatu produk tidak terdistribusi secara merata akan menyulitkan masyarakat dalam mendapatkan produk itu. Kebanyakan perusahaan manufaktur yang sedang mengalami pertumbuhan memanfaatkan agen dan distributor untuk pendistribusian produk mereka. Cara ini lebih efektif dan efisien dan dapat mengurangi biaya distribusi yang tinggi jika mendirikan cabang sendiri.

Kegiatan saluran distribusi secara tidak langsung sebenarnya sudah dijumpai dalam kehidupan sehari-hari. Produsen tidak akan mampu menyalurkan langsung kepada konsumen akhir (pemakai). Ada beberapa faktor yang membatasi penyaluran secara langsung dari produsen ke konsumen, yakni Geographical Gap (perbedaan jarak geografi), Time Gap (perbedaan jarak waktu), Quantity Gap (perbedaan jumlah produksi), Communication and Information Gap (perbedaan komunikasi dan informasi), maka diperlukannya mitra distribusi yang bekerja sama dengan produsen untuk mendistribusikan produknya.

Dijelaskan Suryanot (2016: 5) dalam manajemen distribusi ada dua sistem yang beredar:

- a. Paradigma Lama (*Old Paradigm*), menjelaskan penentuan target penjualan untuk setiap jalur distribusi lebih berorientasi pada produsen. Dalam paradigma ini, produsen (*principal*) memiliki otoritas dalam menjalankan dan menyusun permintaan mitra beserta timnya dalam pendistribusian. Kunci keberhasilan distribusi adalah SCP (*Spreading, Coverage, dan Penetration*)
- b. Paradigma Baru (*New Paradigm*) penentuan permintaan dan penjualan produk atau jasa berasal dari kebutuhan pelanggan (*user*). Jadi, pihak produsen hanya sebagai sarana untuk pemenuhan sejumlah produk atau jasa sesuai dengan permintaan pelanggan. Keberhasilan logistik adalah Diftotef (*Delivery in Full on Time Error Free*)



Gambar 2.1 Alur Distribusi
Sumber (Suryanto, 2016: 6)

2.2 Konsep Dasar Sistem

Dalam Ladjamudin, 2005, dalam mendefinisikan sistem terdapat dua kelompok pendekatan sistem, yaitu sistem yang lebih menekankan prosedur dan elemennya. Prosedur didefinisikan sebagai suatu urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakan, kapan dikerjakan dan bagaimana mengerjakannya (Gerald. J., 1991). Penganut pendekatan elemen adalah Davis (1985) yang mendefinisikan elemen sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai sasaran atau maksud. Sedangkan Lucas (1989) mendefinisikan sistem sebagai suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain dan terpadu. Sebuah sistem memiliki tujuan maupun sasaran. Berdasarkan referensi definisi dari Sistem yang dikemukakan di atas, sistem merupakan kumpulan dari komponen yang saling berintegrasi untuk mencapai satu tujuan. Sistem dikembangkan berdasarkan kebutuhan, tergantung pada kebutuhan *user*.

Konsep sistem terdiri dari tiga dasar implikasi, yaitu:

1. Sebuah sistem dirancang adalah untuk mencapai tujuan yang ditentukan
2. Hubungan timbal - balik dan keterkaitan harus ada pada komponen-komponen
3. Tujuan organisasi bagaimanapun harus memiliki prioritas yang lebih dari tujuan subsistem tersebut.

2.2.1 Karakteristik Sistem

Sistem memiliki karakteristik atau ciri-ciri, yang menggambarkan bagaimana sifat dari sebuah sistem, berikut karakteristik sistem:

1. Organisasi

Organisasi mengacu pada struktur dan golongan. Ini merupakan aturan atau ketetapan terhadap komponen-komponen untuk mencapai tujuan.

2. Interaksi

Hal ini mengacu kepada kewajiban komponen yang mana masing-masing komponen saling memberikan pengaruh terhadap sistem

3. Keterkaitan

Setiap bagian dari organisasi dalam sistem saling terkait satu dengan yang lain. Komponen-komponen tersebut saling berkoordinasi dan terhubung bersama untuk sebuah tujuan. Setiap sup sistem terkait terhadap *output* dari sup sistem lainnya.

4. Integrasi

Integrasi merupakan suatu kesatuan sistem untuk menjadi kesatuan yang utuh. Inilah yang menjadi pusat dari sebuah sistem bagaimana sistem saling mengikat.

5. Satu Tujuan

Organisasi dijalankan secara terintegrasi, meskipun terdiri dari golongan yang memiliki fungsi berbeda tetapi berkerja untuk satu tujuan yang sama.

6. Sebuah sistem harus memiliki tujuan yang sama. Tujuan dapat berupa sebuah pernyataan atau riil. Oleh sebab itu, yang menjadi poin utama dalam sistem yaitu memiliki satu tujuan.

2.2.2 Klasifikasi Sistem

Sistem memiliki tujuan yang berbeda untuk setiap kondisi yang terjadi di dalam sistem tersebut (Ladjamudin, 2005: 6). Oleh karena itu, sistem di klasifikasikan ke dalam beberapa sudut pandang, yaitu:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak yaitu adalah sistem yang tidak tampak memiliki fisik, sistem ini biasanya berupa buah pemikiran atau ide-ide. Sedangkan sistem fisik adalah sistem yang tampak fisiknya, baik dalam bentuk perangkat keras maupun perangkat lunak.

2. Sistem Alamiah dan Sistem Buatan

Sistem alamiah adalah sistem yang tercipta oleh alam itu sendiri, tidak ada sedikit pun campur tangan manusia. Sistem buatan adalah sistem yang dirancang dan dibuat khusus oleh manusia.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik adalah sistem yang mempunyai tujuan tertentu maka hasil untuk sistem ini sudah terlihat jelas hasilnya dan biasanya sistem ini berfungsi untuk tujuan jangka panjang. Sedangkan sistem probabilistik adalah sistem yang tentu sehingga hasil yang keluar tidak dapat dipastikan, karena sistem ini dipengaruhi probabilitas.

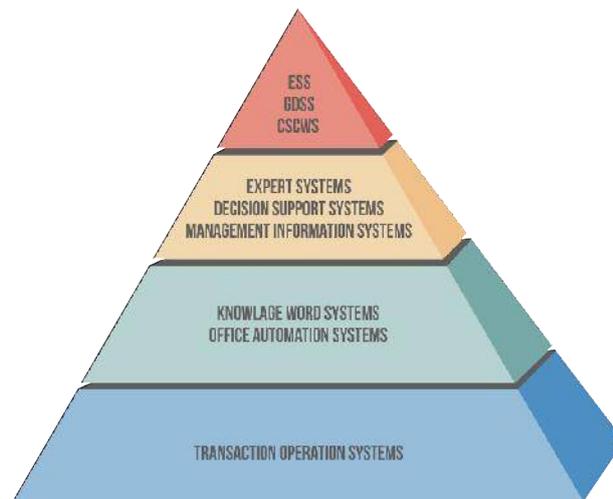
4. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup adalah sistem yang tidak terbuka terhadap lingkungan luar, sistem tertutup tidak sepenuhnya tertutup, relatif. Sistem terbuka adalah sistem yang terbuka, sistem ini juga tidak sepenuhnya terbuka, sistem ini hanya terbuka atau relatif terbuka terhadap sesuatu yang memberikan hasil yang baik (Ladjamudin, 2005: 7).

2.3 Sistem Informasi

Informasi sangat penting sebagai wawasan baik individual maupun organisasi. Informasi adalah sekumpulan data yang diolah atau diproses menjadi sebuah konteks yang berguna dalam pengambilan keputusan. Informasi tidak didapatkan secara langsung oleh penerima informasi, tetapi melalui data-data yang dikumpulkan melalui setiap kejadian dan disusun sedemikian rupa, sehingga data-data tersebut memberikan pengetahuan atau wawasan kepada penerima informasi. Sistem Informasi adalah kumpulan unit-unit program yang tereintegrasi dengan tujuan yaitu memberikan informasi.

Sistem informasi dikembangkan dengan tujuan-tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis (Kendal, 2002: 2). Sistem informasi berdasarkan fungsi dibedakan menjadi beberapa jenis system, untuk perbedaannya dapat dilihat sebagai berikut:



Gambar 2.2 Piramid Jenis Sistem

Sumber : (Kendal, 2002: 4)

1. *Transaction Processing System (TPS)*

TPS adalah sistem informasi terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin.

2. *Office Automation System (OAS)* dan *Knowledge Work System (KWS)*

OAS yang mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru, melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasikannya dengan cara-cara tertentu, sebelum membaginya atau menyebarkannya secara keseluruhan. KWS mendukung para pekerja profesional seperti ilmuwan, insinyur dan doktor dengan membantu mereka menciptakan pengetahuan baru dan memungkinkan mereka memberi kontribusi ke organisasi atau masyarakat.

3. *System Information Manajement (SIM)*

SIM tidak menggantikan TPS melainkan semua sistem SIM mencakup pengolahan informasi. SIM adalah sistem informasi yang sudah terkomputerisasi yang bekerja karena adanya interaksi antara manusia dan

komputer. Dengan bantuan manusia, perangkat lunak (program komputer) dan perangkat keras (komputer, printer dan lain-lain) agar berfungsi dengan baik, System informasi manajemen mendukung spektrum tugas-tugas organisasional yang lebih luas dari TPS, termasuk analisis keputusan dan pembuatan keputusan.

4. *Decision Support System (DSS)*

Kelas sistem informasi terkomputerisasi pada level yang lebih tinggi adalah DSS. DSS hampir sama dengan SIM tradisional karena keduanya sama-sama bergantung pada basis data sebagai sumber data. DSS berangkat dari SIM tradisional karena menekankan pada fungsi keputusan di seluruh tahapannya, meskipun keputusan aktual masih tetap wewenang eksklusif pembuat keputusan, sehingga system hanya berfungsi sebagai pemberi saran dalam pilihan keputusan yang dilakukan. Sehingga, keputusan penuh terdapat pada *Decision Maker*.

5. *Group Decision Support System (GDSS)* dan *Computer Supported Support System (CSWS)*

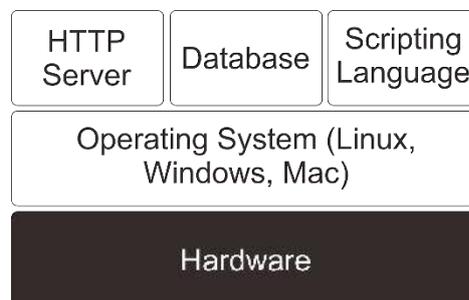
Bila kelompok kerja bersama-sama untuk membuat keputusan semi-terstruktur dan tak-terstruktur, maka GDSS membuat suatu solusi. GDSS digunakan di ruangan khusus yang dilengkapi dengan konfigurasi yang berbeda-beda, memungkinkan anggota kelompok berinteraksi dengan pendukung elektronik. CSWS sama seperti GDSS, keduanya memiliki fungsi yang sama yang mencakup perangkat lunak yang disebut dengan *groupware* untuk kolaborasi tim melalui komputer yang terhubung dengan jaringan.

6. *Executive Support System (ESS)*

ESS membantu para eksekutif mengatur interaksi mereka dengan lingkungan eksternal dengan menyediakan grafik-grafik dan pendukung komunikasi di tempat-tempat yang bisa di akses seperti kantor. Meskipun ESS tergantung pada informasi yang dihasilkan oleh TPS dan SIM, ESS membantu pengguna mengatasi problem keputusan yang tidak terstruktur, yang bukan aplikasi khusus, dengan menciptakan lingkungan yang kondusif untuk memikirkan problem -problem strategis.

2.4 *Website*

Website adalah halaman informasi yang dapat diakses di mana saja dan kapan saja dan hanya dapat diakses jika terhubung dengan internet. Hardware dan Software adalah sebagai dasar dalam struktur *website*, tanpa komponen-komponen tersebut tidak akan membangun sebuah *website*.



Gambar 2.3 Struktur *Software Website*

Sumber : (Mendez, 2014: 14)

2.5 *Hardware*

Dalam Hutchinson, Hardware terbagi dalam beberapa jenis (Sawyer, 1996:5), yaitu:

1. *Input*

Input hardware adalah alat untuk mengumpulkan atau mengoleksi data ke dalam proses komputer.

2. *Processing*

Processing hardware adalah untuk menerima dan melaksanakan instruksi dari *software* atau program yang disediakan dalam komputer.

3. *Storage*

Storage Hardware adalah untuk menyediakan ruang untuk *software* dan juga data, dapat disimpan dalam jangka waktu yang relatif lama.

4. *Output*

Output Hardware adalah untuk menampilkan informasi yang diolah oleh *user* melalui program sesuai dengan kebutuhan.

5. *Communications*

Communications Hardware adalah fasilitator koneksi antar komputer ataupun antar grup komputer agar saling terhubung.

2.6 *Software*

Ada empat elemen yang khas dalam membangun *website*. Empat elemen tersebut adalah *Operating System*, *Web Server*, *Database* dan *Script Language* (Mandez, 2014: 14). *Operating System (OS)* merupakan perantara yang menghubungkan *hardware* dengan aplikasi sehingga komputer dapat difungsikan. OS juga memberikan sarana penyimpanan aplikasi dan sebagai alat komunikasi antara *hardware* dan *software*. *Web Server* adalah untuk mentransfer berkas atau permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Script

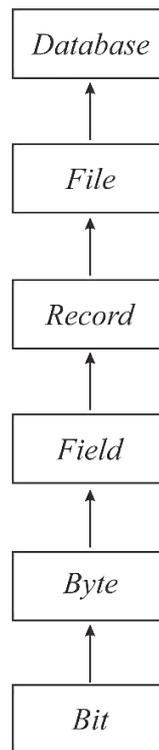
Language (Bahasa pemrograman) adalah skrip khusus yang digunakan untuk membangun aplikasi.

2.7 Database

Database merupakan kumpulan terorganisasi sekelompok data yang ditampilkan dengan beberapa istilah yang saling berhubungan. Kumpulan yang sempurna atau sembarang himpunan bagian yang logis dilihat sebagai satuan yang tunggal (Mahyudin, 1994: 1).

Di dalam *database* terdapat elemen yang membentuk hierarki data yaitu *bit* (*Binary Digit*) yang merupakan satuan terkecil dalam komputasi digital, dan satuan dari bit membentuk 1 *byte* yang terdiri dari 8 bit untuk setiap satuannya dan membentuk satu karakter atau satu angka atau karakter khusus. Butir data atau data (*data item/data*) yang adalah satuan terkecil dari data yang diberi nama, dapat terdiri dari sekelompok angka atau huruf, dapat juga disebut sebagai bidang data (*data field*) atau *field* saja. Kumpulan dari relasi data disebut dengan *record*. Kemudian, kumpulan relasi *record* disebut sebagai berkas (*file*). Oleh sebab itu, kumpulan dari *file* tersebut yang berisi informasi disebut dengan *database*.

Tabel database hanya bersikan data *field* saja, tetapi merupakan dasar dari sekumpulan informasi yang akan diolah. Informasi dibentuk dimulai dari pengumpulan data *field*. Oleh karena itu data *field* yang disimpan di database haruslah benar dan *valid*, sehingga informasi yang dihasilkanpun akan benar.



Gambar 2.6 Hirarki Data

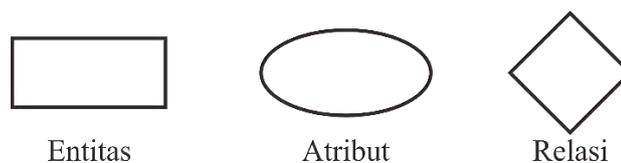
Untuk memulai perancangan *Database Management System* (DBMS), harus dimulai dengan *Entity Relationship* (ER), model ini bersifat grafis menggunakan kotak dan penunjuk arah yang berguna untuk mendefinisikan esensi dan hubungan antar entity. Perancangan DBMS dimulai ide tentang informasi sistem yang akan dirancang dikembangkan di modelkan ke dalam ER.



Gambar 2.7 Perancangan DBMS

Sumber : (Mahyudin, 1994: 1)

Dalam model ER, struktur data dipersentasekan secara grafis sebagai diagram *entity* ER, menggunakan tiga elemen dasar, yaitu:



Gambar 2.8 Element *Entity Relationship*

1. Entitas

Entitas adalah sebuah objek abstrak yang berbeda dan unik, memiliki bentuk yang serupa dalam kumpulan entitas.

2. Atribut

Atribut adalah karakteristik atau ciri yang membedakan antara entitas satu dengan entitas lainnya.

3. Relasi

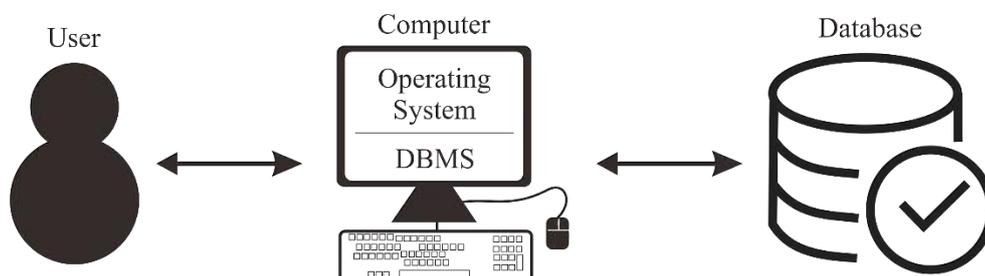
Relasi adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

2.8 *Database Management System (DBMS)*

Informasi merupakan alat pengambilan keputusan dalam dalam sebuah organisasi modern. Melalui penggunaan teknologi digital yang terus meningkat dan memasuki era produksi intensif informasi, memasang dan menggunakan sumber daya informasi adalah strategi yang tepat meningkatkan daya saing industri.

Untuk pengolahan data yang besar diperlukan sebuah perangkat lunak yang dapat memanajemen data. Perangkat lunak tersebut adalah *Database Management System (DBMS)*. Dengan bertambahnya data yang dari waktu ke

waktu akan sangat mudah mengolah data dengan volume yang besar menggunakan DBMS, oleh sebab itu DBMS akan sangat diperlukan.



Gambar 2.9 Pengolahan *Database Management System* (DBMS)

2.8.1 Manfaat DBMS

DBMS memiliki keunggulan yang dapat dimanfaatkan dalam perancangan sebuah sistem. Berikut beberapa keunggulan dalam menggunakan DBMS (Ramakrishnan, 2004:6):

1. Kemandirian data

Program aplikasi idealnya tidak diekspos pada detail representasi dan penyimpanan data. DBMS menyediakan satu pandangan abstrak tentang data yang menyembunyikan detail tersebut.

2. Akses data efisien

DBMS memanfaatkan berbagai teknik yang canggih untuk mengambil dan menyimpan data secara efisien. Fitur utama ini penting jika data disimpan pada alat penyimpanan eksternal.

3. Integritas dan keamanan data

Jika data selalu disimpan melalui DBMS, maka DBMS dapat menggunakan batasan integritas. DBMS juga dapat memanfaatkan kontrol akses yang menentukan data apa yang boleh dilihat oleh kelas pengguna yang berbeda

4. Administrasi data

Ketika beberapa pengguna berbagi data, pemusatan administrasi dapat memberikan perbaikan yang signifikan. Para profesional berpengalaman yang memahami sifat data yang dikelola dan memahami bagaimana kelompok pengguna yang berbeda menggunakan data tersebut, dapat menjadi tanggung jawab untuk mengatur representasi data untuk meminimalkan redundansi dan untuk memfintune penyimpanan data guna melakukan pengambilan data dengan efisien.

5. Akses Konkuren and Crash Recovery

DBMS menjadwalkan akses konkuren pada data dalam cara tertentu sehingga pengguna dapat memandang data sebagai data yang sedang diakses oleh hanya satu pengguna pada satu waktu. Dan juga, DBMS memproteksi pengguna dari efek kegagalan sistem.

6. Waktu pengembangan aplikasi berkurang

Tentu saja DBMS mendukung fungsi penting yang merupakan hal biasa bagi banyak aplikasi untuk mengakses data dalam DBMS. Hal ini, kaitannya dengan *interface* level tinggi pada data, membatasi pengembangan aplikasi yang cepat. Aplikasi DBMS kemungkinan besar menjadi lebih kuat dibandingkan dengan aplikasi yang berdiri sendiri karena banyak tugas penting yang ditangani DBMS.

2.9 *Programing Language (Bahasa Pemrograman)*

Dalam Hutchinson, Sawyer, 1996:220, mendefinisikan bahasa pemrograman adalah aturan yang diterapkan untuk memerintahkan komputer operasi yang harus dikerjakan. Untuk merancang aplikasi Distribusi Berbasis *Web* bahasa pemrograman yang digunakan adalah HTML, PHP, MySQL, CSS, Javascript dan Java yang digunakan untuk membuat aplikasi android sebagai aplikasi untuk mengakses alamat *website* pada platform android.

1. HTML (*Hyper Text Mark-up Language*)

HTML adalah singkatan dari *Hyper Text Mark-up Language*. HTML adalah dasar untuk semua hal-hal *web* dan merupakan keterampilan yang diperlukan untuk setiap pengembang *web*. Hampir semua situs web terdiri dari HTML apakah itu variasi HTML atau HTML versi lama.

2. PHP (*Hypertext Processor*)

Dalam Utomo, 2014:2, sebagai *embedded language*, PHP juga hampir sama dengan bahasa pemrograman lain, yang dapat dimasukkan ke dalam kode HTML seperti *Javascript*. Perbedaannya terletak pada tempat eksekusi. Jika *Javascript* dieksekusi di sisi klien atau *browser*, PHP di eksekusi di sisi *server*. Maka *database* dapat mengambil, mengolah, mengirimkan data dari *database server* ke halaman *browser*.

Dengan adanya pemrograman PHP, sebuah *website* akan menjadi dinamis karena isi halaman dapat diubah sesuai dengan konten dan data yang ada di dalam *database*. Konten juga dapat berasal dari *user* kemudi disimpan ke dalam *database* dan diakses kembali ke dalam *website*.

Keunggulan dari PHP adalah kemampuan mendukung berbagai macam database, MySQL, Oracle, PostgreSQL, DBase, Empress, File Pro, Informix, SQLite, ODBC, Inter Base, Front Base, Solid, Sybase dan Velocis.

3. Javascript

Javascript adalah bahasa pemrograman *website* yang berjalan sebagai klien dan dijalankan dalam *web browser*. Dengan menggunakan javascript fungsionalitas pada *website* menjadi dinamis. Javascript adalah bahasa pemrograman terstruktur sama seperti PHP, dengan menggunakan sintaks *class* akan mempermudah pemanggilan fungsi yang sama.

4. CSS (*Cascading Style Sheets*)

Menurut Wikipedia, 2018, CSS adalah bahasa pemrograman yang digunakan untuk menggambarkan persentase sebuah dokumen yang dimuat dalam HTML. CSS berfungsi mendefinisikan *style* pada *website*, menerapkan desain, layout dan variasi tampilan pada *device* dan ukuran layar yang berbeda. Sehingga, setiap halaman website lebih interaktif dan *user* dapat lebih memahami setiap fungsi *tools* yang ada pada *website* tersebut.

5. Java

Dalam developer.android.com, 2018, Aplikasi *Android* ditulis dalam bahasa pemrograman Java. *Android Software Development Kit (SDK) Tools* mengompilasi kode bersama data dan *file* sumber daya menjadi sebuah *Application Package File (APK)* yang adalah sebuah paket *Android*, yang berupa *file* arsip dengan akhiran .apk. Satu *file* APK berisi semua materi aplikasi *Android* dan merupakan *file* yang digunakan perangkat *Android* untuk

memasang aplikasi. Dikarenakan perancangan program menggunakan basis Bahasa Pemrograman Java, maka perlu mengkonfigurasi kembali *library* Java pada system operasi yang digunakan, sehingga emulator dapat di *running* pada system operasi tersebut.

2.10 Flowchart

Flowchart adalah diagram yang menggunakan standar ANSIsymbol yang menampilkan melalui langkah-langkah aktivitas proses dan logika keputusan untuk menyelesaikan sebuah masalah (Sawyer, 1996:241).

	Direction (Simbol Arus) Untuk menyatakan jalannya alur proses dan juga untuk menyatakan transmisi data dari satu lokasi ke lokasi lainnya.
	Decision (Logika) Untuk menyatakan suatu kondisi tertentu yang menyatakan suatu kemungkinan, ya atau tidak.
	Input/Output Untuk menyatakan proses input atau output tanpa tergantung pada jenis peralatannya input ataupun outputnya.
	Connector Untuk menyatakan sambungan dari suatu proses ke suatu proses lainnya dalam halaman/lembar yang sama.
	Terminal Untuk menyatakan awal ataupun akhir dari sebuah program.
	Processing (Pemrosesan) Untuk menyatakan tindakan (proses) yang dilakukan oleh komputer.
	Offline Connector Untuk menyatakan sambungan dari suatu proses ke suatu proses lainnya dalam halaman/lembar yang berbeda.

Gambar 2.10 Simbol Flowchart dan Fungsinya
Sumber : (Hutchinson, Sawyer, 1996:242)

Programmer sebelum membuat program terlebih dahulu membuat *flowchart*, karena dengan adanya *flowchart*, *Programme* dapat menggambarkan konsep dengan jelas bagaimana alur sistem ataupun program yang dibangun.

2.11 Metode Simple Additive Weighing (SAW)

Dalam Turban, dkk. (2015:137) Sistem Pendukung Keputusan (*Decision Support System/DSS*) sebagai, "sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan", Little (1970). Dia menyatakan untuk sukses, sistem tersebut haruslah sederhana, cepat dan mudah dikontrol, adaptif, lengkap dengan isu-isu penting dan mudah berkomunikasi. DSS dibangun untuk mendukung solusi terhadap suatu masalah atau untuk mengevaluasi suatu peluang. Salah satu metode algoritma yang sering digunakan adalah Metode *Simple Additive Weighing* (SAW).

Metode Simple Additive Weighting (SAW) disebut juga dengan metode penjumlahan terbobot. Konsep metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) (MacCrimmon, 1968).

Metode SAW membutuhkan proses normalisasi matriks keputusan ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi Multiple Attribute Decision Making (MADM). MADM sendiri adalah sebuah metode yang digunakan untuk mencari hasil yang lebih optimal dari beberapa alternatif yang memiliki kriteria tertentu.

$$r_{ij} = \frac{x_{ij}}{\text{Max } x_{ij}}$$

Jika j adalah atribut keuntungan (*benefit*)

$$r_{ij} = \frac{\text{Min } x_{ij}}{x_{ij}}$$

Jika j adalah atribut biaya (*cost*)

Keterangan:

r_{ij} : Rating kerja ternormalisasi

$\text{Max } x_{ij}$: Nilai maksimum dari nilai baris dan kolom

$\text{Min } x_{ij}$: Nilai minimum dari nilai baris dan kolom

x_{ij} : Baris dan kolom matriks

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Rumus akhir dari alternatif

Keterangan:

V_i : Nilai akhir dari alternatif

w_j : Bobot yang telah ditentukan

r_{ij} : Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif lebih terpilih. Rating kinerja ternormalisasi dari Alternatif A_i pada atribut yang ditentukan C_j ; $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, n$. Bobot dari kriteria ditentukan oleh pengambil keputusan,

minimal terdapat dua kriteria yang ditentukan oleh pengambil keputusan, sehingga dapat melakukan perbandingan menggunakan formula tersebut.

BAB III

ANALISIS MASALAH DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini, metode yang akan digunakan adalah Metode Penelitian Tindakan (*Action Research*). Metode Penelitian Tindakan adalah suatu penelitian yang dikembangkan bersama-sama antara peneliti dan *decision maker* tentang variabel -variabel yang dapat dimanipulasikan dan dapat segera digunakan untuk menentukan kebijakan dan pembangunan (Nazir, 2011:79).

3.2 Analisis Permasalahan

PT. Nippon Indosari Corpindo. Tbk, telah memiliki sistem yang distribusi untuk mitra kerja, tetapi belum mampu mendukung kinerja yang lebih efisien. Sistem Informasi yang saat ini digunakan hanya dapat digunakan oleh *admin* kasir, dengan transaksi dan faktur yang masih manual. Dari perhitungan cara kerja seperti ini informasi yang dapat diberikan melalui sistem adalah informasi empat sampai tiga hari yang lalu bahkan sampai tiga puluh hari, dikarenakan data yang harus di entri pada sistem sangat banyak setiap harinya dan juga rangkap kerja tugas dari admin. Hal tersebut menjadi kendala dalam pengambilan keputusan.

Untuk dapat memangkas keterlambatan perolehan informasi adalah mengembalikan peran dan tugas administrasi masing-masing komponen organisasi. Berikut tugas-tugas yang wajib dikerjakan masing-masing amin komponen:

1. Koordinator

Bertugas mengambil keputusan dalam melakukan *Purchase Order*.

2. Admin Kasir

Bertugas mendata piutang *outlet*, menerima tagihan outlet dari salesman, melakukan peyeteran uang ke produsen.

3. *Salesman*

Melakukan penjualan, droping roti dan menagih piutang *outlet* ke *outlet* berdasarkan kunjungan dan area masing-masing *salesman*.

4. Kepala Gudang

Menerima dan mengalokasikan roti *fresh* berdasarkan *salesman*, dan juga menerima dan pendataan roti *bad stok* yang diterima salesman dari *outlet*.

3.3 Perancangan Sistem

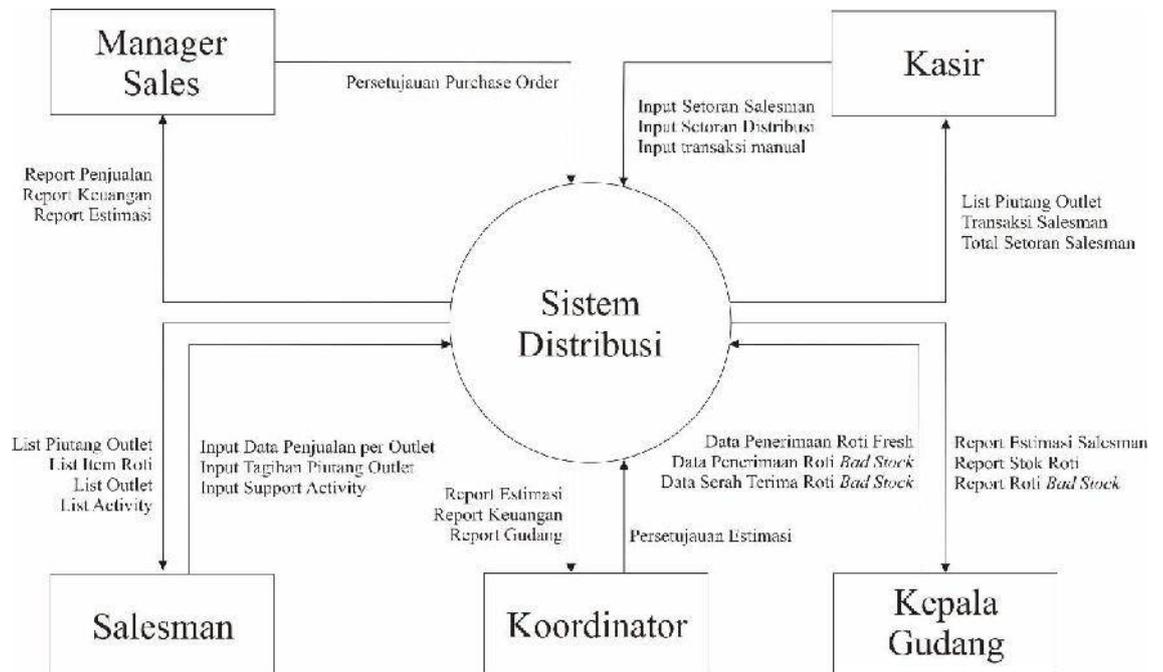
Perancangan sistem merupakan tahapan dari siklus pengembangan sistem. Tahapan ini akan mendefinisikan kebutuhan-kebutuhan fungsional dan yang mendasar dari system untuk menggambarkan bagaimana sistem tersebut dibangun. Sehingga, terlihat gambaran yang jelas sistem yang akan dibangun.

Tahapan perancangan ini meliputi beberapa tahapan, yaitu:

1. Perancangan Proses
2. Perancangan *Database*
3. Perancangan *Interface*

3.3.1 Diagram Konteks Distribusi mitra Kerja PT. NIC, Tbk

Ruang lingkup proses distribusi mitra kerja PT. Nippon Indosari Corpindo, Tbk digambarkan ke dalam diagram konteks. Diagram konteks menggambarkan secara sederhana bagaimana semua proses sistem dan penampilan data distribusi. Berikut diagram konteks distribusi mitra kerja PT. Nippon Indosari Corpindo, Tbk:



Gambar 3.1 Diagram Konteks Distribusi mitra Kerja PT. NIC, Tbk

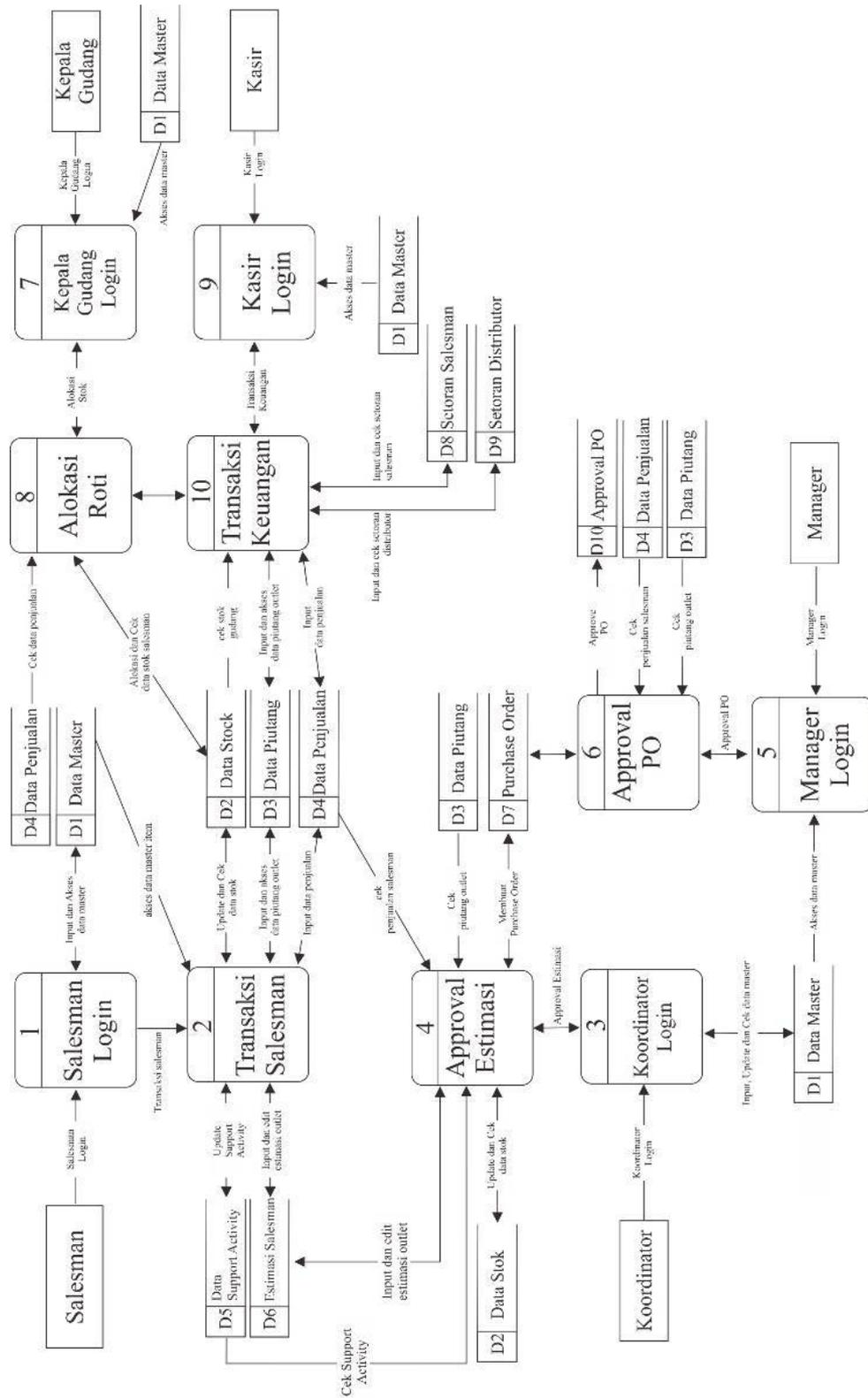
Pada diagram konteks diatas menggambarkan terdapat empat *user* yang dapat mengakses system distribusi tersebut. Diagram konteks disusun berdasarkan tugas masing-masing elemen dalam organisasi distribusi mitra PT. Nippon Indosari Corpindo, Tbk. Tugas digambarkan berdasarkan simbol arah, dari mana dan kemana tugas dilakukan, dan setiap keterangan tugas mengikuti symbol arah, sehingga tidak ada tugas yang dilakukan oleh user lain yang bukan dari tugas dari element tersebut.

3.3.2 DFD Level Nol

Gambar Data Flow Diagram (DFD) Level Nol dibawah adalah alur kerja sistem secara sederhana tetapi lebih kompleks dibandingkan dengan Diagram Konteks, yaitu berada 1 tingkat lebih kompleks dan logika kerja sistem lebih jelas terlihat. DFD sudah menampilkan tidak hanya proses dan user saja, tetapi sudah kepada ruang penyimpanan data, baik alur data yang masuk maupun keluar.

Setiap element distribusi wajib melalui keamanan masing-masing, sehingga proses yang dikerjakan adalah berdasarkan fungsi dan tugas. Tugas dari setiap elemen berbeda dan ada keterbatasan fungsi yang diberikan untuk menciptakan keamanan data yang hanya dapat dibuat berdasarkan fungsi kerja elemen. Tetapi ada beberapa tugas yang dapat dikerjakan oleh elemen lain, seperti transaksi penjualan yang dapat dilakukan oleh salesman tetapi dapat juga dilakukan oleh kasir. Hal ini dikarenakan, konsumen dapat melakukan order secara langsung ke distributor dengan partai besar atau *event* tertentu, sehingga yang melakukan transaksi ini adalah kasir.

Elemen tersebut terbagi menjadi lima elemen, yaitu, *salesman*, kepala gudang, kasir, koordinator dan juga manager. Sebagai puncak *Decision Maker* yaitu manager keputusan akhir berada pada *user* manager. Manager berhak untuk melakukan *adjustment*. Sedangkan kasir dapat melakukan semua transaksi yang dilakukan oleh salesman, dikarenakan untuk mengantisipasi kegagalan transaksi yang terjadi dilapangan oleh salesman atau transaksi langsung yang dilakukan konsumen di kantor distribusi Sari Roti.



Gambar 3.2 DFD Level Nol Sistem Distribusi Mitra PT. NIC, Tbk

3.3.3 Perancangan Database

Database dari sistem distribusi terdiri dari beberapa *entity* dan akan disimpan pada perangkat keras komputer kemudian dimanipulasi sesuai dengan kebutuhan dari *user*. Setiap *entity* memiliki atribut yang berbeda dan dikelompokkan secara restruktur.

Berikut skema tabel yang dirancang untuk membangun sistem distribusi mitra PT. NIC, Tbk.

1. Tabel Daftar Distributor

Tabel daftar distributor adalah tabel data distributor yang bekerja sama sebagai mitra PT. NIC, Tbk. Tabel daftar distributor akan menjadi dasar pembuatan id karyawan, id outlet dan id transaksi di dalam sistem, untuk membedakan data antar distributor. Data-data tersebut didapat berdasarkan data kontrak kerjasama antara perusahaan distribusi dengan PT. NIC, Tbk.

Tabel 3.2.1 Tabel Daftar Distributor

Nama Field	Tipe Data	Ukuran	Deskripsi
entity_distributor	int	8	Entitas distributor
nama_distributor	varchar	35	Nama distributor
nama_perusahaan	varchar	35	Nama perusahaan
titik_distribusi	varchar	35	Titik distribusi
nama_pemilik	varchar	35	Nama pemilik distributor
alamat_distribusi	varchar	35	Alamat distributor
provinsi	varchar	35	Provinsi
kota	varchar	35	Kota

2. Tabel Data Master Karyawan Distribusi

Tabel karyawan adalah tabel yang berisikan atribut data diri karyawan. *id_karyawan* menjadi *primary key* pada Table tersebut. *Field* canvas ditujukan hanya untuk *salesman* saja, salesman dibedakan tugas kerjanya adalah canvas kerja. Ada salesman yang berfungsi hanya sebagai motoris, *droper* dan penjual. Untuk manajer, koordinator dan juga kasir tidak ada perbedaan yang spesifik lagi.

Tabel 3.2.2 Tabel Data Master Karyawan Distributor

Nama Field	Tipe Data	Ukuran	Deskripsi
id_karyawan	bigint	15	Id karyawan
jabatan	varchar	15	Jabatan karyawan
nm_karyawan	varchar	35	Nama karyawan
canvas	varchar	35	Fungsi kerja salesman
kata_kunci	varchar	35	Password user
ktp	bingint	15	KTP Karyawan
kontak	int	12	Kontak Karyawan
alamat	varchar	15	Alamat Karyawana
on_bpjs	bigint	15	Nomor BPJS Karyawan
status_kerja	varchar	11	Status efektif kerja karyawan

3. Tabel Data Master Item

Tabel data master item adalah tabel yang berisi data item-item yang di produksi oleh PT. NIC, Tbk. *kode_item* menjadi *primary key*, karena merupakan kode *uni* dari data tersebut.

Tabel 3.2.3 Tabel data master item

Nama Field	Tipe Data	Ukuran	Deskripsi
kode_item	varchar	8	Kode item roti

nama_item	varchar	35	Nama item roti
singkatan	varchar	6	Singkatan nama item roti
harga	int	5	Harga Roti
tgl_expired	date	null	Tanggal expired roti

4. Table Data Master *Outlet*

Tabel data master outlet adalah tabel yang berisikan data outlet distributor mitra PT. NIC, Tbk. Setiap *outlet* di *cover* oleh salesman berdasarkan area *cover* masing-masing yang telah ditentukan oleh koordinator. Data salesman yang dimuat dalam Tabel tersebut adalah *id_karyawan*, yang diambil dari tabel master data karyawan saat pembuatan *outlet* baru atau NOO (*New Open Outlet*) pada sistem distribusi.

Tabel 3.2.4 Tabel data master *outlet*

Nama Field	Tipe Data	Ukuran	Deskripsi
kode_outlet	bigint	15	Kode <i>outlet</i>
nama_outlet	varchar	35	Nama <i>outlet</i>
owner	varchar	35	Nama pemilik Outlet
kontak	int	12	Kontak outlet
id_karyawan	bigint	15	Id yang digunakan adalah id salesman
tipe_outlet	varchar	35	Jenis <i>outlet</i>
lokasi	varchar	35	Lokasi <i>outlet</i> berada
sup_lokasi	varchar	35	Lokasi sekitar <i>outlet</i>
alamat	varchar	35	Alamat <i>outlet</i>
provinsi	varchar	35	Provinsi lokasi <i>outlet</i>
kota	varchar	35	Kota lokasi <i>outlet</i>
kecamatan	varchar	35	Kecamatan lokasi <i>outlet</i>
kelurahan	varchar	35	Kelurahan lokasi <i>outlet</i>

5. Tabel *Sales Order* (SO)

Tabel *Sales Order* (SO) berisi data order, tetapi tidak secara detail. Tabel tersebut bersikan informasi umum dari SO, untuk tabel SO juga menyimpan nomor PO (*Purchase Order*) sehingga SO yang awalnya berdasarkan *outlet* dan item roti, menjadi data global, yaitu pesanan distributor terhadap PT. NIC, Tbk.

Tabel 3.2.5 Tabel *Sales Order*

Nama Field	Tipe Data	Ukuran	Deskripsi
nomor_so	bigint	15	Nomor pembuatan SO
tgl_so	date	null	Tanggal pembuatan SO
tgl_expectedarrv	date	null	Tanggal pengantaran roti ke <i>outlet</i>
status_so	varchar	10	Status estimasi SO
nomor_po	bigint	15	Nomor pembuatan PO
tgl_po	date	null	Tanggal pembuatan PO
status_po	varchar	10	Status PO

6. Tabel Estimasi *Outlet*

Tabel Estimasi *Outlet* adalah tabel estimasi sales order yang detail. Data tabel tersebut terbentuk berdasarkan pembuatan SO dan tabel tersebut dapat berfungsi berdasarkan status so yang disetujui. Data item dan qty terbentuk berdasarkan penginputan data order yang dilakukan salesman.

Tabel 3.2.6 Tabel Estimasi *Order*

Nama Field	Tipe Data	Ukuran	Deskripsi
nomor_so	bigint	15	Nomor SO
id_salesman	bigint	15	Id Salesman
id_outlet	bigint	15	Id <i>outlet</i>

kode_item	varchar	10	Kode item
qty	int	10	Kuantitas item
status_estmasi	varchar	15	Status estimasi

7. Tabel *Receiving* (Penerimaan) Roti

Tabel *receiving* adalah tabel penerimaan roti dari produsen yaitu PT. NIC, Tbk. *no_dn* menjadi *primary key*, data tersebut diinput berdasarkan nomor *Delivery Number* (DN) yang diterima, dari surat jalan pengantaran roti dari PT. NIC, Tbk. ke distributor. Jumlah roti dan jenis roti yang datang adalah berdasarkan PO yang dilakukan oleh distributor sebelumnya kepada PT. NIC, Tbk.

Tabel 3.2.7 *Tabel Receiving*

Nama Field	Tipe Data	Ukuran	Deskripsi
no_dn	bigint	15	Nomor <i>Delivery Number</i>
tgl_receiving	date	null	Tanggal penerimaan dari PT. NIC, Tbk.
no_po	bigint	15	Nomor PO
kode_item	varchar	15	Kode item roti
qty	int	5	Kuantitas roti yang diterima dari PT. NIC, Tbk.
tgl_expired	date	null	Tanggal <i>expired</i> Roti
status_receiving	varchar	10	Status <i>receiving</i>

8. Tabel Gudang Salesman

Tabel gudang salesman adalah tabel berisi data roti dan jumlah roti yang dialokasikan dari gudang utama distributor ke gudang salesman setelah melakukan penerimaan dari PT. NIC, Tbk.. Data tersebut dibandingkan berdasarkan data tabel estimasi outlet yang di globalkan per salesman.

Table 3.2.8 Tabel Gudang Salesman

Nama Field	Tipe Data	Ukuran	Deskripsi
no_alokasi	bigint	15	Nomor alokasi roti ke gudang salesman
no_dn	bigint	15	Nomor DN
tgl_alokasi	date	null	Tanggal alokasi roti ke gudang salesman
id_salesman	bigint	15	Id salesman
kode_item	varchar	10	Kode Item
qty	int	5	Kuantitas item

9. Tabel Penjualan

Tabel penjualan adalah tabel berisi data penjualan roti ke *outlet*. Data penjualan tersebut termasuk data *droping* dan data penagihan. Data penjualan pada tabel penjualan diinput oleh salesman yang bertransaksi dilapangan, dan digunakan oleh kasir sebagai laporan tansaksi keuangan dan laporan tagihan *outlet*.

Tabel 3.2.9 Tabel Penjualan

Nama Field	Tipe Data	Ukuran	Deskripsi
no_faktur	bigint	15	Nomor faktur penjualan
id_salesman	bigint	15	Id salesman
kode_outlet	bigint	15	Kode <i>outlet</i>
kode_item	varchar	10	Kode item
status_item	varchar	2	Status roti setelah penagihan
qty_droping	int	5	Kuantitas <i>droping</i>

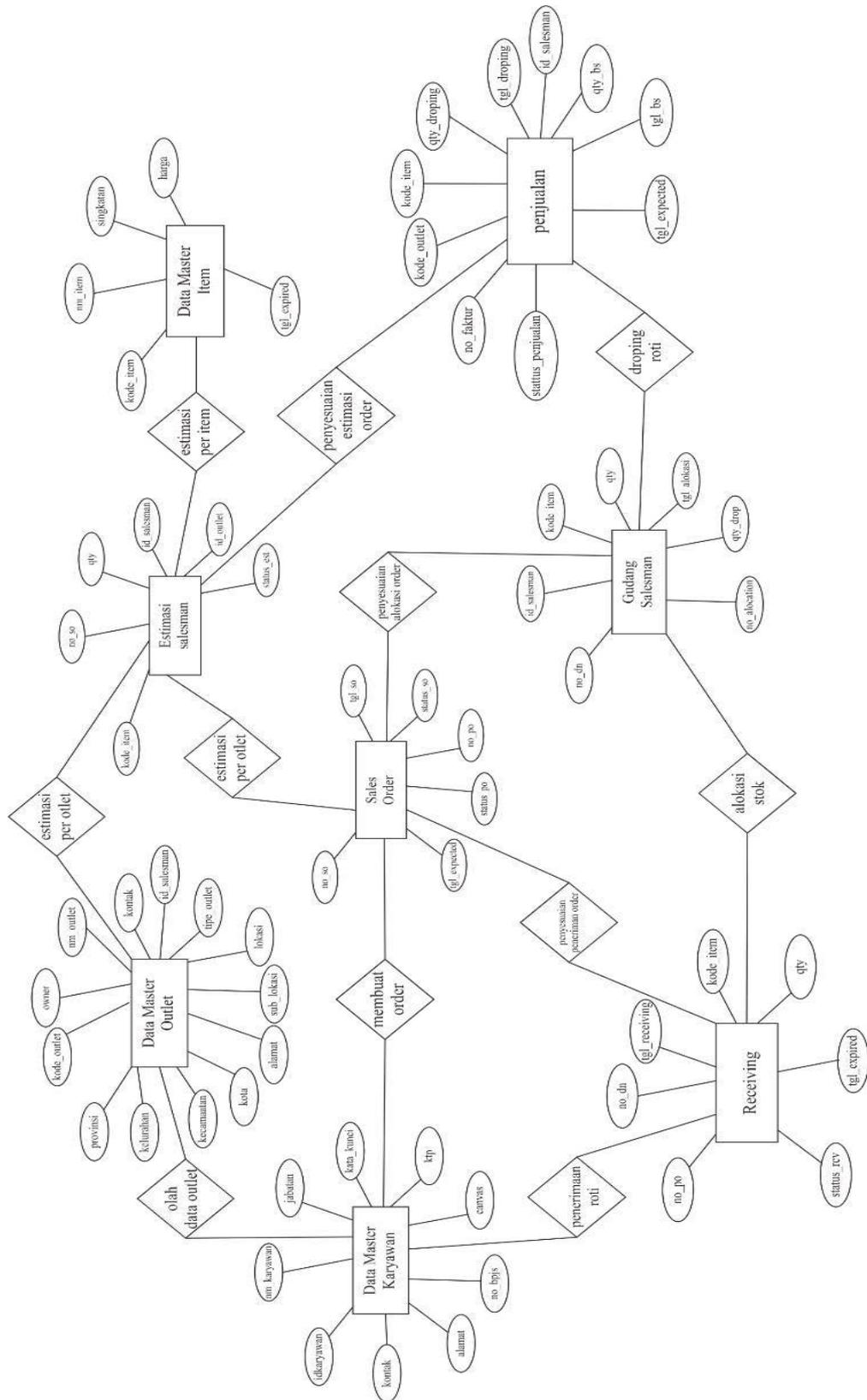
tgl_dropping	date	null	Tanggal <i>dropping</i>
qty_bs	int	5	Kuantitas <i>bad stock</i>
tgl_bs	date	null	Tanggal <i>bad stock</i>
tgl_penagihan	date	null	tanggal penagihan
tgl_pembayaran	date	null	Tanggal pembayaran
status_penjualan	varchar	10	Status penjualan

3.3.4 *Entity Relationship*

Gambar di bawah adalah gambar grafis dari *entity relationship* dari tabel *database*, setiap *entity* mempunyai keunikan dan fungsinya. *Entity* dihubungkan menggunakan relasi, yang isinya adalah aksi yang dilakukan antar *entity*. Prosesnya ada yang melingkupi *input* data ataupun data yang diolah terhadap *entity*, dimana data-data pada *entity* tersebut disimpan dalam ruang penyimpanan primer, kemudian dioalah menjadi informasi atau laporan yang bermanfaat terhadap user.

Entity pada system tersebut terdiri dari Data Master Outlet, Data Master Item, Data Master Karyawan, *Sales Order*, Estimasi Salesman, Receiving, Gudang Salesman dan Penjualan. Setiap *entity* berelasi menghasilkan data yang saling berkaitan dan menjadi informasi yang berguna bagi *user*.

Entity Relationship tersebut sangat kompleks dikarenakan terjadi beberapa kali pemanggilan untuk mengisi data dari *entity* lainnya. Cara tersebut akan memangkas penggunaan ruang untuk data yang sama, karena itu dibuat *entity* khusus yaitu data master karyawan, data master *outlet* dan data master item.



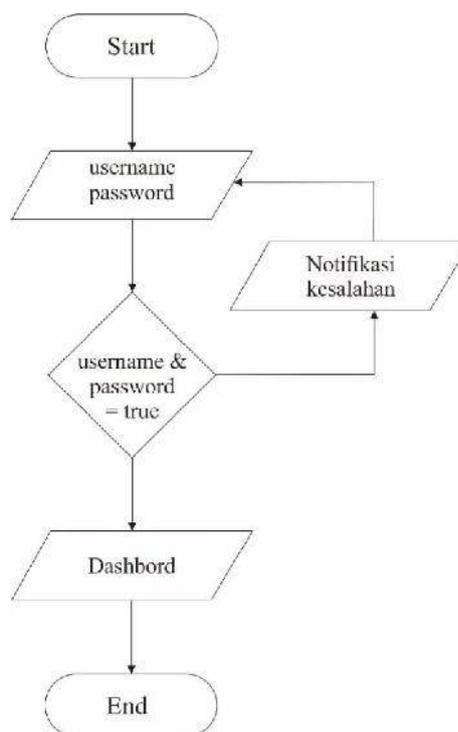
Gambar 3.3 ER Tabel Database Sistem Distribusi Mitra PT. NIC, Tbk

3.3.5 *Flowchart* Sistem Distribusi Mitra PT. NIC, Tbk.

Untuk proses alur kerja program, akan dirancang terlebih dahulu menggunakan *flowchart*, sehingga akan terlihat ilustrasi kerja dari sistem. *Flowchart* akan mempermudah dalam memahami alur kerja ketika melakukan perancangan program, karena *flowchart* dari setiap transaksi kerja menjelaskan secara detail kerja sistem tersebut.

1. *Flowchart* Login

Login adalah proses dari *user* untuk bisa mengakses sistem, di mana sistem yang dapat diproses adalah berdasarkan *user* elemen yang digunakan. Jika, *user* yang digunakan adalah *user* kasir, maka fungsi kerja sistem adalah sebagai *user* kasir, begitu juga dengan *user* elemen lainnya.

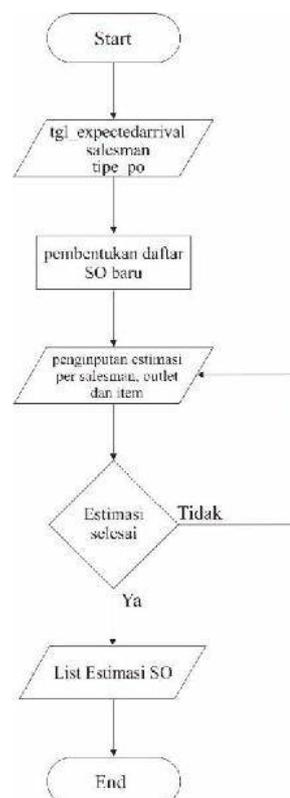


Gambar 3.4 *Flowchart* Login

Pada *flowchart* tersebut *user* memasukkan data *username* dan *password* berdasarkan elemen masing-masing. Setelah di-*input*, maka sistem akan melakukan pemeriksaan data tersebut, jika data tersebut bernilai benar dengan data yang tersimpan dengan data pada tabel *datetabase* karyawan, maka *user* langsung diarahkan ke *dashbord* dan pada halaman sesuai tugas elemen yang di-*input* di awal proses *login*.

2. *Flowchart* Estimasi *Sales Order* (SO)

Estimasi SO adalah proses perencanaan berdasarkan kebutuhan *order* roti yang akan di-*droping* pada kunjungan *salesman* yang akan datang. Estimasi SO dilakukan berdasarkan *salesman*, *outlet* yang akan dikunjungi, dan juga roti.

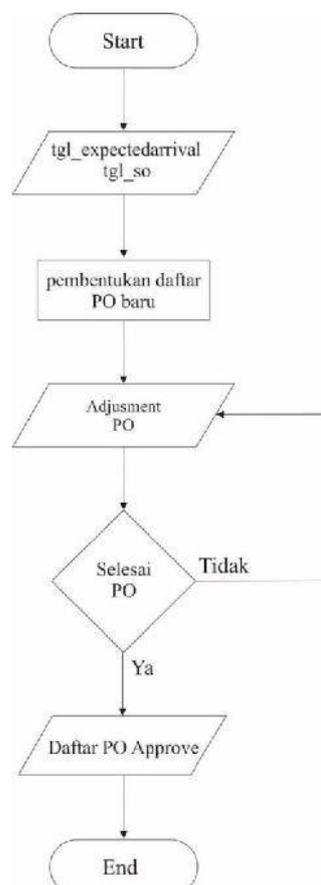


Gambar 3.5 *Flowchart* Estimasi SO

Alur dari *flowchart* estimasi SO dilakukan dimulai dari pembentukan tanggal kedatangan, *salesman*, kemudian tipe PO. Setelah itu, diproses pembentukan *list* SO yang siap diestimasi. Proses selanjutnya estimasi dilakukan *salesman* berdasarkan *outlet* kunjungan dan *item* roti yang akan diantarkan pada tanggal kunjungan sebelumnya.

3. *Flowchart* Purchase Order (PO)

PO merupakan proses pemesanan *order* ke PT. NIC, Tbk. Data PO diambil berdasarkan data global dari Estimasi SO yang telah disetujui oleh koordinator distributor Sariroti.

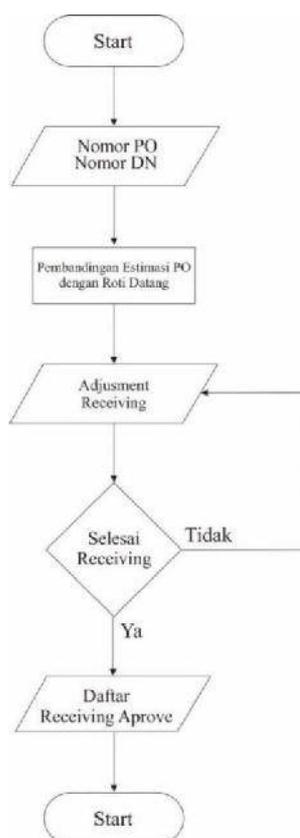


Gambar 3.6 *Flowchart* PO

Purchase Order dibentuk berdasarkan tanggal kedatangan dan tanggal pembentukan SO. Setelah data tersebut di-*input* maka dilakukan proses pembentukan *list* PO tersebut. Koordinator berhak melakukan *adjustment* pada PO meskipun tanpa persetujuan *salesman*. Jika transaksi PO sudah disetujui maka *user* akan diarahkan kepada *list* PO.

4. *Flowchart Receiving* (Penerimaan)

Receiving adalah proses transaksi penerimaan roti dari PT. NIC, Tbk. untuk didistribusikan. Serah terima roti tersebut dilakukan oleh Kepala Gudang distributor Sariroti. Jika serah terima ini telah selesai maka, roti menjadi tanggung jawab kepala gudang, sebelum di serahkan ke gudang masing-masing *salesman*.

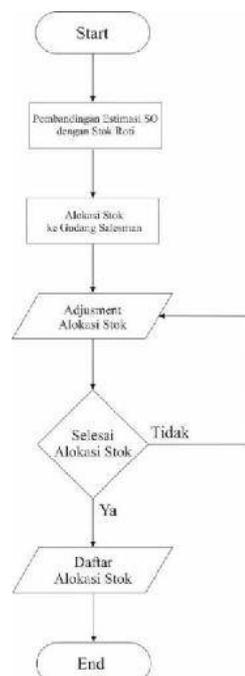


Gambar 3.7 *Flowchart Receiving*

Receiving dimulai dari pemilihan nomor PO dari PO yang dilakukan sebelumnya untuk kedatangan roti tersebut dan peng-*input*-an surat jalan atau *Delivery Note* (DN). Hal ini dilakukan untuk perbandingan *item* roti dan jumlah roti yang diterima oleh kepala gudang. Jika ditemukan ada data yang berbeda dari PO sebelumnya terhadap DN, maka kepala gudang melakukan *adjustment* terhadap data PO sesuai dengan data yang tertera pada DN. Jika selesai dilakukan, maka akan tampil kembali kepada *lis Receiving* roti yang sudah disetujui oleh kepala gudang.

5. *Flowchart* Alokasi Stok

Proses alokasi roti dari gudang utama ke gudang *salesman* dinamakan alokasi stok. Proses ini dilakukan kepala gudang dan *salesman*, agar tidak terjadi kesalahan saat alokasi ke gudang *salesman*, karena roti yang telah dimasukkan ke gudang *salesman* akan menjadi tanggung jawab *salesman*.

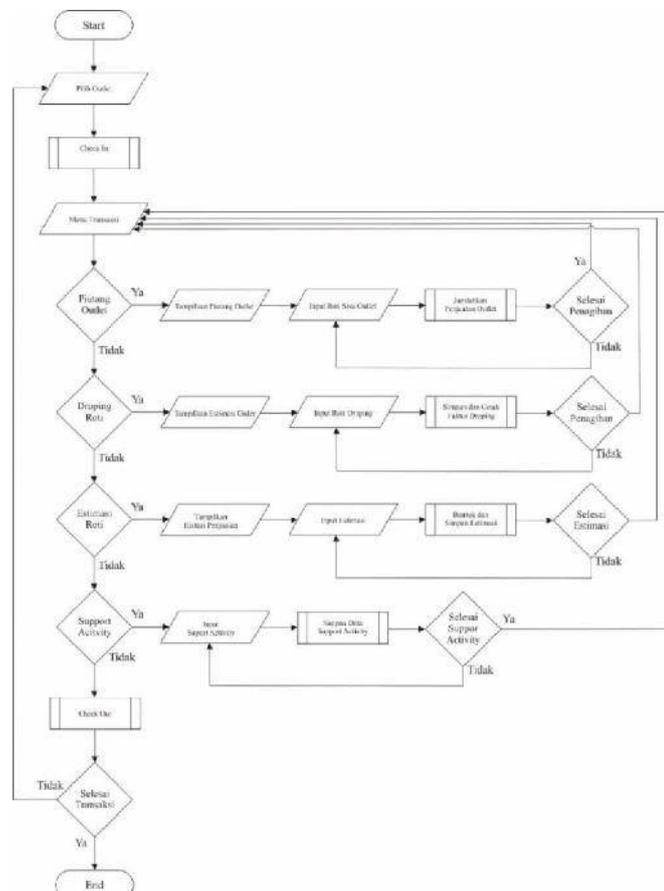


Gambar 3.8 *Flowchart* Alokasi Stok

Alokasi dari gudang utama dan gudang *salesman* dilakukan berdasarkan data estimasi order yang dilakukan oleh *salesman*. Meskipun estimasi yang dilakukan per *outlet* dan per *item*, data yang diambil adalah global dari data estimasi tersebut, hal tersebut dikarenakan pengalokasi adalah data global per *salesman*.

6. Flowchart Trankansi Salesman

Transaksi *salesman* dilakukan di lapangan, saat *salesman* berkunjung dilapangan. Proses kerja ini mencakup waktu kerja *salesman* dikarenakan seorang *salesman* wajib melakukan *Efektif Call* (EC) atau kunjungan *outlet* yang bertransaksi sebanyak lima belas *outlet per hari*.



Gambar 3.9 Flowchart Transaksi Salesman

Proses transaksi salesman dimulai dari *Check In*, Piutang Outlet, *Dropping Roti*, Estimasi Roti, *Support Activity*, *Check Out*, kemudian transaksi dilakukan kembali di outlet yang berbeda. Setiap transaksi di outlet harus mengikuti alur kerja, sehingga tidak ada informasi transaksi yang terlewat.

Dikarenakan transaksi *salesman* ada batasan waktu, yaitu paling lama adalah tujuh menit dimulai dari kedatangan ke *outlet* sampai keberangkatan dari *outlet*. Maka dilakukan proses *check in* diawal transaksi dan *check out* diakhir transaksi. Sebelum itu *salesman* menentukan terlebih dahulu *outlet* yang akan dikunjungi. Setelah *salesman* melakukan *check in*, ada empat tugas yang dikerjakan *salesman* saat di *outlet*, yaitu penagihan piutang, *dropping*, estimasi, dan juga *support activity*. Ketika *salesman* melakukan penagihan piutang *outlet*, *salesman* memeriksa tagihan apakah *outlet* terdapat tagihan atau tidak, jika *outlet* memiliki tagihan, maka *salesman* meng-input data roti sisa yang ada di *outlet* tersebut, kemudian total *dropping* dikurangi dengan roti sisa untuk menampilkan tagihan. Jika transaksi tagihan piutang *outlet* selesai, maka *salesman* melakukan transaksi selanjutnya yaitu *dropping* roti, hingga selesai kemudian kembali pada menu transaksi *outlet*. Estimasi dilakukan dapat berdasarkan histori penjualan, berdasarkan sistem keputusan yang dirancang pada sistem distribusi dan dapat juga berdasarkan *request* dari pemilik *outlet*. Setelah transaksi *dropping* selesai, sistem mengarahkan ke daftar transaksi dan dilanjutkan pada *support activity*, pada tahap ini, *salesman* melakukan *support* yang diperlukan oleh *outlet* untuk mendukung transaksi *outlet*.

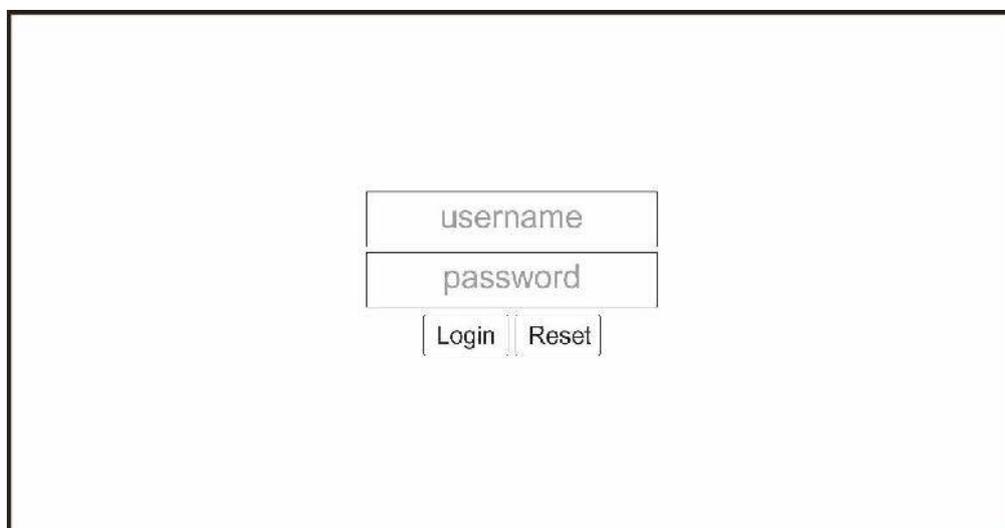
3.4 Perancangan *User Interface* Sistem Distribusi

Perancangan *user interface* bagian yang penting sebagai mekanisme komunikasi *user* terhadap sistem. Melalui *user interface*, sistem dapat menerima dan memberikan informasi terhadap *user*, dan mempermudah alur kerja sistem dapat dipahami dan diakses oleh *user*.

User interface pada sistem distribusi Sariroti menggunakan bahasa yang baku dalam Bahasa Inggris. Tampilan dirancang responsive dikarenakan khusus *user salesman* yang akan mengakses menggunakan *mobile* dengan *platform* android. Berikut *user interface* yang dirancang sebagai media komunikasi sistem distribusi terhadap *user*.

1. *User Interface Login User*

Login user dirancang agar sistem hanya dapat diakses oleh *user* tertentu, yaitu *salesman*, kasir, kepala gudang dan koordinator. Model tersebut didesain karena transaksi dan informasi bersifat rahasia dan hanya digunakan untuk kepentingan operasional distributor.



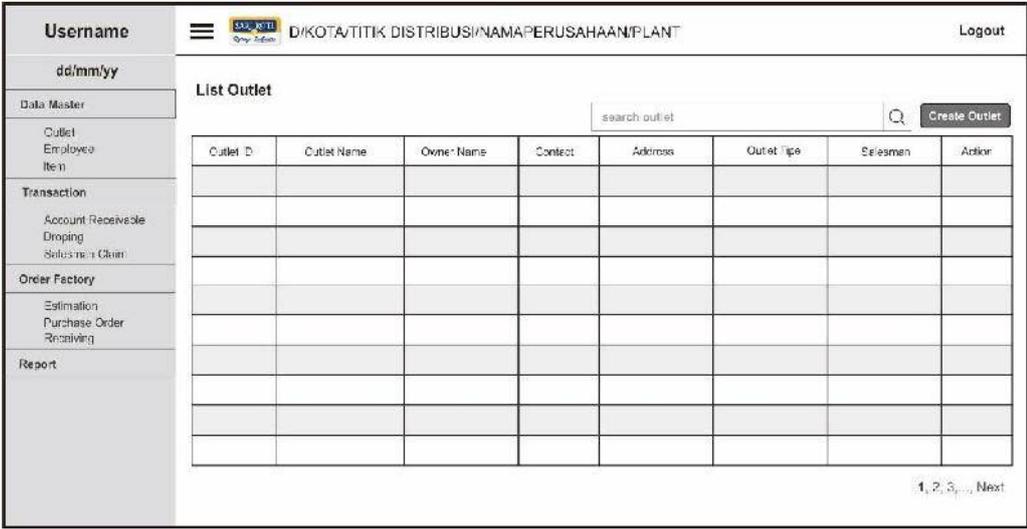
The image shows a login form with two input fields: 'username' and 'password'. Below the 'password' field are two buttons: 'Login' and 'Reset'.

Gambar 3.10 *User Interface Login User*

Pada desain tersebut terdapat *input username* dan *password* yang berfungsi untuk menginput nama pengguna dan kata kunci agar dapat mengakses *user* yang terdaftar dalam *database* karyawan. Tombol *Login* sebagai perintah untuk mengakses *user* tersebut, sedangkan tombol *Reset* untuk mengosongkan *input username* dan *password*.

2. User Interface Master Data

Menu pada sistem distribusi salah satunya adalah Master Data. Pada menu tersebut terbagi atas tiga bagian yaitu *Outlet*, *Employee* dan *Item*. Ketiga bagian tersebut merupakan daftar data fundamental dari sistem distribusi, jadi didesain menjadi satu bagian.



Outlet ID	Outlet Name	Owner Name	Contact	Address	Outlet Type	Salesman	Action

Gambar 3.11 User Interface Master Data List outlet

Dalam *user interface List Outlet* terdapat table yang berisikan informasi umum dari *outlet* yang tertera, untuk melihat informasi detail pada salah satu outlet terdapat tombol edit, didalamnya juga terdapat perintah untuk *cover* dan *uncover* outlet tersebut. Pencarian outlet dapat dilakukan pada bagian pencarian, dapat dicari berdasarkan nama ataupun kode outlet tersebut. Daftar

outlet di desain dengan tabel berjumlah sepuluh baris per halaman, dengan menggunakan membuat *paging* pada halaman *List Outlet*.

Employee ID	Employee Name	Function	Contact	Action

Gambar 3.12 *User Interface Master Data List Employee*

List Employee terdapat lima kolom, berisikan data karyawan Distributor Sariroti.

Untuk penambahan karyawan baru dapat menggunakan tombol *New Employee*.

Daftar tersebut hanya dapat diubah dan ditambahkan hanya oleh koordinator

saja, sebagai penanggung jawab distribusi.

Item ID	Item Name	Short Name	Price

Gambar 3.13 *User Interface Master Data List Item*

List Item merupakan daftar jenis item Sariroti yang didistribusikan di area masing-masing. Item tersebut tidak dapat diubah datanya, karena hal tersebut sudah menjadi ketentuan dari produsen, dan hanya dapat diubah oleh produsen.

3. *User Interface Transaction*

User interface Transaction adalah desain antar muka terhadap kasir dan juga salesman. Dalam menu *Transaction* terdapat tiga bagian tugas yaitu, *Account Receivable (AR)*, *Dropping* dan *Claim Salesman*. Kasir dapat mengakses ketiga tugas tersebut, tetapi *salesman* hanya dapat mengakses dua tugas saja, yaitu *AR* dan *Dropping*.

Outlet ID	Outlet Name	Invoice	Salesman	Due Date

Gambar 3.14 *User Interface Account Receivable*

Account Receivable merupakan *user interface* yang bersikan informasi tagihan outlet yang telah melakukan transaksi. Khusus salesman, ketika mengakses menggunakan *user salesman*, maka informasi yang diterima adalah berdasarkan *outlet* yang di-*cover* oleh *salesman*, sedangkan kasir dapat mengakses semua informasi piutang outlet, agar kasir dapat memantau piutang berdasarkan jatuh tempo atau *Due Date* yang telah lebih dari tiga hari.

Gambar 3.15 *User Interface Check In Outlet*

User interface Check In Outlet bagian dari *dropping* roti ke outlet, dikarenakan tampilan tersebut merupakan *user interface* untuk kasir, maka kasir harus terlebih dahulu mencari daftar *outlet* yang akan dilakukan *dropping* roti sebelum *check in* berdasarkan *salesman* dan tanggal pengantaran roti atau *expected arrival*, sedangkan *salesman* dapat langsung mencari *outlet*, dikarenakan daftar *outlet* yang ditampilkan adalah yang di-*cover* dan kunjungan pada hari itu juga.

Gambar 3.16 *User Interface Transaksi Outlet*

Gambar 3.4.7 merupakan *user interface* dari transaksi salesman di *outlet*. Terdapat empat tugas yang dilakukan *salesman* selama melakukan transaksi, yaitu *AR*, *Dropping*, *Estimation* dan *Support Activity*. Dan transaksi tersebut dilakukan berdasarkan waktu kunjungan yang ditampilkan di bawah dari nama *outlet*. Dan juga terdapat tombol *Check Out*, aksi tersebut dilakukan setelah salesman selesai melakukan transaksi di *outlet*.

Facility	Qty
<input checked="" type="checkbox"/> Allabord	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Crete	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> School Tray	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Rack	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Street Banner	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Flyer	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Poster	<input type="text"/> + <input type="text"/> -
<input type="checkbox"/> Tablecloth	<input type="text"/> + <input type="text"/> -

Gambar 3.17 *User Interface Support Activity*

User Interface Support Activity sebagai *form* yang dirancang untuk memberikan dukungan terhadap *outlet* sehingga dapat meningkatkan penjualan di *outlet*. Dukungan terhadap *outlet* dapat berupa POSM (*Point of Sales Material*) maupun saran dari pemilik *outlet* terhadap distributor atau produsen Sariroti.

4. *User Interface Order to Factory*

Order to Factory merupakan transaksi yang dilakukan oleh mitra distribusi terhadap produsen, mulai dari estimasi yang dilakukan oleh *salesman* atau *Sales Order* sampai dengan penerimaan roti atau *Receiving*. Pada transaksi ini, terdapat *estimation suggest* atau saran estimasi yang diberika terhadap *user*

untuk melakukan estimasi melalui metode DSS yang dirancang di dalam sistem berdasarkan kriteria yang digunakan dalam metode yang digunakan.

SO Number	Salesman	Expected Delivery	Total Cost	Qty Estimation	Value Estimation	Action
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]
						✓ [trash] [edit]

Gambar 3.18 *User Interface Order to Factory*

Pada *user interface Estimation* terdapat tabel daftar estimasi yang dibentuk per salesman dan *Expected Delivery*. *SO Number* sebagai nomor order yang dibentuk menjadi kunci dalam daftar *Sales Order*. Ada terdapat tiga perintah pada kolom *Action* yaitu *Approve*, *Delete* dan *Edit*. *Edit* adalah perintah estimasi yang dilakukan sebagai pengisian estimasi order salesman per *outlet* dan per item, dan totalnya akan ditampilkan pada tabel daftar *Estimation*.

Pada proses pemesanan roti kepada produsen memiliki tiga tahap, yaitu *Create Sales Order (SO)*, *Salesman Estimation* dan *Create Purchase Order*. Pada tahap *Create SO*, user harus menentukan tanggal kedatangan roti ke outlet, sehingga produsen dapat melakukan pengantaran ke distributor sebelum didistribusikan, prosesnya dilakukan per salesman. Kemudian masuk pada tahap *Salesman Estimation*, tahap ini merupakan perencanaan pemesanan yang dilakukan secara detail, yaitu per outlet dan per item roti. Sehingga prosesnya

cukup panjang. Dan terakhir masuk pada tahap *Create Purchase Order*, yaitu pembentukan pemesanan ke produsen dari semua estimasi yang akan datang pada tanggal yang sama, menjadi satu paket yang terakumulasi menjadi total setiap itemnya.

The screenshot displays a web application interface for creating a sales order. At the top, there is a header with 'Username', a navigation menu, the company name 'D/KOTA/TITIK DISTRIBUSI/NAMAPERUSAHAAN/PLANT', and a 'Logout' button. A sidebar on the left contains a menu with categories: 'Data Master' (Outlet, Employee, Item), 'Transaction' (Account Receivable, Dropping, Salesman Claim), 'Order Factory' (Estimation, Purchase Order, Receiving), and 'Report'. The main content area is titled 'Estimation' and features a 'Create Sales Order' modal form. The form includes:

- 'Expected Delivery': A date input field with a calendar icon and a 'dd/mm/yy' placeholder.
- 'Salesman': A dropdown menu with 'salesman name' as the selected option.
- 'PO Type': A dropdown menu with 'po type' as the selected option.
- Buttons: 'Create SO' and 'Cancel' at the bottom of the form.

 To the right of the form is an 'Action' table with columns for checkmarks, delete icons, and edit icons. At the bottom right, there is a pagination indicator '1 2 3... Next'.

Gambar 3.19 User Interface Create Sales Order

Sebelum melakukan estimasi, *user* harus membuat daftar *sales order* melalui tombol *Create SO*, dengan menentukan *Expected Delivery*, *Salesman* dan *PO Type*. *Expected Delivery* adalah tanggal dilakukan pengantaran roti ke *outlet* oleh *salesman*, *salesman* yang dipilih adalah *salesman* yang akan mengantar roti dan *PO Type* adalah jenis *PO* yang di order. Jika *salesman* melakukan order untuk pengantaran roti ke outlet makan jenisnya adalah *regular*, sedangkan untuk dijual langsung ke *end user*, maka *PO Type* adalah *Direct Selling*.

Setelah *SO* telah selesai dibentuk dan dilakukan estimasi, maka *user* harus melakukan persetujuan terhadap *SO*, yaitu *Approve SO*. *SO* yang telah disetujui dapat dibatalkan jika *PO* dengan keterangan *Expected Arrival* dan *SO*

Date yang sesuai dengan SO di bentuk. Setelah SO dibatalkan, maka SO dapat di hapus dan juga di estimasi ulang oleh *user*.

The screenshot displays a web application interface for creating a Purchase Order. At the top, there is a navigation bar with 'Username', a menu icon, the path 'D/KOTA/TITIK.DISTRIBUSI.NAMAPERUSAHAAN/PLANT', and a 'Logout' button. Below this is a sidebar menu with categories like 'Data Master', 'Transaction', 'Order Factory', and 'Report'. The main area shows a 'Purchase Order' table with columns: 'Status PO', 'PO No', 'Value PO', 'PO Type', and 'Action'. A 'Create PO' button is located in the top right of the table area. A modal window titled 'Create Purchase Order' is open in the center, containing two date pickers: 'SO Date' and 'Expected Delivery', both set to 'dd/mm/yy'. The modal also features 'Create PO' and 'Cancel' buttons at the bottom.

Gambar 3.20 *User Interface Create Purchase Order*

Dengan menekan tombol *Create PO*, akan tampil form *Create Purchase Order* dengan isinya adalah *SO Date* yaitu tanggal pembuatan SO dan juga *Expected Arrival* yaitu tanggal pengantaran roti ke *outlet*. Pada form tersebut *user* tidak menentukan tanggal, tetapi memilih tanggal yang telah terbentuk saat persetujuan SO. Oleh sebab itu, pada rentang waktu yang dipilih, akan membentuk semua SO pada tanggal tersebut menjadi satu PO. Dengan menekan tombol *Create PO*, maka daftar PO baru akan terbentuk.

Pembentukan PO harus berbeda dengan PO yang telah ada, oleh sebab itu PO harus memiliki id unik atau *primary key*. Pembentukan id tersebut dibentuk dari nomor entity distributor, tanggal pembentukan kemudian kode untuk PO dan nomor urut pemesanan pada bulan tersebut. Proses ini akan mempermudah *user* dapat membedakan pemesanan satu dengan yang lainnya saat kedatangan roti ke distributor.

Status PO	PO Number	Expected Delivery	Local Outlet	Qty	Value PO	PO Type	Action
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎
							> ☰ ✎

Gambar 3.21 *User Interface List Purchase Order*

User interface Purchase Order terdapat daftar PO yang telah terbentuk, baik yang telah dikirim menjadi pesanan roti terhadap produsen maupun belum terkirim. PO yang belum dikirim sebagai pesanan, *user* koordinator masih dapat melakukan *Adjustment* PO sesuai dengan kebutuhan pemesanan roti pada tanggal pengiriman.

Create New Receiving

PO Number
 ▼

Expected Receiving
 📅

Gambar 3.22 *User Interface Create Receiving*

User interface Create Receiving terdapat PO Number dan Expected Arrival sebagai daftar dalam form yang wajib diisi oleh *user* sebelum melakukan

penerimaan. PO Number telah terbentuk berdasarkan PO yang dikirim kepada produsen. Sedangkan Expected Receiving adalah tanggal ketika distributor melakukan penerimaan secara fisik.

Item ID	Item Name	Qty PO	Actual Receive
			+ -
			+ -
			+ -
			+ -
			+ -
			+ -
			+ -
			+ -
			+ -
			+ -

Gambar 3.23 *User Interface Receiving*

User interface Receiving merupakan penyesuaian penerimaan berdasarkan *Delivery Note* atau surat jalan pengantaran roti dari produsen kepada distributor. Untuk kuantitas yang diterima berdasarkan DN akan di-*input* pada kolom aktual.

Status	DN Number	PO Number	Receiving Date	Qty Receiving	Value Receiving	Action
						+ -
						+ -
						✓ [trash] [refresh]
						✓ [trash] [refresh]
						✓ [trash] [refresh]
						✓ [trash] [refresh]
						✓ [trash] [refresh]
						✓ [trash] [refresh]

Gambar 3.24 *User Interface Allocation Item*

Pada *user interface* tersebut ketika penerimaan telah disetujui oleh *user* maka perintah akan berubah menjadi tombol alokasi roti. Perintah tersebut sebagai langkah selanjutnya, penyesuaian alokasi roti ke gudang salesman, yang di alokasikan berdasarkan fisik roti ke salesman.

5. *User Interface Report*

Report merupakan sumber informasi yang juga dibutuhkan dalam mengambil keputusan bagi user. *Report* adalah informasi yang dapat dikelola sedemikian rupa, untuk merancang strategi, sesuai dengan kehendak user. Karena itu *report* juga harus bersifat dinamis, sehingga dapat dikelola oleh user.

The screenshot shows a web application interface. At the top, there is a header with a logo, the text 'D/KOTA/TITIK DISTRIBUSI/NAMAPERUSAHAAN/PLANT', and a 'Logout' button. Below the header is a navigation menu with categories like 'Data Master', 'Transaction', and 'Report'. The main content area is titled 'Report Salesman Estimation' and contains a form with fields for 'From Date', 'To Date', 'Salesman', and 'Export By'. Below the form is a table with 10 columns and several rows. A 'Show Report' button is located to the right of the table. On the right side of the interface, there is a 'Report List' section with a list of report types: 'Estimation Report', 'Purchase Order Report', 'Sales Order Report', and 'AR Report'.

Gambar 3.25 *User Interface Report*

Pada *user interface report* terdapat daftar *report* yang dapat digunakan untuk mengunduh data transaksi salesman. Daftar tersebut berada disebelah kanan dari halaman *report*. Pada setiap daftar terdapat *form* masing-masing, sebagai ketentuan yang diinput oleh user untuk mendapatkan report yang dibutuhkan.

3.5 Perancangan Algoritma Metode SAW dalam Sistem Distribusi

Dalam manufaktur, produsen atau principal, memiliki target dan mempunyai wewenang dalam pengambilan keputusan terhadap permintaan atau pemesanan mitra distribusi. Sehingga mitra tidak terhindar dari pemesanan yang terus meningkat terhadap produsen. Tetapi, mitra distribusi harus tetap menghindari adanya kerugian akibat pemesanan yang berlebihan. Oleh sebab itu, mitra distribusi harus secara matang dalam pengambilan keputusan dalam melakukan pemesanan. Hal yang menjadi pertimbangan dalam pengambilan keputusan adalah estimasi per per salesman, per outlet dan per item roti, apakah estimasi mengalami *growth*, *constant*, *taking down* atau *stop order*.

Metode SAW atau *Simple Additive Weighthing* digunakan untuk mendukung *Decisin Maker* atau dalam organisasi mitra distribusi pengambil keputusan adalah salesman ataupun koordinator distributor, dalam pengambilan keputusan. Metode ini sebagai *suggest* atau rekomendasi dalam pembuatan *Order Estimation*, tidak secara otomatis menjadi keputusan mutlak.

Metode SAW, memiliki kriteria yang menjadi acuan penilaian dan harus diberikan bobot terhadap kriteria-kriteria tersebut. Daftar kriteria sebagai acuan dalam pengambilan keputusan ada C_i , yaitu:

Tabel 3.4 Daftar Kriteria (C_i)

	Kriteria (C_i)	Bobot/ <i>Weighthing</i> (%)
C1	<i>History Sales</i>	50
C2	Jadwal Kujungan (JWK)	20
C3	Lokasi	20
C4	Jenis Outlet	10

Dalam tabel kriteria, terdapat empat kriteria yang akan menjadi acuan penilaian. Kriteria yang memiliki bobot terbesar adalah C1, ini dikarenakan *history sales* menjadi faktor utama dalam pembuatan *order estimation*.

Langkah kedua yaitu dengan menentukan rangking dari setiap pilihan terhadap kriteria, perengkingan akan diberi nilai dari 1 sampai dengan 10. Ada beberapa kondisi yang menjadikan perangkingan memiliki nilai yang berbeda, dan ini penilaian baku terhadap sistem.

A. History Sales

History Sales menjadi kriteria utama, karena persentase dari penjualan sangat mempengaruhi bertahannya sebuah perusahaan. Dalam organisasi mitra distribusi Sariroti, penjualan harus berada pada persentase 80% dari total pemesanan terhadap produsen yang telah ditentukan oleh principal secara *gross* atau *Costumer Buying Price* senilai Rp 300.000.000,-/bulan, agar mitra distribusi tetap mendapat keuntungan. Kurang lebih 10% dari pemesanan adalah untuk menutupi operasional dan 10% untuk menutupi terjadinya *bad stock* atau roti yang tidak laku di outlet.

Tabel 3.4.1 Table *Rating History Sales* Berdasarkan Persentase Pernjualan

<i>Rating</i>	$95\% \leq x \leq 100\%$	$85\% \leq x \leq 94\%$	$80\% \leq x \leq 84\%$	$0\% \leq x \leq 79\%$
Growth	5	3	2	0
Constant	3	5	3	2
Taking Down	2	2	5	3
Stop Order	0	1	3	5

Penilaian berdasarkan kondisi lapangan yang dilakukan oleh salesman saat melihat data *history sales*. Jika persentase penjualan sudah dibawah 80% maka,

salesman akan melakukan *stop order* dikarenakan tidak mampu menutupi operasional dan kerugian distribusi.

B. Jadwal Kunjungan Salesman (JWK)

Jadwal kunjungan salesman ke outlet terbagi atas tiga jenis, yaitu F_1 , F_2 , dan F_3 . Setiap jenis JWK terbagi atas beberapa bagian berdasarkan periode kunjungan, yaitu, F_1 terbagi atas 1 periode F_{11} , yang artinya adalah kunjungan setiap hari. F_2 terbagi atas dua periode pertama F_{21} artinya kunjungan per dua hari pada periode pertama dari dua hari tersebut, kedua F_{22} artinya kunjungan per dua hari pada periode kedua dari dua hari tersebut. F_3 terbagi atas tiga periode, pertama F_{31} yaitu kunjungan per tiga hari pada periode pertama dari tiga hari tersebut, kedua yaitu kunjungan per tiga hari pada periode ke dua dari tiga hari tersebut, ketiga F_{33} yaitu kunjungan per tiga hari pada periode ke tiga dari tiga hari tersebut.

Tabel 3.4.2 Rating F_1 Berdasarkan Roti Habis Sebelum JWK

Rating	$95\% \leq x \leq 100\%$	$85\% \leq x \leq 94\%$	$75\% \leq x \leq 84\%$	$0\% \leq x \leq 74\%$
<i>Growth</i>	5	3	2	0
<i>Constant</i>	3	5	3	1
<i>Taking Down</i>	2	2	4	2
<i>Stop Order</i>	0	1	3	5

Tabel 3.b. memiliki *rating* dengan pembagian kelompok presentase pada paling rendah adalah pada $0\% \leq x \leq 70\%$. *Rating* pada JWK F_1 tidak jauh beda dengan *rating history sale*, dikarenakan semakin sering salesman melakukan kunjungan semakin besar *cost* yang dikeluarkan. Jadi jika, persentase *history sales* dapat dicapai dengan JWK F_2 , maka JWK tidak perlu dijadikan F_1 .

Tabel 3.4.3 Rating F_2 Berdasarkan Roti Habis Sebelum JWK dan Persentase Sales

Rating	Hari Pertama	Hari Kedua
--------	--------------	------------

(95% ≤ x ≤ 100%)		
<i>Growth</i>	5	5
<i>Constant</i>	3	4
<i>Taking Down</i>	1	1
<i>Stop Order</i>	0	0

Pada Tabel 3.4.3 Memiliki perbedaan *rating* hanya pada pilihan *constant*. Perbedaan tersebut tidak jauh berbeda, karena kunjungan salesman cukup dekat, jadi *sales* roti dapat di pantau per dua hari. Diharapkan semua *dorping* roti dapat dilakukan pada frekuensi tersebut, karena penjualan lebih efisien untuk perputaran dana dan pemantauan roti di outlet.

Table 3.4.4 *Rating F₂* Berdasarkan Roti Yang Tidak Habis Saat JWK dan Persentase Sales

<i>Rating</i> (Roti tidak Habis)	85% ≤ x ≤ 94%	71% ≤ x ≤ 84%	0% ≤ x ≤ 70%
<i>Growth</i>	3	2	0
<i>Constant</i>	5	4	2
<i>Taking Down</i>	4	5	3
<i>Stop Order</i>	0	3	5

Pada Tabel 3.4.4 pengelompokan persentase terendah pada 0% ≤ x ≤ 70%, salesman memberikan *rating* yang besar pada *stop order*, dikarenakan jika roti di *dorping* ulang dalam persentase yang besar, akan mempengaruhi roti *fresh* yang baru saja di *droping* dan sudah pasti tidak akan dibeli karena ada roti dalam kondisi yang lebih *fresh*.

Tabel 3.4.5 *Rating F₃* Berdasarkan Roti Habis Sebelum JWK

<i>Rating</i> (95% ≤ x ≤ 100%)	Hari Pertama	Hari Kedua	Hari ketiga
<i>Growth</i>	5	5	5
<i>Constant</i>	3	3	4
<i>Taking Down</i>	1	1	1
<i>Stop Order</i>	0	0	0

Pada Tabel 3.4.5 perbedaan *rating* terletak pada pilihan *constant* yaitu kolom ketiga. Karena pada hari ketiga, perencanaan estimasi sudah sangat tepat dan memiliki peluang untuk *growth*.

Tabel 3.4.6 *Rating F₃* Berdasarkan Roti Yang Tidak Habis Saat JWK dan Persentase Sales

<i>Rating</i> (Roti tidak Habis)	$85\% \leq x \leq 94\%$	$75\% \leq x \leq 84\%$	$0\% \leq x \leq 74\%$
<i>Growth</i>	3	2	0
<i>Constant</i>	5	4	2
<i>Taking Down</i>	2	5	4
<i>Stop Order</i>	0	3	5

Pada Tabel 3.4.6 presentasi sudah sangat jelas, roti sisa mendekati masa akhir *expired date*, dan dapat dipastikan roti tidak akan terjual. Oleh sebab itu pada persentase yang semakin turun, makan *rating* pada pilihan *growth* akan menurun, dan *rating* pilihan lainnya meningkat.

C. Lokasi Outlet

Lokasi outlet adalah lokasi dimana secara umum outlet berada. Sari Roti memiliki pasar penjualan dengan pembeli ekonomi menengah ke atas dan pasar yang dimana pengonsumsi roti yang cukup besar. Untuk itu setiap lokasi outlet, memiliki *rating* yang jelas berbeda.

Tabel 3.4.7 *Rating Lokasi Outlet*

<i>Rating</i>	Fasilitas Pendidikan	Fasilitas Kesehatan	Pasar	Fasilitas Publik	Perkantoran
<i>Growth</i>	5	5	3	3	4
<i>Constant</i>	4	4	5	4	5
<i>Taking Down</i>	2	2	4	2	2
<i>Stop Order</i>	0	0	2	1	0

Pasar yang utama dari penjualan Sari Roti adalah Fasilitas Pendidikan, Fasilitas Kesehatan, dan Perkantoran. Dari segi kesibukan dan kebutuhan yang cukup mendesak dari konsumen pada pasar tersebut, membuat ketiga lokasi menjadi lokasi yang tepat untuk pemasaran Sari Roti.

D. Sub Lokasi Outlet

Sublokasi outlet menjadi karakter yang mempengaruhi pengambilan keputusan. Penyebabnya adalah lokasi outlet yang lebih spesifik dapat menggambarkan aktivitas yang terjadi di sekitar outlet tersebut. Ada dua pembagian waktu yang menjadi pertimbangan, dalam memberikan *rating* terhadap pilihan yaitu hari normal dan *weekend*.

Pada Sub Lokasi Outlet yang ramai pada *weekend* memiliki *rating* yang lebih tinggi dibanding hari biasa seperti Kursus, Rekreasi/pemandian, Stasiun Kereta Api. Sedangkan untuk Sub Lokasi Outlet lainnya memiliki *rating* yang lebih tinggi di hari biasa. Yang tidak dipengaruhi perbedaan hari adalah Sub Lokasi Outlet yang berada di Rumah Sakit dan Sekitar Pasar, hal itu dikarenakan roti menjadi kebutuhan di lokasi tersebut baik kebutuhan konsumsi keluarga maupun khusus bagi mereka yang hanya sempat mengkonsumsi roti di tengah aktivitas yang sangat padat.

Kriteria Sub Lokasi Outlet outlet lebih banyak dibandingkan yang lainnya karena informasi outlet lebih detail, tetapi bobot untuk kriteria tersebut tidak cukup banyak memberikan pengaruh tetapi tetap menjadi pertimbangan.

Tabel 3.4.8 Rating Sub Lokasi Outlet Terhadap Waktu

Pilihan	Hari Normal										
	SD / SMP / SMA	Kantor	Perumahan	Sekitar Pasar	Perkampungan	Pemandian / Rekreasi	Kursus	Stasiun Kereta Api	Universitas / Kampus	Sport Center	Rumah Sakit
<i>Growth</i>	5	4	4	5	4	3	4	4	5	3	5
<i>Constant</i>	4	3	3	4	3	4	5	5	4	5	4
<i>Taking Down</i>	2	2	2	3	3	4	3	2	2	4	2
<i>Stop Order</i>	0	0	0	0	1	1	1	0	0	0	0
Pilihan	Weekend										
<i>Growth</i>	4	3	3	5	3	5	4	5	3	3	5
<i>Constant</i>	4	5	5	4	4	4	5	4	5	5	4
<i>Taking Down</i>	3	2	2	3	5	3	3	2	3	4	2
<i>Stop Order</i>	0	0	0	0	1	0	1	0	0	0	0

Tabel tersebut menggambarkan aktivitas yang terjadi, berdasarkan *history salesman* dalam melakukan transaksi di lapangan pada waktu normal, aktivitas pembeli banyak terjadi di sekolah, perkantoran, perumahan dan perkampungan, sementara turun di *weekend*.

E. Jenis Outlet

Dalam *history sales* distribusi Sariroti, outlet yang memberikan banyak kontribusi penjualan adalah kantin dan koperasi yang ada di sekolah, kampus maupun perkantoran, ini dikarenakan adanya aktivitas yang mewajibkan pembeli bergerak lebih cepat, sementara minimarket/supermarket dan kantin yang ada di rumah sakit, dikarenakan kondisinya lebih mendukung untuk setiap pengunjung rumah sakit mengonsumsi roti.

Tabel 3.4.9 Tabel *Rating* Berdasarkan Jenis Outlet

Pilihan	Jenis Outlet				
	Kantin	Koperasi	Minimarker/ Supermarket	P&d	Warung/Toko
<i>Growth</i>	5	5	5	5	4
<i>Constant</i>	4	4	4	4	5
<i>Taking Down</i>	2	2	3	3	3
<i>Stop Order</i>	0	0	0	0	0

Rating tersebut diberikan berdasarkan kontribusi penjualan atau sering disebut *outlet pareto*, yang kontribusinya mencapai 80% dari penjualan, kemudian pembeli ekonomi menengah ke atas lebih banyak mengunjungi outlet selain warung/toko, hal tersebut menjadi sasaran penjualan Sariroti. Dan juga outlet selain toko lebih mendukung SOP pendistribusian Sariroti yaitu bersih dan *hygiene*.

Pada Metode SAW terdapat dua formula, yaitu formula dengan atribut *benefit* dan juga formula dengan atribut *cost*.

BAB IV

IMPLEMENTASI PROGRAM

4.1 Implementasi Program

Implementasi dan pengujian program merupakan tahap ketiga dari proses Metode *Waterfall*. Perancangan perangkat lunak direalisasikan sebagai perangkat program atau unit program dan akan dilakukan pengujian unit setelah sistem diimplementasikan. Sistem dalam pengujian harus diverifikasi bahwa sistem yang telah diimplementasikan memenuhi spesifikasi.

Pada tahap perancangan sistem dapat bersamaan juga dikatakan sebagai implementasi sistem, bagaimana model yang sudah dirancang diterapkan menjadi sebuah sistem agar sistem tersebut dapat beroperasi.

4.2 Spesifikasi Perangkat

Spesifikasi perangkat yang digunakan untuk implementasi sistem yang dirancang terbagi atas beberapa bagian besar, yaitu Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*).

4.2.1 Spesifikasi *Hardware*

Hardware yang akan digunakan berupa *Personal Computer (PC)*, *Smartphone Android*, *Printer Thermal POS*. Ketiga perangkat tersebut akan saling terhubung oleh sistem yang diimplementasikan. PC berfungsi sebagai alat

administrasi yang akan dioperasikan oleh admin, sedangkan *smartphone android* dioperasikan oleh salesman ketika melakukan transaksi di *outlet*.

Berikut spesifikasi PC yang akan digunakan:

1. Prosesor : Intel® Core™ i5 7200U Processor
2. Grafis : NVIDIA GeForce 940MX , with 2GB GDDR5
VRAM
3. System Storage : 128GB SATA3 M.2 SSD (Min)
4. Data Storage : 1TB 5400RPM SATA HDD
5. Keyboard
6. Mouse
7. Printer : *Printer Matrix Point*
8. Monitor : 15.6" (16:9) LED backlit FHD (1920x1080)

PC yang digunakan juga dapat berupa *Laptop* ataupun *Komputer*

Berikut spesifikasi *smartphone android* yang akan digunakan:

1. CPU : Octa-core 1.6 GHz Cortex-A53
2. RAM : 3GB
3. OS : Ice Cream Sandwich (Android 14) (Min)
4. Dimensi : 149.9 x 70.8 x 7.7 mm
5. Baterai : Li-Ion 3000 mAh (Min)
6. Jaringan : 4G LTE (Min)
7. Bluetooth : 4.2, A2DP, LE
8. Setting : *Developer Options*

Spesifikasi *Printer Thermal* POS

1. Type : *Portable*
2. Bluetooth : 4.0
3. Paper Width : 58

4.2.2 *Software*

Software yang digunakan untuk implementasi sistem yang telah dirancang adalah sebagai berikut:

1. Sistem Operasi : Windows (32 bit / 64 bit) / Linux
2. *Browser* : Firefox / Chrome
3. *Text Editor* : Notepad / Sublime
4. Tools Buliding Android : Android Studio
5. *Emulator* : Fast Amulator Android Studio
6. Microsoft Office : Microsoft Office 2003 – 2007 (Min)
7. Server Lokal : XAMPP

4.3 **Tampilan Program**

Tampilan sistem diimplementasikan berdasarkan perancangan yang telah dilakukan. Tampilan sistem terbagi atas dua model, yaitu *desktop*, dan juga *mobile*. Untuk model *desktop* akan ditampilkan melalui browser sedangkan untuk model *mobile* akan ditampilkan pada aplikasi *smartphone android*. Dalam mengimplementasikan sistem, tampilan harus mempermudah user untuk

memahami fungsi dan alur kerja dari sistem tersebut. Tampilan harus *friendly*, sehingga user tidak mengalami kejenuhan saat bekerja.

4.3.1 Tampilan *Desktop*

Sistem yang dirancang untuk tampilan desktop akan digunakan dengan menggunakan aplikasi *browser*, yaitu sebuah halaman *web*. Sesungguhnya halaman tersebut tidak hanya dapat diakses melalui *browser*, tetapi juga melalui aplikasi android yang dirancang khusus. Berikut adalah tampilan yang telah diimplementasikan untuk mengerjakan tugas-tugas yang dirancang.

1. Login

Halaman login admin adalah proses masuk ke halaman administrasi distribusi, untuk dapat mengakses berbagai tugas admin. Untuk dapat login diperlukan *username* dan *password* yang ditentukan. Pada proses login, diberikan perintah *session* menggunakan PHP, agar halaman-halaman administrasi dapat diakses berdasarkan kata kunci dari *session* yang ditentukan. Oleh sebab itu tidak akan ada *user* yang dapat mengakses halaman administrasi distribusi.



Gambar 4.1 Halaman Login

Sourcecode:

```

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT * From user WHERE username='$username' AND
password='$password'";
$result = mysqli_query($koneksi, $sql);
$row = mysqli_fetch_assoc($result);
if ($username == $row['username'] && $password == $row['password']) {
    session_start();
    $_SESSION['iduser'] = $row['idUser'];
    $_SESSION['id'] = substr($row['idUser'], 1, 4);
    $_SESSION['entity'] = "7000".substr($row['idUser'], 1, 4);
    header("location: ../dashbord.php");
}else{
    header("location: ../index.php?info=".$row['username'].");
}
}

```

Variabel `$username` dan `$password` adalah variabel untuk menyimpan username dan password milik user ketika diinput untuk *login* ke sistem. Dengan mengaktifkan fungsi *session* dan menggunakan variable *session* `$_SESSION['iduser']`, `$_SESSION['id']` dan `$_SESSION['entity']`, ketika username dan password yang diinput sesuai dengan data yang tersimpan dalam database user, maka user akan diarahkan untuk masuk ke *dashbord* melalui perintah `header("location: ../dashbord.php")` dan jika username atau password salah maka user akan diarahkan kembali ke halaman *login* melalui perintah `header("location: ../index.php?info=".$row['username'].")`.

Dalam *sourcecode* tersebut dibutuhkan konektor untuk dapat mengkoneksikan webserver dengan server database saat akan *login* maupun melakukan transaksi lainnya pada web dan juga aplikasi android, yaitu dengan menggunakan perintah sebagai berikut:

Sourcecode:

```

$servername = "localhost";
$username = "root";

```

```

$password = "";
$db = "distribusi";

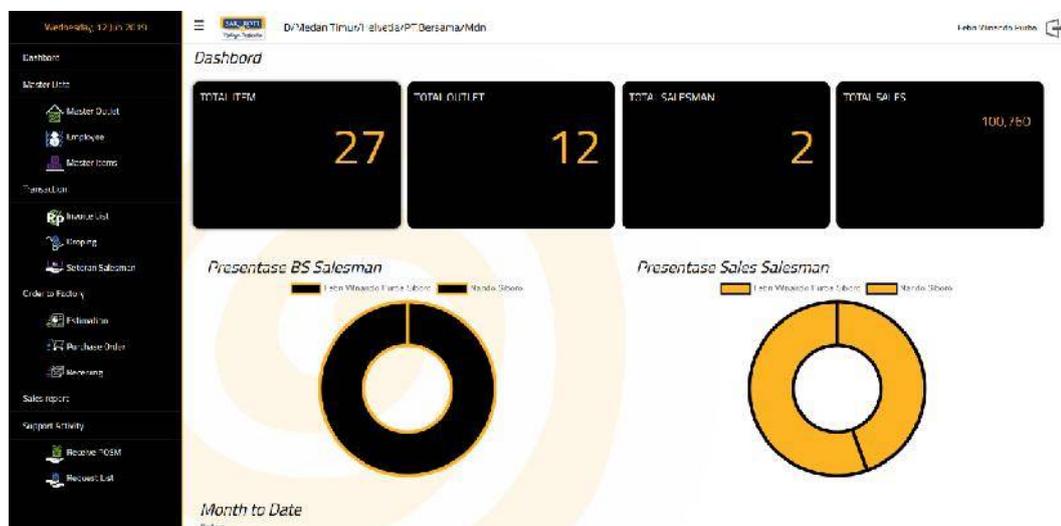
$koneksi = mysqli_connect($servername, $username, $password, "distribusi");

if (!$koneksi) {
    die("Connection failed".mysqli_connect_error());
}

```

2. Dashbord

Dashbord merupakan halaman yang menyediakan informasi umum dari seluruh transaksi salesman di outlet. Konten halaman tersebut berisi informasi item, informasi outlet, salesman, informasi sales, presentasi *sales* salesman, presentasi *bad stcok* salesman dan presentasi *Month to Date* distributor.



Gambar 4.2 Halaman Dashbord

Sourcecode :

```

if($jabatan == "Manager"){
    if($username== $row['username'] && $password ==
        $row['password']) {
        session_start();
        $_SESSION['iduser'] = $row['idUser'];
        $_SESSION['id'] = substr($row['idUser'], 1, 4);
        $_SESSION['entity'] = "7000".substr($row['idUser'], 1,
            4);
        header("location: ../dashbord.php?jabatan=manager");
    }else{
        header("location:../index.php?info=".$row['username'].");
    }
}

```

```

    }
}else{
    if ($username == $row['username'] && $password ==
    $row['password']) {
        session_start();
        $_SESSION['iduser'] = $row['idUser'];
        $_SESSION['id'] = substr($row['idUser'], 1, 4);
        $_SESSION['entity'] = "7000".substr($row['idUser'], 1,
        4);
        header("location: ../dashbord.php?jabatan=user");
    }else{
        header("location:../index.php?info=".$row['username'].");
    }
}
}

```

Sourcecode tersebut adalah perintah untuk menampilkan halaman *Dashbord* sistem informasi. Dengan kondisi tertentu, yaitu variabel jabatan, yaitu dengan nilai “Manager” atau tidak. Sementara jika *username* atau *password* tidak sesuai dengan data yang ada dalam database, maka user akan diarahkan kembali ke halaman *Login*.

3. Master Data

Pada navigasi *Master Data* berisi tiga halaman yang dapat diakses oleh admin, yaitu, *Master Outlet*, *Employee* dan Data Item. Pada ketiga halaman tersebut memiliki menu menambahkan *list* sesuai dengan isi dan fungsi halaman yang ditentukan, kecuali halaman Data Item, karena hanya dapat diubah atau ditambah atau juga di kurang oleh produsen, sesuai ketentuan dari produsen.

Master data, adalah *basic* dari semua informasi yang akan dikumpulkan melalui setiap transaksi. Pada saat menginput ataupun mengubah master data haruslah hati-hati, karena jika melakukan kesalahan, maka data selanjutnya akan salah saat melakukan transaksi.

No	Outlet	Outlet Name	Owner	Contact	Address	Outlet Type	Address	Act
1	211730001	Jaya Jaya	Jaya Jaya	81234567	Jl. Raya No. 10	Minimarket	Jaya Jaya	
2	211730002	Jaya Jaya 2	Jaya Jaya	81234568	Jl. Raya No. 11	Minimarket	Jaya Jaya	
3	211730003	Jaya Jaya 3	Jaya Jaya	81234569	Jl. Raya No. 12	Minimarket	Jaya Jaya	
4	211730004	Jaya Jaya 4	Jaya Jaya	81234570	Jl. Raya No. 13	Minimarket	Jaya Jaya	
5	211730005	Jaya Jaya 5	Jaya Jaya	81234571	Jl. Raya No. 14	Minimarket	Jaya Jaya	
6	211730006	Jaya Jaya 6	Jaya Jaya	81234572	Jl. Raya No. 15	Minimarket	Jaya Jaya	
7	211730007	Jaya Jaya 7	Jaya Jaya	81234573	Jl. Raya No. 16	Minimarket	Jaya Jaya	
8	211730008	Jaya Jaya 8	Jaya Jaya	81234574	Jl. Raya No. 17	Minimarket	Jaya Jaya	
9	211730009	Jaya Jaya 9	Jaya Jaya	81234575	Jl. Raya No. 18	Minimarket	Jaya Jaya	
10	211730010	Jaya Jaya 10	Jaya Jaya	81234576	Jl. Raya No. 19	Minimarket	Jaya Jaya	

Gambar 4.3Halaman Master Data Outlet

Sourcecode 1:

```
<li class="nav-item">
  <a class="nav-link active"
    href="dashbord.php?isi=masteroutlet">
    Master Outlet
  </a>
</li>
```

Sourcecode 2:

```
elseif($isi="masteroutlet"){
  include 'master/outlet.php';
}
```

Sourcecode 1 sebuah *link* untuk mengakses halaman pada *dashbord* yaitu dengan konten yang ditambahkan berdasarkan variable yang dikirim melalui *link* dengan nilai "masteroutlet".



Gambar 4.4 Halaman *Employee*

Sourcecode 1:

```
<li class="nav-item">
  <a class="nav-link active" href="dashbord.php?isi=employee">
    Employee
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "employee") {
    include "master/employee.php";
}
```

Perintah untuk menampilkan halaman *employee* memiliki perintah yang serupa dengan menampilkan tabel item roti dan master outlet. Untuk pemanggilan konten *employee* pada *dashbord* diberikan nilai pada variabel dengan nilai "employee".

No	Item ID	Item Name	Unit Name	Price
1	PS 01 001	ROTI MANIS KUNYIT	KG	8000000
2	PS 01 002	SEMOGATON COLELAT	KG	4500000
3	PS 01 003	MANIS KUNYIT KAJANG	KG	4500000
4	PS 01 004	MANIS KUNYIT KAJANG	KG	4500000
5	PS 01 005	SEMOGATON MANIS KAJANG	KG	4500000
6	PS 01 006	ROTI TAPAK SITESALE	KG	4500000
7	PS 01 007	ROTI TAPAK SITESALE	KG	4500000
8	PS 01 008	ROTI SEMOGATON KAJANG	KG	4500000
9	PS 01 009	ROTI SEMOGATON KAJANG	KG	4500000
10	PS 01 010	ROTI MANIS KUNYIT KAJANG	KG	4500000
11	PS 01 011	ROTI TAPAK SITESALE	KG	4500000
12	PS 01 012	ROTI TAPAK SITESALE	KG	4500000
13	PS 01 013	ROTI TAPAK SITESALE	KG	4500000
14	PS 01 014	ROTI TAPAK SITESALE	KG	4500000
15	PS 01 015	ROTI TAPAK SITESALE	KG	4500000
16	PS 01 016	ROTI TAPAK SITESALE	KG	4500000
17	PS 01 017	ROTI TAPAK SITESALE	KG	4500000
18	PS 01 018	ROTI TAPAK SITESALE	KG	4500000
19	PS 01 019	ROTI TAPAK SITESALE	KG	4500000
20	PS 01 020	ROTI TAPAK SITESALE	KG	4500000
21	PS 01 021	ROTI TAPAK SITESALE	KG	4500000
22	PS 01 022	ROTI TAPAK SITESALE	KG	4500000
23	PS 01 023	ROTI TAPAK SITESALE	KG	4500000
24	PS 01 024	ROTI TAPAK SITESALE	KG	4500000
25	PS 01 025	ROTI TAPAK SITESALE	KG	4500000
26	PS 01 026	ROTI TAPAK SITESALE	KG	4500000
27	PS 01 027	ROTI TAPAK SITESALE	KG	4500000
28	PS 01 028	ROTI TAPAK SITESALE	KG	4500000
29	PS 01 029	ROTI TAPAK SITESALE	KG	4500000
30	PS 01 030	ROTI TAPAK SITESALE	KG	4500000
31	PS 01 031	ROTI TAPAK SITESALE	KG	4500000
32	PS 01 032	ROTI TAPAK SITESALE	KG	4500000
33	PS 01 033	ROTI TAPAK SITESALE	KG	4500000
34	PS 01 034	ROTI TAPAK SITESALE	KG	4500000
35	PS 01 035	ROTI TAPAK SITESALE	KG	4500000
36	PS 01 036	ROTI TAPAK SITESALE	KG	4500000
37	PS 01 037	ROTI TAPAK SITESALE	KG	4500000
38	PS 01 038	ROTI TAPAK SITESALE	KG	4500000
39	PS 01 039	ROTI TAPAK SITESALE	KG	4500000
40	PS 01 040	ROTI TAPAK SITESALE	KG	4500000
41	PS 01 041	ROTI TAPAK SITESALE	KG	4500000
42	PS 01 042	ROTI TAPAK SITESALE	KG	4500000
43	PS 01 043	ROTI TAPAK SITESALE	KG	4500000
44	PS 01 044	ROTI TAPAK SITESALE	KG	4500000
45	PS 01 045	ROTI TAPAK SITESALE	KG	4500000
46	PS 01 046	ROTI TAPAK SITESALE	KG	4500000
47	PS 01 047	ROTI TAPAK SITESALE	KG	4500000
48	PS 01 048	ROTI TAPAK SITESALE	KG	4500000
49	PS 01 049	ROTI TAPAK SITESALE	KG	4500000
50	PS 01 050	ROTI TAPAK SITESALE	KG	4500000

Gambar 4.5 Halaman Daftar Item Roti

Sourcecode 1:

```
<li class="nav-item">
  <a class="nav-link active" href="dashbord.php?isi=item">
    Master Items
  </a>
</li>
```

Sourcecode 2:

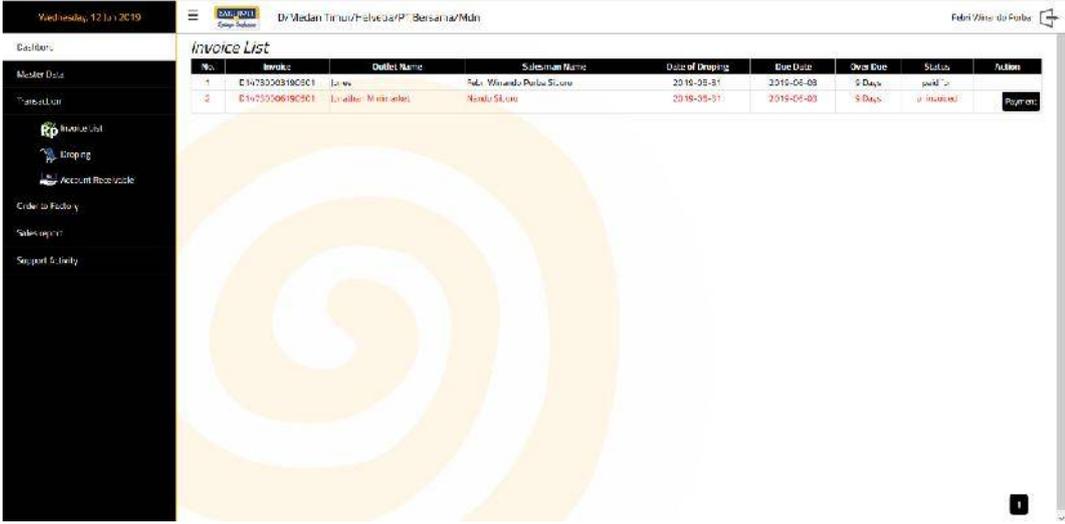
```
elseif ($isi == "item") {
  include "master/itemroti.php";
}
```

Pada saat menampilkan data pada tabel item roti, tabel tidak diberikan batasan baris untuk ditampilkan, karena data pada tabel roti tidak banyak. Pemanggilan konten item pada *dashbord* menggunakan variable dengan nilai “item”.

4. *Transaction*

Halaman Navigasi *Transaction* terbagi atas tiga halaman yaitu, *List Invoice*, *Dropping*, dan *Account Receivable*. *List Invoice* merupakan halaman yang berisikan tabel daftar outlet yang melakukan transaksi, baik transaksi yang telah melakukan pembayaran ataupun belum melakukan pembayaran. *Dropping*, merupakan halaman transaksi salesman, yang diinput oleh admin, jika terjadi kendala dilapangan, saat aplikasi mobile android tidak berfungsi. Pada

halaman tersebut terdapat proses transaksi salesman yaitu, *Salesman Estimastion*, *Dropping*, *Account Receivabe* dan *Support Activity* serta proses aktivitas outlet yang meliputi *check in dan check out* terhadap outlet di sistem. Kemudian *Account Receivable* adalah table daftar tagihan yang dilakukan salesman terhadap outlet yang telah melakukan transaksi pada kunjungan sebelumnya.



No	Invoice	Outlet Name	Salesman Name	Date of Dropping	Due Date	Over Due	Status	Action
1	E1475250819C2C1	Larve	Sd. Yulianto, Parla, Siware	2018-08-31	2018-08-08	9 Days	paid	
2	E1475250819C2C1	Larve/ Miminanai	Nando Sitara	2018-08-31	2018-08-08	9 Days	un-invoiced	Payment

Gambar 4.6 Halaman *Invoice List*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?isi=tagihan">
    Invoice List
  </a>
</li>
```

Sourcecode 2:

```
if ($isi == "transaksi") {
  include "transaksi/tagihan.php";
}
```

Perintah menampilkan konten *Invoice List* pada *Dashbord* yaitu dengan menggunakan variabel bernilai “transaksi”. Jika variabel bernilai “transaksi”, maka, akan dibuka pada konten *dashbord* ling tagihan pada *folder* tagihan.



No.	Outlet ID	Outlet Name	Suboutlet ID	Suboutlet	Action
1	211730003	jemec	314750001	Fabri Winda Purba Sibero	
2	211730004	lubi	314750001	Fabri Winda Purba Sibero	
3	211730005	gapan	314750001	Fabri Winda Purba Sibero	
4	211730006	Sambing	314750001	Fabri Winda Purba Sibero	
5	211730007	Santhan Marambat	314750002	Nurd. Sibero	
6	211730011	Solan Toka	314750002	Nurd. Sibero	
7	211730012	Santhan Marambat	314750007	Nurd. Sibero	
8	211730017	Santhan Marambat	314750002	Nurd. Sibero	

Gambar 4.7 Halaman *Dropping*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?isi=visit">
    Dropping
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "visit") {
  include "transaksi/visit.php";
}
```

Dengan perintah pada *sourcecode 1* dan *2*, konten *Dropping* dapat ditampilkan pada halaman *Dashbord*. Nilai variabel untuk memanggil *link* tersebut adalah “visit”.

No.	AB ID	Salesman ID	Salesman	Receivable Date	Total Outlet	Total Receiver (Rp)
1	912727800001	2142388001	Fachri Alfaridzi Kurniadi, Bero	2019-06-01	1	75.700

Gambar 4.8 Halaman *Account Receivable*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?isi=setoran">
    Account Receivable
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "setoran") {
  include "transaksi/setoransalesman.php";
}
```

Konten *Account Receivable* dapat diakses menggunakan *sourcecode 1*, dengan nilai variabel “setoran”. Jika nilai pada variable `$isi`, bernilai “setoran”, maka user akan diarahkan pada halaman `setoransalesman.php` yang ditampilkan pada konten *dashbord*. Daftar tagihan salesman hanya akan terbentuk ketika user melakukan penerimaan. Karena itu ada perintah penambahan dengan menggunakan *button* dengan kode perintah `button`, sehingga terbuka *.pop up* yang terdapat form daftar salesman yang melakukan penagihan ke outlet.

5. Order to Factory

Halaman Navigasi *Order Victory* adalah transaksi yang berhubungan langsung dengan produsen, yaitu proses pemesanan roti kepada produsen yang akan dikirim berdasarkan tanggal pengiriman yang ditentukan oleh mitra distribusi. Pada navigasi *Order to Vactory* mencakup tiga proses yang dibagi pula ke dalam tiga halaman transaksi pemesanan roti, yaitu *Estimation Salesman*, *Purchase Order* dan *Receiving*.

Halaman *Estimation Salesman* adalah halaman yang berfungsi untuk salesman melakukan estimasi pemesanan untuk kunjungan selanjutnya. Salesman melakukan estimasi berdasarkan outlet yang akan dikunjungi minimal H-2 untuk kedatangan roti ke distributor dan juga estimasi dapat dilakukan mengikuti rekomendasi berdasarkan yang diperhitungkan menggunakan Metode SAW untuk meningkatkan keuntungan berdasarkan Kriteria yang telah ditentukan.

Halaman *Purchase Order* merupakan halaman, dimana setiap *Sales Order* yang telah di-*approve* oleh salesman untuk dipesan ke produsen dan digabung ke dalam satu pemesanan ke produsen. *Purchase Order* yang belum di-*approve* dapat di-*adjustment* berdasarkan perubahan pemesanan oleh distributor ke produsen.

Sedangkan untuk halaman *Receiving* adalah halaman input penerimaan roti yang masuk ke distributor berdasarkan pemesanan yang dilakukan sebelumnya. *Receiving* memiliki *Delivery Note* (DN) sebagai identitas penerimaan roti, dimana surat pengiriman terbagi atas dua lembar. Salah satu lembar tersebut akan menjadi bukti penerimaan kepada distributor dan salah satunya juga menjadi bukti telah terkirimnya roti ke distributor dan kembali ke

produsen sebagai buku tagihan yang akan dilakukan ke distributor. Oleh sebab itu dalam penginputa penerimaan, admin harus menginput data sesuai dengan DN yang telah diterima. Pada halaman *Receiving* juga roti yang telah diterima oleh admin harus dialokasi ke salesman berdasarkan rencana estimasi tanggal kedatangan roti yang dilakukan saat melakukan pemesanan roti.

No.	SO Number	Salesman	Expected Delivery	Total Orders	Qty (kg)	GDP (Rp)	Create Estimation
1	41473190801000	Fedy Winando Purba Sibero	2019-05-30	1	11	101.000,00 PO Created	
2	41473190801000	Fedy Winando Purba Sibero	2019-05-31	3	18	44.000,00 PO Created	
3	41473190801000	Nando Sibero	2019-05-31	2	11	35.000,00 PO Created	
4	41473190801000	Nando Sibero	2019-05-05	1	2	52.500,00	

Gambar 4.9 Halaman *Sales Order*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?jabatan=<?php
  echo $jabatan; ?>&isi=salesorder">
  Estimation
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "salesorder") {
  include "transaksi/salesorder.php";
}
```

Setiap baris dari tabel halaman *Sales Order*, memiliki form masing-masing untuk dapat melakukan *editing*. Untuk setiap nomor *Sales Order* yang telah disetujui

atau approve, perintah *approve*, *delete* dan *edting* akan menghilang, sehingga hanya terdapat perintah pembatalan.

No.	Status	PO no.	Type PO	Expected Delivery	Total Outlets	Qty (po)	CBP (Rp)	Detail
1	Terlirim	51473100310002	Regular	2019-05-31	Jumlah Outlet	30	340.000,00	
2	Terlirim	51473100320002	Regular	2019-05-30	Jumlah Outlet	12	101.000,00	

Gambar 4.10 Halaman *Purchase Order*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?jabatan=?php
    echo $jabatan; ?>&isi=po">
    Purchase Order
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "po") {
  include "transaksi/po.php";
}
```

Nilai pada variabel `$isi` bernilai “po”, sehingga saat pemanggilan *Dashbord* pada nilai variabel tersebut, maka akan terbuka konten *Purchase Order*.

No.	Status	No. DN	No. PO	Receiving Date	Qty (pcs)	CRR (Rp)	Detail
1	Termin	456	B14731902810002	2019-05-21	30	210.000.00	
2	Belanja	121	47671002510002	2019-05-21	12	11.000.000	

Gambar 4.11 Halaman *Receiving*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?jabatan=?php
  echo $jabatan; ?>&isi=receiving">
  Receiving
  </a>
</li>
```

Sourcecode 2:

```
elseif ($isi == "receiving") {
  include "transaksi/receiving.php";
}
```

Konten *Receiving* pada *Dashbord* dipanggil dengan memberikan nilai “receiving” pada variabel `$isi`. Ketika variabel tersebut bernilai “receiving” saat pemanggilan, maka halaman `receiving.php` pada folder `transaksi` akan ditampilkan.

6. Sales Report

Navigasi *Sales Report*, akan menampilkan halaman yang berisikan daftar *report*. Halaman *Sales Report* berfungsi untuk memberikan informasi detail tentang

transaksi yang dilakukan bahkan sampai saat terakhir melakukan transaksi baik di outlet maupun transaksi penyerahan uang tagihan dari salesman ke admin.

The screenshot shows a web application interface. On the left is a dark sidebar with navigation links: 'Home', 'Sales Report', and 'Support Activity'. The main content area is titled 'Report Salesman Estimation' and contains a table with the following columns: 'No', 'Salesman ID', 'Sales ID', 'Sales Name', 'Salesman', 'Item ID', 'Item Name', 'Qty', 'Unit Price', 'Total Price', and 'Status'. The table lists 23 rows of sales data, including salesmen like 'Ika Winda Prita Sari' and 'Ika Winda Prita Sari', and items like 'SABUN LIU - 300 ML' and 'SABUN LIU - 150 ML'. The right sidebar contains a 'Report List' section with links for 'Sales Report', 'Sales Order Report', and 'Sales Report'.

Gambar 4.12 Halaman Report

Sourcecode 1:

```
<a href="dashbord.php?isi=report">
  <strong>Sales report</strong>
</a>
```

Sourcecode 2:

```
elseif ($isi == "report") {
  include "report/report.php";
}
```

Konten *Report* tidak memiliki navigasi khusus untuk daftar *report* seperti tugas-tugas lainnya. Tetapi di dalam konten tersebut terdapat daftar *report* yang dapat digunakan sesuai dengan kebutuhan.

7. Support Activity

Navigasi *Support Activity* merupakan halaman penerimaan dan pemberian *Point Of Sales Material* (POS�). Daftar item POS� tidak dapat ditambah ataupun dikurang, hal tersebut dikarenakan POS� hanya masuk berdasarkan rencana

ataupun program yang ditentukan oleh produsen. Outlet ataupun salesman dapat *me-request* POSM jika sesungguhnya diperlukan untuk digunakan di outlet.



Gambar 4.13 Halaman *Receiving POSM*

Sourcecode 1:

```
<li>
  <a class="nav-link active" href="dashbord.php?jabatan=<?php
  echo $jabatan; ?>&isi=posm">
  Receive POSM
  </a>
</li>
```

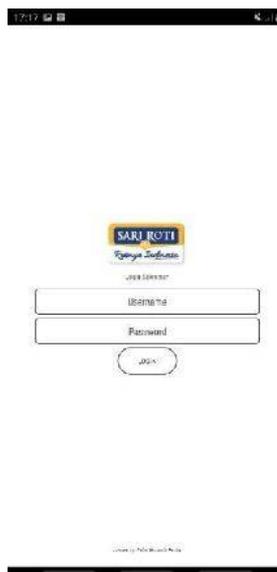
Sourcecode 2:

```
elseif ($isi == "posm") {
  include "transaksi/receivingposm.php";
}
```

Konten Penerimaan atau *Receive POSM* dapat dipanggil dengan pemberian nilai “posm” pada variable `$isi`. Pada *Sourcecode 2*, jika variabel tersebut bernilai sama dengan “posm”, maka akan dibuka halaman `receivingposm.php`.

1. Login Salesman

Pada aplikasi distribusi Sari Roti, salesman harus melakukan login terlebih dahulu sebelum melakukan transaksi. Setiap salesman harus menjaga kerahasiaan user masing-masing, dikarenakan setiap transaksi yang terjadi adalah menjadi tanggung jawab dari user yang melakukan transaksi.



Gambar 4.15 Halaman *Login Salesman* Tampilan *Mobile*

Sourcecode:

```
Call<ResponseLogin> sendData = api.sendUser(stringUsername,
stringPassword);
sendData.enqueue(new Callback<ResponseLogin>() {
    @Override
    public void onResponse(Call<ResponseLogin> call,
        Response<ResponseLogin> response) {

        progressDialog.hide();
        Log.d("Retro", "response: " +
response.body().toString());
        idSalesman = response.body().getIdSalesman();
        nmSalesman = response.body().getNmSalesman();
        usernameSalesman =
response.body().getUsernameSalesman();
        String info = response.body().getInfo();
        String kode = response.body().getKode();

        if (kode.equals("1")){
            Toast.makeText(LoginActivity.this, info,
```

```

        Toast.LENGTH_LONG).show();
        Intent intent = new Intent(getApplicationContext(),
        MainActivity.class);
        startActivity(intent);

    }else{
        Toast.makeText(LoginActivity.this, info,
        Toast.LENGTH_LONG).show();
    }
}

@Override
public void onFailure(Call<ResponseLogin> call, Throwable t)
{
    progressDialog.hide();
    Log.d("Retro", "Failur: "+"Gagal mengirim Request");
}
});

```

2. Home

Pada halaman *Home* aplikasi android, tertera nama salesman sebagai informasi bahwa user yang diakses telah sesuai dengan data salesman, tanggal saat ini, dan juga informasi khusus yang akan ditampilkan jika ada program yang diadakan oleh produsen.



Gambar 4.16 Halaman *Home Salesman* Tampilan *Mobile*

Sourcecode:

```

CardView listOutletVisit =
(CardView)view.findViewById(R.id.btnVisit);
listOutletVisit.setOnClickListener(this);

CardView noo = (CardView)view.findViewById(R.id.btnNoo);
noo.setOnClickListener(this);

CardView stockSalesman =
(CardView)view.findViewById(R.id.btnStock);
stockSalesman.setOnClickListener(this);

CardView estimasi =
(CardView)view.findViewById(R.id.btnEstimasi);
estimasi.setOnClickListener(this);

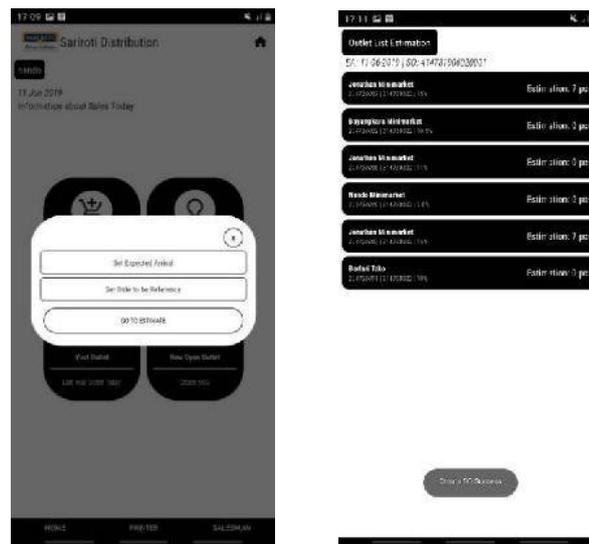
```

Pada *Home* terapat tiga menu utama yang dikerjakan oleh salesman, yaitu *Estimation*, *Outlet Visit*, *New Open Outlet* (NOO) dan juga check stok roti yang ada pada gudang salesman. Menu *Estimation* berfungsi untuk salesman mengisi estimasi pemesanan roti, *Outlet Visit* berfungsi untuk mengarahkan salesman melakukan kunjungan outlet, berdasarkan Jadwal Kunjungan (JWK), menu *Stock* berfungsi untuk menampilkan *stock* yang ada di gudang salesman, dan juga untuk melakukan transfer item roti, terakhir menu NOO berfungsi untuk membuat outlet baru.

3. Estimasi

Halaman Estimasi dapat diakses selemas melalui menu *Estimastion* yang ada pada halaman *Home*. Terlebih dahulu salesma wajib menginput tanggal kedatangan pada kolom *Set Expected Arrival* dan juga tanggal referensi untuk *suggest order* pada kolom *Set Date to be Reference*.

Halaman estimasi pada *mobile* memerlukan *Date Reference*, yang menjadi acuan untuk data *suggest* pada metode SAW. Penginputan data tersebut wajib dilakukan, sehingga sistem dapat memberikan *suggest qty* pada saat salesman melakukan estimasi per item dan per outlet.



(a)

(b)

Gambar 4.17 (a)Halaman *Insert Expected Arrival* dan (b) Halaman *Outlet List* Tampilan *Mobile*

Sourcecode:

```
if (kode.equals("1")) {
    Toast.makeText(getApplicationContext(), info,
    Toast.LENGTH_LONG).show();
    Intent intent = new Intent(getApplicationContext(),
    ListOutletEstimasiActivity.class);
    startActivity(intent);
} else {
    Toast.makeText(getApplicationContext(), "gagal",
    Toast.LENGTH_LONG).show();
}
```

Jika salesman telah melanjutkan estimasi, maka salesman akan diarahkan ke datar outlet yang akan dikunjungi berdasarkan tanggal kedatangan roti atau *Expepcted Arrival*. Salesman akan diarahkan kepada halaman estimasi item berdasarkan outlet yang dipilih untuk diinput estimasi itemnya.

4. Outlet Visit

Halaman *Outlet Visit* dapat diakses melalui menu *Outlet Visit* yang tertera pada *Home*. Halaman *Outlet Visit* berisi daftar outlet berdasarkan JWK tanggal tersebut yang telah dilakukan estimasi sebelumnya. Ketika salesman melakukan visit,

salesman hanya dapat membuka outlet pada daftar *visit*, karena ketika salesman membuka list tersebut, maka salesman melakukan *check in* pada outlet.

Halaman Outlet terdapat nama outlet yang dikunjungi dan tanggal kunjungan pada saat itu, terdapat tiga menu, yaitu *Claim*, *Dropping*, *Estimation* dan *Support Activity*, terdapat keterangan dari outlet tersebut.



Gambar 4.18 Halaman Outlet *Visit* Tampilan *Mobile*

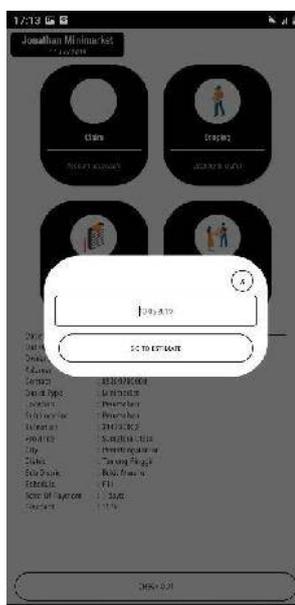
Sourcecode:

```
public void onClick(View v) {
    switch (v.getId()){
        case R.id.btnDroppingVisit:
            droppingVisit();
            break;
        case R.id.btnEstimateOutletVisit:
            estimasiVisit();
            break;
        case R.id.btnAROutlet:
            piutangOutlet();
            break;
        case R.id.btnSupport:
            supportActivity();
            break;
    }
}
```

Ketika salesman selesai melakukan transaksi pada outlet, salesman wajib melakukan *check out* pada aplikasi. Jika salesman tidak melakukan *check out* ketika akan melanjutkan kunjungan ke outlet selanjutnya, maka salesman akan terdata memiliki waktu yang cukup lama ketika melakukan transaksi, dan hal tersebut dapat menjadi pertanyaan bagi salesman.

5. Estimasi on Visit

Halaman *Estimasi on Visit*, memiliki fungsi dan cara kerja yang sama dengan menu *Estimation* di Home, hanya perbedaannya adalah tanggal *Expected Arrival* ditentukan berdasarkan JKW outlet, oleh sebab itu tidak diperlukan input tanggal *Expected Arrival*.



Gambar 4.19 Halaman *Insert Suggest Date* Tampilan Mobile

Sourcecode:

```

date1
= (EditText) suggestDialog.findViewById(R.id.etReferenceDateSuggest);

showPopUpEA = (Button)
suggestDialog.findViewById(R.id.btnSuggest);

```

```

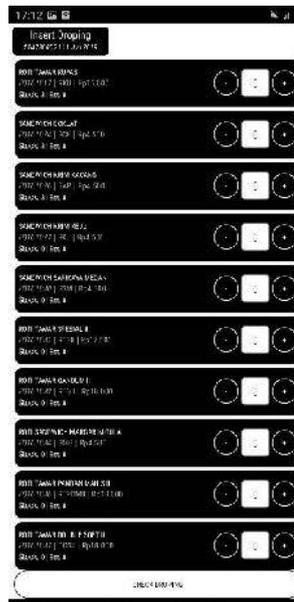
txtClose.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        suggestDialog.dismiss();
    }
});

date1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showDateDialogReference();
    }
});

```

6. Dropping

Menu *Dropping* pada halaman Outlet adalah menu untuk melakukan *drop* roti pada outlet tersebut. Dropping roti dapat dilakukan berdasarkan estimasi pemesanan atau juga dapat dilakukan *adjustment* sesuai dengan kebutuhan.



Gambar 4.20 Halaman *Insert Kuantitas Dropping* Tampilan *Mobile*

Sourcecode:

```

public void droppingVisit() {
    Intent intent = new Intent(OutletActivity.this,

```

```

DroppingActivity.class);
    droppingDate =

DateFormat.getDateInstance(DateFormat.DEFAULT).format(calendar.g
etTime());
    startActivity(intent);
}

```

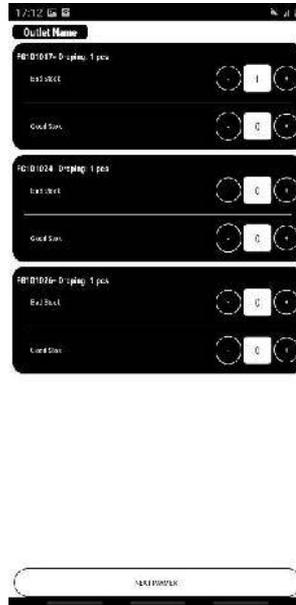
Pada halaman *dropping* tertera daftar item dan juga keterangan item tersebut, informasi estimasi dan juga stok salesman yang ada pada saat itu. Pada daftar item terdapat kolom input data dan juga tombol kurang dan tambah, untuk mengisi kuantitas item yang akan di drop pada outlet. Ketika rencana dropping item telah selesai, salesman harus melakukan *Check Dropping* melalui tombol *CHECK DROPPING* yang ada pada halaman *Dropping*. Sehingga tidak terjadi kesalahan sebelum melakukan dropping.

Pada halaman *Check Dropping* salesman dapat melihat rencana dropping item yang akan di *drop* ke outlet beserta dengan total secara kuantiti maupun *value*. Ketika salesman akan melakukan perubahan rencana, salesman dapat kembali kepada halaman *Dropping* untuk mengubah kuantitas dropping. Jika data telah sesuai dengan fisik yang akan di *drop* maka salesman harus terlebih dahulu mencetak *invoice* sebanyak dua rangkap, *invoice* akan menjadi bukti transaksi *dropping* bagi outlet dan juga distributor untuk penagihan selanjutnya. Setelah itu, maka salesman dapat melakukan penyimpanan data pada sistem.

7. Account Receivable

Halaman *Account Receivable* terdapat daftar tagihan dari outlet yang dikunjungi oleh salesman, sebagai keterangan tagihan maka pada daftar tagihan ditampilkan

berdasarkan invoice yang belum ditagih dan juga belum lunas atau belum melakukan pembayaran tetapi telah dilakukan penagihan.



Gambar 4.21 Halaman *Account Receivable* Tampilan *Mobile*

Sourcecode:

```
public void piutangOutlet() {
    Intent intent = new Intent(OutletActivity.this,
        ListTagihanActivity.class);

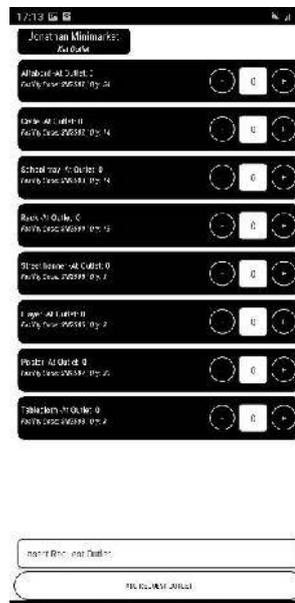
    startActivity(intent);
}
```

Halaman Tagihan item salesman berisi daftar item yang di dropping pada kunjungan sesuai dengan *invoice*, kemudian pada setiap item terdapat baris *Bad Stock* dan *Good Stock*. Pada proses penagihan yang dapat dihitung adalah roti yang tersisa di outlet untuk melihat berapa total roti yang telah terjual. Penginputan dapat dilakukan melalui kolom kuantiti dan juga melalui tombol tambah atau kurang. Barang sisa tersebut harus diinput berdasarkan kondisi roti, apakah roti masih dapat *resufle* atau telah kadaluarsa. Jika roti telah kadaluarsa

maka, kuantiti roti sisa tersebut harus dimasukkan pada kolom *Bad Stock* , dan jika roti masih dapat di dropping ulang atau *resufle* maka roti harus dimasukkan ke dalam kolom *Good Stock*.

8. Support Activity

Menu *Support Activity* berisikan daftar POSM yang terdaftar di produsen. Jika pada kuantiti POSM, maka salesman tidak dapat melakukan *request* item. Pada halaman *Support Activity*, outlet dapat memberikan masukan kepada distributor mengenai kebutuhan outlet.



Gambar 4.22 Halaman *Support Activity* Tampilan *Mobile*

Sourcecode:

```
public void supportActivity() {
    Intent intent = new Intent(OutletActivity.this,
    SupportOutlet.class);
    droppingDate =
    DateFormat.getDateInstance(DateFormat.YEAR_FIELD).format(calenda
    r.getTime());
    startActivity(intent);
}
```

Pada halaman *request*, saat salesman melakukan *request* dengan menekan tombol SEND REQUEST OUTLET, halaman akan diarahkan pada halaman Outlet, tidak perlu melakukan pengecekan ulang.

9. NOO (*New Open Outlet*)

Halaman *form* NOO, berfungsi untuk menambahkan outlet baru ke distributor.

Halaman tersebut dapat diakses melalui menu NOO di halaman *Home*.

Gambar 4.23 Halaman NOO Tampilan *Mobile*

Sourcecode:

```
public void btnVisit() {
    Intent intent = new Intent(getActivity(),
    ListOutletVisitActivity.class);
    startActivity(intent);
}
```

Halaman *form* NOO berisikan komlom input *text*, yaitu *Outlet Name*, *Outlet Address*, *Owner*, *Contact*, *Outlet Type*, *Outlet Location*, *Sub Location*, *Province*, *City*, *District*, *Sub District*, *Salesman*, *Visit Plan*, *Term of Payment*, dan *Discount*.

Seluruh *form* tersebut harus diisi dengan benar, karena data yang diinput sangat berguna digunakan.

10. Stock Salesman

Halaman *Stock Salesman* diakses melali menu *Stock* pada halaman *Home*. Halaman *Stock Salesman* terdapat daftar item dan kuantiti item yang ada di gudang salesman.



Gambar 4.24 Halaman *Stock Salesman* Tampilan *Mobile*

Sourcecode:

```
public void btnStock() {
    Intent intent = new Intent(getActivity(),
    StockSalesmanAcitvity.class);
    startActivity(intent);
}
```

Pada halaman *Stock Salesaman*, salesman dapat melakukan transfer item ke salesman lain. Dengan menekan tombol *TRANSFER ITEM*, salesman akan diarahkan kepada halaman transfer. Salesman dapat menentukan item dan salesman yang dituju.

BAB V

PENUTUP

5.1 Kesimpulan

Sistem dapat memaksimalkan kerja operasional distributor sebagai mitra kerja dari PT. Nippon Indosari mulai dari transaksi salesman dilapangan sampai setoran uang ke admin kasir.

Sistem yang dirancang dapat membantu salesman untuk dapat melakukan pengambilan keputusan yang tepat saat melakukan pemesanan, dengan memberikan data rekomendasi pemesanan, dengan menggunakan menggunakan Metode SAW atribut keuntungan.

5.2 Saran

Sistem dapat diintegrasikan dengan sistem *System Application and Processing* (SAP) sebagai software internal PT. NIC, Tbk. Agar data dapat langsung diterima oleh produsen dan diproses melalui SAP dengan satu langkah.

Sistem yang dirancang untuk digunakan oleh salesman dilapangan yaitu aplikasi android, dapat menyimpan data di *temporaary storage*, sehingga jika terjadi kegagalan jaringan dapat dilakukan secara *offline*.

DAFTAR PUSTAKA

- Fuad, R. N., & Winata, H. N. (2017). Aplikasi keamanan file audio wav (waveform) dengan terapan algoritma RSA. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(2), 113-119.
- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Hafni, Layla, and Rismawati Rismawati. "Analisis faktor-faktor internal yang mempengaruhi nilai perusahaan pada perusahaan manufaktur yang terdaftar di BEI 2011-2015." *Bilancia: Jurnal Ilmiah Akuntansi* 1.3 (2017): 371-382.
- Hamdi, Muhammad Nurul, Evi Nurjanah, and Latifah Safitri Handayani. "Community development based on Ibnu Khaldun thought, sebuah interpretasi program pemberdayaan UMKM di bank zakat el-zawa." *EL MUHASABA: Jurnal Akuntansi (e-journal)* 5.2 (2014): 158-180.
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode MFEP pada CV. Sapo Durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Hutchinson, Sawyer. 1996. *Computers and Information Systems*. United States of America: Richard D. Darwin, a Times Mirror Higher Education Group, Inc.
- Indra Permana, A. M. I. N. U. D. D. I. N. "Sistem Pakar Mendeteksi Hama dan Penyakit Tanaman Kelapa Sawit pada PT. Moeis Kebun Sipare-pare Kabupaten Batubara." (2013).
- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Kendal dan Kendal. 2002. *Analisis dan Perancangan Sistem*. Jakarta: Pearson Education Asia Pte. Ltd. Dan PT. Prenhallindo

- Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 279-284.
- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Ladjamudin, Al. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu
- Mahyuddin, 1994. *Pengantar Sistem Pengelolaan Basis Data*. Medan: USU Press
- Mariance, U. C. (2018). Analisa dan Perancangan Media Promosi dan Pemasaran Berbasis Web Menggunakan Work System Framework (Studi Kasus di Toko Mandiri Prabot Kota Medan). *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(1).
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." *Int. J. Recent Trends Eng. Res* 2.12 (2016): 140-151.
- Nazir, Moh. 2011. *Metode Penelitian*. Bogor: Ghalia Indonesia
- Ramakrishnan dan Gehrke. 2003. *Sistem Manajemen Database*. Yogyakarta: ANDI
- Somerville, Ian. 2011. *Software Enginaring*. United States of America: Pearson Education, Inc.
- Suryanto, Mikael. 2016. *Sistem Operasional Manajemen Distribusi*. Jakarta: Grasindo
- Developers. 2018. Mengenal Android Studio.
https://developer.android.com/studio/intro/?hl=id#antarmuka_pengguna
- Siaran Pers. 2018. *Empat Strategi Indonesia Masuk Revolusi Industri Keempat*.
<http://www.kemenperin.go.id/artikel/17565/Empat-Strategi-Indonesia-Masuk-Revolusi-Industri-Keempat>
- Turban, dkk. 2005. *Decision Support System and Intelligent Systems*. New Jersey: Pearson Education, Inc. Upper Saddle River
- Wikipedia. 2018. *Cascading Style Sheets*.
https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- W3School.com. 2018. *CSS Introduction*.
https://www.w3schools.com/css/css_intro.asp