

**RANCANG BANGUN APLIKASI PENJUALAN DAN  
*INVENTORY MINIMARKET* DENGAN MENGGUNAKAN  
*SHORTEST REMAINING TIME FIRST ALGORITHM***

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer**



**CHANDRA SETIAWAN GIMON**

**1112001043**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS BAKRIE  
JAKARTA  
2016**

## HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Chandra Setiawan Gimon  
NIM : 1112001043  
Program Studi : Informatika  
Fakultas : Teknik dan Ilmu Komputer  
Judul Skripsi : Rancang Bangun Aplikasi Penjualan & *Inventory*  
*Minimarket* Dengan Menggunakan *Shortest*  
*Remaining Time First Algorithm*.

Telah disetujui oleh pembimbing tugas akhir untuk diajukan ke seminar tugas akhir.

Jakarta, Agustus 2016

Menyetujui,

Pembimbing Tugas Akhir,

Yusuf Lestanto, S.T., M.Sc.

**RANCANG BANGUN APLIKASI PENJUALAN DAN INVENTORY  
MINIMARKET DENGAN MENGGUNAKAN *SHORTEST REMAINING  
TIME FIRST ALGORITHM***

Chandra Setiawan Gimon

---

**ABSTRAK**

Semakin banyaknya bisnis waralaba swalayan yang menjual berbagai jenis barang keperluan sehari-hari memberikan dampak negatif terhadap toko-toko yang berada disekitarnya sehingga mengakibatkan sepiya pelanggan dan penumpukan barang dagangan. Maka harus dilakukan proses perhitungan ulang terhadap stok barang yang ada. Namun terkadang mengalami kesulitan jika harus melakukan proses pengecekan secara manual. Berdasarkan masalah tersebut diperlukan aplikasi *inventory*, aplikasi ini merupakan pengelolaan informasi yang tepat dan akurat yang dirancang untuk memberikan kemudahan dalam menghasilkan informasi persediaan dan penjualan yang mampu mengontrol proses keluar-masuk barang. Aplikasi *inventory* ini menerapkan metode algoritma *shortest remaining time first*, tiap barang memiliki prioritas yang disesuaikan dengan tanggal kadaluarsa barang, sehingga memudahkan pemilik toko saat proses keluarnya barang untuk mengurangi kerugian dalam proses bisnis. Metodologi yang digunakan adalah metode *waterfall*, model ini menggambarkan perancangan perangkat lunak seperti aliran air terjun, mulai dari *analysis requirement* sebagai awal proses sampai dengan *coding* dan *testing* di akhir proses. Hasil dari aplikasi yang dibuat yaitu informasi *inventory* yang saling terintegrasi dengan penjualan sehingga proses bisnis menjadi lebih efektif dan efisien, menyediakan informasi barang yang harus didahulukan dalam proses penjualan berdasarkan batas waktu ketahanan produk serta fitur tambahan lokasi tempat barang tersebut disimpan.

Kata Kunci:

SRTF, *Inventory*, Kadaluarsa, Laporan Penjualan, Laporan Persediaan

**DESIGN OF SALES AND INVENTORY FOR MINIMARKET  
APPLICATION USING SHORTEST REMAINING TIME FIRST  
ALGORITHM**

Chandra Setiawan Gimon

---

**ABSTRACT**

The increasing number of self-service franchise businesses that sell various kinds of daily necessities brought negative impact to stores that are nearby thus resulting in deserted customers and cumulation of items. Hence must be done process recalculation of the stocks availability. But sometimes have trouble of having to perform manual checking process. Based on these problems required inventory application, this application is the appropriate management and accurate information that is designed to provide ease of generating inventory and sales information are able to control the process in and out of items. This inventory application applying methods shortest remaining time first, in which each item has a priority that is adapted to expiry date of items, making it easier for the owner of the store when the discharge of items to reduce losses in business process. The methodology used is the waterfall method, it describes the design of software such as the flow of waterfalls, ranging from requirements analysis as the beginning of the process up to coding and testing at the end of the process. Results of this application is the inventory of information systems that are integrated with sales so that business processes become more effective and efficient, providing information on items that should take precedence in the sales process based on expiry date of items as well as additional features of the location where the items are stored.

Keywords:

SRTF, Inventory, Expiry Date, Sales Report, Inventory Report

## UNGKAPAN TERIMA KASIH

Puji dan syukur kehadiran Allah SWT karena atas rahmat-Nya dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir dengan judul “Rancang Bangun Aplikasi Penjualan dan *Inventory Minimarket* Dengan Menggunakan *Shortest Remaining Time First Algorithm*” ini ditulis untuk memenuhi salah satu syarat dalam menyelesaikan perkuliahan pendidikan strata satu (S1) pada Program Studi Informatika, Universitas Bakrie.

Banyak pihak yang telah membantu penulis dalam penelitian dan penulisan Tugas Akhir ini, baik itu berupa bimbingan, saran, maupun dukungan secara moril dan materil. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih dan penghargaan yang setulus-tulusnya kepada:

1. Bapak Prof. Dr. Hoga Saragih S.T., M.T., selaku Kepala Program Studi Informatika, yang senantiasa memberikan masukan dan motivasi kepada penulis;
2. Bapak Yusuf Lestanto, S.T., M.Sc., selaku dosen pembimbing, yang telah meluangkan waktunya serta memberikan bimbingan, saran, dan perbaikan dalam menyelesaikan penelitian ini;
3. Bapak Berkah I. Santoso, S.T., M.TI., selaku pembahas yang memberikan saran dan perbaikan terhadap penelitian ini;
4. Seluruh Bapak/Ibu dosen Program Studi Informatika UB, yang telah memberikan banyak ilmu, pengetahuan, wawasan kepada penulis selama perkuliahan;
5. Keluarga tercinta, yang telah memberikan dukungan dan doa yang sangat berarti bagi penulis. Kedua orang tua penulis (Dicky Gimon dan Suryati Suratinoyo), kakak penulis (dr. Indriaty Gimon). Serta keluarga besar di Manado dan Jakarta;
6. Family: Evi Margaretha, Faiz Faidurrahman, Rahmad Dita, Rien Pratama, Rizky Akbarie, Sarah Putri Mardhatillah, Sawitri Sadanti, dan Steffany

Uliarta. Terima kasih telah memberikan semangat, motivasi, dukungan, suka cita dan kebersamaan selama ini;

7. Bad Boys: Alvian Aditya Kanzi dan Sairam Salim, terima kasih atas saran dan dukungan selama proses penyusunan;
8. Teman-teman KARLOTA MAMPOS (Indira, Olan, Ayu, Seri, Dewi, Suharti, Pipit, Wati, Ipul) terima kasih atas perhatian, dukungan dan keceriaannya selama ini;
9. Teman-teman TIF 2011 senasib dan seperjuangan. Terima kasih sudah menemani dan bekerja sama selama 4 tahun masa studi di UB;
10. Seluruh pihak yang terlibat dalam penyusunan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu;

Dengan segala keterbatasan yang ada, penulis menyadari bahwa penyusunan Tugas Akhir ini masih jauh dari kesempurnaan. Untuk itu, saran dan kritik akan selalu diterima agar penulis dapat memperbaiki setiap kekurangan untuk kesempurnaan dimasa mendatang.

Akhirnya, penulis menyampaikan ucapan terima kasih dan semoga Allah SWT membalas segala kebaikan serta melimpahkan berkat dan rahmay-Nya kepada semua pihak yang telah membantu selama ini. Penulis berharap Tugas Akhir ini berguna dan bermanfaat bagi kita semua.

Jakarta, Agustus 2016

Chandra Setiawan Gimon

## DAFTAR ISI

<b>HALAMAN PENGESAHAN.....</b>	<b>i</b>
<b>ABSTRAK .....</b>	<b>ii</b>
<b>ABSTRACT .....</b>	<b>iii</b>
<b>UNGKAPAN TERIMA KASIH .....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR GAMBAR.....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR SINGKATAN.....</b>	<b>xii</b>
<b>BAB I.....</b>	<b>1</b>
1.1    Latar Belakang Masalah .....	1
1.2    Rumusan Masalah .....	2
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian.....	3
1.5    Manfaat Penelitian.....	4
<b>BAB II .....</b>	<b>5</b>
2.1    Penelitian Terdahulu.....	5
2.2    Sistem, Data dan Informasi Manajemen .....	8
2.2.1    Pengertian Sistem.....	8
2.2.2    Data .....	9
2.2.3    Informasi .....	10
2.3    Konsep Dasar <i>Inventory</i> .....	11
2.4    Konsep Dasar <i>Minimarket</i> .....	12
2.5    Algoritma Shortest Job First.....	13
2.6    Model Waterfall .....	17
<b>BAB III.....</b>	<b>21</b>
3.1    Requirements analysis .....	21
3.2    System and software design .....	21

3.3	Implementation.....	21
3.3.1	Implementasi Algoritma Shortest Remaining Time First .....	22
3.4	System testing.....	24
3.4.1	Rencana Pengujian Sistem .....	25
3.4.2	Rencana Pengujian Algoritma <i>Shortest Job First</i> .....	25
3.5	Maintenance .....	26
3.6	Metode Pengumpulan Data .....	26
3.7	Evaluasi dan Pengujian .....	26
<b>BAB IV</b>	<b>.....</b>	<b>28</b>
4.1	<i>Requirement Analysis</i> .....	28
4.2	<i>System and Software Design</i> .....	29
4.2.1	Diagram <i>Use case</i> .....	29
4.2.2	Class Diagram .....	43
4.2.3	Perancangan <i>Database</i> .....	43
4.2.4	Perancangan <i>User Interface</i> .....	51
4.2.5	Perancangan Visualisasi.....	54
4.2.6	Pseudocode sistem .....	55
4.3	<i>Implementation</i> .....	58
4.3.1	Implementasi Sistem.....	58
4.3.2	Implementasi GUI ( <i>Graphical User Interface</i> ) .....	59
4.3.3	Implementasi Algoritma SRTF .....	68
4.4	<i>System Testing</i> .....	70
4.4.1	Pengujian Sistem.....	70
4.4.2	Pengujian Algoritma .....	78
4.5	<i>Results and Reports</i> .....	84
<b>BAB V</b>	<b>.....</b>	<b>85</b>
5.1	Simpulan.....	85
5.2	Saran .....	86
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>87</b>



## DAFTAR GAMBAR

Gambar 2. 1 <i>Data, Proses, dan Informasi</i> .....	11
Gambar 2. 2 <i>Waterfall Phase</i> (Pressman, 2010) .....	17
Gambar 2. 3 <i>Waterfall Phase</i> (Sommerville, 2011).....	19
Gambar 4.1 <i>Use Case Diagram</i> .....	30
Gambar 4.2 <i>Class Diagram</i> .....	43
Gambar 4.3 <i>Conceptual Database Design</i> .....	44
Gambar 4.4 <i>Logical Database Design</i> .....	46
Gambar 4.5 <i>Physical Database Design</i> .....	49
Gambar 4.6 GUI Halaman <i>Login</i> .....	51
Gambar 4.7 GUI Halaman <i>Utama</i> .....	52
Gambar 4.8 GUI Halaman <i>Stok Barang</i> .....	52
Gambar 4.9 GUI Halaman <i>Penjualan</i> .....	53
Gambar 4.10 GUI Halaman <i>Laporan</i> .....	54
Gambar 4.11 GUI Halaman <i>Visualisasi</i> .....	55
Gambar 4.12 Halaman <i>Login</i> .....	59
Gambar 4.13 Halaman <i>Awal</i> .....	60
Gambar 4.14 Halaman <i>Data Utama</i> .....	60
Gambar 4.15 Halaman <i>Daftar Pengguna</i> .....	61
Gambar 4.16 Halaman <i>Tambah Data Pengguna</i> .....	61
Gambar 4.17 Halaman <i>Ubah Data Pengguna</i> .....	62
Gambar 4.18 Halaman <i>Daftar Barang / Produk</i> .....	62
Gambar 4.19 Halaman <i>Tambah Data Barang / Produk</i> .....	63
Gambar 4.20 Halaman <i>Daftar Rak</i> .....	63
Gambar 4.21 Halaman <i>Tambah Data Rak Barang</i> .....	64
Gambar 4.22 Halaman <i>Stok Produk</i> .....	64
Gambar 4.23 Halaman <i>Daftar Stok Barang / Produk</i> .....	65
Gambar 4.24 Halaman <i>Pembelian Produk</i> .....	65
Gambar 4.25 Halaman <i>Penjualan</i> .....	66
Gambar 4.26 Halaman <i>Penjualan Barang / Produk</i> .....	66
Gambar 4.27 Halaman <i>Penjualan Barang Hari Ini</i> .....	67

Gambar 4.28 Halaman Laporan .....	67
Gambar 4.29 Halaman Keluar - Masuk Barang .....	67
Gambar 4.30 Halaman Laporan Keuangan .....	68
Gambar 4.31 Halaman Visualisasi .....	68
Gambar 4.32 Implementasi Algoritma SRTF .....	69
Gambar 4.33 Grafik <i>Usability Testing</i> .....	74
Gambar 4.34 Grafik <i>Interface Testing</i> .....	77
Gambar 4.35 Hasil <i>Test Case</i> 1 (A) .....	82
Gambar 4.36 Hasil <i>Test Case</i> 1 (B) .....	82
Gambar 4.37 Hasil Test Case 2 (A) .....	83
Gambar 4.38 Hasil Test Case 2 (B) .....	83

## DAFTAR TABEL

Tabel 2. 1 Rangkuman Penelitian Terdahulu .....	6
Tabel 2. 2 Definisi Data .....	9
Tabel 2. 3 Definisi Informasi .....	10
Tabel 2. 4 Perbandingan <i>Minimarket</i> dan Toko/Warung Kelontong .....	13
Tabel 2. 5 Contoh Proses .....	14
Tabel 2. 6 <i>Job</i> Berdasarkan Urutan .....	14
Tabel 2. 7 <i>Job</i> Berdasarkan SJF .....	15
Tabel 2. 8 Perbandingan Algoritma Penjadwalan .....	15
Tabel 3. 1 Contoh Proses 1 .....	23
Tabel 3. 2 Contoh Proses 2 .....	23
Tabel 4.1 <i>Use Case Scenario Login</i> .....	30
Tabel 4.2 <i>Use Case Scenario</i> Melihat Informasi Stok Barang .....	31
Tabel 4.3 <i>Use Case Scenario</i> Melihat Transaksi Penjualan .....	32
Tabel 4.4 <i>Use Case Scenario</i> Melakukan Transaksi Penjualan .....	32
Tabel 4.5 <i>Use Case Scenario</i> Melihat Laporan Keluar – Masuk Barang .....	33
Tabel 4.6 <i>Use Case Scenario</i> Melihat Laporan Keuangan .....	34
Tabel 4.7 <i>Use Case Scenario</i> Melihat Lokasi Barang .....	34
Tabel 4.8 <i>Use Case Scenario Add</i> Barang .....	35
Tabel 4.9 <i>Use Case Scenario Update</i> Barang .....	36
Tabel 4.10 <i>Use Case Scenario Delete</i> Barang .....	37
Tabel 4.11 <i>Use Case Scenario Add</i> User .....	37
Tabel 4.12 <i>Use Case Scenario Update</i> User .....	38
Tabel 4.13 <i>Use Case Scenario Delete</i> User .....	39
Tabel 4.14 <i>Use Case Scenario Add</i> Rak .....	40
Tabel 4.15 <i>Use Case Scenario Update</i> Rak .....	40
Tabel 4.16 <i>Use Case Scenario Delete</i> Rak .....	41
Tabel 4.17 <i>Use Case Scenario Logout</i> .....	42
Tabel 4.18 <i>Database Entity</i> .....	44
Tabel 4.19 Hubungan Antar <i>Entity</i> .....	45
Tabel 4.20 Definisi Entitas dan Atribut .....	46

Tabel 4.21 Deskripsi Atribut <i>Entity Database</i> .....	49
Tabel 4.22 Hasil Pengujian <i>Usability</i> .....	71
Tabel 4.23 Hasil Pengujian <i>Interface</i> .....	75
Tabel 4.24 Hasil <i>Whitebox Testing</i> .....	78
Tabel 4.25 Hasil <i>Blackbox Testing</i> .....	81

## DAFTAR SINGKATAN

SRTF *Shortest Remaining Time First*

CPU *Central Processing Unit*

GUI *Graphical User Interface*

PDM *Physical Database Model*

UML *Unified Modeling Language*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Kebutuhan akan informasi saat ini merupakan kebutuhan yang sangat mendasar untuk menentukan rencana kedepan. Akses internet yang sangat mudah dan tidak terbatas, dapat menjawab semua keingintahuan masyarakat akan informasi yang dibutuhkannya. Sistem informasi merupakan hal yang penting dalam suatu organisasi atau perusahaan. Dengan adanya sistem informasi, organisasi atau perusahaan dapat menjamin kualitas informasi yang disajikan dan dapat mengambil keputusan berdasarkan informasi tersebut (Zebua & Mustikasari, 2012).

Dalam sistem *inventory*, proses masuk dan keluar barang merupakan hal yang umum terjadi. Dengan jenis barang dan stok barang yang mencapai ratusan dalam satuan kecil dan besar, kedua proses tersebut merupakan hal yang tidak mudah untuk dilaksanakan. Hal ini semakin sulit mengingat pencatatan gudang yang masih bersifat manual. Untuk menangani masalah ini harus dibuat suatu basis data, perancangan basis data yang berguna untuk menjamin keakuratan data agar dapat meningkatkan kinerja organisasi atau perusahaan.

Proses untuk mendapatkan jumlah stok barang yang tersisa akan mengalami kesulitan jika harus melakukan pengecekan secara manual menggunakan berkas fisik. Apabila dilihat dari sisi efisiensi kerja, hal ini membuat operasional terlihat kurang handal dan memiliki banyak resiko, karena berkas-berkas yang mudah rusak dan apabila rusak tidak tersedia data cadangan (Saputra, Falahah, & Siswanto, 2012). Kegiatan pengelolaan barang dari tahun ke tahun yang terus berlangsung ini bukan hanya melibatkan aset barang lama saja tapi aset barang yang baru. Sehingga dari tahun ke tahun jumlah barang tidak berkurang tapi terus akan bertambah. Dengan bertambahnya jumlah barang-barang yang tersimpan, tentunya menambah tingkat kesulitan tersendiri dalam proses pengelolaan. Agar supaya proses

penyimpanan barang dapat dikelola serta tertata dengan baik, maka perlu dikembangkan suatu aplikasi Penjualan dan *Inventory* barang, karena dengan cara biasa (banyak proses manual) seperti sekarang, cukup menyulitkan dalam hal pengarsipan dan pengecekan data barang karena proses pengeluaran barang, penempatan dan perubahan posisi barang yang kurang terdokumentasi dan terkontrol dengan baik. Berdasarkan hasil wawancara yang dilakukan secara mendalam terhadap beberapa pemilik toko/minimarket 100% menyatakan membutuhkan aplikasi penjualan yang dapat mengatur persediaan barang serta mampu memberikan informasi kadaluarsa barang untuk mencegah kerugian dalam proses bisnis.

Aplikasi Penjualan dan *Inventory* barang ini akan menampung semua data dan informasi tentang barang-barang yang tersimpan. Data dan informasi ini kemudian terakumulasi dan diarsipkan secara terpusat pada suatu *database*. Terpusatnya data dan informasi ini digunakan untuk mempermudah pengelolaan barang. Pekerjaan seperti pencarian data dan status barang menjadi lebih cepat, mudah, dan efisien. *Database* ini bersifat digital, akurasi informasi data yang diberikan sangat tinggi, karena mengurangi faktor kesalahan manusia. Akurasi data dan informasi ini bukan untuk penyajian data saja tetapi dalam hal penyimpanannya. Akurasi data bergantung pada proses *input*, semakin baik proses *input* data maka akan semakin tinggi akurasi data yang tersimpan dalam *database*, dan sebaliknya, apabila proses *input* data kurang baik maka akurasi data tidak dapat diandalkan. Kemudahan lain apabila data ini telah tersimpan dalam *database*, secara berkala, maka akan dapat digunakan untuk proses evaluasi dan analisis. Sehingga dapat diambil data untuk digunakan pada *forecasting* kebutuhan dan penyediaan barang.

## 1.2 Rumusan Masalah

Adapun rumusan masalah dari permasalahan ini adalah:

1. Bagaimana membuat aplikasi penjualan dan *inventory* pada *minimarket* yang terintegrasi dan dapat mengontrol item barang yang masuk dan

keluar untuk mencegah kadaluarsa terhadap barang karena adanya penumpukan barang di gudang ?

2. Bagaimana menyediakan informasi barang yang harus dijual terlebih dahulu untuk mencegah adanya kerusakan atau terjadi kadaluarsa pada barang tersebut dengan merancang bangun aplikasi penjualan dan inventory dengan karakteristik tanggal kadaluarsa sebagai acuan ?
3. Bagaimana menyediakan informasi lokasi dan posisi dari barang yang terdapat dalam *inventory* ?

### 1.3 Batasan Masalah

Berdasarkan perumusan masalah diatas maka permasalahan dibatasi sebagai berikut:

1. Sistem informasi ini tidak mengatur tentang hubungan manajemen penjualan dan hubungan pemasaran.
2. Sistem informasi ini bersifat sebagai manajemen stok barang untuk warung atau toko atau *minimarket*. Warung/toko yang dimaksud adalah yang menjual kebutuhan pangan manusia, yang mempunyai batas waktu ketahanan (kadaluarsa).

### 1.4 Tujuan Penelitian

Tujuan dibuatnya aplikasi penjualan dan inventory ini adalah:

1. Membuat aplikasi *inventory* pada *minimarket* integrasi antara proses *inventory* dengan penjualan dapat tercapai, sehingga proses bisnis menjadi lebih efektif dan efisien serta mencegah terjadinya penumpukan barang lama dalam gudang.
2. Menyediakan informasi barang yang harus didahulukan dalam proses penjualan berdasarkan batas waktu ketahanan atau kadaluarsa produk, untuk mencegah kerugian yang disebabkan kadaluarsa barang dan barang rusak, akibat penumpukan barang dalam gudang.



3. Menyediakan informasi untuk mempermudah pemilik mengetahui stok barang yang dimiliki dan lokasi tempat barang tersebut disimpan.

## 1.5 Manfaat Penelitian

Manfaat yang diharapkan dalam pembuatan aplikasi penjualan dan *inventory* adalah sebagai berikut:

### 1. Bagi Pemilik

- Mendapatkan informasi yang lebih cepat mengenai stok barang yang terdapat dalam *inventory* setiap hari dan secara *real time*.
- Mempermudah dalam pencatatan stok masing masing barang, meliputi kronologis keluar masuknya barang.
- Membantu dalam mencegah kerugian dikarenakan barang yang rusak akibat kesalahan pada proses penjualan.
- Membantu pemilik warung atau toko atau *minimarket* mengetahui evaluasi hasil penjualan.

### 2. Bagi Pembaca

- Menambah wawasan tentang pentingnya transaksi penjualan yang saling intergrasi antar proses bisnis dan solusi dari beberapa permasalahan yang ada.

### 3. Bagi Penulis

- Dapat menambah pengetahuan dan wawasan serta dapat mengaplikasikan dan mensosialisasikan teori yang telah diperoleh selama perkuliahan.

### 4. Bagi Pengembangan Keilmuan

- Dengan penelitian ini diharapkan dapat menjadi referensi dalam pengembangan aplikasi bagi peneliti selanjutnya yang tertarik untuk meneliti tentang aplikasi penjualan dan *inventory* yang terintegrasi.

### 5. Bagi Universitas

- Dapat menambah referensi dan masukan bagi pihak-pihak yang bersangkutan dan memerlukan informasi mengenai penelitian ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian tentang perancangan aplikasi *point of sale* dengan arsitektur *client/server* berbasis linux dan windows dilakukan oleh (Kosasi, 2014). Pada penelitian ini aplikasi *point of sale* adalah aplikasi untuk mengelola transaksi bisnis ritel swalayan yang berhubungan dengan pengolahan data pembelian, penjualan dan laporan untuk pihak manajemen. Aplikasi *point of sale* membuat transaksi di dalam proses menjadi lebih cepat dan efisien. Memiliki integrasi sistem antara gudang dan penjualan merupakan faktor yang penting. Hal ini menjadi persoalan dalam penelitian ini karena segala proses transaksi ritel masih menggunakan sistem manual dengan menggunakan alat bantu *cash register* yang dalam prosesnya mesin ini masih memiliki keterbatasan dari sisi item barang, pengolahan data transaksi dan informasi. Hasil yang dihasilkan dari penelitian ini meliputi modul aplikasi kasir, modul aplikasi gudang dan modul aplikasi laporan. Modul aplikasi antara lain mencakup pengelolaan penjualan, edit data penjualan, pengisian data barang, penentuan harga barang dan menghasilkan laporan penjualan.

Penelitian sejenis yang dilakukan (Setiawan & Suryana, 2006) sebelumnya menggunakan bahasa pemrogram *C#* dengan teknologi *.Net Framework* dengan *database* MS Office 2003 dan masih bersifat semi integrasi antar proses bisnis. Selain itu fitur-fitur aplikasi yang dihasilkan masih terbatas hanya untuk pengolahan data transaksi dan dengan informasi yang terbatas. Tujuan dari penelitian ini adalah untuk mengelola *administrasi* stok, maupun keuangan secara komputerisasi pada toko kecil atau menengah. Aplikasi yang dihasilkan dapat melakukan pencatatan dan perhitungan secara otomatis. Aplikasi ini diharapkan dapat semaksimal mungkin mengurangi proses kerja secara manual sehingga para pemilik toko swalayan berskala kecil menengah dapat melakukan kegiatan bisnisnya dengan lebih efisien dalam segi waktu maupun biaya, serta memudahkan dalam melakukan pemeriksaan terhadap stok barang dan transaksi pembelian/penjualan.

Penelitian dilakukan oleh Dwiprastio, dkk. (Dwiprastio, Karismariyanti, & Sukawati, 2013) tentang aplikasi penjualan dan persediaan barang dagang dengan metode perpetual FIFO berbasis web (studi kasus pada PD Anugerah) dengan permasalahan menangani transaksi penjualan dan persediaan barang pada PD. Anugerah dan bagaimana menghasilkan jurnal penjualan dan kartu persediaan. Penelitian ini bertujuan untuk membuat aplikasi yang dapat menangani transaksi penjualan dan persediaan barang dagang pada PD. Anugerah serta membuat aplikasi yang dapat menangani jurnal penjualan dan kartu persediaan.

Tabel 2. 1 *Rangkuman Penelitian Terdahulu*

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
1	Perancangan Aplikasi Point of Sale dengan Arsitektur Client/Server Berbasis Linux dan Windows	Sandy Kosasi	2014	Pada proses transaksi ritel masih menggunakan sistem manual dengan menggunakan alat bantu <i>cash register</i> yang dalam prosesnya mesin ini masih memiliki keterbatasan	Modul aplikasi kasir, modul aplikasi gudang dan modul aplikasi laporan. Modul aplikasi antara lain mencakup pengelolaan penjualan, edit data penjualan, pengisian data barang, penentuan harga barang dan menghasilkan laporan penjualan
2	Aplikasi Terintegrasi Toko Swalayan	Elisabet Setiawan & Erwin Suryana	2006	Masih bersifat semi integrasi antar proses bisnis	Mengelola <i>administrasi</i> stok, maupun keuangan secara komputerisasi pada toko kecil atau menengah, mengurangi proses

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
					kerja secara manual, serta memudahkan dalam melakukan pemeriksaan terhadap stok barang dan transaksi pembelian/penjualan
3	Aplikasi Penjualan Dan Persediaan Barang Dagang Dengan Metode Perpetual FIFO Berbasis Web (Studi Kasus Pada PD Anugerah)	Chandra Dwiprastio, Magdalena Karismariyanti & Renny Sukawati	2013	Pembuatan jurnal penjualan dan kartu persediaan PD. Anugerah belum menggunakan basis data yang terpusat. Data – data yang berkaitan dengan jurnal penjualan dan kartu persediaan masih tercatat dalam berbagai <i>file</i> yang terpisah di Microsoft Excel	Aplikasi ini memberikan kemudahan dalam menghasilkan jurnal penjualan dan kartu persediaan untuk setiap transaksi secara bersamaan dalam satu <i>file</i> , dan dibuat berbasis web dengan teknik pemograman terstruktur, menggunakan bahasa pemograman PHP dan basis data MySQL
4	Aplikasi Penjualan dan Inventory Pada Minimarket Dengan Menggunakan Shortest Remaining Time First Algorithm	Penulis	2015	Banyaknya kompetitor membuat terjadinya penumpukan barang dagangan,	Aplikasi <i>inventory</i> pada <i>minimarket</i> integrasi antara proses <i>inventory</i> dengan penjualan, informasi barang

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
				stok barang yang terus bertambah tiap tahun serta masih menggunakan pencatatan secara manual dalam proses data transaksi	yang harus didahulukan dalam proses penjualan berdasarkan batas waktu ketahanan atau kadaluarsa produk, informasi untuk mempermudah pemilik mengetahui stok barang yang dimiliki dan lokasi tempat barang tersebut disimpan berbasis <i>web base</i>

## 2.2 Sistem, Data dan Informasi Manajemen

### 2.2.1 Pengertian Sistem

Sistem adalah Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersamasama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Hartono, 2006)

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2004). Saat ini penggunaan konsep sistem sangat luas dan meliputi berbagai bidang sehingga timbul berbagai definisi tentang sistem, masing-masing menekankan pada sudut pandang dan kebutuhan sendiri.

Dengan kata lain, sistem adalah satuan (*entity*) yang terdiri dari dua atau lebih komponen (subsistem) atau suatu kerangka kerja terpadu yang terjalin satu sama lain untuk mencapai suatu tujuan/satu sasaran.

Ada 4 unsur utama dalam suatu sistem, yaitu:

- 1) Terdiri dari elemen-elemen atau bagian-bagian.
- 2) Mempunyai interaksi antar elemen-elemen tersebut.
- 3) Adanya suatu yang mengikat elemen-elemen tersebut menjadi suatu kesatuan.
- 4) Mempunyai tujuan sebagai hasil akhir.

### 2.2.2 Data

Terkait dengan *database* terdapat istilah dasar yang disebut data. Awalnya data menyatakan fakta yang dapat direkam dan disimpan pada media komputer; misalnya *hard disk*. Nama barang, harga barang, jumlah barang menyatakan suatu data. Namun perlu diketahui bahwa data pada masa sekarang tidak sekedar hanya berupa teks seperti itu, tetapi juga bisa berupa dokumen, gambar, suara, ataupun potongan video (Kadir, Dasar Perancangan & Implementasi Database Relational, 2008).

Tabel 2. 2 *Definisi Data*

Definisi Data	Sumber
Fakta-fakta mentah yang mewakili kejadian-kejadian yang berlangsung dalam organisasi atau lingkungan fisik sebelum ditata dan diatur kedalam bentuk yang dapat dipahami dan digunakan orang	(Laudon & Laudon, 1998)
Deskripsi tentang benda, kejadian, aktivitas dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai	(Kadir, Pengenalan Sistem Informasi, 2003)
Fakta, angka, bahkan symbol mentah. Secara bersama-sama merupakan masukan bagi suatu sistem informasi	(Wilkison, 1992)

Dapat dikatakan bahwa data adalah suatu bahan mentah yang dapat diolah lebih lanjut untuk menjadi sesuatu yang lebih bermakna. Data inilah yang nantinya akan disimpan dalam *database*.

### 2.2.3 Informasi

Informasi adalah istilah lain yang seringkali rancu dengan data. Kedua istilah ini seringkali saling dipertukarkan. Walaupun demikian, perbedaan kedua istilah tersebut perlu untuk dijelaskan.

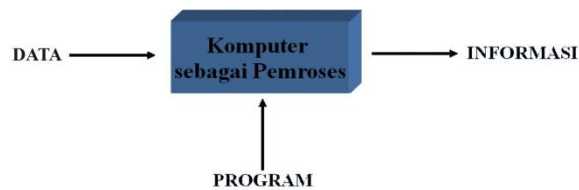
Tabel 2. 3 *Definisi Informasi*

Definisi Informasi	Sumber
Data yang telah diolah menjadi bentuk yang bermakna dan berguna bagi manusia	(Laudon & Laudon, 1998)
Data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya	(Hoffer, Prescot , & McFadden, 2005)
Data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang	(Davis, 1999)

Dapat disimpulkan bahwa:

- 1) Informasi bermuara pada data.
- 2) Memberikan suatu nilai tambah atau pengetahuan bagi yang menggunakannya.
- 3) Dapat digunakan untuk pengambilan keputusan.

Agar dapat menjadi informasi, perlu dilakukan penambahan item-item lain

Gambar 2. 1 *Data, Proses, dan Informasi*

dan penyediaan suatu kerangka sehingga memberikan suatu makna. Seringkali dinyatakan bahwa informasi adalah hasil pemrosesan data. Prosesnya sendiri dapat berupa peringkasan, perataan, penyajian ke bentuk grafik, ataupun bentuk lain, dengan tujuan untuk memudahkan interpretasi manusia.

### 2.3 Konsep Dasar *Inventory*

Persediaan diterjemahkan dari kata *inventory* yang merupakan timbunan barang (bahan baku, komponen, produk setengah jadi, atau produk jadi, dan lain-lain) yang secara sengaja disimpan sebagai cadangan (*safety* atau *buffer-stock*) untuk menghadapi kelangkaan pada saat proses produksi yang sedang berlangsung atau penimbunan barang dikarenakan kurangnya permintaan.

*Inventory* merupakan proses mengelola pengadaan atau persediaan barang di gudang. Dalam sistem akuntansi, *inventory* adalah salah satu aktiva lancar perusahaan yang pada dasarnya merupakan suatu pengolahan barang yang meliputi penjualan, pembelian dan kontrol stok gudang. Sistem *inventory* barang jadi disediakan untuk memenuhi permintaan dari konsumen setiap waktu yang disimpan dan di rawat menurut aturan tertentu dalam keadaan siap pakai dan tersimpan dalam *database* (Yuhendra & Poerwanta, 2013).

Dalam sistem informasi *inventory*, informasi-informasi yang dapat diambil seperti: (Damayanti, 2013)

- 1) Informasi mengenai jumlah barang yang telah terjual kepada pelanggan/customer selama periode tertentu (harian, minggu, bulan, tahun)
- 2) Jumlah barang yang telah dibeli dari pemasok selama periode tertentu
- 3) Keadaan stok yang terakhir dalam gudang untuk semua barang



Untuk lebih jelasnya mengenai persediaan, maka akan dipaparkan pengertian persediaan. Pengertian persediaan akan dijelaskan dari beberapa definisi berikut.

3. (Vrat, 2014) menyatakan bahwa persediaan atau stok dianggap sebagai dasar pokok dalam mengelola barang. *Inventory Turnover Ratio* (ITR) adalah barometer untuk mengukur kinerja dari fungsi manajemen barang. Pada umumnya, *inventory* berarti stok fisik barang yang disimpan di *minimarket*/toko untuk memenuhi atau mengantisipasi proses permintaan atas kebutuhan.
4. Menurut (McLeod, 1998) persediaan adalah sebagai suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha yang normal.
5. Menurut (Ristono, 2009) inventori dapat diartikan sebagai barang-barang yang disimpan untuk digunakan atau dijual pada masa atau periode yang akan datang.

Jadi dapat disimpulkan bahwa *inventory* adalah unsur aktiva yang berupa barang-barang yang tersedia untuk dijual dalam kegiatan bisnis normal atau barang-barang yang akan dikonsumsi dalam pengolahan produk yang akan dijual.

## 2.4 Konsep Dasar *Minimarket*

*Minimarket* adalah semacam “toko kelontong atau yang menjual segala macam barang dan makanan, tapi tidak selengkap dan sebesar *supermarket*. Berbeda dengan toko kelontong, *minimarket* menerapkan sistem swalayan, pembeli mengambil sendiri barang yang dibutuhkan (Jawaldi, 2014).

*Minimarket* digolongkan sebagai pasar/toko moderen, sehingga dalam hal ini pengertian *minimarket* dipersamakan dengan pengertian pasar/toko moderen. Dalam Peraturan Presiden RI No 112 Tahun 2007 Tentang Penataan dan Pembinaan Pasar Traditional, Pusat Perbelanjaan dan Toko Moderen dikatakan bahwa toko moderen adalah toko dengan sistem pelayanan mandiri, menjual berbagai jenis barang secara eceran yang berbentuk *minimarket*, *supermarket*, *department store*, *hypermarket* ataupun grosir yang berbentuk perkulakan.

Menurut (Ma'ruf, 2005), pengertian *minimarket* adalah toko yang mengisi kebutuhan masyarakat akan warung yang berformat moderen yang dekat dengan pemukiman penduduk sehingga dapat mengungguli toko atau warung.

Perbandingan antara minimarket dan toko/warung kelontong dengan melihat beberapa aspek meliputi: suasana belanja, pelayanan, jenis barang dan harga.

Tabel 2. 4 Perbandingan Minimarket dan Toko/Warung Kelontong

	<i>Minimarket</i>	<b>Toko/Warung Kelontong</b>
<b>Suasana Belanja</b>	Suasana belanja yang nyaman dan rata-rata berpendingin udara. Bebas memilih barang, baik mengentuh dan mengambilnya	Suasana belanja tidak nyaman, barang yang ingin dibeli di ambilkan oleh pemilik
<b>Pelayanan</b>	Pelayanan di minimarket dikemas melalui bisnis waralaba yang profesional	Terkadang menemukan warung kosong dan tidak ditunggu dan harus memanggil sang pemilik
<b>Jenis Barang</b>	Barang kebutuhan sehari-hari lengkap dan dengan stok yang cukup banyak	Belum menjamin semua jenis barang tersedia serta adanya barang tetapi umumnya stok terbatas.
<b>Harga</b>	Sering mengadakan diskon barang, lebih hemat dengan pembelian dalam jumlah banyak	Tidak ada diskon pada barang, pada umumnya tidak ada potongan untuk pembelian dalam jumlah banyak

## 2.5 Algoritma *Shortest Job First*

Algoritma *Shortest Job First Scheduling* sangat optimal (Nugrahanto, 2002), karena memberikan rata-rata waktu tunggu lebih kecil dibandingkan algoritma penjadwalan yang lain dengan cara memindahkan *job-job* pendek di depan *job-job* yang panjang, sehingga akan mengurangi waktu tunggu.

Mekanismenya adalah menjadwalkan proses dengan waktu jalan terpendek lebih dulu sampai selesai, sehingga memberikan efisiensi yang tinggi dan *turn around time* rendah (Julianto, 2014)

Algoritma ini dapat dibagi menjadi dua skema, yaitu: (Silberschatz, Gagne, & Galvin, 2002)

1. *Preemptive*, jika ada proses yang sedang dieksekusi oleh CPU dan terdapat proses di *ready queue* dengan *burst time* yang lebih kecil daripada proses yang sedang dieksekusi sebelumnya, maka proses yang sedang dieksekusi oleh CPU akan digantikan oleh proses yang berada di *ready queue* tersebut. *Preemptive Shortest Job First* juga sering disebut *Shortest Remaining Time First* (SRTF).
2. *Non-preemptive*, CPU tidak mengizinkan proses yang ada di *ready queue* untuk menggeser proses yang sedang dieksekusi oleh CPU sebelumnya, meskipun proses yang baru memiliki *burst time* yang lebih kecil.

Misalkan ada 4 *job* yaitu A, B, C, D masing-masing waktu kedatangan sama yaitu pada  $t = 0$ , dan lama proses *job* berturut-turut: 8,4,4,4

Tabel 2. 5 Contoh Proses

Proses	Waktu
A	8
B	4
C	4
D	4

Jika urutan pengerjaannya:

- a. *Job* A, B, C, D
- b. *Job* B, C, D, A

Maka proses pengerjaannya adalah sebagai berikut:

Tabel 2. 6 Job Berdasarkan Urutan

8	4	4	4
---	---	---	---

A	B	C	D
---	---	---	---

Tabel 2. 7 Job Berdasarkan SJF

4	4	4	8
B	C	D	A

- Dengan pengerjaan *job* berdasarkan urutan (Tabel 2.6) maka berturut-turut waktu yang dibutuhkan untuk proses A, B, C, D adalah 8, 12, 16, 20 sehingga dapat dihitung waktu rata-rata =  $(8+12+16+20)/4 = 14$
- Bila *job* dikerjakan berdasarkan (Tabel 2.7), yaitu dengan *shortest job first*, maka waktu yang dibutuhkan untuk proses B, C, D, A adalah 4, 8, 12, 20 atau dengan rata-rata =  $(4+8+12+20)/4 = 11$

Berdasarkan contoh di atas, bahwa algoritma *shortest job first scheduling* lebih optimal karena waktu pengerjaan *job* kecil cepat tanpa harus antri menunggu pekerjaan *job* yang lebih besar sehingga dapat mengoptimalkan kinerja (Alie, Widodo, & Sutanto, 2013).

Penjadwalan bertugas untuk memutuskan proses yang harus berjalan, kapan dan selama berapa proses itu berjalan. Berikut adalah perbandingan algoritma penjadwalan

Tabel 2. 8 Perbandingan Algoritma Penjadwalan

Algoritma	Karakteristik
<i>First In First Out</i>	Algoritma ini merupakan algoritma paling sederhana karena prinsip penjadwalan tidak berprioritas, proses-proses diberi jatah waktu pemroses berdasarkan waktu kedatangan, pada saat proses mendapat jatah waktu pemroses, proses dijalankan sampai selesai
Algoritma	Karakteristik

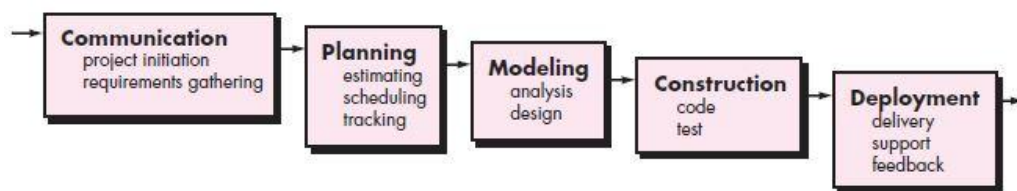
<i>Priority Scheduling</i>	Algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing. Prioritas suatu dapat ditentukan melalui beberapa karakteristik. <i>Priority scheduling</i> dapat dijalankan secara <i>preemptive</i> maupun <i>non-preemptive</i>
<i>Shortest Job First</i>	Pada algoritma ini proses dieksekusi berdasarkan burst time terkecil. <i>Waiting time</i> rata-ratanya juga menjadi pendek, sehingga dapat dikatakan bahwa algoritma ini adalah algoritma yang optimal. <i>Shortest Job First</i> dapat dijalankan secara <i>preemptive</i> maupun <i>non-preemptive</i>
<i>Round Robin</i>	Algoritma ini hampir mirip dengan FCFS akan tetapi terdapat proses perpindahan antar proses yaitu satu proses melakukan interupsi terhadap lainnya atau disebut <i>preemptive</i> . Proses <i>preemptive</i> menggunakan <i>time quantum</i> atau <i>time slice</i>
<i>Multiple Feedback Queues</i>	Penjadwalan berprioritas dinamis. Penjadwalan ini adalah untuk mencegah banyaknya <i>swapping</i> dan mencegah proses-proses interaktif yang singkat harus menunggu lama.
<i>Guaranteed Scheduling</i>	Algoritma penjadwalan ini memberikan daya pemroses yang sama untuk membuat dan menyesuaikan kinerja. Algoritma yang memiliki kinerja yang cukup bagus akan menjanjikan kelangsungan yang baik pula. Algoritma ini menjalankan proses dengan rasio yang paling rendah dulu sampai proses tersebut mendapatkan rasio melebihi rasio proses yang sebelumnya mempunyai rasio satu tingkat lebih tinggi darinya.
<i>Highest Response Ratio Next</i>	Penjadwalan dengan prioritas proses tidak hanya merupakan fungsi waktu layanan tetapi juga jumlah

	waktu tunggu proses. Begitu proses mendapat jatah pemroses, proses berjalan sampai selesai. karena waktu tunggu ditambah waktu layanan adalah waktu tanggap, yang berarti waktu tanggap tertinggi yang harus dilayani
--	---

## 2.6 Model Waterfall

Menurut Pressman (Pressman, 2010) model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Berikut ini ada dua gambaran dari *waterfall* model.

Fase-fase dalam model *waterfall* menurut referensi Pressman:



Gambar 2. 2 Waterfall Phase (Pressman, 2010)

### 1. Communication

Langkah ini merupakan analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun internet.

### 2. Planning

Dalam proses ini merupakan proses lanjutan dari proses sebelumnya. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*.

### 3. *Modeling*

Proses *modeling* ini menerjemahkan syarat kebutuhan ke dalam sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) procedural. Pada tahapan ini menghasilkan dokumen *software requirement*.

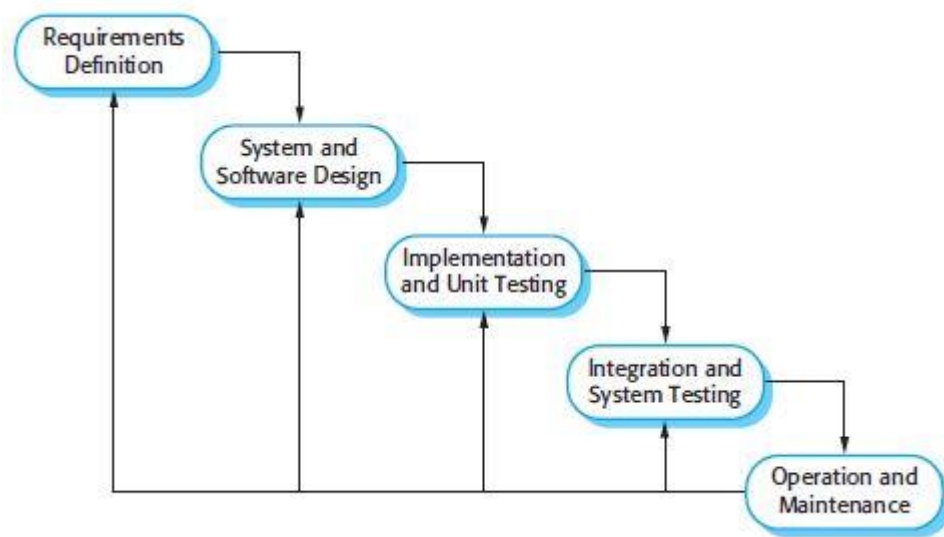
### 4. *Construction*

*Construction* adalah proses membuat kode. *Coding* merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* menerjemahkan permintaan yang diminta oleh *user*. Pada tahapan inilah yang merupakan tahapan nyata dalam menyerjakan suatu *software*. Setelah proses pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan *error* terhadap sistem untuk kemudian bisa diperbaiki.

### 5. *Deployment*

Tahapan ini merupakan tahapan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan proses analisis, desain dan pengkodean maka sistem yang sudah jadi bisa digunakan oleh *user*. *Software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Sedangkan fase-fase *waterfall* menurut referensi Sommerville (Sommerville, 2011)



Gambar 2. 3 Waterfall Phase (Sommerville, 2011)

### 1. *Requirements Analysis and Definition*

Kebutuhan secara lengkap dikumpulkan kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun. Hal ini sangat penting, karena *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. Tahap ini disebut dengan *project definition*.

### 2. *System and Software Design*

Proses pencarian difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*. Proses *software design* adalah untuk mengubah kebutuhan-kebutuhan menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai.

### 3. *Implementation and Unit Testing*

Pada tahapan ini desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.



#### 4. *Integration and System Testing*

Agar dapat dimengerti oleh komputer, maka desain harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini implementasi dari tahap *design* yang nantinya dikerjakan oleh *programmer*. Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

#### 5. *Operation and maintenance*

Semua fungsi-fungsi software harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Proses *maintenance* atau pemeliharaan *software* sangat diperlukan, termasuk pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lunak.

Kelebihan dari model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh dan benar di awal pengerjaan, maka *software engineering* (*SE*) dapat berjalan dengan baik dan tanpa masalah.

## BAB III

### METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian Rancang Bangun Aplikasi *Inventory Minimarket* menggunakan *shortest remaining time first algorithm* adalah sebagai berikut:

#### 3.1 *Requirements Analysis*

Tahap pertama ini diawali dengan pengidentifikasian kebutuhan yang menganalisis aturan bisnis yang berisikan aturan-aturan yang berlaku pada proses bisnis penjualan, proses bisnis persediaan dan proses pencatatan laporan serta data pendukung lainnya yang mengevaluasi permasalahan dan hambatan yang sering terjadi pada proses bisnis yang tidak sesuai dengan keinginan pemilik bisnis.

#### 3.2 *System and Software Design*

Tujuan utama pada tahap ini adalah proses pemodelan dengan menggunakan *Data Flow Diagram* (DFD) dan pemodelan data dengan menggunakan *Entity Relationship Diagram* (ERD) termasuk perancangan dan pembuatan *database* serta seluruh proses yang ada pada aplikasi penjualan dan *inventory* menjadi sebuah desain *user interface*.

#### 3.3 *Implementation*

Setelah proses desain dilakukan, penulis melakukan *coding* program dari hasil desain yang telah dirancang sebelumnya. Bahasa pemrograman yang digunakan adalah Visual Basic dengan melakukan penerapan algoritma *Shortest Remaining Job First* dalam aplikasi yang dibuat berdasarkan kebutuhan pada aplikasi penjualan dan *inventory*.

### 3.3.1 Implementasi Algoritma *Shortest Remaining Time First*

Penjadwalan konvensional yaitu penjadwalan *first in first out* (barang yang pertama dibeli yang akan dijual terlebih dulu) sering mengabaikan batas waktu ketahanan produk sehingga ketidaktauan informasi untuk barang mana yang harus dikeluarkan terlebih dahulu yang terkadang dapat menghambat dan menimbulkan kerugian pada proses bisnis. Pada sistem aplikasi penjualan dan *inventory* ini diterapkan algoritma penjadwalan *Shortest Remaining Time First* pada sistem *inventory* yang bekerja pada saat pihak toko dituntut untuk menjual produk yang mempunyai batas waktu ketahanan lebih pendek daripada produk lainnya terlebih dahulu. Algoritma *Shortest Remaining Time First* pada penelitian ini mengasumsikan waktu ketahanan produk sudah diketahui sebelumnya.

Mekanismenya adalah menjadwalkan proses dengan lama waktu ketahanan terpendek lebih dulu sampai selesai sehingga memberikan efisiensi yang tinggi dalam proses bisnis. *Shortest Job First* memiliki konsep *Non-preemptive* (tidak dapat diinterupsi) pemilihan proses dalam antrian yang sedang berjalan tidak memperbolehkan proses baru yang memiliki nilai prioritas tinggi untuk menggeser proses yang sedang dieksekusi dan *preemptive* (dapat diinterupsi) proses yang dipilih ada yang memiliki waktu sisa eksekusi terkecil dan jika ada proses baru yang datang dengan nilai prioritas lebih tinggi dari proses sebelumnya maka diperbolehkan menggeser proses yang sedang dieksekusi, versi *preemptive* ini disebut *Shortest Remaining Time First*.

Dalam sistem ini, elemen-elemen direpresentasikan dalam algoritma *Shortest Remaining Time First* adalah *burst time* (lama eksekusi) dalam kasus ini *burst time* adalah lama waktu ketahanan yang dihitung dari *arrival time* (waktu kedatangan) sampai waktu kadaluarsa dari proses yang dilakukan. Awal analisis dimulai dengan pengolahan data barang dengan tujuan mencari total waktu ketahanan atau *burst time* dan kemudian dilanjutkan berdasarkan waktu *burst time* terkecil. Penjadwalan *Shortest Remaining Time First* adalah kasus khusus untuk algoritma penjadwalan prioritas.

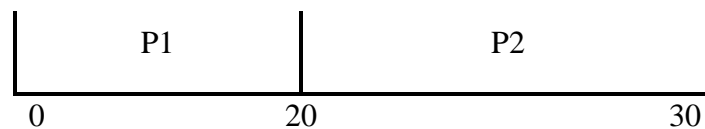
*Shortest Remaining Time First* adalah yang paling optimal karena mengeksekusi waktu yang paling pendek terlebih dahulu baru waktu yang lama. Berikut dijabarkan contoh kasus dalam pengimplementasian algoritma.

Jika barang P1 dan P2 datang pada saat yang sama maka akan dilihat prioritasnya terlebih dahulu yang memiliki nilai waktu ketahanan paling pendek.

Tabel 3. 1 *Contoh Proses 1*

<b>Proses</b>	<b>Waktu Kedatangan</b>	<b><i>Burst Time</i></b>
P1	0	20
P2	0	30

*Gantt chart:*



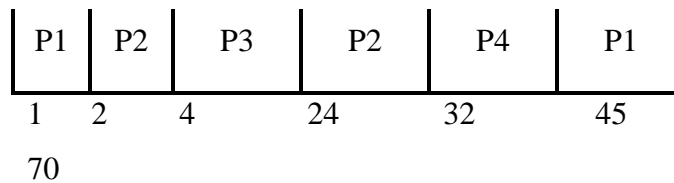
P1 memiliki burst time 20 dan P2 memiliki *burst time* 30 karena *burst time* terkecil memiliki prioritas maka P1 akan diproses terlebih dahulu baru P2.

Misalnya terdapat empat barang P1, P2, P3, dan P4 dengan waktu kedatangan tiap barang masing-masing berturut-turut adalah 1, 2, 4, 5 dengan *burst time* 70, 30, 20, 40

Tabel 3. 2 *Contoh Proses 2*

<b>Proses</b>	<b>Waktu Kedatangan</b>	<b><i>Burst Time</i></b>
P1	1	70
P2	2	30
P3	4	20
P4	5	40

*Gantt chart:*



Bila proses dikerjakan berdasarkan *Shortest Remaining Time First* maka urutan penyelesaian proses berdasarkan prioritas adalah P1, P2, P3, P2, P4 dan P1. Pada hari pertama masuk barang P1 dengan waktu ketahanan 70 hari maka proses P1 dikerjakan dahulu karena tidak ada proses yang sedang mengantri, berselang sehari masuk barang P2 dengan waktu ketahanan 30 hari, oleh karena proses penyelesaian menggunakan *Shortest Remaining Time First* yang mengizinkan adanya interupsi terhadap proses yang sedang berjalan, maka proses P2 dikerjakan dan proses P1 mengantri. Pada hari keempat masuk barang P3 dengan waktu ketahanan 20 hari, oleh karena itu proses P2 yang sedang berjalan digantikan dengan proses P3 sebab prioritas P3 lebih tinggi. Barang P4 masuk pada hari kelima dengan waktu ketahanan 40 hari, berdasarkan SRTF P4 memiliki prioritas lebih rendah dibanding dengan P3, maka proses P4 masuk dalam antrian. Setelah proses P3 selesai, proses P2 dijalankan karena P2 memiliki prioritas lebih tinggi dalam antrian kemudian dilanjutkan dengan proses P4 dan P1.

### 3.4 *System Testing*

Pengujian dilakukan untuk menguji apakah sistem sudah sesuai dengan yang diinginkan atau dibutuhkan. Aplikasi yang telah jadi dan diimplementasikan dengan algoritma *Shortest Remaining Time First* untuk penjadwalannya, akan dicoba seberapa efisien penjadwalan yang dihasilkan oleh algoritma ini. Pengujian ini dilakukan oleh *admin* dengan menggunakan aplikasi yang telah dibuat. *Admin* yang menjadi *user* akan memasukkan beberapa barang yang sudah ada dengan memasukkan tanggal barang masuk serta tanggal kadaluarsa barang tersebut. Kemudian akan tersimpan dalam tabel *inventory* dan setelah itu dapat ditampilkan barang yang memiliki prioritas tertinggi untuk di keluarkan atau dijual terlebih dahulu berdasarkan ketahanan produk dari implementasi algoritma *Shortest*

*Remaining Time First*. Laporan tersebut memberikan urutan barang yang harus dijual terlebih dahulu dan dapat menjadi lebih efisien untuk mencegah kerugian yang disebabkan kerusakan barang karena kadaluarsa.

### 3.4.1 Rencana Pengujian Sistem

Pengujian aplikasi *web* dapat dilakukan dengan beberapa rangkaian pengujian secara lebih detail, yang terdiri dari (Vijay, 2015). Pada penelitian ini, penulis melakukan beberapa detail pengujian yaitu *functionality testing*, *usability testing*, dan *interface testing* dikarenakan tujuan dari penelitian yang hanya menyelesaikan permasalahan pada ruang lingkup berdasarkan pertimbangan peneliti yang tidak memerlukan kebutuhan keamanan dan performa aplikasi sehingga penulis tidak melakukan *security*, *compatibility* dan *performance*.

### 3.4.2 Rencana Pengujian Algoritma *Shortest Job First*

Pengujian algoritma pada penelitian ini akan digunakan untuk membuktikan bahwa algoritma dalam aplikasi berhasil diterapkan. Pengujian dilakukan berdasarkan pengujian integration melalui *whitebox testing* dan *blackbox testing* :

#### 1. *Whitebox testing*

*Whitebox testing* dilakukan dengan menguji struktur internal yang terdapat pada *code* program. Penulis akan menggunakan draft pengujian berdasarkan jurnal dari Lemos dkk. (Lemos, Franchin, & Masiero, 2009) dengan menambah kolom *actual result* dan *status*.

#### 2. *Blackbox testing*

Penulis melakukan pengujian dengan simulasi 2 skenario yang dibuat berdasarkan perkiraan kondisi yang memungkinkan terjadi dalam menjalankan aplikasi berdasarkan bisnis proses penjualan dan inventory pada minimarket pada umumnya.

### 3.5 *Maintenance*

Karena penelitian ini dilakukan sebagai syarat kelulusan tugas akhir, maka pemeliharaan sistem selanjutnya akan dilakukan oleh *admin* pada masing masing *minimarket/toko*

### 3.6 Metode Pengumpulan Data

Pengumpulan data selain melihat secara langsung keadaan yang ada di lapangan juga dilakukan dengan cara apa dan bagaimana data yang diperlukan dapat dikumpulkan sehingga akhir penelitian bisa menyajikan informasi yang valid dan reliable. Dalam pengumpulan data dilakukan beberapa metode sebagai berikut:

#### 1. Studi literatur

Tahap pertama dalam pengumpulan data adalah dengan melakukan studi literatur menggunakan referensi buku, jurnal dan artikel mengenai penelitian yang dilakukan, serta berbagai sumber yang mendukung dalam pembuatan dan pengembangan aplikasi. Literatur tersebut digunakan sebagai pedoman dalam melakukan penelitian

#### 2. Survei dan observasi

Dengan melakukan survei dan observasi di lapangan penulis dapat melihat hal-hal yang kurang atau tidak diamati orang lain serta dapat menemukan hal-hal yang di luar persepsi sehingga penulis memperoleh gambaran yang komperhensif. Dalam penelitian ini observasi dibutuhkan untuk dapat memahami masalah-masalah atau gejala dalam objek penelitian.

### 3.7 Evaluasi dan Pengujian

Pengujian dilakukan untuk mendeteksi adanya perbedaan antara kondisi *software* yang ada dengan kondisi yang diinginkan, untuk melihat kerusakan sistem

dengan melakukan evaluasi dari fitur-fitur *software*. Pengujian sistem dilakukan dengan mengidentifikasi adanya ketidaksesuaian hasil sebuah sistem dengan apa yang diharapkan, menilai dan mengevaluasi terhadap *output* atau hasil dari sistem, serta menguji terhadap *input* yang dimasukkan.

Evaluasi dilakukan dengan teknik *white box testing* dan *black box testing*. Pengujian *white box* perangkat lunak didasarkan pada hasil pengamatan secara teliti terhadap desain prosedural untuk memperoleh *test case*. Jalur-jalur logika yang melewati perangkat lunak diuji dengan memberikan *test case* yang menguji serangkaian kondisi atau *loop* tertentu untuk menentukan apakah kondisi yang diinginkan sudah sesuai dengan kondisi aktual. Kondisi aktual merupakan kondisi yang terjadi pada saat pengujian dengan menggunakan *input* data dari pengguna, untuk proses validasi dan verifikasi, sistem ketika diimplementasikan sesuai dengan yang diharapkan. Pengujian *black box* berfokus pada desain fungsional perangkat lunak serta berkaitan dengan *interface* perangkat lunak. Pada pengujian ini digunakan untuk memperlihatkan bahwa fungsi-fungsi *input* diterima dengan baik dan *output* dihasilkan dengan tepat, dengan parameter sistem *database* untuk pengecekan *input* data dan laporan data barang untuk pengecekan hasil *output*.



## BAB IV

### PEMBAHASAN

Bab ini membahas tentang perancangan dan pengembangan aplikasi penjualan dan *inventory minimarket* dengan menggunakan *shortest remaining time first algorithm*. Pengembangan aplikasi dilakukan dengan menggunakan metode waterfall.

#### 4.1 *Requirement Analysis*

Tahap ini diawali dengan pengidentifikasian kebutuhan sistem yang akan dikembangkan dalam bentuk *software* berbasis *web base*. Tahapan pertama dilakukan dengan observasi, survei, dan beberapa wawancara mendalam pada pihak yang terkait. Tahapan ini menghasilkan beberapa elemen yang berinteraksi dengan sistem yaitu:

1. Pelaku Sistem

*User* atau pengguna dalam aplikasi ini yang mengelola semua data disebut *user admin*. *User* aplikasi dibagi menjadi dua tipe yaitu *user admin* yang merupakan pemilik dari usaha dan *user kasir* yang merupakan pekerja. *User kasir* hanya dapat melakukan transaksi penjualan.

2. Identifikasi Kebutuhan Informasi

Kebutuhan informasi yang diperlukan dalam sistem penjualan dan *inventory* adalah data stok barang, laporan keuangan, laporan keluar – masuk barang, laporan penjualan.

3. Kebutuhan Pengguna

Pengguna pada aplikasi terdiri dari *owner* dan kasir. Kemampuan pengguna yang tidak terlalu memahami aplikasi, maka aplikasi dibuat sederhana mungkin agar pengguna mudah menggunakannya dalam melakukan transaksi.

#### 4. Kebutuhan Sistem

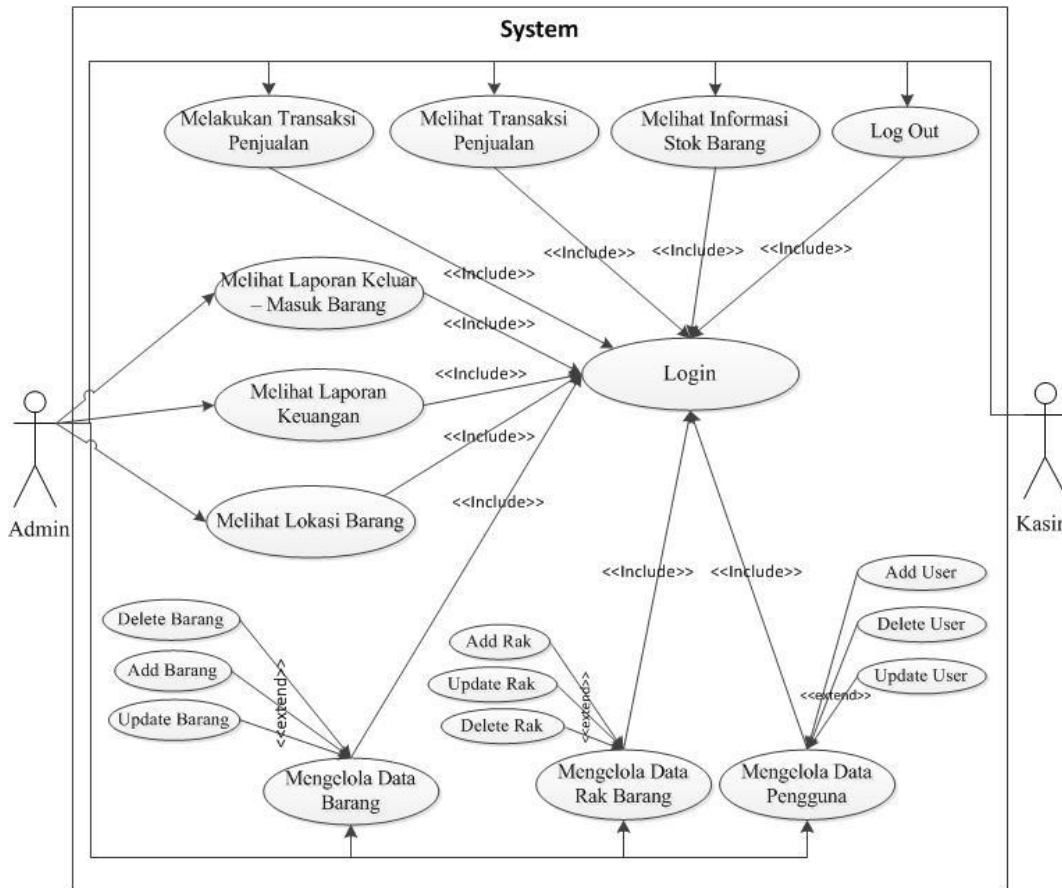
Pada penelitian ini terdapat kebutuhan fungsional yang menunjukkan fasilitas dan aktivitas yang dapat dilakukan oleh sistem. Kebutuhan fungsional sistem dibuat berdasarkan hasil pengumpulan data melalui observasi dan wawancara yang dibuat dalam elisitasi tahap I, II, III yang terlampir pada lampiran 4.

### 4.2 *System and Software Design*

Tujuan utama pada tahap ini adalah pemodelan yang sangat dibutuhkan dalam sebuah pengembangan *software*. Hasil dalam tahapan ini dijelaskan dalam model UML seperti diagram *use case*, *class* diagram, dan data model serta desain *user interface*. Adapun model perancangan aplikasi sebagai berikut.

#### 4.2.1 *Diagram Use case*

*Use case* merupakan suatu diagram yang menggambarkan fungsionalitas yang diharapkan dalam perancangan aplikasi. Berikut gambar rancangan *use case* diagram dari perancangan aplikasi penjualan dan *inventory*.



Gambar 4.1 Use Case Diagram

Pada gambar 4.1 ada beberapa *use case* yang memiliki relasi *include* dan *extend*. *Include* adalah relasi *use case* membutuhkan *use case* relasinya untuk menjalankan fungsinya, sedangkan *extend* adalah *use case* yang dapat berdiri sendiri tanpa relasinya, namun *use case* dapat digunakan untuk mengkategorikan *use case* – *use case* tersebut. Diagram *use case* akan dijelaskan secara lebih detail dalam tabel *use case scenario* berikut:

Tabel 4.1 Use Case Scenario Login

<b>Use case Name</b>	Login
<b>Use case ID</b>	1
<b>Actor</b>	Admin dan Kasir
<b>Description</b>	Use case ini menggambarkan kegiatan login ke dalam aplikasi penjualan dan inventory

<b><i>Pre-Condition</i></b>	Actor membuka aplikasi penjualan dan <i>inventory</i>	
<b><i>Trigger</i></b>	<i>Use case</i> ini dilakukan agar Actor dapat masuk ke dalam sistem aplikasi	
<b><i>Typical of events</i></b>	<b><i>Actor Action</i></b>	<b><i>System Response</i></b>
	1. Membuka Aplikasi	
	2. <i>Login</i>	
	3. Input inisial pengguna dan <i>password</i>	4. Cek inisial pengguna dan <i>password</i> 5. Menampilkan menu utama
<b><i>Alternate Course</i></b>	Jika inisial pengguna dan <i>password</i> salah, maka Actor harus <i>input</i> inisial pengguna dan <i>password</i> kembali.	
<b><i>Post-Condition</i></b>	Aplikasi menampilkan menu utama	

Tabel 4.2 *Use Case Scenario* Melihat Informasi Stok Barang

<b><i>Use case Name</i></b>	Melihat Informasi Stok Barang	
<b><i>Use case ID</i></b>	2	
<b><i>Actor</i></b>	Admin dan Kasir	
<b><i>Description</i></b>	<i>Use case</i> ini menggambarkan kegiatan melihat halaman informasi stok barang pada aplikasi	
<b><i>Pre-Condition</i></b>	Actor berada pada halaman utama aplikasi	
<b><i>Trigger</i></b>	<i>Use case</i> ini dilakukan agar Actor dapat melihat laporan stok barang	
<b><i>Typical of events</i></b>	<b><i>Actor Action</i></b>	<b><i>System Response</i></b>
	1. Pilih menu informasi stok	2. Menampilkan informasi stok barang dalam bentuk tabel
<b><i>Alternate Course</i></b>	Actor memilih menu stok barang dan memilih menu informasi stok	

<b>Post-Condition</b>	Aplikasi menampilkan menu utama, <i>Actor</i> dapat menambahkan / mengurangi stok barang, melakukan <i>filtering</i>
-----------------------	--

Tabel 4.3 *Use Case Scenario* Melihat Transaksi Penjualan

<b>Use case Name</b>	Melihat Transaksi Penjualan	
<b>Use case ID</b>	3	
<b>Actor</b>	Admin dan Kasir	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan melihat halaman transaksi penjualan hari ini	
<b>Pre-Condition</b>	<i>Actor</i> berada pada halaman utama aplikasi	
<b>Trigger</b>	<i>Use case</i> ini dilakukan agar <i>Actor</i> dapat melihat laporan transaksi penjualan hari ini	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu penjualan hari ini	2. Menampilkan informasi transaksi penjualan
<b>Alternate Course</b>	<i>Actor</i> memilih menu penjualan kemudian memilih menu penjualan hari ini	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.4 *Use Case Scenario* Melakukan Transaksi Penjualan

<b>Use case Name</b>	Melakukan Transaksi Penjualan	
<b>Use case ID</b>	4	
<b>Actor</b>	Admin dan Kasir	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk melakukan transaksi penjualan	
<b>Pre-Condition</b>	<i>Actor</i> berada pada halaman utama aplikasi	
<b>Trigger</b>	<i>Use case</i> ini dilakukan agar <i>Actor</i> dapat melakukan transaksi penjualan	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>

	1. Pilih menu penjualan baru	2. Menampilkan formulir penjualan
	3. <i>Input</i> kode barang	4. Menampilkan barang yang sesuai dengan <i>input</i>
	5. <i>Input</i> jumlah barang	6. Melakukan kalkulasi total harga barang yang di beli
	7. <i>Input</i> total bayar	8. Menampilkan jumlah uang kembali
	9. Menekan tombol simpan	10. Menyimpan semua data penjualan dalam <i>database</i> ketika proses berhasil, jika tidak berhasil memberikan notifikasi terhadap kesalahan
<b>Alternate Course</b>	Actor memilih menu penjualan kemudian memilih menu penjualan baru, menekan tombol ulangi untuk mengosongkan <i>field</i> pada formulir penjualan	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.5 Use Case Scenario Melihat Laporan Keluar – Masuk Barang

<b>Use case Name</b>	Melihat Laporan Keluar – Masuk Barang	
<b>Use case ID</b>	5	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan melihat halaman laporan keluar – masuk barang	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat melihat laporan keluar – masuk barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>

	1. Pilih menu laporan keluar – masuk barang	2. Menampilkan informasi laporan keluar – masuk barang dalam bentuk tabel
<b>Alternate Course</b>	Actor memilih menu laporan kemudian memilih menu laporan keluar – masuk barang	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama, melakukan <i>filtering</i> , mencetak laporan	

Tabel 4.6 Use Case Scenario Melihat Laporan Keuangan

<b>Use case Name</b>	Melihat Laporan Keuangan	
<b>Use case ID</b>	6	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan melihat halaman laporan keuangan	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat melihat laporan keuangan	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu laporan keuangan	2. Menampilkan informasi laporan keuangan
<b>Alternate Course</b>	Actor memilih menu laporan kemudian memilih menu laporan keuangan	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama, melakukan <i>filtering</i> , mencetak laporan, melihat rincian laporan	

Tabel 4.7 Use Case Scenario Melihat Lokasi Barang

<b>Use case Name</b>	Melihat Lokasi Barang
<b>Use case ID</b>	7
<b>Actor</b>	Admin
<b>Description</b>	Use case ini menggambarkan kegiatan melihat halaman laporan keuangan

<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat melihat lokasi barang disimpan	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu visualisasi	2. Menampilkan informasi barang dalam bentuk <i>chart</i>
<b>Alternate Course</b>	Actor memilih menu laporan kemudian memilih menu visualisasi	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.8 Use Case Scenario Add Barang

<b>Use case Name</b>	Add Barang	
<b>Use case ID</b>	8	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan untuk menambahkan data barang	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat menambah data barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu barang / produk	2. Menampilkan informasi barang yang tersedia
	3. Menekan tombol tambah	4. Menampilkan formulir pengisian tambah data barang
	5. <i>Input</i> nama, merek, satuan, harga beli, harga jual, stok, tanggal kadaluarsa, memilih rak untuk	



	6. Menekan tombol simpan	7. Menyimpan data dalam <i>database</i> dan memberikan notifikasi jika ada kesalahan dalam proses <i>input</i> barang
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.9 Use Case Scenario Update Barang

<b>Use case Name</b>	Update Barang	
<b>Use case ID</b>	9	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan untuk mengubah informasi barang	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat mengubah informasi barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu barang / produk	2. Menampilkan informasi barang yang tersedia
	3. Menekan <i>icon</i> atur ulang info	4. Menampilkan formulir <i>edit</i> data barang
	5. <i>Update</i> informasi yang ada dalam formulir	
	6. Menekan tombol simpan	7. <i>Update</i> data barang
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan halaman daftar barang / produk	

Tabel 4.10 *Use Case Scenario Delete Barang*

<b>Use case Name</b>	<i>Delete Barang</i>	
<b>Use case ID</b>	10	
<b>Actor</b>	<i>Admin</i>	
<b>Description</b>	<i>Use case ini menggambarkan kegiatan untuk menghapus data barang</i>	
<b>Pre-Condition</b>	<i>Actor berada pada halaman utama aplikasi</i>	
<b>Trigger</b>	<i>Use case ini dilakukan agar Actor dapat menghapus data barang</i>	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu barang / produk	2. Menampilkan informasi barang yang tersedia
	3. Menekan tombol <i>delete</i>	4. Menampilkan pesan peringatan untuk penghapusan barang
	5. Menekan tombol “Ya”	6. Menghapus barang dari <i>database</i>
<b>Alternate Course</b>	<i>Actor dapat membatalkan use case ini dengan menekan tombol tidak</i>	
<b>Post-Condition</b>	<i>Aplikasi menampilkan halaman daftar barang / produk</i>	

Tabel 4.11 *Use Case Scenario Add User*

<b>Use case Name</b>	<i>Add User</i>	
<b>Use case ID</b>	11	
<b>Actor</b>	<i>Admin</i>	
<b>Description</b>	<i>Use case ini menggambarkan kegiatan untuk menambahkan data pengguna</i>	
<b>Pre-Condition</b>	<i>Actor berada pada halaman utama aplikasi</i>	
<b>Trigger</b>	<i>Use case ini dilakukan agar Actor dapat menambah data pengguna</i>	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>

	1. Pilih menu pengguna	2. Menampilkan informasi seluruh pengguna
	3. Menekan tombol tambah	4. Menampilkan formulir pengisian tambah data pengguna
	5. <i>Input</i> inisial, nama pengguna, kata sandi, ulangi kata sandi, memilih hak akses pengguna	
	6. Menekan tombol simpan	7. Menyimpan data dalam <i>database</i> dan memberikan notifikasi jika ada kesalahan dalam proses <i>input</i> data pengguna
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.12 Use Case Scenario Update User

<b>Use case Name</b>	Update User	
<b>Use case ID</b>	12	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan untuk mengubah informasi pengguna	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat mengubah informasi pengguna	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu pengguna	2. Menampilkan informasi data pengguna yang tersedia

	3. Menekan <i>icon</i> atur ulang info	4. Menampilkan formulir ubah data pengguna
	5. <i>Update</i> informasi yang ada dalam formulir	
	6. Menekan tombol simpan	7. <i>Update</i> data pengguna
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan halaman daftar pengguna	

Tabel 4.13 Use Case Scenario Delete User

<b>Use case Name</b>	<i>Delete User</i>	
<b>Use case ID</b>	13	
<b>Actor</b>	<i>Admin</i>	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk menghapus data pengguna	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	<i>Use case</i> ini dilakukan agar Actor dapat menghapus data pengguna	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu pengguna	2. Menampilkan informasi daftar pengguna yang tersedia
	3. Menekan tombol <i>delete</i>	4. Menampilkan pesan peringatan untuk penghapusan data pengguna
	5. Menekan tombol “Ya”	6. Menghapus data pengguna dari <i>database</i>
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol tidak	

<b>Post-Condition</b>	Aplikasi menampilkan halaman daftar pengguna
-----------------------	--

Tabel 4.14 *Use Case Scenario Add Rak*

<b>Use case Name</b>	Add Rak	
<b>Use case ID</b>	14	
<b>Actor</b>	Admin	
<b>Description</b>	Use case ini menggambarkan kegiatan untuk menambahkan data rak barang	
<b>Pre-Condition</b>	Actor berada pada halaman utama aplikasi	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat menambah data rak barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu rak	2. Menampilkan informasi daftar rak barang
	3. Menekan tombol tambah	4. Menampilkan formulir pengisian tambah data rak barang
	5. Input nama rak	
	6. Menekan tombol simpan	7. Menyimpan data dalam <i>database</i> dan memberikan notifikasi apabila nama rak sudah tersedia
<b>Alternate Course</b>	Actor dapat membatalkan use case ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan menu utama	

Tabel 4.15 *Use Case Scenario Update Rak*

<b>Use case Name</b>	Update Rak
<b>Use case ID</b>	15
<b>Actor</b>	Admin

<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengubah informasi data rak barang	
<b>Pre-Condition</b>	<i>Actor</i> berada pada halaman utama aplikasi	
<b>Trigger</b>	<i>Use case</i> ini dilakukan agar <i>Actor</i> dapat mengubah informasi data rak barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Pilih menu rak	2. Menampilkan informasi data rak barang yang tersedia
	3. Menekan <i>icon</i> atur ulang info	4. Menampilkan formulir ubah data rak barang
	5. <i>Update</i> informasi yang ada dalam formulir	
	6. Menekan tombol simpan	7. <i>Update</i> data rak barang
<b>Alternate Course</b>	<i>Actor</i> dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<b>Post-Condition</b>	Aplikasi menampilkan halaman daftar rak barang	

Tabel 4.16 *Use Case Scenario Delete Rak*

<b>Use case Name</b>	<i>Delete Rak</i>	
<b>Use case ID</b>	16	
<b>Actor</b>	<i>Admin</i>	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk menghapus data rak barang	
<b>Pre-Condition</b>	<i>Actor</i> berada pada halaman utama aplikasi	
<b>Trigger</b>	<i>Use case</i> ini dilakukan agar <i>Actor</i> dapat menghapus data rak barang	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>

	1. Pilih menu rak	2. Menampilkan informasi daftar rak barang yang tersedia
	3. Menekan tombol <i>delete</i>	4. Menampilkan pesan peringatan untuk penghapusan data rak barang
	5. Menekan tombol “Ya”	6. Menghapus data rak barang dari <i>database</i>
<b>Alternate Course</b>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol tidak	
<b>Post-Condition</b>	Aplikasi menampilkan halaman daftar rak barang	

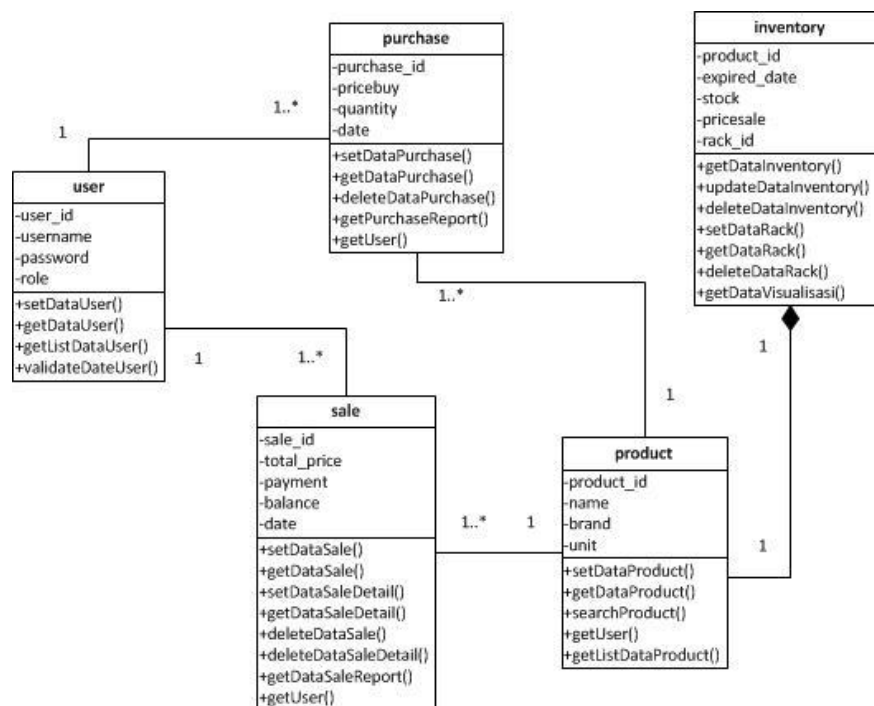
Tabel 4.17 Use Case Scenario Logout

<b>Use case Name</b>	Logout	
<b>Use case ID</b>	17	
<b>Actor</b>	Admin dan Kasir	
<b>Description</b>	Use case ini menggambarkan kegiatan <i>logout</i> dari dalam aplikasi penjualan dan <i>inventory</i>	
<b>Pre-Condition</b>	Actor menekan tombol pengguna pada halaman utama dan memilih “Keluar”	
<b>Trigger</b>	Use case ini dilakukan agar Actor dapat keluar dari dalam sistem	
<b>Typical of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Lakukan use case <i>login</i> ( <i>use case ID</i> 1)	
	2. Menekan tombol pengguna	3. Menampilkan pilihan “My Profile” dan “Keluar”
	4. Memilih pilihan “Keluar”	5. Menampilan menu <i>login</i>
<b>Alternate Course</b>		

<b>Post-Condition</b>	Aplikasi menampilkan halaman <i>login</i>
-----------------------	---

#### 4.2.2 Class Diagram

*Class* diagram menggambarkan kelas yang dibuat dengan hubungannya terhadap *class* lainnya. *Class* diagram tersebut bertujuan untuk merepresentasikan hubungan antar kelas pada aplikasi yang dibangun menggunakan *framework CodeIgniter*. Berikut adalah rancangan *class* diagram dari aplikasi penjualan dan *inventory minimarket*:



Gambar 4.2 *Class Diagram*

#### 4.2.3 Perancangan Database

Perancangan *database* adalah untuk menentukan isi dan pengaturan data yang dibutuhkan dalam suatu perancangan sistem. Rancangan *database* dimulai dari *conceptual design*, *logical design*, dan *physical design*.

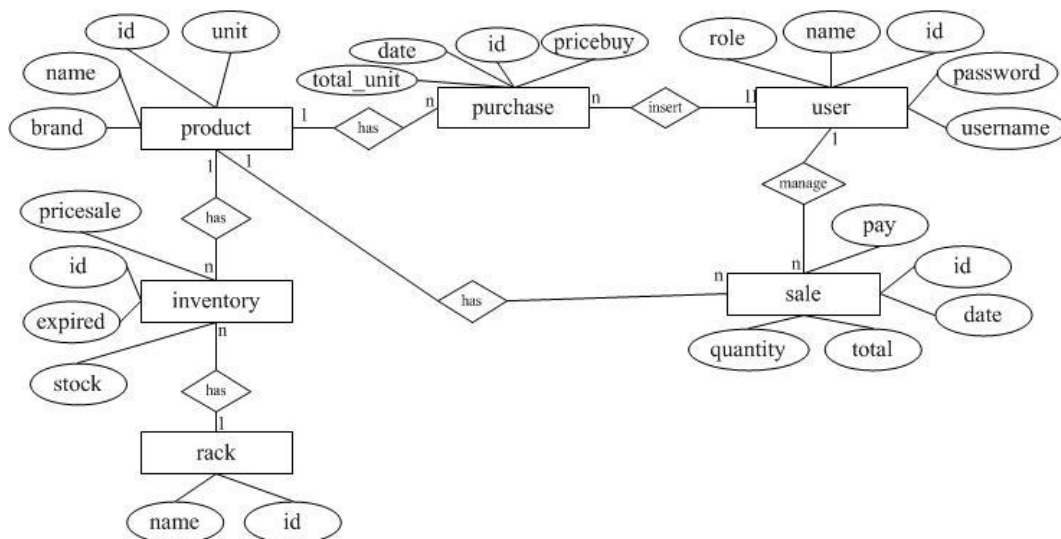
##### 1. *Conceptual Database Design*

Beberapa *entity* yang digunakan oleh aplikasi dijelaskan pada tabel *database entity* berikut ini.



Tabel 4.18 Database Entity

Nama Entitas	Deskripsi
<i>User</i>	Digunakan untuk menyimpan data <i>user</i> dari aplikasi
<i>Product</i>	Digunakan untuk menyimpan data barang yang ada dalam aplikasi
<i>Rack</i>	Digunakan untuk mengetahui tempat barang disimpan
<i>Inventory</i>	Digunakan untuk mengetahui rincian persediaan barang
<i>Purchase</i>	Digunakan untuk menyimpan data transaksi pembelian barang
<i>Sale</i>	Digunakan untuk menyimpan data transaksi penjualan barang.



Gambar 4.3 Conceptual Database Design

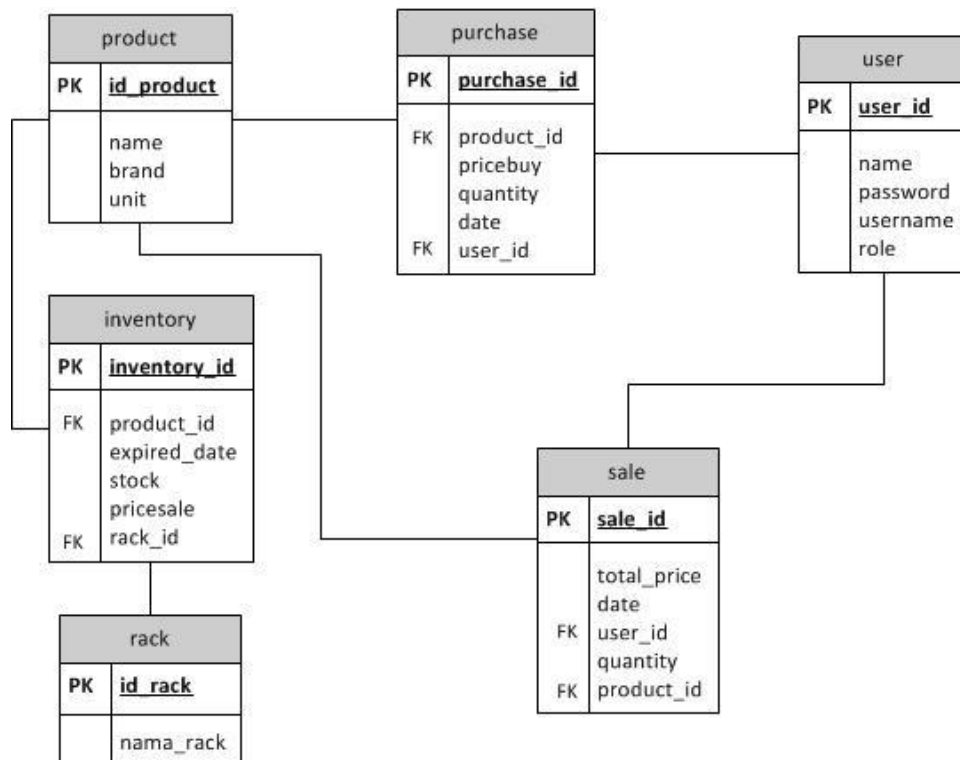
Setelah *entitiy* diidentifikasi, berikut ini adalah relasi antar *entitiy* yang dijelaskan pada tabel 4.19.

Tabel 4.19 Hubungan Antar *Entity*

No.	Nama <i>Entity</i>	Hubungan Antar <i>Entity</i>
1	<i>User</i>	<i>one-to many</i> dengan <i>purchase</i>
		<i>one-to many</i> dengan <i>sale</i>
2	<i>Product</i>	<i>one-to many</i> dengan <i>purchase</i>
		<i>one-to-many</i> dengan <i>sale</i>
		<i>one-to-many</i> dengan <i>inventory</i>
3	<i>Rack</i>	<i>one-to many</i> dengan <i>inventory</i>
4	<i>Inventory</i>	<i>many-to-one</i> dengan <i>product</i>
		<i>many-to-one</i> dengan <i>rack</i>
5	<i>Purchase</i>	<i>many-to-one</i> dengan <i>product</i>
		<i>many-to-one</i> dengan <i>user</i>
6	<i>Sale</i>	<i>many-to-one</i> dengan <i>product</i>
		<i>many-to-one</i> dengan <i>user</i>

## 2. Logical Database Design

Tahapan ini menampilkan skema *logical entity* dan *relationship* yang telah dijelaskan pada tahap *conceptual design* dengan menentukan *primary key*, *foreign key*, dan atribut pada masing-masing entitas. Berikut adalah gambaran dari *logical design* dari sistem yang akan dibuat.



Gambar 4.4 Logical Database Design

Definisi entitas dan atribut dapat dilihat pada tabel 4.20.

Tabel 4.20 Definisi Entitas dan Atribut

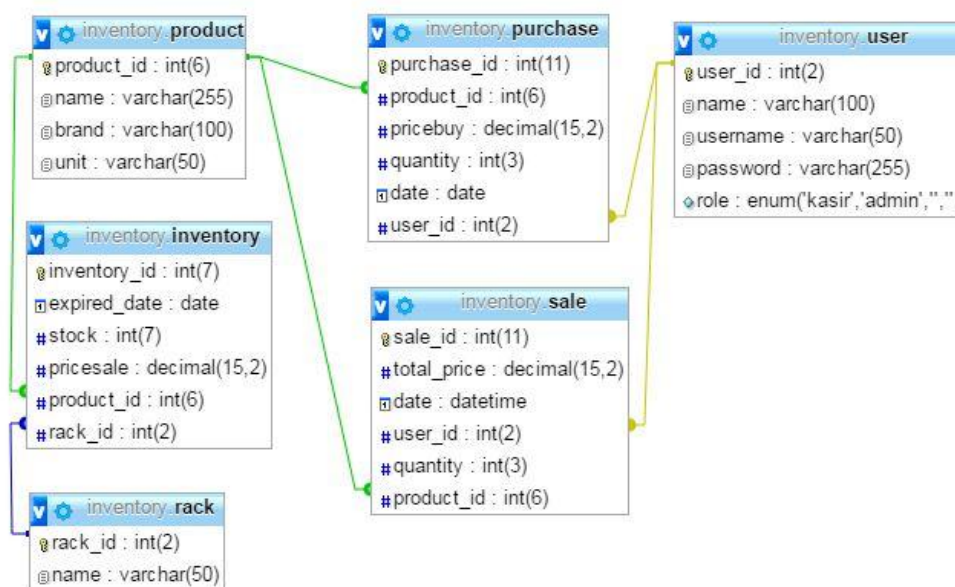
No.	Entitas	Atribut
1.	<i>User</i> entitas yang menyimpan data <i>user</i> yang berhak <i>login</i> ke aplikasi untuk mengakses aplikasi	<i>User_id</i> atribut yang menjadi identitas <i>user</i>
		<i>Username</i> atribut untuk melakukan proses <i>login</i>
		<i>Password</i> Atribut sebagai kata sandi untuk melakukan <i>login</i>
		<i>name</i> atribut nama <i>user</i>

		<i>Role</i>  Atribut untuk mengetahui hak akses <i>user</i> yang berhak mengelola data atau tidak (disebut sebagai <i>admin</i> dan kasir)
2.	<i>Product</i>  Entitas untuk menyimpan data barang	<i>Product_id</i>  Atribut yang menjadi identitas barang  <i>Name</i>  Atribut nama barang  <i>Brand</i>  Atribut merk barang  <i>Unit</i>  Atribut satuan barang
3.	<i>Rack</i>  Entitas untuk tempat penyimpanan barang	<i>Rack_id</i>  Atribut yang menjadi identitas tabel <i>rack</i>  <i>Name</i>  Atribut untuk nama rak barang
4.	<i>Inventory</i>	<i>Inventory_id</i>  Atribut yang menjadi identitas tabel <i>inventory</i>  <i>Expired_date</i>  Atribut tanggal kadaluarsa barang  <i>Stock</i>  Atribut untuk jumlah persediaan yang tersisa
5.	<i>Purchase</i>	<i>Purchase_id</i>

	Entitas untuk menyimpan transaksi pembelian yang dilakukan oleh <i>user</i>	Atribut yang menjadi identitas tabel <i>purchase</i>
		<i>pricebuy</i> Atribut untuk menyimpan harga barang beli
		<i>quantity</i> Atribut untuk menyimpan jumlah barang yang dibeli
		<i>date</i> Atribut untuk tanggal transaksi pembelian
6.	Sale  Entitas untuk menyimpan transaksi penjualan yang dilakukan oleh user	<i>Sale_id</i> Atribut yang menjadi identitas table <i>sale</i>
		<i>Total_price</i> Atribut untuk harga total barang belanja
		<i>date</i> Atribut untuk tanggal transaksi penjualan
		<i>Quantity</i> Atribut untuk mengetahui rincian jumlah barang

### 3. Physical Database Design

Model relasional atau *physical* data model adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data. Setiap tabel memiliki sejumlah kolom yang memiliki nama dan tipe datanya. PDM ini merupakan bentuk fisik perancangan basis data yang akan diimplementasikan ke DBMS (*Database Management System*). Berikut adalah gambar *physical database design* yang telah dilengkapi dengan tipe data yang digunakan pada masing-masing atribut.



Gambar 4.5 Physical Database Design

Tabel 4.21 Deskripsi Atribut Entity Database

Nama Tabel	Deskripsi
<i>User</i>	Tipe data untuk <i>primary key</i> pada tabel ini menggunakan tipe data <i>integer</i> karena berupa bilangan bulat dengan panjang dua karakter karena user dalam aplikasi ini tidak terlalu banyak. Tipe data <i>varchar</i> digunakan untuk data yang panjangnya tidak

	tetap. Tipe data untuk <i>user_role</i> menggunakan tipe <i>enum</i> yang memiliki tipe yang sudah ditentukan.
<i>Product</i>	Tipe data untuk <i>primary key</i> tabel ini menggunakan tipe <i>integer</i> dengan panjang enam karakter agar dapat menampung banyak <i>product</i> . Tabel ini merupakan tabel master yang sifatnya tidak dapat berubah. Terdiri dari <i>name</i> , <i>brand</i> dan <i>unit</i> menggunakan tipe data <i>varchar</i> karena data yang dimasukkan memiliki panjang yang tidak tetap.
<i>Rack</i>	Tipe data <i>primary key</i> menggunakan tipe data <i>integer</i> dengan panjang dua karakter karena perkiraan jumlah maksimal rak yang ada. Tipe data <i>name</i> menggunakan <i>varchar</i> karena data yang dimasukkan memiliki panjang yang tidak tetap.
<i>Inventory</i>	Tipe data <i>primary key</i> menggunakan tipe data <i>integer</i> dengan panjang tujuh karakter untuk menampung banyak prsediaan. Atribut yang menjadi <i>foreign key</i> juga menggunakan tipe data <i>integer</i> yang menyesuaikan dengan <i>primary key</i> . <i>Stock</i> menggunakan tipe data <i>integer</i> karena merupakan bilangan bulat. <i>Expired date</i> menggunakan tipe data <i>date</i> untuk menyimpan informarsi waktu dan tanggal atau dalam format tanggal.
<i>Purchase</i>	Tipe data <i>primary key</i> menggunakan tipe data <i>integer</i> dengan panjang karakter sebelas, agar dapat menampung banyak transaksi pembelian sama dengan atribut yang menjadi <i>foreign key</i> juga menggunakan tipe data <i>integer</i> yang menyesuaikan dengan <i>primary key</i> nya. Tipe data <i>pricebuy</i> menggunakan tipe data <i>decimal</i> . Tipe data <i>quantity</i> menggunakan tipe data <i>integer</i> karena berupa bilangan bulat dan tipe data <i>date</i>

	menggunakan tipe data <i>datetime</i> untuk menyimpan informasi waktu dan tanggal atau dalam format tanggal.
<i>Sale</i>	Tipe data <i>primary key</i> menggunakan tipe data <i>integer</i> dengan panjang karakter sebelas, agar dapat menampung banyak transaksi penjualan sama dengan atribut yang menjadi <i>foreign key</i> juga menggunakan tipe data <i>integer</i> yang menyesuaikan dengan <i>primary key</i> nya. Tipe data <i>total_price</i> menggunakan tipe data <i>decimal</i> karena menampung data pecahan. Tipe data <i>date</i> menggunakan tipe data <i>datetime</i> untuk menyimpan informasi waktu dan tanggal atau dalam format tanggal.

#### 4.2.4 Perancangan *User Interface*

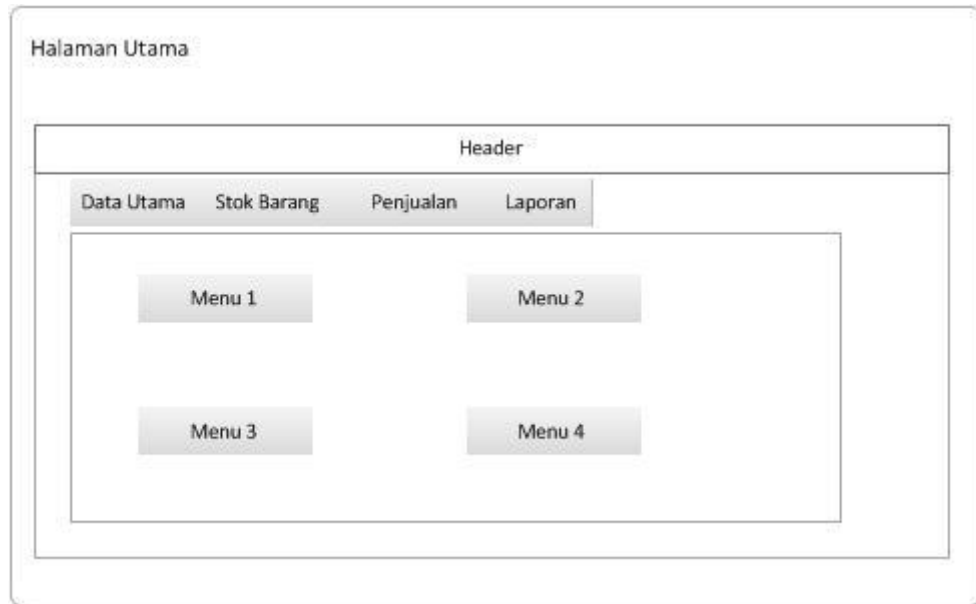
Berikut adalah rancangan *Graphic User Interface* (GUI) dari aplikasi penjualan dan *inventory* yang akan dibuat.

The image shows a wireframe for a login page. It consists of a main container box labeled 'Halaman Login' at the top left. Inside this container is a smaller box containing the following elements from top to bottom: the text 'Selamat Datang', a text input field labeled 'Username', another text input field labeled 'Password', and a button labeled 'Masuk'.

Gambar 4.6 GUI Halaman *Login*

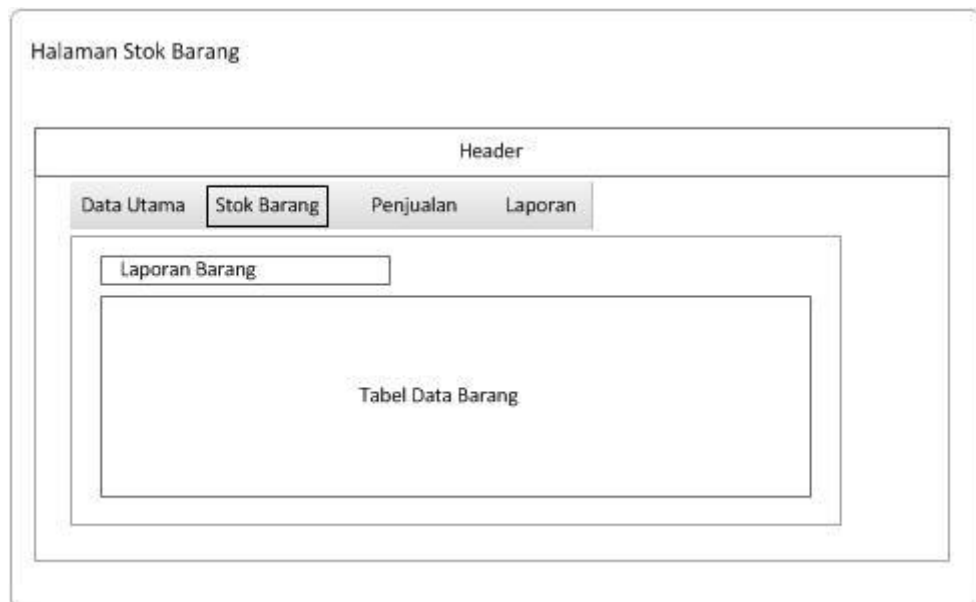


Gambar 4.6 menampilkan halaman ketika *user* membuka aplikasi saat pertama kali sistem dijalankan. *User* harus memasukkan *username* dan *password* dengan benar agar dapat masuk ke dalam menu utama.



Gambar 4.7 GUI Halaman Utama

Pada halaman utama Gambar 4.7 terdapat beberapa menu navigasi yang dapat dipilih yaitu menu data utama, stok barang, penjualan dan laporan.



Gambar 4.8 GUI Halaman Stok Barang

Tampilan pada Gambar 4.8 adalah tampilan menu stok barang. Pada saat *user* memilih menu stok barang, aplikasi menampilkan seluruh data barang dalam bentuk tabel. Dalam halaman ini *user* dapat menambahkan atau mengurangi stok barang yang dalam *inventory*.

Halaman Penjualan

Header

Data Utama Stok Barang **Penjualan** Laporan

Rincian Barang

Total

Total Bayar

Simpan

Gambar 4.9 GUI Halaman Penjualan

Gambar 4.9 menggambarkan halaman menu penjualan pada saat *user* memilih *tab* menu penjualan. Halaman ini berisi formulir untuk melakukan transaksi penjualan untuk diisi oleh *user*. *User* harus memasukkan jumlah total bayar sebelum menyimpan transaksi.



Gambar 4.10 GUI Halaman Laporan

Tampilan pada Gambar 4.10 adalah tampilan menu laporan, *user* dapat melihat laporan keluar – masuk barang, laporan keuangan, dan laporan visualisasi yang dilengkapi dengan rincian yang jelas per laporan.

Untuk sistem pengeluaran barang dipergunakan metode SRTF (Shortest Remaining Time First) atau barang dengan kadaluarsa tercepat akan dikeluarkan paling awal. Hal ini ditujukan untuk mencegah agar tidak terjadi perbedaan kualitas antara produk yang sama serta mencegah kerusakan pada barang tersebut.

#### 4.2.5 Perancangan Visualisasi

Konsep perancangan visual dalam aplikasi ini adalah visualisasi terhadap posisi barang yang berada di dalam gudang dengan menggunakan visualisasi 2D yaitu visualisasi statis (tak bergerak) seperti pada Gambar 4.11. Peletakan barang menggunakan aturan penempatan secara berkelompok yaitu ke dalam rak, dalam rak terdiri dari beberapa blok, barang yang sama akan diletakkan pada tempat (blok) yang sama. Barang dengan frekuensi kadaluarsa tercepat diletakkan pada posisi depan. Tujuannya agar mempermudah proses pengeluaran barang yang harus dijual terlebih dahulu.

Untuk sistem pengeluaran barang dipergunakan metode SRTF (*Shortest Remaining Time First*) atau barang dengan kadaluarsa tercepat akan dikeluarkan paling awal. Hal ini ditujukan untuk mencegah agar tidak terjadi perbedaan kualitas antara produk yang sama serta mencegah kerusakan pada barang tersebut.

<u>Nama Rak 1</u>	<u>Nama Rak 2</u>	<u>Nama Rak 3</u>	<u>Nama Rak 4</u>	
Barang 1	Barang 6	Barang 11	Barang 16	Barang 21
Barang 2	Barang 7	Barang 12	Barang 17	Barang 22
Barang 3	Barang 8	Barang 13	Barang 18	Barang 23
Barang 4	Barang 9	Barang 14	Barang 19	Barang 24
Barang 5	Barang 10	Barang 15	Barang 20	Barang 25

Gambar 4.11 GUI Halaman Visualisasi

#### 4.2.6 Pseudocode sistem

Berikut adalah penjabaran alur sistem dengan pseudocode dari aplikasi penjualan dan *inventory*:

```
// User login
declare username, password
input username
input password
function login()
GET username
GET password
    if (username==username &&password==password)
    then login-success
    if (login==success)
    then menuUtama()
    end if
```

```

        else login-failed
        end if

// Menu Utama
start
input pilih menu
if (pilih menu == master) then
    function master()
else if (pilih menu == laporan) then
    function laporan()
else if (pilih menu == lokasi) then
    function visualisasi()
end if
end

// Master Barang
start
function master()
input pilih menu
if (pilih menu == dataBarangMasuk) then
    input pilih submenu
    if (pilih submenu == inputBarang) then
        input kodeBarang
        input namaBarang
        input satuan
        input stok
        input harga
    else if (pilih submenu == delete) then
        get barang
        delete barang
    else if (pilih submenu == edit) then
        get barang
        edit barang
    end if
end if
end

```

```

        end if
        save
    else if (pilih menu == dataBarangKeluar) then
        input barangKeluar
        save
    end if
end
// Laporan
start
function laporan()
input pilih menu
if (pilih menu == laporanBarangMasuk) then
    print laporanBarangMasuk
else if (pilih menu == laporanBarangKeluar) then
    print laporanBarangKeluar
else if (pilih menu == dataBarang) then
    print dataBarang
end if
end

// Visualisasi
start
function visualisasi()
int n
n = dataBarang[]
for (int i = 0; i < n; i++)
    print dataBarang
    print visualisasi
end for
end

```

### 4.3 Implementation

Tahap *implementation* adalah tahap implementasi dari model perancangan. Pada tahapan ini desain program diterjemahkan ke dalam kode – kode dengan menggunakan bahasa pemrograman. Tahap implementasi dibagi menjadi tiga bagian yaitu implementasi sistem, implementasi *user interface* dan implementasi visualisasi yang mengacu pada algoritma *shortest remaining time first*.

#### 4.3.1 Implementasi Sistem

Tahapan implementasi sistem adalah mengimplementasikan semua aktifitas yang dilakukan pada saat tahap perancangan. Pada tahapan ini terdapat beberapa proses yang dilakukan untuk membuat aplikasi penjualan dan *inventory*. Berikut adalah informasi *hardware* dan *software* yang digunakan dalam tahapan implementasi.

Informasi *hardware* yang digunakan dalam pengembangan sistem adalah sebagai berikut:

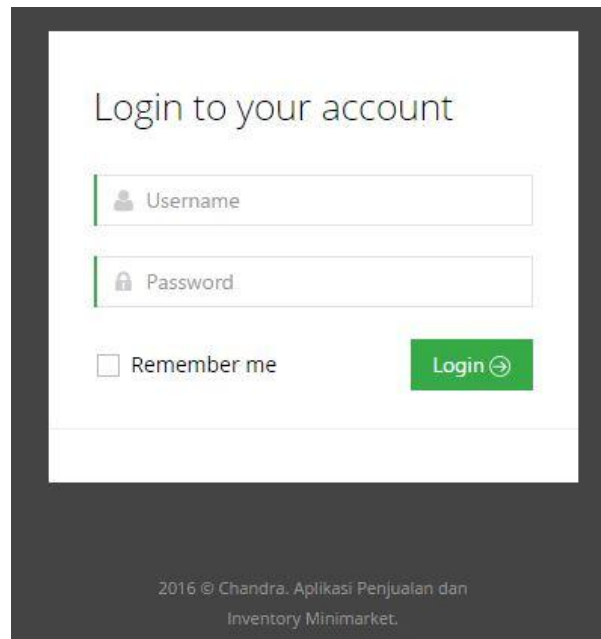
1. *Device* : Notebook PC Asis N46VJ
2. *Operating System* : Windows® 7 Ultimate™ 64-bit
3. *Processor* : Intel® Core™ i5-3210M CPU @ 2.5 GHz
4. *Memory* : 4.00 GB RAM

Informasi *software* untuk pengembangan aplikasi adalah sebagai berikut:

1. XAMPP™ versi 3.2.1, sebagai sebagai *web server*, *database server* dan *application server*.
2. Google® Chrome™ versi 48.0.2564.109 m, sebagai web browser
3. Notepad++™ versi 6.8.6 sebagai IDE untuk membangun aplikasi

### 4.3.2 Implementasi GUI (*Graphical User Interface*)

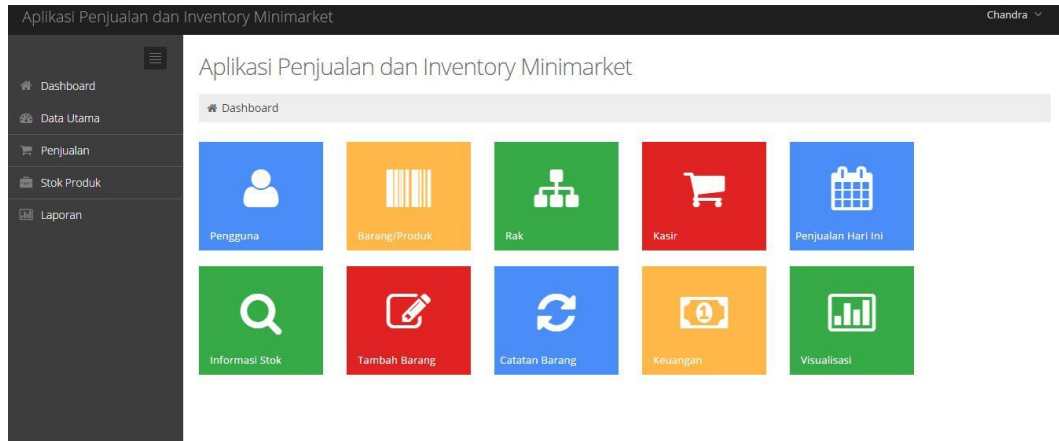
GUI merupakan suatu antar muka yang memungkinkan pengguna berinteraksi secara grafis dengan program komputer untuk memudahkan pengguna dalam menggunakan aplikasi penjualan dan *inventory minimarket* melalui gambar-gambar grafik, ikon, menu dan menggunakan perangkat penunjuk (*pointing device*). Berikut adalah tampilan dari GUI yang telah diimplementasikan:



Gambar 4.12 Halaman *Login*

Halaman pertama yang ditampilkan pada saat *admin* (pengguna) membuka aplikasi penjualan dan *inventory minimarket* adalah tampilan halaman *login*. *User* harus memasukkan “*username*” dan “*password*” dengan benar sehingga *user* dapat akses masuk ke dalam aplikasi. Terdapat dua tipe *user* dalam aplikasi ini yaitu “*Admin*” dan “*Kasir*”. *User admin* dapat mengakses semua menu dalam aplikasi sedangkan *user kasir* hanya bisa mengakses menu “*Stok Barang*” dan “*Penjualan*”.





Gambar 4.13 Halaman Awal

Halaman di atas adalah halaman awal aplikasi Penjualan dan *Inventory Minimarket*. Terdapat menu navigasi pada bagian samping seperti “Data Utama”, “Stok Produk”, “Penjualan”, dan “Laporan”.



Gambar 4.14 Halaman Data Utama

Halaman data utama menampilkan tiga sub menu yaitu “Pengguna”, “Barang / Produk”, dan “Rak”. Pada halaman ini *user admin* dapat melihat pengguna, daftar barang dan daftar rak yang ada dalam aplikasi.

## Aplikasi Penjualan dan Inventory Minimarket

Data Utama > Pengguna

+ Tambah

Daftar Pengguna

Nama	Username	Hak Akses	Pengaturan
Chandra Setiawan	admin2	admin	View Edit Delete
Sawitri	sawitri	admin	View Edit Delete

Gambar 4.15 Halaman Daftar Pengguna

Bagi *user admin*, mereka dapat melihat daftar pengguna dalam aplikasi. Informasi dan status hak akses dapat dilihat dalam bentuk tabel seperti Gambar 4.15

## Tambah Data

Data Utama > Pengguna > Tambah Pengguna

Tambah Data Pengguna

Nama

Username

Password

Hak Akses

Save Cancel

Gambar 4.16 Halaman Tambah Data Pengguna

Gambar 4.16 menampilkan halaman untuk menambah data pengguna untuk aplikasi Penjualan dan Inventory Minimarket. Terdapat formulir yang harus diisi untuk kelengkapan data pengguna berupa “Nama Pengguna”, “*username*”, “*password*”, dan “Hak Akses”. Semua baris yang ada dalam formulir harus diisi oleh *user*.

## Ubah Data

Ubah Data Pengguna Chandra Setiawan

Nama

Username

Password

Hak Akses

Gambar 4.17 Halaman Ubah Data Pengguna

Gambar 4.17 menampilkan halaman data pengguna untuk melakukan perubahan “Nama Pengguna”, “*username*”, “*password*”, dan “Hak Akses”. *User admin* bisa melakukan perubahan *password* apabila dibutuhkan.

## Aplikasi Penjualan dan Inventory Minimarket

Daftar Barang / Produk

10 records per page Search:

Kode Barang	Nama	Merek	Unit	Pengaturan
4	Minuman Botol	Aqua	Minuman Botol	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Deterjen Cair 25ml	Rinso	Deterjen Cair 25ml	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 2 of 2 entries  1

Gambar 4.18 Halaman Daftar Barang / Produk

*User admin* diberikan akses untuk melihat daftar barang / produk yang ada dalam *inventory*. Informasi lengkap suatu barang dapat dilihat pada halaman ini yang dibuat dalam bentuk tabel seperti Gambar 4.18.

## Tambah Data

Gambar 4.19 Halaman Tambah Data Barang / Produk

Gambar 4.19 menampilkan formulir untuk penambahan barang / produk. *User* harus mengisi semua baris yang ada dalam formulir agar proses penambahan barang berhasil. Dalam menambah barang / produk baru yang pertama adalah mengisi info nama produk, nama merek dan satuan.


## Aplikasi Penjualan dan Inventory Minimarket

No	Nama	Pengaturan
1	Rak A2rr	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	Rak BCD	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	Rak EFG	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Gambar 4.20 Halaman Daftar Rak

*User admin* dapat melihat daftar rak yang tersedia. Informasi daftar rak ditampilkan dalam bentuk tabel seperti Gambar 4.20

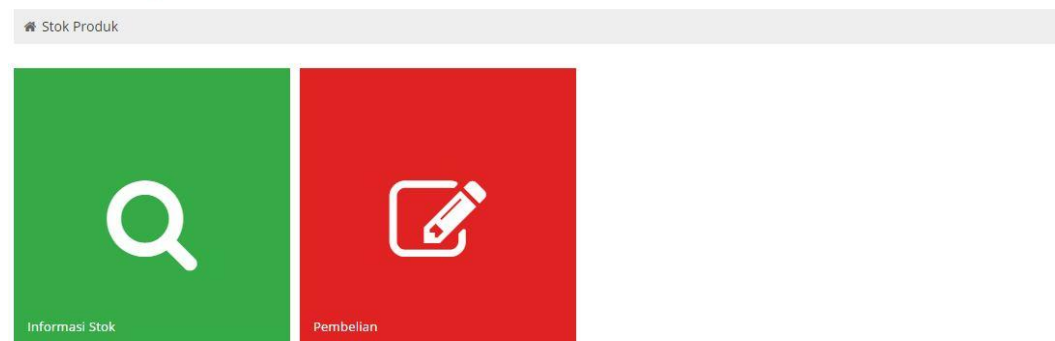
## Tambah Data



Gambar 4.21 Halaman Tambah Data Rak Barang

Untuk menambah data rak barang, *user* harus mengisi “Nama Rak” yang tersedia pada halaman tambah data rak barang pada Gambar 4.21.

## Aplikasi Penjualan dan Inventory Minimarket



Gambar 4.22 Halaman Stok Produk

Gambar 4.22 adalah halaman stok produk, halaman ini dapat diakses oleh *user admin* dan *user kasir*. Tersedia dua menu yaitu “Informasi Stok” dan “Pembelian”. Menu informasi stok menyediakan informasi daftar stok produk yang ada dalam *inventory* dan menu pembelian berfungsi untuk melakukan penambahan barang yang dibeli.

## Aplikasi Penjualan dan Inventory Minimarket

Stok Produk > Stok Barang

Daftar Informasi Stok

10 records per page Search:

No	Rak	Kode	Produk	Stok	Satuan	Kadaluarsa	Pengaturan
1	Rak A2rr	4	Minuman Botol	23	liter	2016-09-22	<a href="#">Edit</a> <a href="#">Delete</a>
2	Rak A2rr	4	Minuman Botol	323	liter	2016-09-30	<a href="#">Edit</a> <a href="#">Delete</a>
3	Rak BCD	5	Deterjen Cair 25ml	332	pack	2016-09-24	<a href="#">Edit</a> <a href="#">Delete</a>
4	Rak EFG	5	Deterjen Cair 25ml	23	pack	2016-09-30	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Gambar 4.23 Halaman Daftar Stok Barang / Produk

Gambar 4.23 menampilkan daftar stok produk yang ada dalam *inventory*. Informasi ditampilkan dalam bentuk tabel.

## Aplikasi Penjualan dan Inventory Minimarket

Stok Produk > Pembelian

Tambah Data Produk

Produk:

Jumlah:

Harga Beli:  dalam Rupiah

Harga Jual:  dalam Rupiah

Tanggal Kadaluarsa:

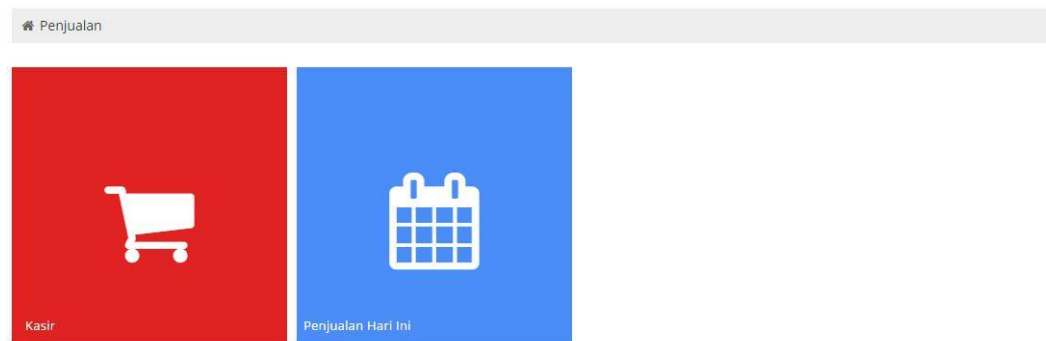
Rak:

[Save](#) [Cancel](#)

Gambar 4.24 Halaman Pembelian Produk

Halaman ini digunakan untuk menambah data pembelian apabila dilakukan transaksi pembelian. Halaman ini berisi *form* yang digunakan untuk memasukkan data pembelian ke dalam *database*. Setelah semua data telah diisi, kemudian tekan tombol *save* yang digunakan untuk mengirim data ke dalam *database*. seperti Gambar 4.24

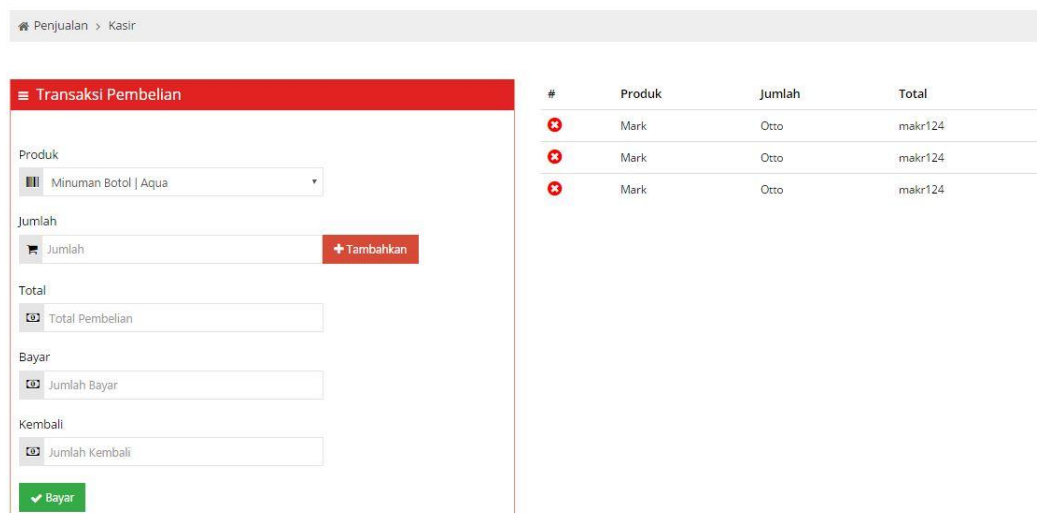
## Aplikasi Penjualan dan Inventory Minimarket



Gambar 4.25 Halaman Penjualan

Gambar 4.25 merupakan tampilan halaman penjualan, halaman penjualan dapat diakses oleh *user admin* dan *user kasir*. Ada dua menu yang terdapat pada halaman ini yaitu “Kasir” dan “Penjualan Hari Ini”. Menu kasir digunakan untuk melakukan transaksi penjualan dan menu penjualan hari ini menyediakan informasi detail transaksi pada hari itu yang disajikan dalam bentuk tabel.

## Aplikasi Penjualan dan Inventory Minimarket



Gambar 4.26 Halaman Penjualan Barang / Produk

Halaman ini digunakan untuk menambah data penjualan apabila dilakukan transaksi penjualan yang baru. Halaman ini berisi *form* yang digunakan untuk memasukkan data penjualan ke dalam *database*. Terdapat 3 tahapan dalam melakukan transaksi penjualan, yang pertama adalah memilih barang / produk. Tahap kedua adalah menambah jumlah barang / produk yang dibeli, lalu tahap ketiga memasukkan jumlah bayar.

## Aplikasi Penjualan dan Inventory Minimarket

Penjualan > Data Penjualan

Daftar Penjualan Hari Ini

10 records per page Search:

No Nota	Nama Penanggung Jawab	Waktu	Total Penjualan	Pengaturan
1	Chandra	23.00	760000	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

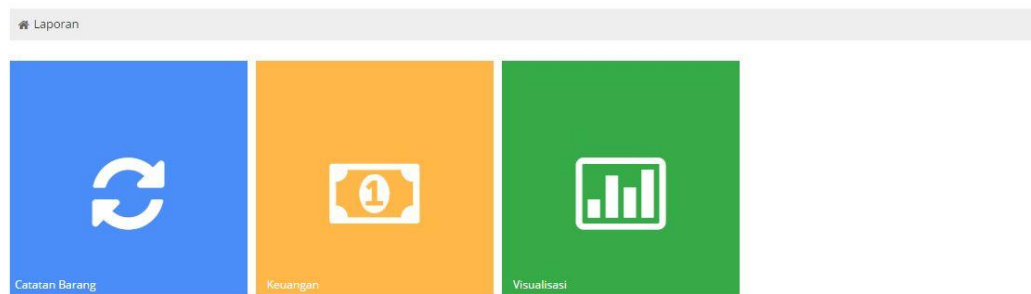
Showing 1 to 1 of 1 entries

← Previous 1 Next →

Gambar 4.27 Halaman Penjualan Barang Hari Ini

Gambar 4.27 menampilkan penjualan barang / produk hari ini, informasi ditampilkan dalam bentuk tabel.

## Aplikasi Penjualan dan Inventory Minimarket



Gambar 4.28 Halaman Laporan

Gambar 4.28 merupakan tampilan halaman laporan, *user* dapat melihat catatan barang, laporan keuangan dan visualisasi barang dalam halaman ini.

## Aplikasi Penjualan dan Inventory Minimarket

Laporan > Catatan

Daftar Catatan Barang

10 records per page Search:

No	Tanggal	Jenis	Penanggung Jawab	Pengaturan
1	23 Agus 2016	Pembelian	Chandra	<a href="#">View</a> <a href="#">Delete</a>

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Gambar 4.29 Halaman Keluar - Masuk Barang

Halaman di atas menampilkan laporan keluar – masuk barang, terdapat dua jenis informasi yang ada dalam laporan ini yaitu penjualan dan pembelian. Informasi lengkap ditampilkan dalam bentuk tabel seperti Gambar 4.29.



## Aplikasi Penjualan dan Inventory Minimarket

Laporan > Keuangan

Daftar Laporan Keuangan

10 records per page Search:

No	Tanggal	Penanggung Jawab	Total Penjualan	Total Pembelian	Pengaturan
1	23 Agus 2016	Chandra	230000	23330000	<a href="#">View</a> <a href="#">Delete</a>

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Gambar 4.30 Halaman Laporan Keuangan

Gambar 4.30 menampilkan laporan keuangan yang digunakan untuk melihat status keuangan dari total penjualan, total pembelian. Laporan keuangan dibuat dalam bentuk tabel.

## Aplikasi Penjualan dan Inventory Minimarket

Laporan > Visualisasi

Rak EFG Rak BCD Rak A Visualisasi Stok Dalam Rak

Minuman Botol Aqua 2016-09-22	23
Minuman Botol Aqua 2016-09-30	323
Minuman Botol Aqua 2016-10-14	34
Minuman Botol Aqua 2016-10-29	12
Minuman Botol Aqua 2016-11-24	1

Gambar 4.31 Halaman Visualisasi

Gambar 4.31 menampilkan visualisasi stok produk yang ada pada rak, informasi yang disediakan berupa nama barang, jumlah barang dan tanggal kadaluarsa. Visualisasi membantu *user* untuk menemukan posisi dan letak barang dengan kadaluarsa paling cepat sehingga *user* bisa menjual produk itu terlebih dahulu.

### 4.3.3 Implementasi Algoritma SRTF

Algoritma SRTF (*Shortest Remaining Time First*) adalah algoritma yang diimplementasikan ke dalam aplikasi untuk menyelesaikan sebagian permasalahan yang terdapat dalam penelitian ini. Pada aplikasi ini, algoritma *shortest remaining*

*time first* dijadikan alur algoritma dalam menyediakan informasi barang dalam bentuk visualisasi dalam memberikan informasi lokasi, posisi, dan barang yang harus dijual terlebih dahulu.

Algoritma *shortest remaining time first* diimplementasikan ke bentuk kode dalam program untuk menghasilkan visualisasi susunan barang yang harus dijual terlebih dahulu. Penerapan algoritma SJF di terapkan dalam aplikasi ini dalam sebuah fungsi yang diberi nama *bar\_char()*, fungsi tersebut merepresentasikan alur logika dari algoritma SJF yang disesuaikan dengan kebutuhan dan kondisi pada aplikasi. Pertama variable data dideklarasikan sebagai *array*, kemudian data diambil dari *database* melalui *query* yang dimasukkan ke dalam variabel *racks* data yang diambil dari table *ukm\_rack* yang memiliki status *rack\_available* adalah 1, hasil tersebut kemudian akan dimasukkan ke dalam *array* dua dimensi yaitu *\$racks[\$i]['products']* data *list product* diambil dari *database* *ukm\_product* dengan kondisi *product\_inventory* lebih besar dari nol artinya masih ada jumlah tersedianya, *rack\_id* yang sesuai dengan *rack\_id* tiap *product*, dan diurutkan berdasarkan urutan tanggal kedaluarsa pada *field product\_expired*. Selanjutnya hasil *array* tersebut akan dikirim melalui variabel data, saat *load view* *template\_visual2* untuk ditampilkan kepada *user* dalam GUI.

```
function bar_chart()
{
    $data = array();
    $racks = $this->db->get_where('ukm_rack', array('rack_available' => 1))->
    result_array();
    foreach($racks as $i => $rack) {
        if($this->year) {
            $this->db->like('product_expired', $this->year, 'both');
        }
        $racks[$i]['products'] = $this->db->order_by('product_expired', 'ASC')->
        get_where('ukm_product', array('product_inventory >'=>0, 'rack_id'=>$rack[
        'rack_id']))->result_array();
    }
    $data['racks'] = json_decode(json_encode($racks));
    return $this->load->view('template_visual2', $data, true);
}
```

Gambar 4.32 Implementasi Algoritma SRTF

#### 4.4 *System Testing*

Pengujian merupakan bagian penting dalam pengembangan aplikasi untuk memastikan bahwa aplikasi telah memenuhi *requirement*. Pada tahapan ini, penulis melakukan pengujian sistem dan pengujian algoritma yang diterapkan pada aplikasi. Pengujian sistem terdiri dari pengujian *functionality* dan pengujian *usability*.

##### 4.4.1 Pengujian Sistem

Pengujian sistem ini menggunakan pengujian komperhensif berdasarkan pedoman pengujian aplikasi *web* (Vijay, 2015). Teknik sampling menggunakan sampel purposive (Palys, 2008) dengan 10 potential *user*. Terdapat 10 *user* yang memiliki usaha *minimarket* yang berada di daerah menteng atas dan pasar rumput. Pada penelitian ini pengujian yang dilakukan adalah pengujian *functionality*, *usability*, dan *interface*. Semua potensial *user* tersebut melakukan pengujian *functionality*, *usability*, dan *interface* menggunakan *acceptance testing* atau yang sering disebut *User Acceptance Testing* (UAT).

##### 1. *Functionality Testing*

Hasil dari pengujian *functionality* pada aplikasi penjualan dan *inventory minimarket* dapat dilihat pada halaman lampiran. Pengujian dilaksanakan oleh 10 *user* potential. Semua penguji berhasil melakukan pengujian pada semua fungsi yang diujikan. Berdasarkan hasil tersebut maka dapat disimpulkan bahwa semua fungsi dalam aplikasi telah berhasil dengan baik.

##### 2. *Usability Testing*

Pada penelitian ini, pengujian *usability* merupakan analisis perilaku *user* dan mengarah pada temuan bagaimana mudahnya *user interface* digunakan. Penyusunan soal dalam pengujian *usability* dibuat berdasarkan lima kualitas komponen aspek *usability*. Terdapat 20 buah pertanyaan untuk merepresentasikan komponen *usability* menggunakan grafik diagram yang mengidentifikasi prespektif *user*. Analisis hasil pengujian ini dilakukan dengan menggunakan skala Likert (Boone, 2012).

Dalam penentuan hasil akhir pengujian (Simamora, 2005) ditentukan nilai skala dan skor dengan rumus sebagai berikut:

$$\text{Total Skor} = \sum(\text{skor} \times \text{jumlah responden})$$

$$\text{Min Range} = \text{skor minimal} \times \text{jumlah seluruh responden}$$

$$\text{Max Range} = \text{skor maksimal} \times \text{jumlah seluruh responden}$$

$$\text{Median} = (\text{Max Range} + \text{Min Range}) : 2$$

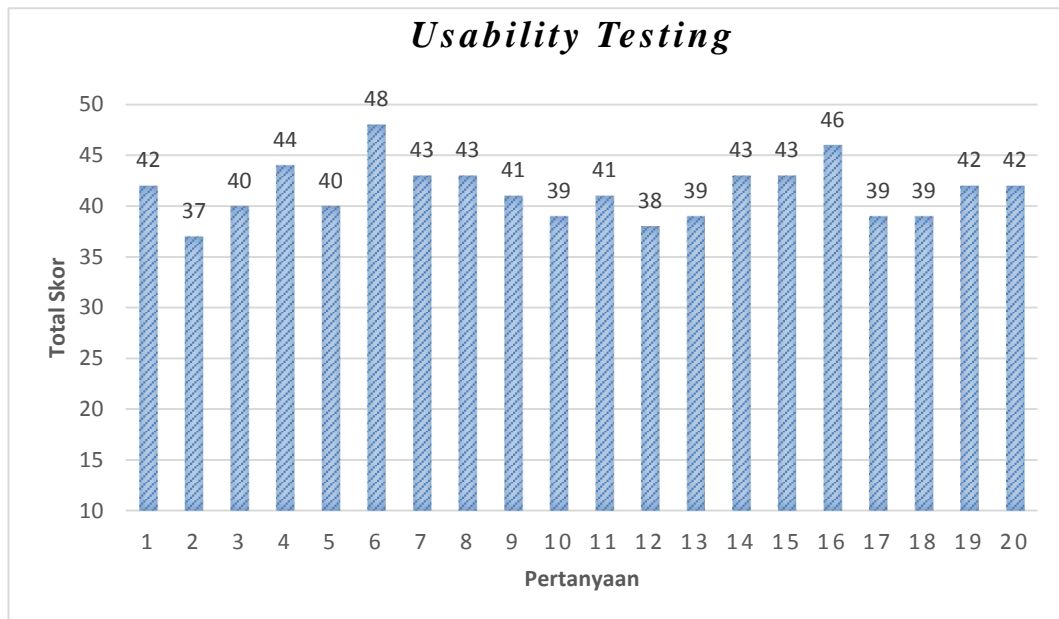
Berikut adalah hasil analisis pengujian *usability*:

Tabel 4.22 Hasil Pengujian *Usability*

Pertanyaan	Poin Skala Penilaian					Total Skor	Skala			Ket.
	1	2	3	4	5		Min	Med	Max	
1	0	0	2	4	4	42	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
2	0	1	3	4	2	37	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
3	0	0	2	6	2	40	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
4	0	0	2	2	6	44	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
5	0	0	2	6	2	40	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median

Pertanyaan	Poin Skala Penilaian					Total Skor	Skala			Ket.
	1	2	3	4	5		Min	Med	Max	
6	0	0	1	4	5	48	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
7	0	0	1	5	4	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
8	0	0	2	3	5	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
9	0	0	2	5	3	41	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
10	0	0	3	5	2	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
11	0	0	3	3	4	41	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
12	0	1	2	5	2	38	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
13	0	0	3	5	2	39	10	30	50	Semua <i>user</i> memiliki kecenderungan

Pertanyaan	Poin Skala Penilaian					Total Skor	Skala			Ket.
	1	2	3	4	5		Min	Med	Max	
										<i>positive</i> , karena nilai skor $\geq$ Median
14	0	0	0	7	3	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
15	0	0	2	3	5	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
16	0	0	0	4	6	46	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
17	0	0	3	5	2	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
18	0	0	1	9	0	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
19	0	0	1	6	3	42	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
20	0	1	1	1	7	44	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median

Gambar 4.33 Grafik *Usability Testing*

Tabel di atas menunjukkan hasil perhitungan skala Likert di setiap pertanyaan yang menggambarkan interpretasi prespektif *usability* yang dirasa oleh *user*. Hasil menyatakan bahwa semua *user* memberikan skor atau nilai diantara median dan maksimal yang menyatakan bahwa respon penilaian *usability* memiliki kecenderungan positif. Grafik pada gambar juga menyatakan bahwa hasil respon pada tiap pertanyaan berada dalam zona kecenderungan positif. Maka dapat disimpulkan bahwa aplikasi telah memenuhi syarat *usability*.

### 3. *Interface Testing*

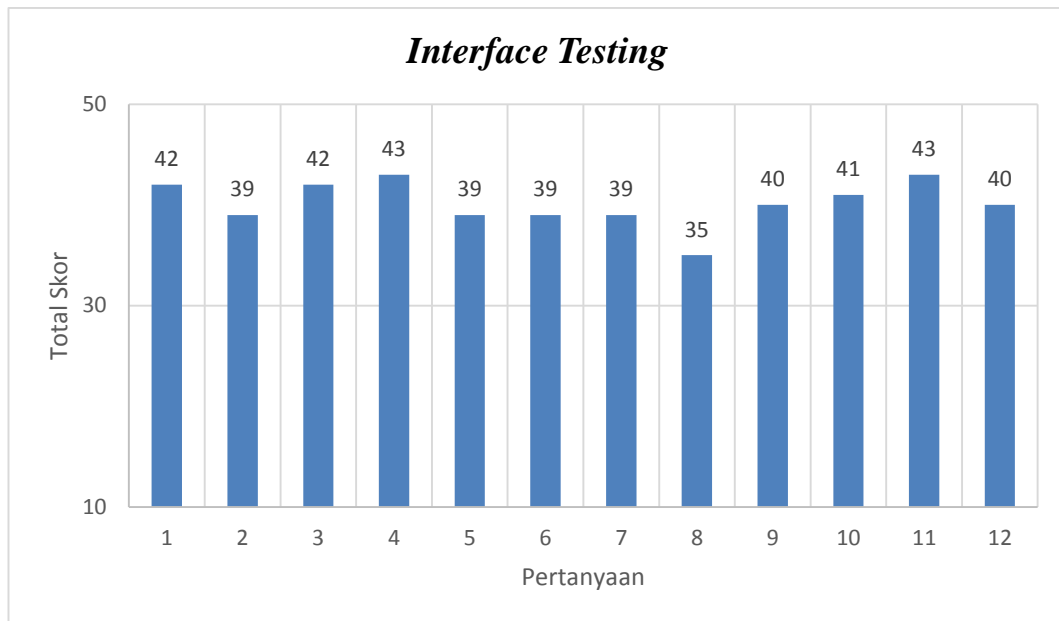
Pengujian *interface* merupakan analisis perilaku *user* dan mengarah pada temuan bagaimana pendapat *user* terhadap tampilan yang disajikan aplikasi yang secara langsung berinteraksi dengan *user*. Penyusunan soal dalam pengujian *interface* dibuat berdasarkan empat kualitas komponen aspek *usability*. Terdapat 12 buah pertanyaan untuk merepresentasikan komponen *interface* menggunakan grafik diagram yang mengidentifikasi prespektif *user*. Berikut adalah tabel yang merepresentasikan hasil pengujian *interface*:

Tabel 4.23 Hasil Pengujian *Interface*

Pertanyaan	Poin Skala Penilaian					Total Skor	Skala			Ket.
	1	2	3	4	5		Min	Med	Max	
1	0	0	3	2	5	42	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
2	0	1	2	4	3	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
3	0	0	1	6	3	42	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
4	0	0	0	7	3	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
5	0	0	3	5	2	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
6	0	0	3	5	2	39	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
7	0	1	2	4	3	39	10	30	50	Semua <i>user</i> memiliki



Pertanyaan	Poin Skala Penilaian					Total Skor	Skala			Ket.
	1	2	3	4	5		Min	Med	Max	
										kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
8	0	0	5	5	0	35	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
9	0	1	2	3	4	40	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
10	0	0	3	3	4	41	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
11	0	0	0	7	3	43	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median
12	0	0	3	4	3	40	10	30	50	Semua <i>user</i> memiliki kecenderungan <i>positive</i> , karena nilai skor $\geq$ Median

Gambar 4.34 Grafik *Interface Testing*

Tabel diatas menunjukkan hasil perhitungan skala Likert di setiap pertanyaan yang menggambarkan interpretasi prespektif *interface* yang dirasa *user*. Hasil menyatakan bahwa semua *user* memberikan skor atau nilai diantara median dan maksimal yang menyatakan bahwa respon penilaian *interface* memiliki kecenderungan positif. Grafik pada gambar juga menyatakan bahwa hasil respon pada tiap pertanyaan berada dalam zona kecenderungan positif. Maka dapat disimpulkan bahwa aplikasi telah memenuhi syarat *interface*.

#### 4.4.2 Pengujian Algoritma<sup>[C1]</sup>

##### 1. Whitebox Testing

Tabel 4.24 Hasil Whitebox Testing

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
1	<i>bar_chart</i>	<i>Product :</i> [1]Keju; Craft;Rak A.1;10;Batang;2016-07-31 [2]Susu;FrisianFlag;Rak A.1;30;Kaleng;2016-08-05 [3]Minyak;Sania;Rak A.1;50;Liter;2016-12-12 [4]San Miguel Pin 640 ml;San Miguel;RakA.1;100;Ctn;2017-02-19	Array ( [ <i>racks</i> ] => Array ( [0] => stdClass Object ( [ <i>rack_id</i> ] => 1 [ <i>rack_name</i> ] => Rak A.1 [ <i>rack_available</i> ] => 1 [ <i>products</i> ] => Array ( [0] => stdClass Object ( [ <i>product_id</i> ] => 54 [ <i>product_code</i> ] => J023 [ <i>product_name</i> ] => Keju [ <i>product_brand</i> ] => Craft [ <i>product_unit</i> ] => Batang [ <i>product_pricebuy</i> ] => 30000 [ <i>product_pricesale</i> ] => 32000 [ <i>product_inventory</i> ] => 10 [ <i>product_image</i> ] => unknown.gif [ <i>product_available</i> ] => 1 [ <i>product_expired</i> ] => 2016-07-31 [ <i>rack_id</i> ] => 1 ) [1] => stdClass Object ( [ <i>product_id</i> ] => 52 [ <i>product_code</i> ] => B001 [ <i>product_name</i> ] => Susu	Array ( [ <i>racks</i> ] => Array ( [0] => stdClass Object ( [ <i>rack_id</i> ] => 1 [ <i>rack_name</i> ] => Rak A.1 [ <i>rack_available</i> ] => 1 [ <i>products</i> ] => Array ( [0] => stdClass Object ( [ <i>product_id</i> ] => 54 [ <i>product_code</i> ] => J023 [ <i>product_name</i> ] => Keju [ <i>product_brand</i> ] => Craft [ <i>product_unit</i> ] => Batang [ <i>product_pricebuy</i> ] => 30000 [ <i>product_pricesale</i> ] => 32000 [ <i>product_inventory</i> ] => 10 [ <i>product_image</i> ] => unknown.gif [ <i>product_available</i> ] => 1 [ <i>product_expired</i> ] => 2016-07-31 [ <i>rack_id</i> ] => 1 ) [1] =>	True

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
			[product_brand] => Frisian Flag [product_unit] => kaleng [product_pricebuy] => 3000 [product_pricesale] => 5000 [product_inventory] => 30 [product_image] => unknown.gif [product_available] => 1 [product_expired] => 2016-08-05 [rack_id] => 1 ) [2] => stdClass Object ( [product_id] => 53 [product_code] => C301 [product_name] => Minyak [product_brand] => Sania [product_unit] => Liter [product_pricebuy] => 20000 [product_pricesale] => 22000 [product_inventory] => 50 [product_image] => unknown.gif [product_available] => 1 [product_expired] => 2016-12-12 [rack_id] => 1 ) [3] => stdClass Object ( [product_id] => 50 [product_code] => SMP1D [product_name] => San Miguel Pin 640 ml [product_brand] => San Miguel [product_unit] => Ctn [product_pricebuy] => 300000 [product_pricesale] => 325000 [product_inventory] => 100 [product_image] => logos_production-000002088-	stdClass Object ( [product_id] => 52 [product_code] => B001 [product_name] => Susu [product_brand] => Frisian Flag [product_unit] => kaleng [product_pricebuy] => 3000 [product_pricesale] => 5000 [product_inventory] => 30 [product_image] => unknown.gif [product_available] => 1 [product_expired] => 2016- 08-05 [rack_id] => 1 ) [2] => stdClass Object ( [product_id] => 53 [product_code] => C301 [product_name] => Minyak [product_brand] => Sania [product_unit] => Liter [product_pricebuy] => 20000 [product_pricesale] => 22000 [product_inventory] => 50 [product_image] => unknown.gif [product_available] => 1 [product_expired] => 2016- 12-12 [rack_id] => 1 ) [3] => stdClass Object ( [product_id] => 50	

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
			1426838915.jpg [product_available] => 0 [product_expired] => 2017-02-19 [rack_id] => 1 ) )	[product_code] => SMP1D [product_name] => San Miguel Pin 640 ml [product_brand] => San Miguel [product_unit] => Ctn [product_pricebuy] => 300000 [product_pricesale] => 325000 [product_inventory] => 100 [product_image] => logos_production-000002088-1426838915.jpg [product_available] => 0 [product_expired] => 2017-02-19 [rack_id] => 1 ) )	

## 2. Blackbox Testing

Tabel 4.25 Hasil *Blackbox Testing*

<i>Test case</i>	<i>Test case Condition</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
1	Produk dimasukkan secara acak dan terdiri dari beberapa urutan tanggal kedaluarsa. Tanggal 16 September 2016, 19 Februari 2017, 12 Desember 2016, dan 1 Januari 2017	Visualisai produk tersedia sesuai urutan yaitu 16 September 2016, 12 Desember 2016, 1 Januari 2016, dan 19 Februari 2017. Tampilan muncul dengan <i>layout vertical</i> , atas ke bawah.	Visualisai produk tersedia sesuai urutan yaitu 16 September 2016, 12 Desember 2016, 1 Januari 2016, dan 19 Februari 2017. Tampilan muncul dengan <i>layout vertical</i> , atas ke bawah.	<i>True</i>
2	Produk dimasukkan secara acak dan terdiri dari beberapa urutan tanggal kedaluarsa. Tanggal 16 September 2016, 19 Februari 2017, 12 Desember 2016, 1 Januari 2017, dan 1 Juli 2016 (pengujian dilakukan pada tanggal 30 Juli 2016)	Visualisai produk tersedia sesuai urutan yaitu 16 September 2016, 12 Desember 2016, 1 Januari 2016, dan 19 Februari 2017. Tampilan muncul dengan <i>layout vertical</i> , atas ke bawah.	Visualisai produk tersedia sesuai urutan yaitu 16 September 2016, 12 Desember 2016, 1 Januari 2016, dan 19 Februari 2017. Tampilan muncul dengan <i>layout vertical</i> , atas ke bawah.	<i>True</i>

Tabel di atas menyatakan hasil ‘true’ untuk semua *test case* karena mengeluarkan hasil ‘*Actual Output*’ sesuai dengan hasil ‘*Expected Output*’. Hasil dari pengujian algoritma di atas dapat dilihat pada Gambar 4.35 dan Gambar 4.36 di bawah ini:

#### Aplikasi Penjualan dan Inventory Minimarket

Stok Produk > Stok Barang

Daftar Informasi Stok

10 records per page Search:

No	Rak	Kode	Produk	Stok	Satuan	Kadaluarsa	Pengaturan
1	Rak A	9	Keju	10	Box	2016-09-16	<a href="#">Edit</a> <a href="#">Delete</a>
2	Rak A	7	Susu	35	Box	2016-12-12	<a href="#">Edit</a> <a href="#">Delete</a>
3	Rak A	8	Minyak	40	Botol	2017-01-01	<a href="#">Edit</a> <a href="#">Delete</a>
4	Rak A	6	San Miguel Pin 640 MI	30	Botol	2017-02-19	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Gambar 4.35 Hasil *Test Case* 1 (A)

Gambar di atas merupakan hasil *list* data produk yang dimasukkan berdasarkan *test case* pertama yang ditampilkan dalam bentuk tabel.

#### Aplikasi Penjualan dan Inventory Minimarket

Laporan > Visualisasi

Rak EFG Rak BCD Rak A Visualisasi Stok Dalam Rak

Keju 10 2016-09-16	35
Susu Dancow 2016-12-12	40
Minyak Bimoli 2017-01-01	30
San Miguel Pin 640 MI San Miguel 2017-02-19	

Gambar 4.36 Hasil *Test Case* 1 (B)

Gambar diatas merupakan hasil visualisasi susunan produk yang telah diurutkan berdasarkan tanggal kedaluarsa . Terlihat bahwa susunan produk paling atas adalah yang memiliki tanggal kedaluarsa paling cepat, dengan tujuan agar menyarankan produk paling atas untuk dijual terlebih dahulu.

## Aplikasi Penjualan dan Inventory Minimarket

Stok Produk > Stok Barang

Daftar Informasi Stok

10 records per page Search:

No	Rak	Kode	Produk	Stok	Satuan	Kadaluarsa	Pengaturan
1	Rak A	9	Keju	10	Box	2016-09-16	<a href="#">Edit</a> <a href="#">Delete</a>
2	Rak A	7	Susu	35	Box	2016-12-12	<a href="#">Edit</a> <a href="#">Delete</a>
3	Rak A	8	Minyak	40	Botol	2017-01-01	<a href="#">Edit</a> <a href="#">Delete</a>
4	Rak A	6	San Miguel Pin 640 MI	30	Botol	2017-02-19	<a href="#">Edit</a> <a href="#">Delete</a>
5	Rak A	10	Kacang	90	Bungkus	2016-07-01	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 4.37 Hasil Test Case 2 (A)

Gambar di atas merupakan hasil list data produk yang diinput berdasarkan test case pertama yang ditampilkan dalam bentuk tabel. Yaitu menampilkan produk sesuai data yang diinput oleh *user*. Dengan memasukkan data yang memiliki tanggal kedaluarsa yang telah lewat.

## Aplikasi Penjualan dan Inventory Minimarket

Laporan > Visualisasi

Rak EFG Rak BCD Rak A Visualisasi Stok Dalam Rak

Keju Kraft 2016-09-16	10
Susu Danisco 2016-12-12	35
Minyak Bimoli 2017-01-01	40
San Miguel Pin 640 MI San Miguel 2017-02-19	30

Gambar 4.38 Hasil Test Case 2 (B)

Gambar diatas merupakan hasil visualisasi susunan produk yang telah diurutkan berdasarkan tanggal kedaluarsa. Terlihat bahwa susunan berdasarkan tanggal kedaluarsa. dan data produk yang telah kedaluarsa tidak masuk ke dalam visualisasi rak.

Berdasarkan hasil pengujian yang telah dilakukan maka dapat disimpulkan bahwa algoritma SRTF dapat diimplementasikan dalam aplikasi dan berjalan baik untuk



memberikan visualisasi produk dengan benar sesuai dengan tujuan dari penelitian ini.

#### **4.5 *Results and Reports***

Tahap akhir setelah analisis dari hasil pengujian sistem dan algoritma adalah hasil dan laporan. Hasil dan laporan ini berupa aplikasi penjualan dan *inventory* yang telah lolos semua syarat pengujian dan telah siap untuk digunakan. Laporan dari penelitian ini dibuat berupa dokumentasi yang disusun dalam sebuah laporan penulisan tugas akhir. Penulisan laporan dibuat berdasarkan pedoman penulisan dan penyusunan laporan tugas akhir di Universitas Bakrie.

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Berdasarkan tujuan dari penelitian ini maka dapat disimpulkan hasil dari penelitian sebagai berikut:

1. Perancangan dan pembangunan aplikasi penjualan dan *inventory* menggunakan metode *waterfall*, pada tahap perancangan pertama kali dilakukan survei dan observasi dengan menggunakan beberapa sumber sesuai dengan permasalahan yang diajukan dan ditinjau dari tingkat eksplanasinya kemudian dilakukan wawancara secara mendalam terhadap *user* potential untuk mendapatkan data analisa kebutuhan sistem secara fungsional. Dari data analisa tersebut menghasilkan sebuah rancangan yang direpresentasikan dalam bentuk *UML Diagram* yang terdiri dari *use case* diagram, *class diagram*, *database* model, dan *pseudocode* aplikasi. Hasil perancangan digunakan sebagai dasar pembuatan aplikasi yang diterjemahkan ke dalam bahasa pemrograman. Penulis menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*), *PhpMyAdmin* untuk implementasi *database* dan XAMPP™ versi 3.2.1 sebagai *web server*, *database server* dan aplikasi *server*. Setelah melalui proses *coding*, program kemudian diuji secara keseluruhan. Semua fungsi–fungsi *software* diujicobakan, agar *software* bebas dari *error*, dan hasilnya benar-benar sesuai dengan kebutuhan yang ada. Untuk mengetahui apakah aplikasi berjalan sesuai dengan rancangan dilakukan dua pengujian, yaitu pengujian *white box testing* yang dilakukan oleh penulis sendiri dan pengujian *black box* yang dilakukan kepada *user* potential sebanyak 10 orang. Dari hasil pengujian menunjukkan bahwa seluruh *codes* dan kebutuhan fungsional aplikasi berjalan dengan benar.

2. Implementasi *algortima shortest remaining time first* pada laporan visualisai barang membantu menghasilkan informasi akurat tentang lokasi, posisi tempat barang disimpan serta mengatur urutan barang sesuai kadaluarsa yang memudahkan *user* untuk menjual barang dengan kadaluarsa tercepat terlebih dahulu.

## 5.2 Saran

Untuk pengembangan penelitian lebih lanjut, penulis memberikan beberapa saran untuk kedepannya, sebagai berikut:

1. Aplikasi penjualan dan inventory minimarket akan membutuhkan beberapa perubahan untuk menyesuaikan ketika aplikasi digunakan oleh toko/minimarket dengan jumlah barang dan transaksi yang berbeda antara setiap toko/minimarket serta dapat dikembangkan dengan menambah beberapa fitur baru, seperti adanya notifikasi ketika barang sudah kadaluarsa, penjualan barang menggunakan *barcode scan*, dan beberapa fitur yang sesuai dengan kebutuhan pengguna
2. Penelitian ini belum komprehensif, karena hanya melihat kinerja dari segi pendekatan proses, maka untuk kebutuhan penelitian berikutnya bagi yang berminat meneliti kinerja minimarket dapat menggunakan pendekatan output, yaitu mengukur produk barang berdasarkan jumlah barang berdasarkan tingkat jumlah keluar-masuk barang, penjualan dan konsumen yang dilayani.

## DAFTAR PUSTAKA

- Alie, D. F., Widodo, A. P., & Sutanto, T. (2013). Rancang Bangun Aplikasi Informasi Penjadwalan Produksi Sandal Dengan Metode Dynamic Priority Dan Shortest Job First. *Skripsi*, Semarang : STIKOM.
- Boone, H. N. (2012). Analyzing Likert Data. *Journal of Extention*, 50(2).
- Connolly, T., & Begg, C. (2010). *Database System, Fifth Edition*. Boston: Pearson.
- Damayanti, B. Y. (2013). Analisis Dan Perancangan Sistem Informasi Inventory Pada PT. Paloma Agung. *Naskah Publikasi*.
- Davis, G. B. (1999). A Research Perspective for Information Systems and Example of Emerging Area of Research. *Information Systems Frontiers Volume 1, Issue 3*, 195 - 203.
- Dwiprastio, C., Karismariyanti, M., & Sukawati, R. (2013). Aplikasi Penjualan Dan Persediaan Barang Dagang Dengan Metode Perpetual FIFO Berbasis Web (Studi Kasus Pada PD ANUGERAH). *Jurnal*.
- Hartono, J. (2006). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
- Hoffer, J. A., Prescott, M. B., & McFadden, F. R. (2005). *Modern Database Management*. New Jersey: Pearson Education, Inc.
- Jawaldi, H. (2014). Tinjauan Hukum Pengaruh Waralaba Minimarket Terhadap Pedagang Kecil Di Kecamatan Tampan Kota Pekanbaru. *Skripsi*, Pekanbaru: Ilmu Hukum, UIN Sultan Syarif Kasim Riau.
- Julianto, T. (2014). Penjadwalan Produksi Percetakan Dengan Metode Shortest Job First Untuk Optimalisasi Waktu Proses Produksi Studi Kasus CV. Syauqi Press. *Skripsi*, Fakultas Ilmu Komputer, UDINUS.
- Kadir, A. (2003). *Pengenalan Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Kadir, A. (2008). *Dasar Perancangan & Implementasi Database Relational*. Yogyakarta: Penerbit ANDI.
- Kosasi, S. (2014). Perancangan Aplikasi Point of Sale dengan Arsitektur Client/Server Berbasis Linux dan Windows. *Citec Journal*, Vol. 1, No. 2.
- Kurata, D. (2001). *Object-Oriented Programming in Visual Basic .NET*. InStep Technologies, Inc.

- Laudon, K., & Laudon, J. (1998). *Management Information System New Approaches to Organization & Technology*. New Jersey: Prentice Hall, Inc.
- Lemos, O. A., Franchin, I. G., & Masiero, P. C. (2009). Integration testing of Object-Oriented and Aspect-Oriented programs. *Science of Computer Programming*, 74, 861-878.
- Ma'ruf, H. (2005). *Pemasaran Ritel*. Jakarta: PT Gramedia Pustaka Utama.
- McLeod, R. (1998). *Management Information Systems*. United States: Prentice Hall.
- Nugrahanto, Y. (2002). Rancang Bangun Sistem Penjadwalan Produksi Dengan Kombinasi Algoritma Shortest Job First Dan Dynamic Priority Scheduling. *Skripsi*, Semarang : STIKOM.
- Palys, T. (2008). Purposive Sampling. In *The Sage Encyclopedia of* (pp. 697-698). Los Angeles: In L. M. Given.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach Seventh Edition*. New York: McGraw-Hill.
- Ristono, A. (2009). *Manajemen Persediaan Edisi I*. Yogyakarta: Graha Ilmu.
- Saputra, H., Falahah, & Siswanto, B. (2012). Aplikasi Inventori Gudang Berbasis WEB Pada Bengkel Maman Jaya Bandar Lampung. *Jurnal*.
- Setiawan, E., & Suryana, E. (2006). Aplikasi Terintegrasi Toko Swalayan. *Jurnal Sistem Informasi UKM, Vol 1, No1*, hal 17-26.
- Silberschatz, A., Gagne, G., & Galvin, P. B. (2002). CPU Scheduling. In *Operating System Concepts, Ninth Edition* (p. Chapter 6).
- Simamora, B. (2005). *Analisis Multivariat Pemasaran*. Jakarta: Gramedia Pustaka Utama.
- Smith, B. (2015). *Object-Oriented Programming*. United States: Apress.
- Sommerville, I. (2011). *Software Engineering Ninth Edition*. Pearson.
- Sonhaji, D. (2012, Oktober 18). *Pengantar Teknologi Informasi*. Retrieved Mei 24, 2015, from <http://ptisonhaji.blogspot.com>:  
<http://ptisonhaji.blogspot.com/2012/10/bab-1-dasar-teknologi-informasi.html>
- Sutabri, T. (2004). *Analisa Sistem Informasi*. Yogyakarta: Andi Offset.

- Vijay. (2015, June 26). *Web Testing: Complete guide on testing web applications*. Retrieved July 31, 2015, from <http://www.softwaretestinghelp.com/web-application-testing/>
- Vrat, P. (2014). *Materials Management An Integrated Systems Approach*. India: Springer India.
- Wilkison, J. (1992). *Accounting and Information*. John Wiley & Sons, Inc.
- Williams, L. (2006). Testing Overview and Black-Box Testing Techniques. *williams2006testing*. Retrieved December 2, 2015, from [agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf](http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf)
- Yuhendra, D. M., & Poerwanta, R. (2013). Perancangan Sistem Inventory Spare Parts Mobil Pada CV. Auto Parts Toyota Berbasis Aplikasi Java. *Jurnal TEKNOIF Vol.1 No.2*.
- Zebua, J., & Mustikasari, M. (2012). Aplikasi Pencarian Buku Perpustakaan Dengan Menggunakan RDF dan SPARQL. *UG Jurnal Vol. 6 No. 04*.

## Lampiran 1 : Software Requirements Specification

IEEE 830

### **Aplikasi Penjualan dan Inventory**

### **Software Requirements Specification**

Versi 1.0

21 Agustus 2015

Chandra Setiawan Gimon

Software Engineer

Dipersiapkan untuk:

Kelengkapan Tugas Akhir Informatika Universitas Bakrie

Dosen pembimbing: Yusuf Lestanto

## 1. Pendahuluan

Penulisan dokumen SRS ini akan menggambarkan penjelasan seluruh kebutuhan pengembangan aplikasi penjualan dan *inventory* sesuai dengan spesifikasi kebutuhan perangkat lunak. Dokumen ini dibuat berdasarkan standar sesuai dengan SRS IEEE – 830

### 1.1 Tujuan

Tujuan spesifikasi ini adalah menjelaskan secara menyeluruh tentang aplikasi penjualan dan *inventory* dengan penerapan algoritma SRTF. Dokumen ini menjelaskan tujuan dan fitur dari aplikasi, *interface*, serta apa saja yang dapat dilakukan dalam aplikasi. Dokumen ini diperuntukan untuk desainer dan pengembang sebagai pedoman penerapan sistem di lapangan.

### 1.2 Ruang Lingkup

Ruang lingkup aplikasi penjualan dan inventory ini adalah menampung semua data dan informasi tentang barang-barang tersebut. Data dan informasi ini nantinya akan terakumulasi dan tersimpan (diarsipkan) secara terpusat pada suatu *database* dan saling terintegrasi. Aplikasi ini memberikan informasi barang yang harus didahulukan dalam proses penjualan berdasarkan batas waktu ketahanan atau kadaluarsa produk serta mempermudah pemilik mengetahui stok barang yang dimiliki dan lokasi tempat barang tersebut disimpan.



### 1.3 Daftar Istilah

Istilah	Definisi
<i>Database</i>	Kumpulan informasi yang dimonitor oleh sistem
<i>Software Requirements Specification (SRS)</i>	Dokumen yang menjelaskan seluruh fungsi-fungsi sistem yang dibuat dan batasan-batasannya.
<i>Interface</i>	Tampilan dari aplikasi sebagai penghubung antara pengguna dengan aplikasi yang mudah untuk dipahami
<i>Admin</i>	Orang yang mengelola sistem informasi secara keseluruhan

### 1.4 Referensi

- IEEE. IEEE Std 830-1998 *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society; 1998

### 1.5 Gambaran Umum Dokumen

Penulisan dokumen ini dibagi menjadi beberapa bab sebagai berikut:

**Bab 1** berisi pendahuluan menjelaskan tujuan, ruang lingkup, daftar istilah, referensi, serta deskripsi/gambaran umum dokumen.

**Bab 2** menjelaskan tentang keseluruhan gambaran dari aplikasi penjualan dan *inventory*.

**Bab 3** berisi kebutuhan spesifikasi *requirement* yang ditulis untuk pengembang dan menjelaskan hal teknis rincian fungsi produk.

## 2. Deskripsi Umum

### 2.1 Perspektif Produk

Aplikasi penjualan dan *inventory* ini merupakan aplikasi yang digunakan untuk mempermudah proses bisnis dibidang ritel serta memberikan informasi barang yang harus didahulukan dalam proses penjualan berdasarkan batas waktu ketahanan atau kadaluarsa produk untuk mencegah kerugian karena kerusakan pada barang. Pada aplikasi ini diimplementasikan algoritma SRTF pada sistem inventornya yang bekerja pada saat pihak toko dituntut untuk menjual produk yang mempunyai batas waktu ketahanan lebih pendek daripada produk lainnya.

### 2.2 Fungsi Produk

Aplikasi penjualan dan *inventory* menampilkan laporan daftar barang masuk dan keluar serta laporan stok barang terakhir yang ada di dalam gudang. Aplikasi ini juga dapat menampilkan laporan barang yang akan kadaluarsa dalam periode waktu yang ditentukan. Selain itu aplikasi ini memberikan visualisasi pada proses peletakkan dalam gudang dengan aturan secara berkelompok dengan menempatkan barang dengan frekuensi kadaluarsa tercepat pada posisi depan.

### 2.3 Karakteristik Pengguna

*User* dalam aplikasi ini sebagai *administrator* atau operator yang dapat melihat, memanipulasi serta mengelola data transaksi dengan bertanggung jawab yang adalah sebagai pemilik bisnis.

## 2.4 Batasan – Batasan

Pengembangan aplikasi penjualan dan *inventory* ini memiliki batasan yaitu sebagai berikut:

- Aplikasi bersifat *web base*.
- Seluruh informasi yang ada dalam aplikasi di perbarui oleh *administrator*.
- Aplikasi penjualan dan inventory dibangun dengan menggunakan bahasa pemrograman PHP “Hypertext Preprocessor”.

## 2.5 Asumsi dan Ketergantungan

Asumsi – asumsi pada aplikasi penjualan dan *inventory* adalah:

- Pengguna untuk aplikasi ini minimal memiliki kemampuan dasar dalam mengoperasikan sebuah komputer.
- Tidak ada kerusakan pada *hardware*.

### 3. Spesifikasi Kebutuhan

#### 3.1 Kebutuhan Antarmuka Eksternal

##### 3.1.1 Antarmuka Pengguna

Antarmuka pengguna (*user interface*) menggunakan antarmuka berbasis form dan *desktop* GUI (*Grafic User Interface*). Persyaratan tampilan antarmuka yang diharapkan oleh pengguna adalah tampilan yang *user-friendly* dan mudah dimengerti penggunaannya.

##### 3.1.2 Antarmuka Perangkat Keras

Aplikasi penjualan dan *inventory* ini berbasis *object oriented programming*. Menggunakan antarmuka *desktop* GUI untuk komputer *standalone* (tidak terhubung dengan jaringan).

#### 3.2 Kebutuhan Fungsional

Berikut ini menjelaskan kebutuhan fungsional produk. Kebutuhan fungsional menggambarkan fasilitas yang dibutuhkan serta kegiatan yang terjadi dalam aplikasi.

##### *Login*

Nama <i>Use case</i>	<i>Login</i>
Trigger	Pengguna mengakses aplikasi
Precondition	Aplikasi menampilkan halaman <i>login</i>
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan <i>username</i> dan password</li> <li>2. Sistem melakukan pengecekan</li> </ol>

	3. <i>Login</i> berhasil, sistem menampilkan halaman menu utama
Alternative Path	Tidak ada

### Add Barang

Nama <i>Use case</i>	Add Barang
Trigger	Pengguna memilih menu untuk menambahkan barang
Precondition	Menampilkan menu add produk
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna sudah <i>login</i></li> <li>2. Melakukan pengisian data tentang spesifikasi produk</li> <li>3. Menekan tombol save jika selesai</li> <li>4. Produk berhasil di tambahkan</li> </ol>
Alternative Path	Tidak ada

### Edit Barang

Nama <i>Use case</i>	Edit Barang
Trigger	Pengguna memilih menu untuk mengedit barang
Precondition	Menampilkan menu edit barang
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna sudah <i>login</i></li> <li>2. Melakukan pengisian data yang dicari</li> </ol>

	3. Sistem menampilkan data yang sesuai dengan permintaan 4. Merubah informasi data 5. Produk berhasil diubah
Alternative Path	Tidak ada

### Delete Barang

Nama <i>Use case</i>	Delete Barang
Trigger	Pengguna memilih menu untuk menghapus barang
Precondition	Menampilkan menu delete barang
Basic Path	1. Pengguna sudah <i>login</i> 2. Melakukan pengisian data yang dicari 3. Sistem menampilkan data yang sesuai dengan permintaan 4. Hapus barang 5. Sistem menampilkan pesan pop up “Apakah anda ingin melakukan penghapusan” 6. Menekan yes 7. Barang berhasil di hapus
Alternative Path	Tidak Ada

### Laporan Barang Masuk

Nama <i>Use case</i>	Laporan Barang Masuk
----------------------	----------------------

Trigger	Pengguna ingin melihat transaksi barang masuk
Precondition	Aplikasi menampilkan menu utama
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna sudah <i>login</i></li> <li>2. Pilih menu laporan barang masuk</li> <li>3. Masukkan periode waktu tertentu</li> <li>4. Sistem menampilkan produk sesuai dengan pilihan</li> </ol>
Alternative Path	Pengguna dapat langsung menggunakan tombol laporan barang masuk untuk melihat seluruh barang

### Laporan Barang Keluar

Nama <i>Use case</i>	Laporan Barang Keluar
Trigger	Pengguna ingin melihat transaksi barang keluar
Precondition	Aplikasi menampilkan menu utama
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna sudah <i>login</i></li> <li>2. Pilih menu laporan barang keluar</li> <li>3. Masukkan periode waktu tertentu</li> <li>4. Sistem menampilkan produk sesuai dengan pilihan</li> </ol>

Alternative Path	Pengguna dapat langsung menggunakan tombol laporan barang keluar untuk melihat seluruh barang
------------------	---

### Laporan Stok Barang

Nama <i>Use case</i>	Laporan Stok Barang
Trigger	Pengguna ingin melihat jumlah stok yang tersisa
Precondition	Aplikasi menampilkan menu utama
Basic Path	<ol style="list-style-type: none"> <li>1. Pengguna sudah <i>login</i></li> <li>2. Menekan tombol stok barang</li> <li>3. Sistem menampilkan jumlah stok barang yang tersedia</li> </ol>
Alternative Path	Tidak ada

### 3.3 Kebutuhan Non-fungsional

Terdapat beberapa aspek yang juga dibutuhkan pada aplikasi penjualan dan inventory yaitu :

- Aspek Kemudahan

Menampilkan interface sederhana dan mudah dipahami dengan detail penjelasan untuk informasi tiap fungsi yang jelas

- Aspek Performansi

Agar dapat digunakan dengan baik, perangkat lunak yang dibangun harus didukung komputer dengan spesifikasi minimum processor 1.1 GHz, RAM 256 MB, *Harddisk* 80 Gb, *Keyboard* dan *Mouse*.



Serta perangkat lunak yang di bangun mampu menampilkan data *inventory* dengan perkiraan waktu maksimal 180 detik.

- Aspek Keamanan

Perlu diperhatikan mengenai proses *back up database* secara berkala untuk keamanan data di dalamnya, melalui SQL Server Management Studio atau Transact-SQL

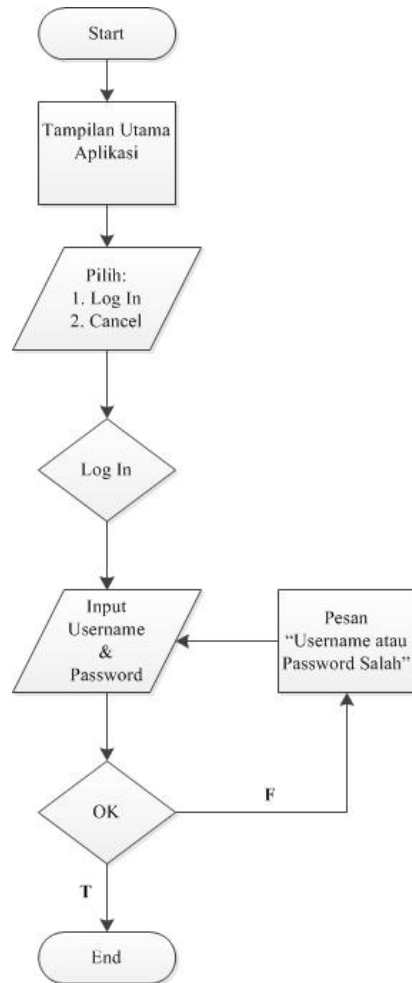
### **3.4 Kebutuhan Lain**

#### **3.4.1 Database**

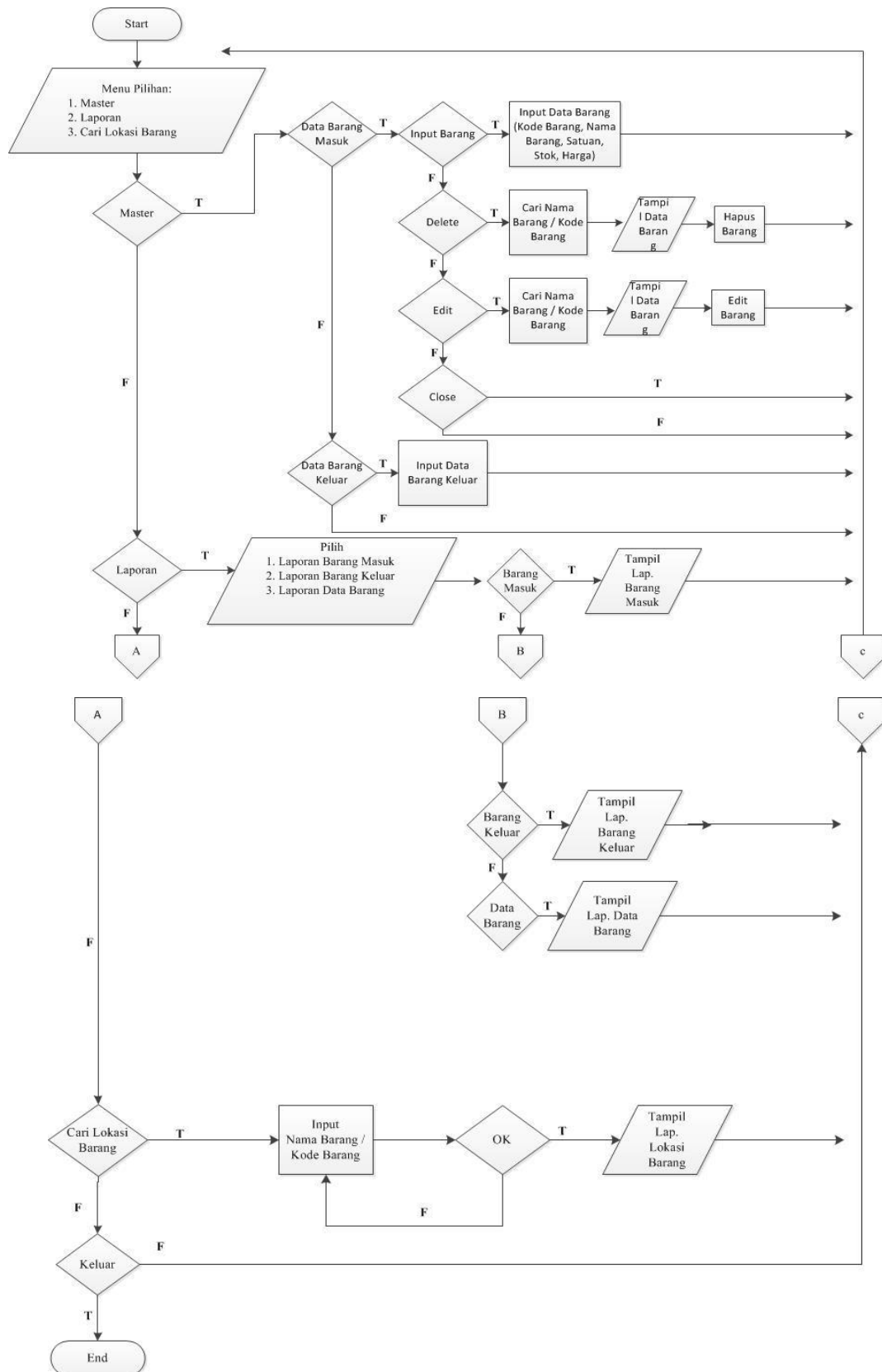
*Database* yang akan digunakan dalam aplikasi ini adalah Microsoft® SQL Server™. SQL Server merupakan Relational *Database* Management System (RDBMS) yang dikembangkan oleh Microsoft®. Fungsi utama SQL Server adalah menampung dan menggunakan data yang terintegrasi dengan aplikasi.

#### **3.4.2 Flowchart Sistem**

- *Flowchart* form login



- *Flowchart menu utama user/admin*



### 3.4.3 Pseudocode Sistem

Berikut adalah gambaran aplikasi penjualan dan *inventory* dalam bentuk *pseudocode*:

```
// User login
declare username, password
input username
input password
function login()
GET username
GET password
    if (username==username &&password==password)
    then login-success
    if (login==success)
    then menuUtama()
    end if
    else login-failed
    end if

// Menu Utama
start
input pilih menu
if (pilih menu == master)then
    function master()
else if (pilih menu == laporan) then
    function laporan()
else if (pilih menu == lokasi) then
    function visualisasi()
end if
end
```

```
// Master Barang
start
function master()
input pilih menu
if (pilih menu == dataBarangMasuk) then
    input pilih submenu
    if (pilih submenu == inputBarang) then
        input kodeBarang
        input namaBarang
        input satuan
        input stok
        input harga
    else if (pilih submenu == delete) then
        get barang
        delete barang
    else if (pilih submenu == edit) then
        get barang
        edit barang
    end if
    save
else if (pilih menu == dataBarangKeluar) then
    input barangKeluar
    save
end if
end

// Laporan
start
function laporan()
input pilih menu
if (pilih menu == laporanBarangMasuk) then
    print laporanBarangMasuk
else if (pilih menu == laporanBarangKeluar) then
```

```
        print laporanBarangKeluar
    else if (pilih menu == dataBarang) then
        print dataBarang
    end if
end

// Visualisasi
start
function visualisasi()
    int n
    n = dataBarang[]
    for (int i = 0; i < n; i++)
        print dataBarang
        print visualisasi
    end for
end
```

## Lampiran 2 : Hasil Survei dan Observasi

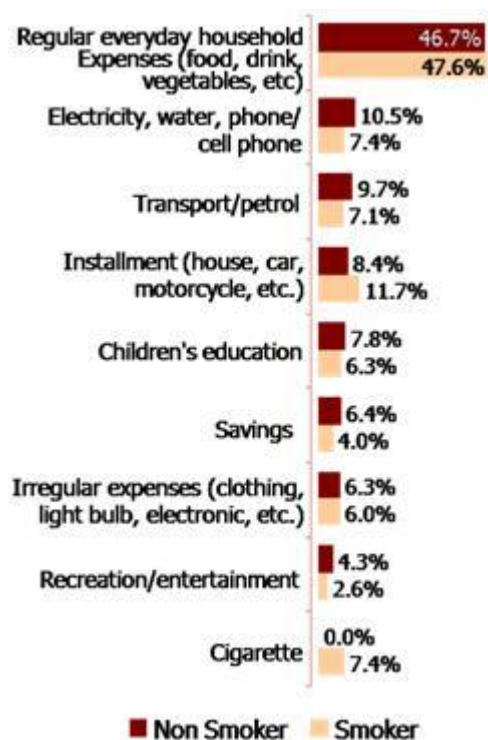
Pada penelitian survei dan observasi, data diambil secara *online* menggunakan beberapa sumber media massa sesuai dengan permasalahan yang di ajukan dan ditinjau dari tingkat ekplanasinya.

1. Kementerian Perdagangan (Kemendag) mengungkapkan, saat ini jumlah pasar modern yang ada diseluruh Indonesia mencapai 23.000 unit. Jumlah tersebut mengalami peningkatan sebesar 14 persen dalam tiga tahun terakhir. Direktur Jenderal Perdagangan Dalam Negeri Kemendag, Srie Agustina mengatakan "Pasar modern ada 23.000 dan dari jumlah itu sebanyak 14.000 lebih di antaranya merupakan kelompok usaha *minimarket*, sedangkan sisanya adalah *supermarket*," saat ini pola beli masyarakat sudah mulai berubah, di mana yang tadinya membeli barang kebutuhan di pasar tradisional sedikit beralih ke pasar modern, baik *supermarket* maupun *minimarket*. Pasar modern saat ini terus tumbuh dan berkembang tidak hanya terpusat pada satu daerah saja. Selain itu, pembangunan pasar modern selalu diatur izin pembangunan agar mampu tersebar ke berbagai daerah di Indonesia dengan pertimbangan jarak dan jumlah pertumbuhan penduduk suatu daerah. Kemendag juga sudah mengatur mengenai produk-produk apa saja yang bisa dan tidak bisa dijual di pasar modern. Hal tersebut dilakukan agar keberadaan pasar rakyat tidak tergerus oleh keberadaan pasar modern.

Sumber:(<http://bisniskeuangan.kompas.com/read/2014/08/08/025100726/Jumlah.Pasar.Modern.di.Indonesia.Capai.23.000.Unit> diakses pada tanggal 26 Juni 2015)

2. Perkembangan industri retail sangat menarik untuk di amati, salah satu yang mengalami perkembangan pesat adalah *modern market*. Berdasarkan data Nielsen, jumlah outlet *modern market* mencapai lebih dari 11,300 di 2009. Bahkan perkembangan ritel di prediksi hingga Rp 2,32 triliun di 2015 dari Rp 1,39 triliun di 2011 menurut Business Monitor International. Ada beberapa hal yang memicu perkembangan retail di Indonesia seperti pendapatan per

kapita, *lifestyle*, daya beli, kemudahan, dan infrastruktur retail yang terus berkembang. Bila kita mencermati demografi Indonesia, jumlah penduduk 237 juta jiwa, dengan total household berkisar 61. Dengan komposisi SES A sekitar 6%, SES B 19%, SES C 30%, dan sisanya SES D dan E. Sehingga dengan jumlah kemampuan *household* lebih dari 50% total *household*, ditambah berdasarkan hasil mitigasi *spending* Frontier Consulting Group seperti dibawah ini:



Grafik 1. Pengeluaran Rumah Tangga Menurut Jenis (Sumber: Frontier Consulting Group)

Bila dilihat dari mitigasi *spending*, sekitar 50% pengeluaran *household* mampu dipenuhi oleh produk-produk yang dijual di *minimarket*, tentu ini menjadi daya tarik bagi para investor. *Modern market* berupa *minimarket* menjamur disekitar pemukiman penduduk. Menurut data yang dilansir Nielsen, pertumbuhan gerai *minimarket* mencapai 42% di 2011. Bahkan di daerah DKI Jakarta, ada lebih dari 2000 gerai *minimarket*.

Sumber: (<http://www.frontier.co.id/kunci-sukses-ritel-membangunmerek.html> diakses pada tanggal 26 Juni 2015)



### **Lampiran 3 : Hasil Wawancara**

## **Lampiran 4 : Dokumen Elisitasi**

## **Lampiran 5 : *Functionality Testing***

## **Lampiran 6 : *Usability Testing***

## **Lampiran 7 : *Interface Testing***