

**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 diseminarkan dan disetujui oleh pembimbing tugas akhir untuk diajukan ke sidang 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: Sawitri Sadanti, Rahma Mualifa, Faiz Faidurrahman, Rahmad Dita, Rien Pratama, Rizky Akbarie, Sarah Putri Mardhatillah, Steffany Uliarta

dan Evi Margaretha. 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

HALAMAN PENGESAHAN	i
ABSTRAK	ii
ABSTRACT	iii
UNGKAPAN TERIMA KASIH	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
DAFTAR SINGKATAN.....	xi
BAB I.....	1
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.....	3
BAB II	5
2.1 Penelitian Terdahulu.....	5
2.2 Sistem, Data dan Informasi Manajemen	9
2.2.1 Pengertian Sistem.....	9
2.2.2 Data	10
2.2.3 Informasi	11
2.3 Konsep Dasar <i>Inventory</i>	12
2.4 Konsep Dasar <i>Minimarket</i>	13
2.5 Algoritma <i>Shortest Job First</i>	15
2.6 Model <i>Waterfall</i>	18
BAB III.....	22
3.1 Alat Penelitian	22
3.2 Metode Pengumpulan Data	23
3.3 Metode Pengembangan Sistem	23
3.3.1 Analisis dan Definisi Kebutuhan	24

3.3.2	Perancangan Sistem	24
3.3.3	Implementasi dan Pengujian Unit	28
3.3.4	Integrasi dan Pengujian Sistem	29
3.3.5	<i>Operation and Maintenance</i>	29
3.4	Jadwal Penelitian	30
BAB IV	31
4.1	Pengumpulan Informasi	31
4.2	Perancangan Sistem.....	32
4.2.1	<i>Use Case Diagram</i>	32
4.2.2	<i>Class Diagram</i>	49
4.2.3	Desain Database	50
4.3	Implementasi	58
4.3.1	Implementasi Algoritma <i>Shortest Remaining Time First</i>	58
4.3.2	Implementasi Antarmuka	59
4.4	Pengujian Algoritma.....	68
4.4.1	Pengujian <i>White Box</i>	69
4.4.2	Pengujian <i>Black Box</i>	69
BAB V	74
5.1	Simpulan.....	74
5.2	Saran	75
DAFTAR PUSTAKA	76

DAFTAR GAMBAR

Gambar 2. 1 <i>Waterfall Phase</i> (Pressman, 2010).....	19
Gambar 2. 2 <i>Waterfall Phase</i> (Sommerville, 2011).....	20
Gambar 3. 1 GUI Halaman <i>Login</i>	25
Gambar 3. 2 GUI Halaman Utama.....	26
Gambar 3. 3 GUI Halaman Stok Barang	26
Gambar 3. 4 GUI Halaman Penjualan.....	27
Gambar 3. 5 GUI Halaman Laporan	27
Gambar 3. 6 GUI Halaman Visualisasi.....	28
Gambar 4. 1 <i>Use Case Diagram</i>	33
Gambar 4. 2 <i>Class Diagram</i>	49
Gambar 4. 3 <i>Conceptual Database Design</i>	51
Gambar 4. 4 <i>Logical Database Design</i>	52
Gambar 4. 5 <i>Physical Database Design</i>	56
Gambar 4. 6 Implementasi Algoritma SRTF.....	59
Gambar 4. 7 Halaman <i>Login</i>	59
Gambar 4. 8 Halaman Awal.....	60
Gambar 4. 9 Halaman Data Utama	60
Gambar 4. 10 Halaman Daftar Pengguna	61
Gambar 4. 11 Halaman Tambah Data Pengguna	61
Gambar 4. 12 Halaman Ubah Data Pengguna	62
Gambar 4. 13 Halaman Daftar Barang / Produk.....	62
Gambar 4. 14 Halaman Tambah Data Barang / Produk.....	63
Gambar 4. 15 Halaman Daftar Rak.....	63
Gambar 4. 16 Halaman Tambah Data Rak Barang.....	64
Gambar 4. 17 Halaman Stok Produk	64
Gambar 4. 18 Halaman Daftar Informasi Stok	65
Gambar 4. 19 Halaman Pembelian Produk.....	65
Gambar 4. 20 Halaman Penjualan.....	66
Gambar 4. 21 Halaman Transaksi Penjualan	66
Gambar 4. 22 Halaman Daftar Penjualan Hari Ini	67

Gambar 4. 23 Halaman Laporan	67
Gambar 4. 24 Halaman Catatan Barang	67
Gambar 4. 25 Halaman Laporan Keuangan	68
Gambar 4. 26 Halaman Visualisasi	68
Gambar 4. 27 Hasil <i>Test Case</i> 1.A	71
Gambar 4. 28 Hasil <i>Test Case</i> 1.B	71
Gambar 4. 29 Hasil <i>Test Case</i> 2.A	72
Gambar 4. 30 Hasil <i>Test Case</i> 2.B	72

DAFTAR TABEL

Tabel 2. 1 Rangkuman Penelitian Terdahulu	6
Tabel 2. 2 Definisi Data	10
Tabel 2. 3 Definisi Informasi	11
Tabel 2. 4 Perbandingan <i>Minimarket</i> dan Toko/Warung Kelontong	14
Tabel 2. 5 Contoh Proses	15
Tabel 2. 6 <i>Job</i> Berdasarkan Urutan	16
Tabel 2. 7 <i>Job</i> Berdasarkan SJF	16
Tabel 2. 8 Perbandingan Algoritma Penjadwalan	17
Tabel 4. 1 <i>Use Case Scenario Login</i>	33
Tabel 4. 2 <i>Use Case Scenario</i> Melihat Informasi Stok Barang	34
Tabel 4. 3 <i>Use Case Scenario</i> Melihat Transaksi Penjualan	35
Tabel 4. 4 <i>Use Case Scenario</i> Melakukan Transaksi Penjualan	36
Tabel 4. 5 <i>Use Case Scenario</i> Melihat Laporan Keluar – Masuk Barang	37
Tabel 4. 6 <i>Use Case Scenario</i> Melihat Laporan Keuangan	38
Tabel 4. 7 <i>Use Case Scenario</i> Melihat Lokasi Barang	38
Tabel 4. 8 <i>Use Case Scenario Add</i> Barang	39
Tabel 4. 9 <i>Use Case Scenario Update</i> Barang	40
Tabel 4. 10 <i>Use Case Scenario Delete</i> Barang	41
Tabel 4. 11 <i>Use Case Scenario Add User</i>	42
Tabel 4. 12 <i>Use Case Scenario Update User</i>	43
Tabel 4. 13 <i>Use Case Scenario Delete User</i>	44
Tabel 4. 14 <i>Use Case Scenario Add</i> Rak	45
Tabel 4. 15 <i>Use Case Scenario Update</i> Rak	46
Tabel 4. 16 <i>Use Case Scenario Delete</i> Rak	47
Tabel 4. 17 <i>Use Case Scenario Logout</i>	48
Tabel 4. 18 <i>Database Entity</i>	50
Tabel 4. 19 Hubungan Antar <i>Entity</i>	51
Tabel 4. 20 Definisi Entitas dan Atribut	53
Tabel 4. 21 Deskripsi Atribut Entitas <i>Database</i>	56
Tabel 4. 22 Hasil Pengujian <i>Black Box</i>	69

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 wawancara yang dilakukan secara mendalam terhadap pemilik *minimarket* 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 visualisasi barang yang harus dijual terlebih dahulu berdasarkan tanggal kadaluarsa pada barang tersebut

dengan merancang bangun aplikasi penjualan dan *inventory* dengan karakteristik kadaluarsa barang sebagai acuan ?

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 barang, untuk mencegah kerugian yang disebabkan kadaluarsa barang atau barang rusak, akibat penumpukan barang dalam gudang.

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 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 <i>Point of Sale</i> dengan Arsitektur <i>Client/Server</i> 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

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
2	Aplikasi Terintegrasi Toko Swalayan	Elisabet Setiawan & Erwin Suryana	2006	Masih bersifat semi integrasi antar proses bisnis	Mengelola administrasi stok, maupun keuangan secara komputerisasi pada toko kecil atau menengah, mengurangi proses kerja secara manual, serta memudahkan dalam melakukan pemeriksaan terhadap stok barang dan transaksi pembelian/penjualan
3	Aplikasi Penjualan Dan Persediaan Barang Dagang Dengan Metode	Chandra Dwiprastio, Magdalena Karismariyanti & Renny Sukawati	2013	Pembuatan jurnal penjualan dan kartu persediaan PD. Anugerah belum menggunakan	Aplikasi ini memberikan kemudahan dalam menghasilkan jurnal penjualan dan kartu

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
	<i>Perpetual</i> FIFO Berbasis Web (Studi Kasus Pada PD Anugerah)			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	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 <i>Inventory</i> Pada <i>Minimarket</i> Dengan Menggunakan <i>Shortest Remaining Time First Algorithm</i>	Penulis	2015	Banyaknya kompetitor membuat terjadinya penumpukan barang dagangan, stok barang yang terus bertambah tiap tahun serta masih menggunakan pencatatan	Aplikasi <i>inventory</i> pada <i>minimarket</i> integrasi antara proses <i>inventory</i> dengan penjualan, informasi barang yang harus didahulukan dalam proses

No	Judul	Pengarang	Tahun	Permasalahan	Hasil
				secara manual dalam proses data transaksi	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 bersama-sama 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 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.

1. (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.
2. Menurut (McLeod, 1998) persediaan adalah sebagai suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha yang normal.
3. 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>	Toko/Warung Kelontong
Suasana Belanja	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 ambikan oleh pemilik
Pelayanan	Pelayanan di <i>minimarket</i> dikemas melalui bisnis waralaba yang profesional	Terkadang menemukan warung kosong dan tidak ditunggu dan harus memanggil sang pemilik
Jenis Barang	Barang kebutuhan sehari-hari lengkap dan dengan stok yang cukup banyak	Belum menjamin semua jenis barang tersedia serta adanya barang tetapi umumnya stok terbatas.
Harga	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

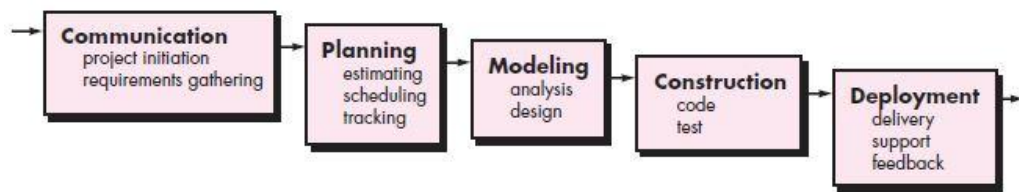
Algoritma	Karakteristik
<i>First In First Out</i>	Algoritma ini merupakan algoritma paling sederhana karena prinsip penjadwalan tidak berprioritas, proses-proses diberi jatah waktu pemroses berdasarkan waktu kedatangan, pada saat proses mendapat jatah waktu pemroses, proses dijalankan sampai selesai
<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. 1 *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) prosedural. Pada tahapan ini menghasilkan dokumen *software requirement*.

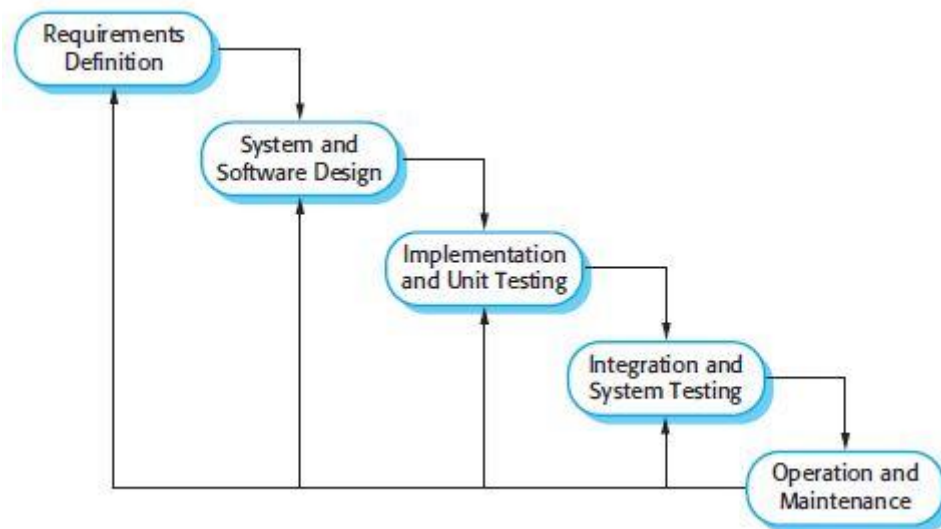
4. *Construction*

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. 2 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

METODOLOGI PENELITIAN

3.1 Alat Penelitian

Dalam melakukan penelitian ini, dibutuhkan alat penelitian yang digunakan dalam pembuatan sistem ini. Alat yang digunakan dalam pembuatan sistem dibagi menjadi dua bagian, yaitu:

1. Perangkat Keras (*Hardware*)

Informasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

- *Device* : Notebook PC Asus N46VJ
- *Operating Systems* : Windows® 7 Ultimate™ 64-bit
- *Processor* : Intel® Core™ i5-3210M CPU @ 2.5 GHz
- *Memory* : 4.00 GB RAM

2. Perangkat Lunak (*Software*)

Informasi perangkat lunak yang digunakan dalam penelitian ini antara lain:

- Microsoft® Office™ Word 2013, digunakan untuk pengolahan kata untuk mencatat informasi dan data-data yang digunakan untuk membuat sistem.
- Google® Chrome™ versi 48.0.2564.109 m, sebagai web browser
- XAMPP™ versi 3.2.1, sebagai sebagai *web server apache*, *database server* dan *application server*
- Notepad++™ versi 6.8.6 sebagai IDE, digunakan untuk penulisan dan penyuntingan bahasa pemrograman. Bahasa pemrograman yang digunakan pada penelitian ini adalah PHP.

2.2 Metode Pengumpulan Data

Proses peneliti dalam pengumpulan data merupakan salah satu komponen penting dalam penelitian ini. Data-data dan informasi digunakan untuk menganalisa permasalahan yang akan dibahas agar sesuai dengan tujuan akhir penelitian ini. Dalam pengumpulan data dilakukan dengan menggunakan dua metode, yaitu penelitian lapangan dan studi pustaka.

1. Penelitian lapangan, yaitu pengumpulan data secara langsung ke lapangan dengan melakukan wawancara dan observasi. Hal ini dilakukan untuk mendapatkan data yang *valid* dan terpercaya, sehingga sesuai dengan kebutuhan yang diperlukan untuk mendukung penelitian ini agar hasil dan kesimpulan penelitian tidak diragukan. Data yang diperoleh secara langsung di lapangan dengan menggunakan metode wawancara dilakukan pada tanggal 21 Juli 2015 dengan pemilik *minimarket* sebagai narasumber. Wawancara dilakukan dengan tujuan untuk mendapatkan informasi mengenai proses penjualan, proses persediaan dan proses pencatatan laporan serta data pendukung lainnya seperti permasalahan dan hambatan yang sering terjadi pada proses bisnis yang tidak sesuai dengan keinginan pemilik bisnis.
2. Studi pustaka, yaitu mengambil data dari literatur, modul, buku panduan, dan juga beberapa kepustakaan yang dapat digunakan untuk mendukung penelitian ini. Metode ini dilakukan sebagai pedoman dan pendukung untuk menyelesaikan penelitian ini.

2.3 Metode Pengembangan Sistem

Dalam sebuah penelitian diperlukan sebuah model perancangan sistem yang digunakan untuk dapat memenuhi kebutuhan pengguna. Prosedur, konsep dan aturan digunakan sebagai pedoman bagaimana dan apa yang harus dikerjakan selama pengembangan sistem ini. Oleh karena itu dalam pengembangan sistem ini menggunakan metode *waterfall* sebagai model acuan dalam pembuatan sistem. Metode *waterfall* dipilih karena pengerjaan sistem ini dilakukan secara berurutan atau secara *linear*, sehingga kualitas dari sistem yang dihasilkan akan baik, ini dikarenakan oleh pelaksanaan yang secara bertahap.

Sesuai dengan tahapan di dalam model *waterfall*, perancangan sistem yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.3.1 Analisis dan Definisi Kebutuhan

Pada tahap ini diawali dengan melakukan pengumpulan kebutuhan untuk perangkat lunak yang akan dibuat yang berupa data *input*, proses yang terjadi dan *output* yang diharapkan. Pengidentifikasian kebutuhan menganalisis aturan bisnis yang berisikan aturan-aturan yang berlaku pada proses bisnis penjualan, proses bisnis persediaan dan proses pencatatan laporan serta data pendukung lainnya untuk menyelesaikan permasalahan dan hambatan yang sering terjadi pada proses bisnis. Analisa kebutuhan sangatlah penting untuk mengetahui kebutuhan-kebutuhan yang nantinya akan digunakan untuk mendukung proses penelitian ini.

Dalam definisi kebutuhan dilakukan pendefinisian dari semua kebutuhan yang dapat diberikan oleh sistem yang akan dibangun. dengan melakukan studi pustaka, wawancara dan observasi. Studi pustaka dilakukan sebagai acuan dalam melakukan penelitian dengan mempelajari penelitian sejenis yang telah dilakukan sebelumnya. Studi pustaka dilakukan untuk mempelajari teori-teori yang mendasari masalah dari jurnal-jurnal, buku dan dokumen yang relevan dengan penelitian ini. Metode wawancara dan observasi dilakukan untuk melihat kebutuhan dari pihak yang diteliti, sehingga kebutuhan dari pengguna dapat terpenuhi dan sesuai dengan keinginan.

Tujuan dari tahapan ini adalah untuk mengumpulkan kebutuhan sistem dengan lebih spesifik. Kebutuhan fungsional dan non-fungsional sistem disertakan pada dokumen SRS di lampiran 1.

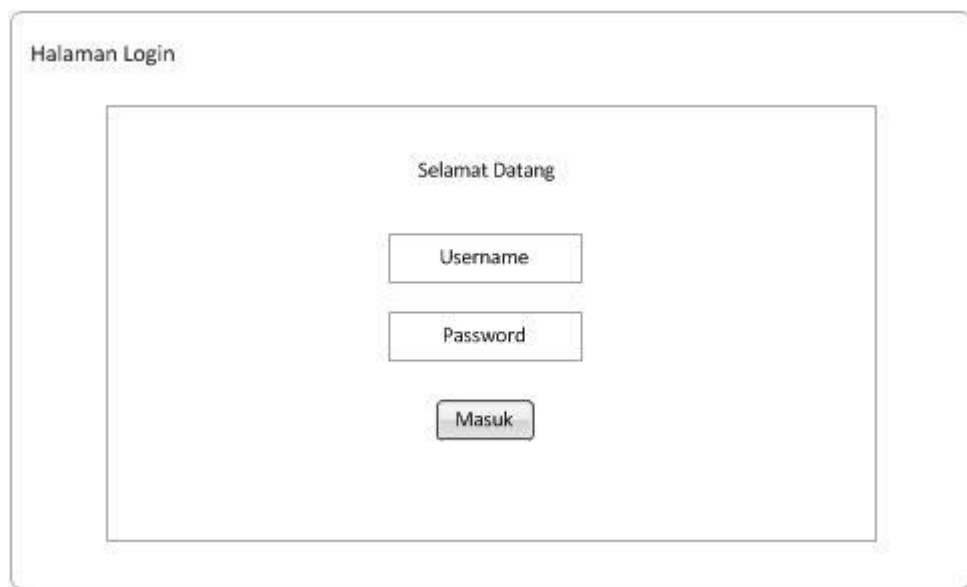
3.3.2 Perancangan Sistem

Pada tahap ini dilakukan perancangan dan penggambaran tentang sistem yang akan dibuat. Hasil yang didapatkan dari analisis dan definisi kebutuhan dijadikan sebagai dasar perancangan sistem. Perancangan sistem ini terdiri dari perancangan *graphic user interface*, perancangan visualisasi, pembuatan *use case*, *class diagram* serta rancangan basis data. Hal ini dilakukan untuk mempermudah

proses pemahaman akan alur kerja dan struktur dari sistem. Perencanaan sistem dilakukan untuk memastikan bahwa rancangan sistem sudah sesuai dengan kebutuhan pengguna. Berikut adalah rancangan *user interface* dan rancangan visualisasi dalam penelitian ini.

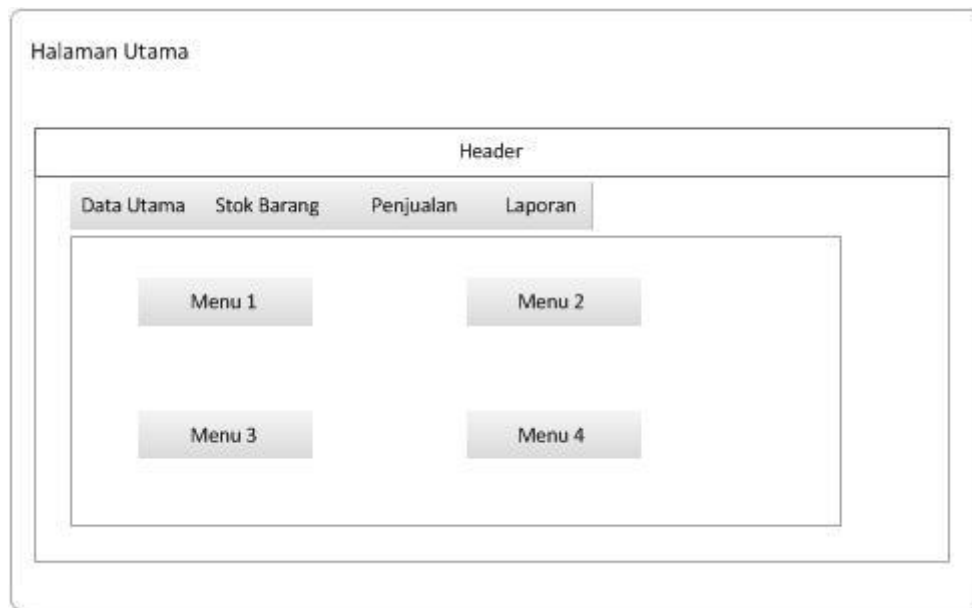
1. Perancangan *user interface*

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



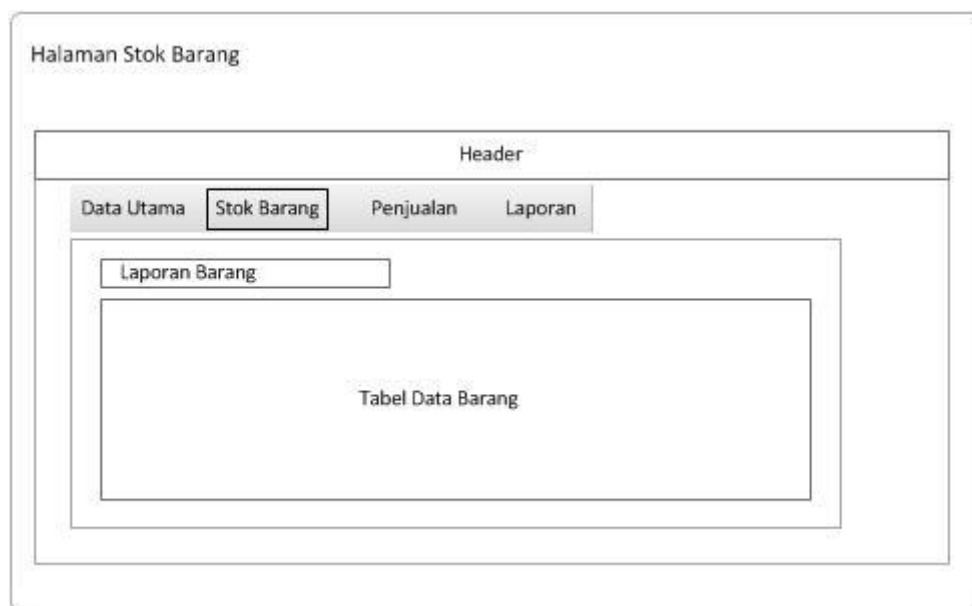
Gambar 3. 1 GUI Halaman *Login*

Gambar 3.1 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 3. 2 GUI Halaman Utama

Pada halaman utama Gambar 3.2 terdapat beberapa menu navigasi yang dapat dipilih yaitu menu data utama, stok barang, penjualan dan laporan.



Gambar 3. 3 GUI Halaman Stok Barang

Tampilan pada Gambar 3.3 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*.

Gambar 3. 4 GUI Halaman Penjualan

Gambar 3.4 menggambarkan halaman menu penjualan pada saat *user* memilih *tab* menu penjualan. Halaman ini berisi formulir untuk melakukan transaksi penjualan untuk diisi oleh *user*.

Gambar 3. 5 GUI Halaman Laporan

Tampilan pada Gambar 3.5 adalah tampilan menu laporan, *user* dapat melihat laporan keluar – masuk barang, laporan keuangan, dan laporan visualisasi.

2. 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 3.6. 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. Berikut adalah rancangan visualisasi barang.

<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 3. 6 GUI Halaman Visualisasi

Untuk sistem pengeluaran barang menggunakan 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.

3.3.3 Implementasi dan Pengujian Unit

Setelah tahap perancangan sistem dilakukan, penulis melakukan *coding* program dari hasil perancangan ke dalam perangkat keras. Penulisan kode menggunakan bahasa pemrograman PHP dan menggunakan MySQL sebagai

penyimpan data. Dalam tahap ini juga memaparkan pengimplementasian algoritma *shortest remaining time first* yang digunakan dalam penelitian ini.

Pada tahap ini dilakukan pengujian dari unit sistem yang telah dibangun. Pengujian dilakukan dengan menguji setiap unit dari sistem. Pengujian dari unit sistem ini digunakan untuk memverifikasi bahwa semua bagian unit dari sistem telah berjalan dengan baik dan telah memenuhi spesifikasinya.

3.3.4 Integrasi dan Pengujian Sistem

Pada tahap ini, setiap unit sistem yang telah diuji diintegrasikan satu sama lain. Setelah diintegrasikan, maka akan dilakukan pengujian pada sistem secara keseluruhan. Pengujian dilakukan untuk menguji apakah sistem sudah sesuai dengan kebutuhan dengan menggunakan dua teknik pengujian yaitu *black-box* dan *white-box*. 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 dikeluarkan 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.

3.3.5 Operation and Maintenance

Tahapan ini tidak dilakukan dikarenakan pada tahap ini memerlukan waktu yang lama karena penelitian ini dilakukan sebagai syarat kelulusan tugas akhir. Oleh karena itu, tahapan ini tidak dilakukan dalam penelitian ini.

2.4 Jadwal Penelitian

Penelitian ini dilakukan pada pertengahan tahun 2015, tepatnya pada bulan Juli. Untuk lebih jelas, rencana kegiatan dituangkan dalam bentuk gambar yang dapat ditemukan pada lampiran 2.

BAB IV

PEMBAHASAN

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

4.1 Pengumpulan Informasi

Tahap ini diawali dengan pengidentifikasian kebutuhan sistem yang akan dikembangkan dalam bentuk *software* berbasis *web base*. Tahapan pertama dilakukan dengan observasi dan 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 produk, laporan keuangan, laporan catatan barang, laporan penjualan dan laporan visualisasi.

3. Kebutuhan Pengguna

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

4. Kebutuhan Sistem

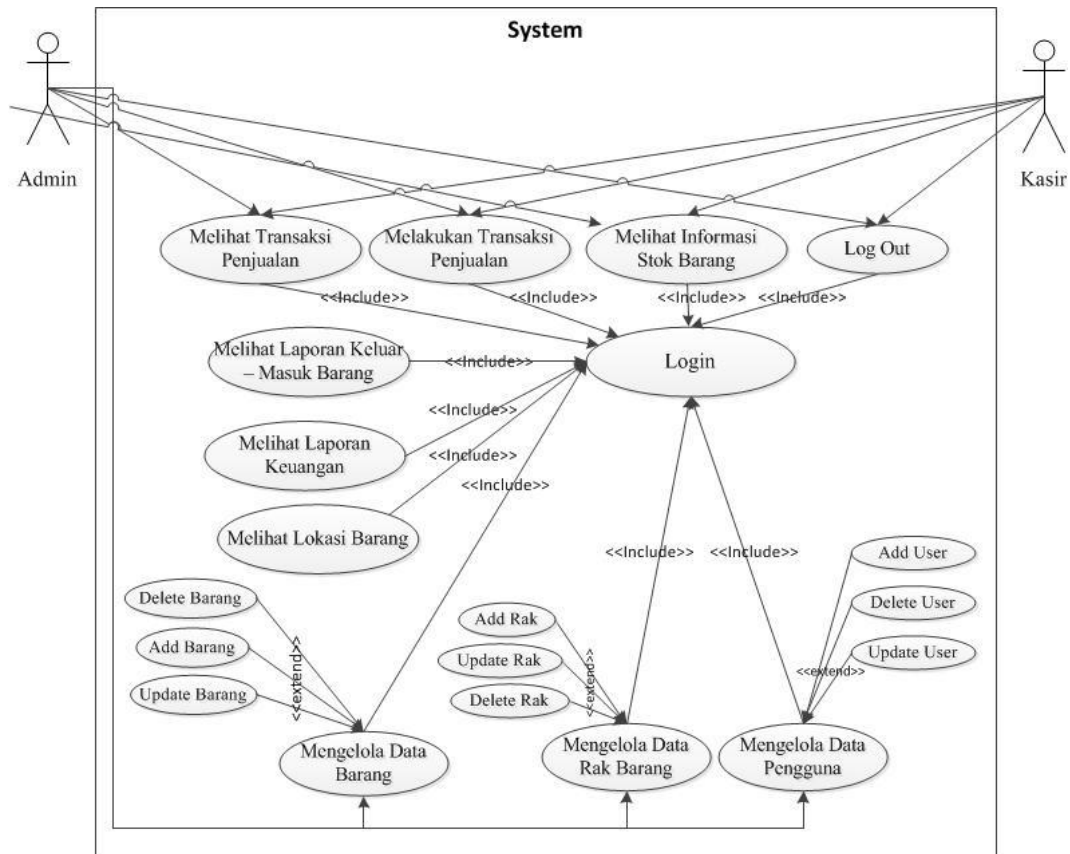
Pada penelitian ini terdapat kebutuhan fungsional yang menunjukkan fasilitas dan aktivitas yang dapat dilakukan oleh sistem. Kebutuhan fungsional sistem dibuat berdasarkan hasil pengumpulan data melalui observasi dan wawancara yang dibuat dalam elisitasi tahap I, II, III yang terlampir pada lampiran 3.

4.2 Perancangan Sistem

Tujuan utama pada tahap ini adalah pemodelan yang sangat dibutuhkan dalam sebuah pengembangan *software*. Hasil dalam tahapan ini dijelaskan dalam model UML seperti *use case diagram*, *class diagram*, dan data model. Pemaparan tiap elemen dapat ditemukan pada subbab ini secara berurutan.

4.2.1 Use Case Diagram

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

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

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

Tabel 4. 2 *Use Case Scenario* Melihat Informasi Stok Barang

Use case Name	Melihat Informasi Stok Barang	
Use case ID	2	
Actor	Admin dan Kasir	
Description	<i>Use case</i> ini menggambarkan kegiatan melihat halaman informasi stok barang pada aplikasi	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	<i>Use case</i> ini dilakukan agar Actor dapat melihat laporan stok barang	
Typical of events	Actor Action	System Response

	1. Pilih menu informasi stok	2. Menampilkan informasi stok barang dalam bentuk tabel
Alternate Course	Actor memilih menu stok barang dan memilih menu informasi stok	
Post-Condition	Aplikasi menampilkan menu utama, Actor dapat menambahkan / mengurangi stok barang, melakukan <i>filtering</i>	

Tabel 4. 3 Use Case Scenario Melihat Transaksi Penjualan

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

Tabel 4. 4 *Use Case Scenario* Melakukan Transaksi Penjualan

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

<i>Alternate Course</i>	<i>Actor</i> memilih menu penjualan kemudian memilih menu penjualan baru, menekan tombol ulangi untuk mengosongkan <i>field</i> pada formulir penjualan
<i>Post-Condition</i>	Aplikasi menampilkan menu utama

Tabel 4. 5 *Use Case Scenario* Melihat Laporan Keluar – Masuk Barang

<i>Use case Name</i>	Melihat Laporan Keluar – Masuk Barang	
<i>Use case ID</i>	5	
<i>Actor</i>	<i>Admin</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan melihat halaman laporan keluar – masuk barang	
<i>Pre-Condition</i>	<i>Actor</i> berada pada halaman utama aplikasi	
<i>Trigger</i>	<i>Use case</i> ini dilakukan agar <i>Actor</i> dapat melihat laporan keluar – masuk barang	
<i>Typical of events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. Pilih menu laporan keluar – masuk barang	2. Menampilkan informasi laporan keluar – masuk barang dalam bentuk tabel
<i>Alternate Course</i>	<i>Actor</i> memilih menu laporan kemudian memilih menu laporan keluar – masuk barang	
<i>Post-Condition</i>	Aplikasi menampilkan menu utama, melakukan <i>filtering</i> , mencetak laporan	

Tabel 4. 6 *Use Case Scenario* Melihat Laporan Keuangan

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

Tabel 4. 7 *Use Case Scenario* Melihat Lokasi Barang

<i>Use case Name</i>	Melihat Lokasi Barang	
<i>Use case ID</i>	7	
<i>Actor</i>	<i>Admin</i>	
<i>Description</i>	<i>Use case ini menggambarkan kegiatan melihat halaman laporan keuangan</i>	
<i>Pre-Condition</i>	<i>Actor berada pada halaman utama aplikasi</i>	

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

Tabel 4. 8 Use Case Scenario Add Barang

Use case Name	Add Barang	
Use case ID	8	
Actor	Admin	
Description	Use case ini menggambarkan kegiatan untuk menambahkan data barang	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	Use case ini dilakukan agar Actor dapat menambah data barang	
Typical of events	Actor Action	System Response
	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
<i>Alternate Course</i>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<i>Post-Condition</i>	Aplikasi menampilkan menu utama	

Tabel 4. 9 *Use Case Scenario Update Barang*

<i>Use case Name</i>	<i>Update</i> Barang	
<i>Use case ID</i>	9	
<i>Actor</i>	<i>Admin</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengubah informasi barang	
<i>Pre-Condition</i>	Actor berada pada halaman utama aplikasi	
<i>Trigger</i>	<i>Use case</i> ini dilakukan agar Actor dapat mengubah informasi barang	
<i>Typical of events</i>	<i>Actor Action</i>	<i>System Response</i>

	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
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
Post-Condition	Aplikasi menampilkan halaman daftar barang / produk	

Tabel 4. 10 Use Case Scenario Delete Barang

Use case Name	Delete Barang	
Use case ID	10	
Actor	Admin	
Description	Use case ini menggambarkan kegiatan untuk menghapus data barang	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	Use case ini dilakukan agar Actor dapat menghapus data barang	
Typical of events	Actor Action	System Response
	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>
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol tidak	
Post-Condition	Aplikasi menampilkan halaman daftar barang / produk	

Tabel 4. 11 Use Case Scenario Add User

Use case Name	<i>Add User</i>	
Use case ID	11	
Actor	<i>Admin</i>	
Description	<i>Use case</i> ini menggambarkan kegiatan untuk menambahkan data pengguna	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	<i>Use case</i> ini dilakukan agar Actor dapat menambah data pengguna	
Typical of events	Actor Action	System Response
	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
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
Post-Condition	Aplikasi menampilkan menu utama	

Tabel 4. 12 *Use Case Scenario Update User*

Use case Name	<i>Update User</i>	
Use case ID	12	
Actor	<i>Admin</i>	
Description	<i>Use case</i> ini menggambarkan kegiatan untuk mengubah informasi pengguna	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	<i>Use case</i> ini dilakukan agar Actor dapat mengubah informasi pengguna	
Typical of events	Actor Action	System Response

	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
<i>Alternate Course</i>	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
<i>Post-Condition</i>	Aplikasi menampilkan halaman daftar pengguna	

Tabel 4. 13 *Use Case Scenario Delete User*

<i>Use case Name</i>	<i>Delete User</i>	
<i>Use case ID</i>	13	
<i>Actor</i>	<i>Admin</i>	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk menghapus data pengguna	
<i>Pre-Condition</i>	Actor berada pada halaman utama aplikasi	
<i>Trigger</i>	<i>Use case</i> ini dilakukan agar Actor dapat menghapus data pengguna	
<i>Typical of events</i>	<i>Actor Action</i>	<i>System Response</i>

	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>
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol tidak	
Post-Condition	Aplikasi menampilkan halaman daftar pengguna	

Tabel 4. 14 Use Case Scenario Add Rak

Use case Name	Add Rak	
Use case ID	14	
Actor	Admin	
Description	Use case ini menggambarkan kegiatan untuk menambahkan data rak barang	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	Use case ini dilakukan agar Actor dapat menambah data rak barang	
Typical of events	Actor Action	System Response
	1. Pilih menu rak	2. Menampilkan informasi daftar rak barang

	3. Menekan tombol tambah	4. Menampilkan formulir pengisian tambah data rak barang
	5. <i>Input</i> nama rak	
	6. Menekan tombol simpan	7. Menyimpan data dalam <i>database</i> dan memberikan notifikasi apabila nama rak sudah tersedia
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
Post-Condition	Aplikasi menampilkan menu utama	

Tabel 4. 15 *Use Case Scenario Update Rak*

Use case Name	Update Rak	
Use case ID	15	
Actor	Admin	
Description	Use case ini menggambarkan kegiatan untuk mengubah informasi data rak barang	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	Use case ini dilakukan agar Actor dapat mengubah informasi data rak barang	
Typical of events	Actor Action	System Response
	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
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol kembali	
Post-Condition	Aplikasi menampilkan halaman daftar rak barang	

Tabel 4. 16 *Use Case Scenario Delete Rak*

Use case Name	<i>Delete Rak</i>	
Use case ID	16	
Actor	<i>Admin</i>	
Description	<i>Use case</i> ini menggambarkan kegiatan untuk menghapus data rak barang	
Pre-Condition	Actor berada pada halaman utama aplikasi	
Trigger	<i>Use case</i> ini dilakukan agar Actor dapat menghapus data rak barang	
Typical of events	Actor Action	System Response
	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>
Alternate Course	Actor dapat membatalkan <i>use case</i> ini dengan menekan tombol tidak	
Post-Condition	Aplikasi menampilkan halaman daftar rak barang	

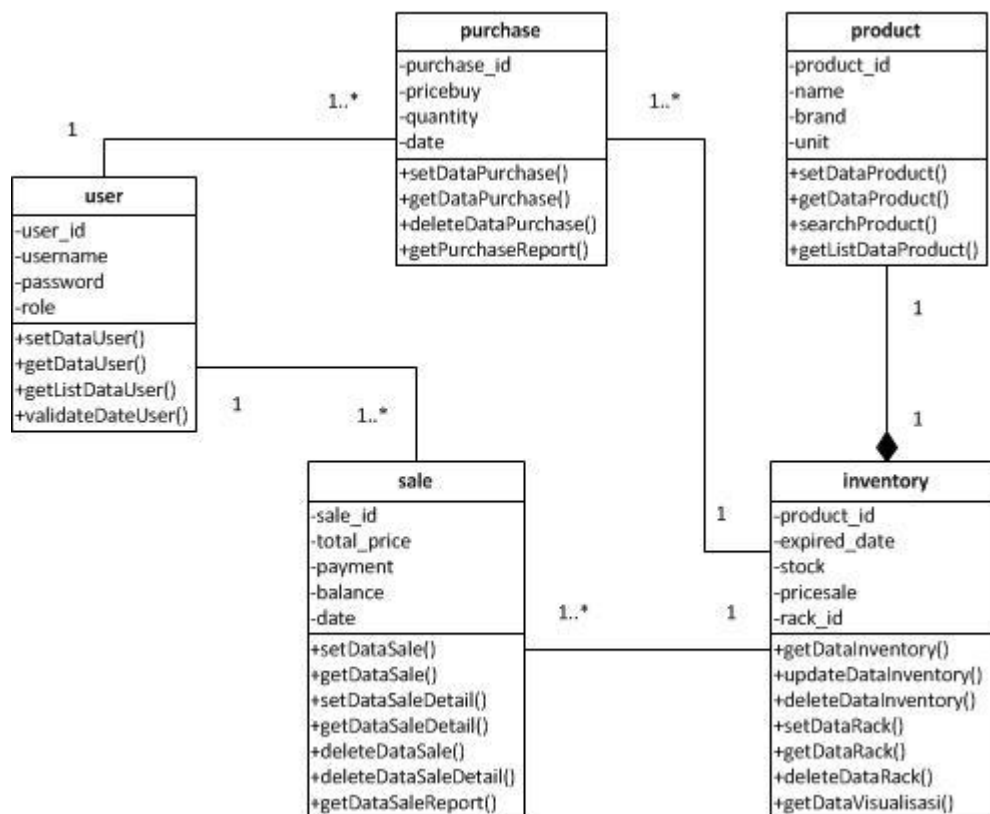
Tabel 4. 17 Use Case Scenario Logout

Use case Name	Logout	
Use case ID	17	
Actor	Admin dan Kasir	
Description	Use case ini menggambarkan kegiatan <i>logout</i> dari dalam aplikasi penjualan dan <i>inventory</i>	
Pre-Condition	Actor menekan tombol pengguna pada halaman utama dan memilih “Keluar”	
Trigger	Use case ini dilakukan agar Actor dapat keluar dari dalam sistem	
Typical of events	Actor Action	System Response
	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. Menampilkan menu <i>login</i>
Alternate Course		
Post-Condition	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. Berikut adalah rancangan *class* diagram dari aplikasi penjualan dan *inventory minimarket*:



Gambar 4. 2 Class Diagram

4.2.3 Desain Database

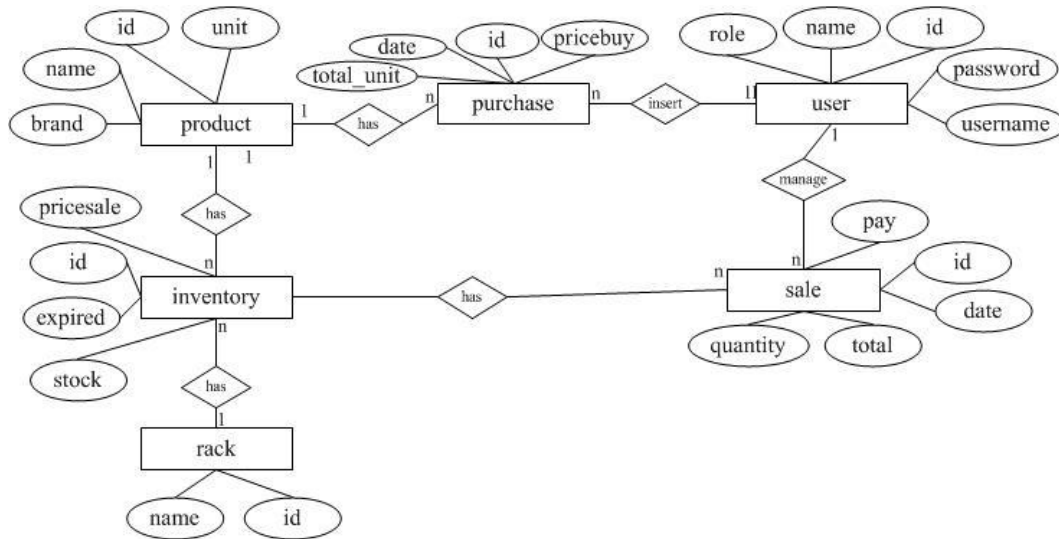
Perancangan *database* adalah untuk menentukan isi dan pengaturan data yang dibutuhkan dalam suatu perancangan sistem. Rancangan *database* dimulai dari *conceptual design*, *logical design*, dan *physical design*.

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.

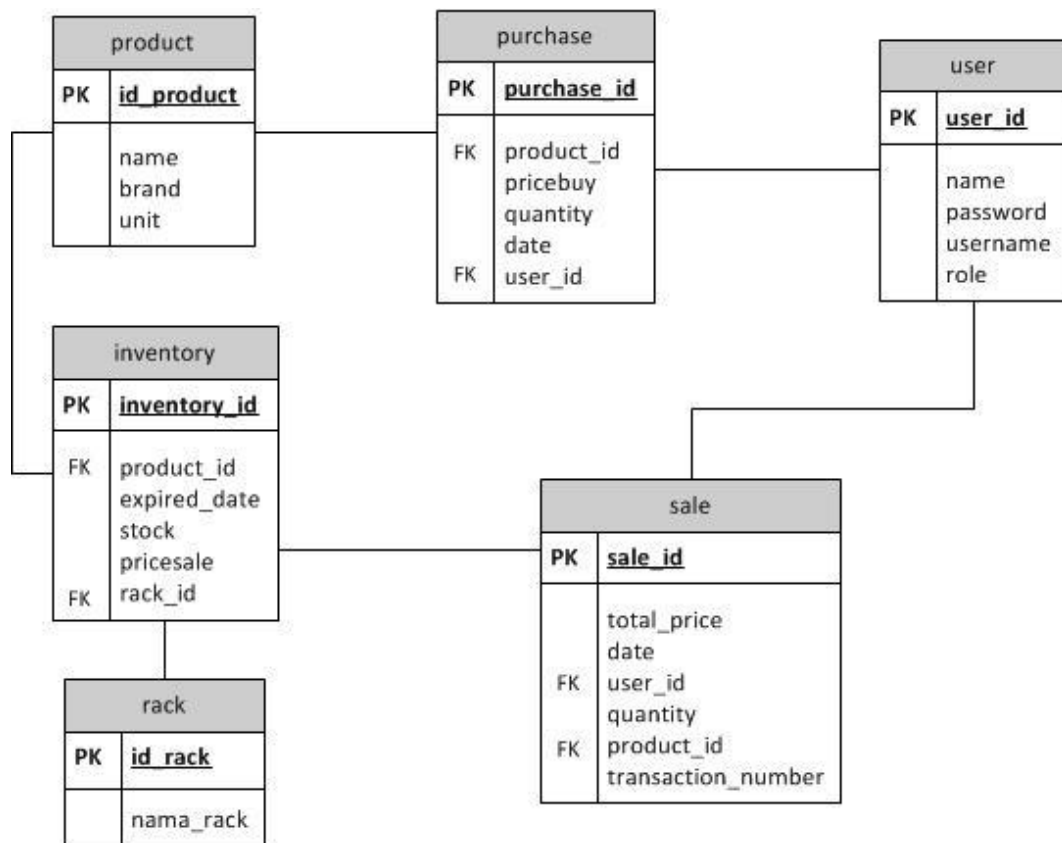
Tabel 4. 19 Hubungan Antar *Entitiy*

No.	Nama <i>Entitiy</i>	Hubungan Antar <i>Entitiy</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	Sale	many-to-one dengan inventory
		many-to-one dengan user

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

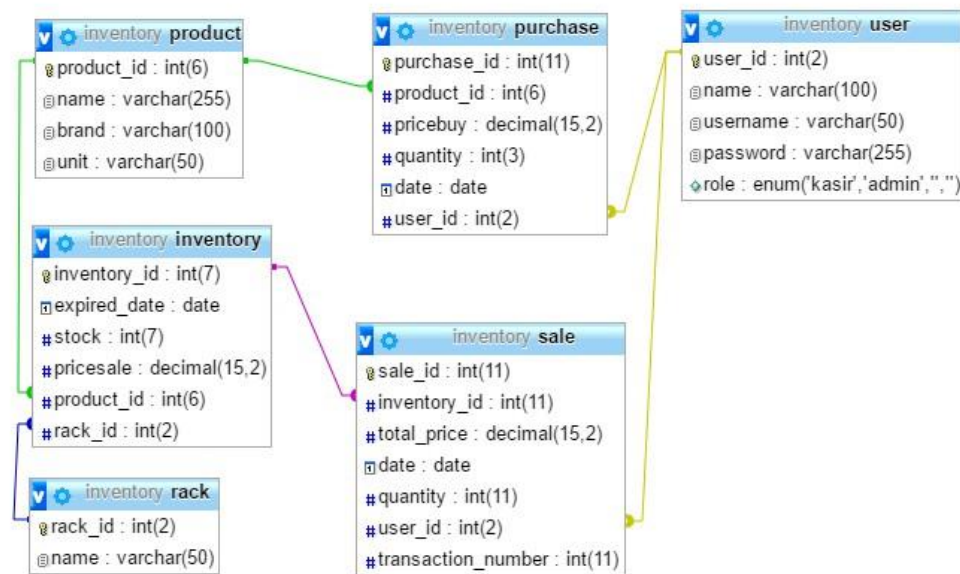
No.	Entitas	Atribut
1.	<i>User</i> entitas yang menyimpan data <i>user</i> yang berhak <i>login</i> ke aplikasi untuk mengakses aplikasi	<i>User_id</i> atribut yang menjadi identitas <i>user</i>
		<i>Username</i> atribut untuk melakukan proses <i>login</i>
		<i>Password</i> Atribut sebagai kata sandi 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 identiras 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> Entitas untuk menyimpan transaksi pembelian yang dilakukan oleh <i>user</i>	<i>Purchase_id</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.	<i>Sale</i> 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
		<i>Transaction_number</i> Atribut yang digunakan untuk membedakan setiap transaksi penjualan

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

Deskripsi setiap entitas yang ada dalam dabatase dapat dilihat pada Tabel 4. 21

Tabel 4. 21 Deskripsi Atribut Entitas Database

Nama Tabel	Deskripsi
<i>User</i>	Tipe data untuk <i>primery 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>primery 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 prediaan. 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 informarsi 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.
--	--

4.3 Implementasi

Tahap implementasi adalah tahap rancangan sistem yang telah dibuat sebelumnya diwujudkan ke dalam bentuk sistem yang sesungguhnya. Dengan berpatokan pada hasil perancangan, dilakukan pengembangan sistem untuk membangun fitur-fitur sistem agar dapat digunakan oleh pengguna. Pemaparan tahap implementasi dibagi menjadi dua bagian, yaitu implementasi algoritma shortest remaining time first dan implementasi antarmuka.

4.3.1 Implementasi Algoritma *Shortest Remaining Time First*

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 SRTF diterapkan dalam aplikasi ini dalam sebuah fungsi yang diberi nama *getListDataVisualisasi()*, dengan *variable \$rack* sebagai parameter yang menampilkan daftar rak yang akan diambil datanya. Fungsi ini merepresentasikan alur logika dari algoritma *shortest remaining time first* yang disesuaikan dengan kebutuhan dan kondisi dalam aplikasi. Pertama *variable \$data* dideklarasikan sebagai *array*, kemudian data diambil dari *database* melalui *query* yang mengambil data dari daftar *inventory* yang diurutkan berdasarkan tanggal kadaluarsa dan tidak melewati batas hari ini. Dapat dilihat pada gambar 4.6 *query* melakukan *syntax order by*, kemudian *list* data yang telah disortir dimasukkan ke

dalam *array* yang diberi nama *\$data*, selanjutnya hasil *array* akan dikirim melalui variable *\$data* untuk ditampilkan ke dalam halaman visualisasi.

```

public function getListDataVisualisasi($rack)
{
    $sql = "SELECT *, product.name as product_name , rack.name as rack_name FROM inventory JOIN
    product ON inventory.product_id = product.product_id JOIN rack ON inventory.rack_id =
    rack.rack_id WHERE inventory.rack_id = '$rack' AND expired_date >= CURDATE() ORDER BY
    expired_date ASC";
    $sqlResult = $this->myconn->query($sql);

    if ($sqlResult->num_rows > 0)
    {
        while ($row = $sqlResult->fetch_array())
        {
            $data[]=$row;
        }
        return $data;
    }
}

```

Gambar 4. 6 Implementasi Algoritma SRTF

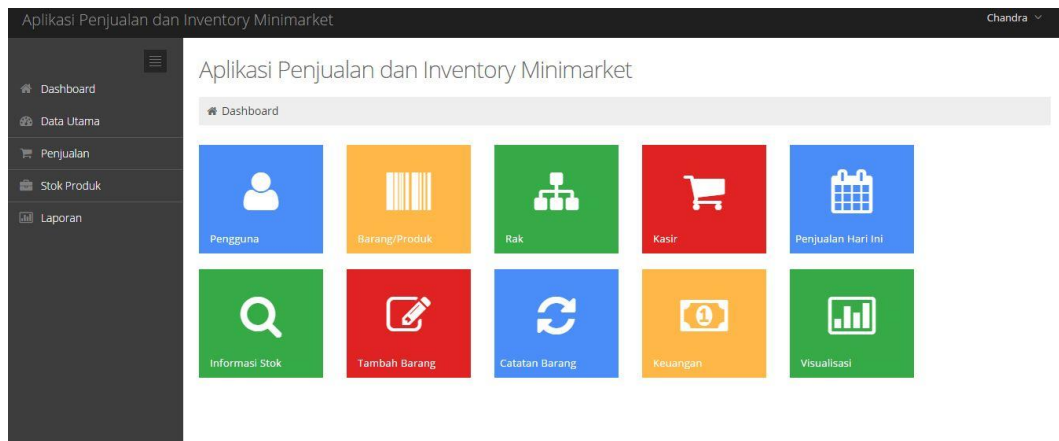
4.3.2 Implementasi Antarmuka

GUI merupakan suatu antarmuka 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. 7 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. 8 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. 9 Halaman Data Utama

Halaman data utama pada Gambar 4.9 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. 10 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 4.10.

Tambah Data

Data Utama > Pengguna > Tambah Pengguna

Tambah Data Pengguna

Nama

Username

Password

Hak Akses

Gambar 4. 11 Halaman Tambah Data Pengguna

Gambar 4.11 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: Chandra Setiawan

Username: admin2

Password: admin

Hak Akses: Admin

Save Cancel

Gambar 4. 12 Halaman Ubah Data Pengguna

Gambar 4.12 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	View Edit Delete
5	Deterjen Cair 25ml	Rinso	Deterjen Cair 25ml	View Edit Delete

Showing 1 to 2 of 2 entries Previous 1 Next

Gambar 4. 13 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 disajikan dalam bentuk tabel seperti Gambar 4.13.

Tambah Data

Gambar 4. 14 Halaman Tambah Data Barang / Produk

Gambar 4.14 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	View Edit Delete
2	Rak BCD	View Edit Delete
3	Rak EFG	View Edit Delete

Gambar 4. 15 Halaman Daftar Rak

User admin dapat melihat daftar rak yang tersedia. Informasi daftar rak ditampilkan dalam bentuk tabel seperti Gambar 4.15

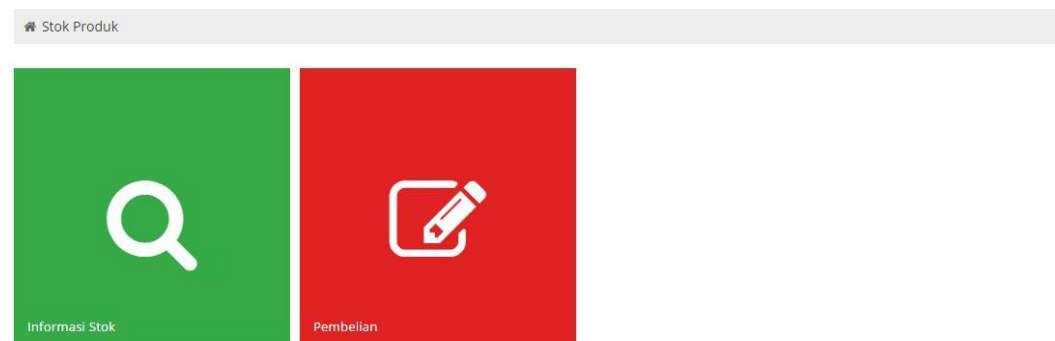
Tambah Data



Gambar 4. 16 Halaman Tambah Data Rak Barang

Untuk menambah data rak barang, *user* harus mengisi “Nama Rak” yang tersedia pada halaman tambah data rak barang seperti Gambar 4.16

Aplikasi Penjualan dan Inventory Minimarket



Gambar 4. 17 Halaman Stok Produk

Gambar 4.17 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 telah 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	Edit Delete
2	Rak A2rr	4	Minuman Botol	323	liter	2016-09-30	Edit Delete
3	Rak BCD	5	Deterjen Cair 25ml	332	pack	2016-09-24	Edit Delete
4	Rak EFG	5	Deterjen Cair 25ml	23	pack	2016-09-30	Edit Delete

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Gambar 4. 18 Halaman Daftar Informasi Stok

Gambar 4.18 menampilkan daftar informasi stok 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:

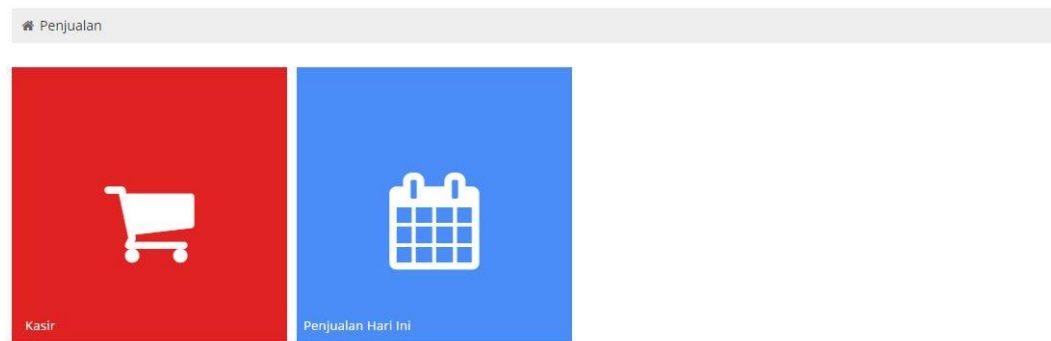
Rak:

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

Gambar 4. 19 Halaman Pembelian Produk

Halaman ini digunakan untuk menambah data pembelian terhadap produk yang telah dibeli. Halaman ini berisi *form* yang digunakan untuk memasukkan data pembelian ke dalam *database*. Setelah semua data telah diisi, kemudian tekan tombol *save* untuk mengirim data ke dalam *database*. seperti Gambar 4.19

Aplikasi Penjualan dan Inventory Minimarket



Gambar 4. 20 Halaman Penjualan

Gambar 4.20 merupakan tampilan halaman penjualan, halaman penjualan dapat diakses oleh *user admin* dan *user kasir*. Ada dua menu yang terdapat dalam 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. 21 Halaman Transaksi Penjualan

Halaman ini digunakan untuk melakukan transaksi penjualan terhadap produk yang dijual. Halaman ini berisi *form* yang digunakan untuk memasukkan data penjualan ke dalam *database*. Terdapat 3 tahapan dalam melakukan transaksi penjualan, yang pertama adalah memilih produk. Tahap kedua adalah menambah jumlah 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	View Edit Delete

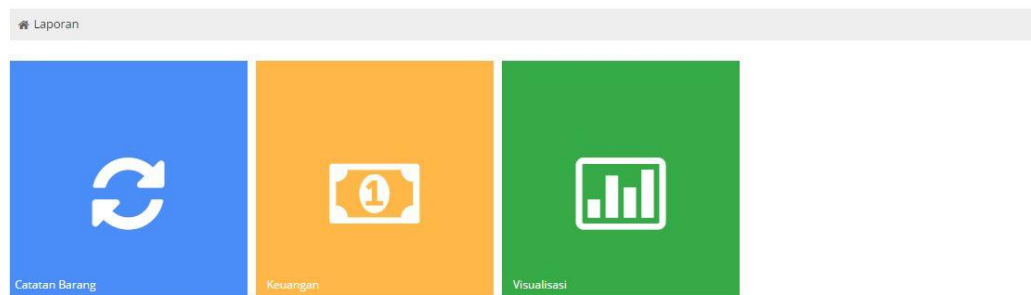
Showing 1 to 1 of 1 entries

← Previous 1 Next →

Gambar 4. 22 Halaman Daftar Penjualan Hari Ini

Gambar 4.22 menampilkan daftar penjualan hari ini, informasi ditampilkan dalam bentuk tabel lengkap dengan nama *user* yang melakukan transaksi penjualan.

Aplikasi Penjualan dan Inventory Minimarket



Gambar 4. 23 Halaman Laporan

Gambar 4.23 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	View Delete

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Gambar 4. 24 Halaman Catatan Barang

Halaman di atas menampilkan laporan catatan barang, terdapat dua jenis informasi yang ada dalam laporan ini yaitu penjualan dan pembelian. Informasi lengkap ditampilkan dalam bentuk tabel seperti Gambar 4.24

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	View Delete

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 4. 25 Halaman Laporan Keuangan

Gambar 4.25 menampilkan laporan keuangan yang digunakan untuk melihat status keuangan dari total penjualan, total pembelian. Laporan keuangan dibuat dalam bentuk tabel dilengkapi dengan informasi pendukungnya.

Aplikasi Penjualan dan Inventory Minimarket

Laporan > Visualisasi

Rak EFG Rak BCD Rak A Visualisasi Stok Dalam Rak

Minuman Botol Aqua 2015-09-22	23
Minuman Botol Aqua 2015-09-30	323
Minuman Botol Aqua 2015-10-14	34
Minuman Botol Aqua 2015-10-29	12
Minuman Botol Aqua 2015-11-24	1

Gambar 4. 26 Halaman Visualisasi

Gambar 4.26 menampilkan visualisasi stok produk yang ada dalam 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.

4.4 Pengujian Algoritma

Pengujian merupakan bagian penting dalam pengembangan aplikasi untuk memastikan bahwa aplikasi telah memenuhi *requirement*. Pada tahapan ini, penulis melakukan serangkaian pengujian sistem yang dibangun, khususnya pada implementasi algoritma yang diterapkan pada aplikasi. Hal ini dilakukan untuk memastikan bahwa algoritma *shortest remaining time first* yang diimplementasikan

ke dalam sistem telah berjalan sebagaimana mestinya. Pengujian dilakukan dengan dua metode, yaitu *white box* dan *black box*. Berikut adalah pemaparan dari hasil pengujian yang dilakukan

4.4.1 Pengujian *White Box*

Berikut adalah hasil pengujian *white box* dari implementasi algoritma *shortest remaining time first* dengan pengujian data pada fungsi *getListDataVisualisasi()*. Hasil *output* sesuai dengan yang diharapkan, maka dapat disimpulkan bahwa algoritma *shortest remaining time first* berhasil diimplementasi pada aplikasi dan menyelesaikan permasalahan dalam penelitian ini. Secara lengkap hasil pengujian *white box* dapat dilihat pada lampiran 6.

4.4.2 Pengujian *Black Box*

Pengujian *black box* merupakan salah satu cara untuk menguji sejauh mana sistem dapat digunakan dengan baik sesuai dengan yang telah direncanakan sebelumnya. Pengujian ini dilakukan agar *bugs* dan *errors* dapat ditemukan dan diminimalisasi. Hasil pengujian *blackbox* dijelaskan pada tabel 4.22 sebagai berikut:

Tabel 4. 22 Hasil Pengujian *Black Box*

<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</i>	Visualisai produk tersedia sesuai urutan yaitu 16 September 2016, 12 Desember 2016, 1 Januari 2016, dan 19 Februari 2017. Tampilan	<i>True</i>

		<i>vertical</i> , atas ke bawah.	muncul dengan <i>layout vertical</i> , atas ke bawah.	
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 4.27, Gambar 4.28, Gambar 4.29 dan Gambar 4.30 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	Edit Delete
2	Rak A	7	Susu	35	Box	2016-12-12	Edit Delete
3	Rak A	8	Minyak	40	Botol	2017-01-01	Edit Delete
4	Rak A	6	San Miguel Pin 640 Ml	30	Botol	2017-02-19	Edit Delete

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Gambar 4. 27 Hasil *Test Case* 1.A

Gambar di atas merupakan hasil daftar informasi stok 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 Kraft 2016-09-16	10
Susu Buncow 2016-12-12	35
Minyak Bimoli 2017-01-01	40
San Miguel Pin 640 Ml San Miguel 2017-02-19	30

Gambar 4. 28 Hasil *Test Case* 1.B

Gambar di atas 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	Edit Delete
2	Rak A	7	Susu	35	Box	2016-12-12	Edit Delete
3	Rak A	8	Minyak	40	Botol	2017-01-01	Edit Delete
4	Rak A	6	San Miguel Pin 640 MI	30	Botol	2017-02-19	Edit Delete
5	Rak A	10	Kacang	90	Bungkus	2016-07-01	Edit Delete

Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 4. 29 Hasil *Test Case 2.A*

Gambar di atas merupakan hasil daftar informasi stok yang dimasukkan berdasarkan *test case* kedua yang ditampilkan dalam bentuk tabel. Gambar menampilkan produk sesuai data yang dimasukkan oleh *user* dengan menambahkan produk 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 Dancow 2016-12-12	35
Minyak Birel 2017-01-01	40
San Miguel Pin 640 MI San Miguel 2017-02-19	30

Gambar 4. 30 Hasil *Test Case 2.B*

Gambar di atas merupakan hasil visualisasi susunan produk yang telah diurutkan berdasarkan tanggal kedaluarsa. Terlihat bahwa susunan produk berdasarkan tanggal kedaluarsa. Data produk yang telah kedaluarsa tidak masuk ke dalam visualisasi.

Berdasarkan hasil pengujian yang telah dilakukan maka dapat disimpulkan bahwa algoritma *shortest remaining time first* dapat diimplementasikan dalam

aplikasi dan berjalan dengan baik untuk memberikan visualisasi produk dengan benar sesuai tujuan dari penelitian ini.

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 wawancara dan observasi. Dari data analisa tersebut menghasilkan sebuah rancangan yang direpresentasikan dalam bentuk *UML Diagram* yang terdiri dari *use case diagram*, *class diagram*, dan *database model*. 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* yang dilakukan oleh penulis sendiri dan pengujian *black box* yang dilakukan oleh pengguna. 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 *minimarket* dengan jumlah barang dan transaksi yang berbeda antara setiap *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. Pembuatan laporan transaksi yang lebih spesifik dan mudah agar memudahkan pengguna. Untuk kebutuhan penelitian dapat menggunakan pendekatan output, yaitu mengukur produk barang berdasarkan jumlah barang berdasarkan tingkat jumlah keluar-masuk barang, jumlah transaksi penjualan per hari 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.
- 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*.
- Laudon, K., & Laudon, J. (1998). *Management Information System New Approaches to Organization & Technology*. New Jersey: Prentice Hall, Inc.
- 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.
- 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).
- Sommerville, I. (2011). *Software Engineering Ninth Edition*. Pearson.
- Sutabri, T. (2004). *Analisa Sistem Informasi*. Yogyakarta: Andi Offset.
- Vrat, P. (2014). *Materials Management An Integrated Systems Approach*. India: Springer India.
- Wilkison, J. (1992). *Accounting and Information*. John Wiley & Sons, Inc.
- 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 minimarket* 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 *inventory* yang bekerja pada saat pihak *minimarket* 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 *admin* yang dapat melihat, memanipulasi serta mengelola data transaksi yang adalah sebagai pemilik bisnis. Berikut merupakan karakteristik pengguna pada sistem:

Pengguna	Aksi
<i>User</i>	<ol style="list-style-type: none"> 1. <i>User</i> dapat melakukan <i>login</i> ke dalam sistem. 2. <i>User</i> dapat membuat, melihat, mengubah, dan menghapus data <i>user</i>. 3. <i>User</i> dapat membuat, melihat, mengubah, dan menghapus data barang. 4. <i>User</i> dapat membuat, melihat, mengubah, dan menghapus data rak.

	<p>5. <i>User</i> dapat membuat, melihat, mengubah, dan menghapus data penjualan.</p> <p>6. <i>User</i> dapat membuat, melihat, mengubah, dan menghapus data pembelian.</p> <p>7. <i>User</i> dapat melakukan <i>logout</i> ke dalam sistem.</p>
--	--

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 (*User Interface*)

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 (*Hardware*)

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

- *Device* : Notebook PC Asus N46VJ
- *Operating Systems* : Windows® 7 Ultimate™ 64-bit
- *Processor* : Intel® Core™ i5-3210M CPU @ 2.5 GHz
- *Memory* : 4.00 GB RAM

3.1.3 Antarmuka Perangkat Lunak (*Software*)

Antarmuka *software* (perangkat lunak) yang dibutuhkan untuk membantu kelengkapan dari pembangunan aplikasi yang sedang dirancang meliputi:

- Google® Chrome™ versi 48.0.2564.109 m, sebagai web browser
- XAMPP™ versi 3.2.1, sebagai *web server apache, database server* dan *application server*
- Notepad++™ versi 6.8.6 sebagai IDE, digunakan untuk penulisan dan penyuntingan bahasa pemrograman. Bahasa pemrograman yang digunakan pada penelitian ini adalah PHP.

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>
<i>Trigger</i>	Pengguna mengakses aplikasi
<i>Precondition</i>	Aplikasi menampilkan halaman <i>login</i>
<i>Basic Path</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan <i>username</i> dan <i>password</i> 2. Sistem melakukan pengecekan 3. <i>Login</i> berhasil, sistem menampilkan halaman menu utama
<i>Alternative Path</i>	Tidak ada

Add Barang

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

	4. Produk berhasil ditambahkan
<i>Alternative Path</i>	Tidak ada

Edit Barang

Nama <i>Use case</i>	<i>Edit</i> Barang
<i>Trigger</i>	Pengguna memilih menu untuk mengubah barang
<i>Precondition</i>	Menampilkan menu <i>edit</i> barang
<i>Basic Path</i>	<ol style="list-style-type: none"> 1. Pengguna sudah <i>login</i> 2. Melakukan pengisian data yang dicari 3. Sistem menampilkan data yang sesuai dengan permintaan 4. Merubah informasi data 5. Produk berhasil diubah
<i>Alternative Path</i>	Tidak ada

Delete Barang

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

	<ol style="list-style-type: none"> 3. Sistem menampilkan data yang sesuai dengan permintaan 4. Hapus barang 5. Sistem menampilkan pesan <i>pop up</i> “Apakah anda ingin melakukan penghapusan” 6. Menekan tombol <i>yes</i> 7. Barang berhasil dihapus
<i>Alternative Path</i>	Tidak Ada

Laporan Barang Masuk

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

Laporan Barang Keluar

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

Laporan Stok Barang

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

<i>Alternative Path</i>	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*

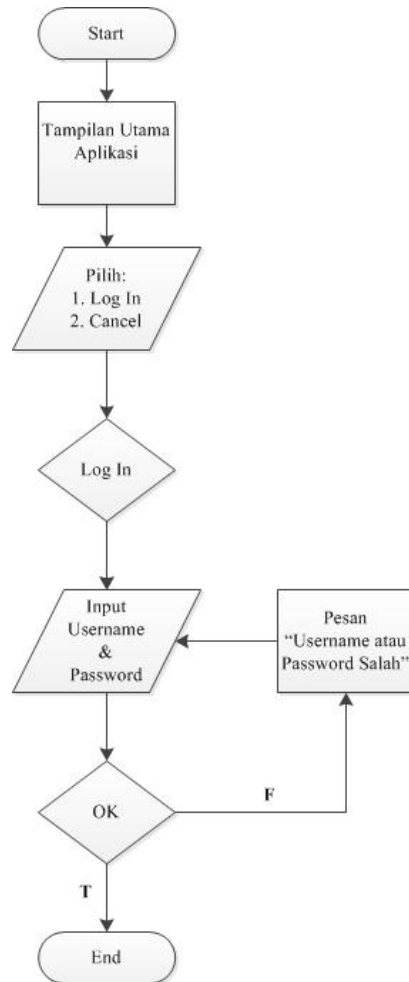
3.4 Kebutuhan Lain

3.4.1 Database

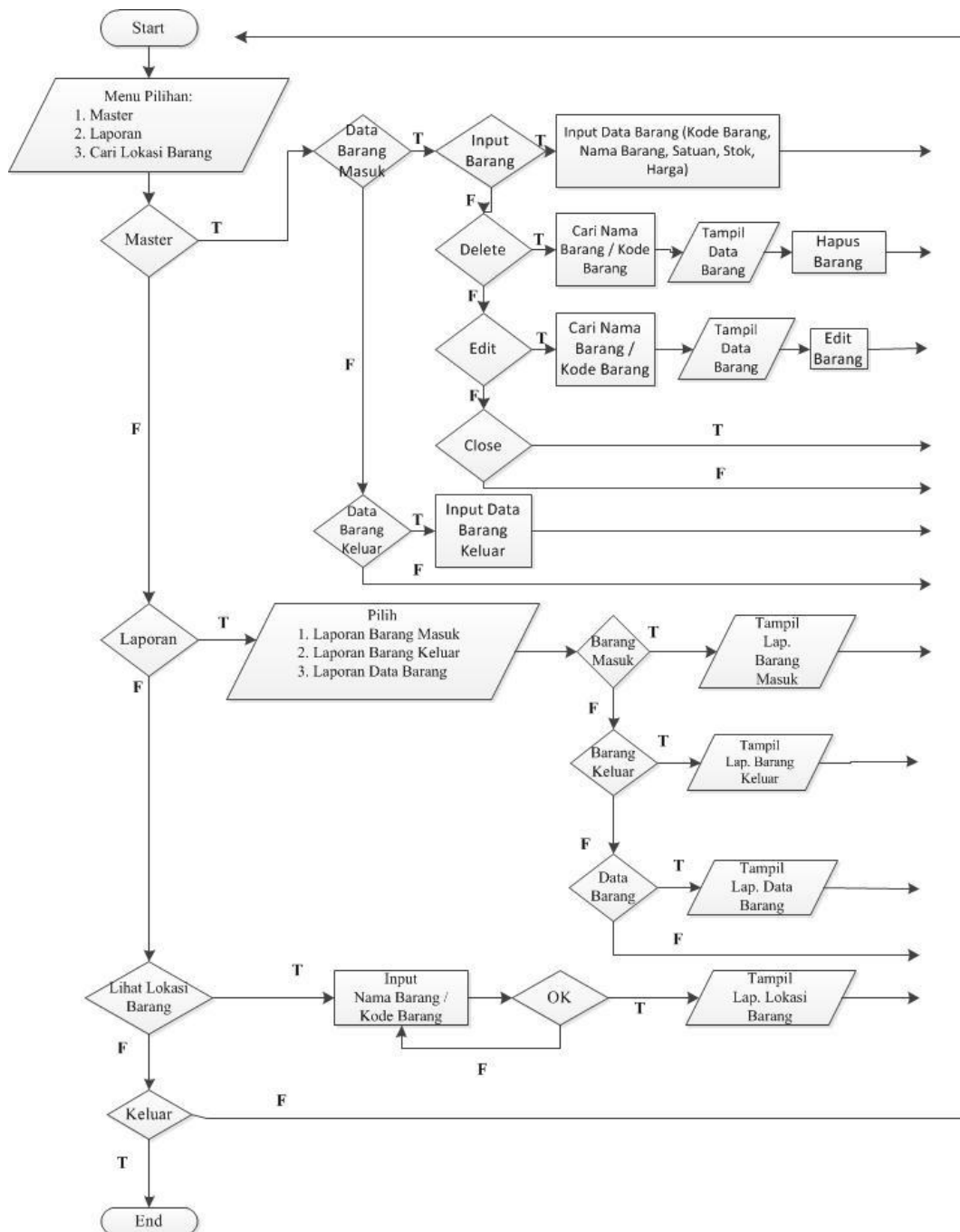
Database yang akan digunakan dalam aplikasi ini adalah Microsoft® SQL Server™. *SQL Server* merupakan *Relational Database Management System* (RDBMS) yang dikembangkan oleh Microsoft®. Fungsi utama *SQL Server* adalah menampung dan menggunakan data yang terintegrasi dengan aplikasi.

3.4.2 Flowchart Sistem

- *Flowchart 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 : Jadwal Penelitian

No	Jenis Kegiatan	2015																2016																														
		Jul				Aug				Sep				Oct, Nov Dec				Jan				Feb				Mar				April				Mei				June				Juli						
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4							
1	Analisis persyaratan																																															
2	Definisi persyaratan																																															
3	Perancangan sistem																																															
4	Perancangan <i>software</i>																																															
5	Implementasi unit																																															
6	Pengujian unit																																															
7	Integrasi sistem																																															
7	Pengujian sistem																																															
8	Penulisan tugas akhir																																															

Lampiran 3 : Dokumen Elisitasi

Requirement Elicitation Aplikasi Penjualan dan *Inventory Minimarket* Dengan Menggunakan *Shortest Remaining Time First Algorithm*

Requirement Elicitation Tahap 1

Functional	
No.	Analisis Kebutuhan
Saya ingin sistem dapat	
1.	Mengijinkan <i>User</i> dapat melakukan <i>login</i> ke dalam sistem
2.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke masing – masing halaman sesuai dengan hak akses yang diberikan
3	Mengijinkan <i>User</i> (<i>Admin</i>) masuk ke halaman pengisian formulir data barang
4.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke halaman pengisian formulir penjualan barang
5.	Mengijinkan <i>User</i> (<i>Admin</i>) masuk ke halaman pengisian formulir penambahan data pengguna
6.	Mengijinkan <i>User</i> (<i>Admin</i>) masuk ke halaman pengisian formulir penambahan data rak barang
7.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke halaman penambahan stok barang
8.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke halaman pengurangan stok barang
9.	Mengijinkan <i>User</i> (<i>Admin</i>) dapat melihat daftar pengguna

10.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) mengubah data pengguna inisial dan kata sandi
11.	Mengijinkan <i>User</i> (<i>Admin</i>) mengubah dan menghapus data barang
12.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) dapat melihat daftar barang yang ada dalam <i>inventory</i>
13.	Menampilkan pesan peringatan pada saat menghapus data pengguna, data barang, dan data rak barang
14.	Menampilkan pesan peringatan pada saat penambahan data pengguna, data barang dan data rak barang
15.	Mengijinkan <i>User</i> (<i>Admin</i>) dapat melihat daftar rak barang
16.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) dapat melihat daftar stok barang yang dilengkapi informasi kode barang, nama barang, jumlah, dan tanggal kadaluarsa
17.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) melakukan transaksi penjualan
18.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) melihat keseluruhan transaksi penjualan hari ini
19.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan kode barang
20.	Menampilkan pesan peringatan pada saat <i>User</i> memasukkan jumlah barang beli lebih besar dari jumlah stok yang tersedia
21.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan jumlah total bayar
22.	Menginjinkan <i>User</i> (<i>Admin</i>) dapat melihat rincian laporan keluar – masuk barang
23.	Mengijinkan <i>User</i> (<i>Admin</i>) dapat melihat rincian laporan keuangan

24.	Mengjinkan <i>User (Admin)</i> dapat melihat laporan keseluruhan mencakup laporan barang terjual, laporan penambahan stok, laporan pengurangan stok dan laporan aset
25.	Mengijinkan <i>User (Admin)</i> dapat melihat visualisasi barang
26.	Mengijinkan <i>User (Admin dan kasir)</i> logout dari sistem

<i>Non Functional</i>	
No.	Analisis Kebutuhan
Saya ingin sistem dapat	
1.	Menyajikan tampilan <i>User interface</i> yang <i>User friendly</i> .
2.	Mampu berjalan dengan berbasis web.
3.	Menampilkan warna tampilan yang tegas dan jelas.

Requirement Elicitation Tahap 2

Elisitasi Tahap II dibentuk berdasarkan Elisitasi Tahap I yang diklasifikasikan melalui metode MDI.

<i>Functional</i>				
No.	Analisis Kebutuhan	M	D	I
Saya ingin sistem dapat				
1.	Mengijinkan <i>User</i> dapat melakukan <i>login</i> ke dalam sistem	√		

2.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke masing – masing halaman sesuai dengan hak akses yang diberikan		√	
3	Mengijinkan <i>User (Admin)</i> masuk ke halaman pengisian formulir data barang	√		
4.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke halaman pengisian formulir penjualan barang	√		
5.	Mengijinkan <i>User (Admin)</i> masuk ke halaman pengisian formulir penambahan data pengguna	√		
6.	Mengijinkan <i>User (Admin)</i> masuk ke halaman pengisian formulir penambahan data rak barang	√		
7.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke halaman penambahan stok barang		√	
8.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke halaman pengurangan stok barang		√	
9.	Mengijinkan <i>User (Admin)</i> dapat melihat daftar pengguna	√		
10.	Mengijinkan <i>User (Admin dan kasir)</i> mengubah data pengguna inisial dan kata sandi	√		
11.	Mengijinkan <i>User (Admin)</i> mengubah dan menghapus data barang	√		
12.	Mengijinkan <i>User (Admin dan kasir)</i> dapat melihat daftar barang yang ada dalam <i>inventory</i>	√		
13.	Menampilkan pesan peringatan pada saat menghapus data pengguna, data barang, dan data rak barang		√	

14.	Menampilkan pesan peringatan pada saat penambahan data pengguna, data barang dan data rak barang		√	
15.	Mengijinkan <i>User (Admin)</i> dapat melihat daftar rak barang	√		
16.	Mengijinkan <i>User (Admin dan kasir)</i> dapat melihat daftar stok barang yang dilengkapi informasi kode barang, nama barang, jumlah, dan tanggal kadaluarsa		√	
17.	Mengijinkan <i>User (Admin dan kasir)</i> melakukan transaksi penjualan	√		
18.	Mengijinkan <i>User (Admin dan kasir)</i> melihat keseluruhan transaksi penjualan hari ini		√	
19.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan kode barang		√	
20.	Menampilkan pesan peringatan pada saat <i>User</i> memasukkan jumlah barang beli lebih besar jumlah stok yang tersedia		√	
21.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan jumlah total bayar		√	
22.	Menginjinkan <i>User (Admin)</i> dapat melihat rincian laporan keluar – masuk barang		√	
23.	Mengijinkan <i>User (Admin)</i> dapat melihat rincian laporan keuangan	√		
24.	Mengijinkan <i>User (Admin)</i> dapat melihat laporan keseluruhan mencakup laporan barang		√	

	terjual, laporan penambahan stok, laporan pengurangan stok dan laporan aset			
25.	Mengijinkan <i>User (Admin)</i> dapat melihat visualisasi barang	√		
26.	Mengijinkan <i>User (Admin dan kasir)</i> logout dari sistem	√		

<i>Non Functional</i>				
No.	Analisis Kebutuhan	M	D	I
Saya ingin sistem dapat				
1.	Menyajikan tampilan <i>User interface</i> yang <i>User friendly</i> .	√		
2.	Mampu berjalan dengan berbasis web.	√		
3.	Menampilkan warna tampilan yang tegas dan jelas.		√	

Keterangan :

M = *Mandatory* (yang diinginkan),

D = *Desirable* (diperlukan),

I = *Inessential* (yang tidak diinginkan)

Requirement Elicitation Tahap 3

Berdasarkan Elisitasi Tahap II, dibentuklah Elisitasi Tahap III yang diklasifikasikan kembali dengan menggunakan metode TOE dengan opsi LMH. Berikut adalah *requirement* tersebut:

Functional										
Feasibility		T			O			E		
		L	M	H	L	M	H	L	M	H
No.	Analisis Kebutuhan									
Saya ingin sistem dapat										
1.	Mengijinkan <i>User</i> dapat melakukan <i>login</i> ke dalam sistem	√			√			√		
2.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke masing – masing halaman sesuai dengan hak akses yang diberikan		√			√			√	
3	Mengijinkan <i>User</i> (<i>Admin</i>) masuk ke halaman pengisian formulir data barang		√		√				√	
4.	Mengijinkan <i>User</i> (<i>Admin</i> dan kasir) masuk ke halaman pengisian formulir penjualan barang		√		√				√	
5.	Mengijinkan <i>User</i> (<i>Admin</i>) masuk ke halaman pengisian formulir penambahan data pengguna		√		√				√	

6.	Mengijinkan <i>User (Admin)</i> masuk ke halaman pengisian formulir penambahan data rak barang		√			√			√	
7.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke halaman penambahan stok barang		√			√			√	
8.	Mengijinkan <i>User (Admin dan kasir)</i> masuk ke halaman pengurangan stok barang		√			√			√	
9.	Mengijinkan <i>User (Admin)</i> dapat melihat daftar pengguna		√				√	√		
10.	Mengijinkan <i>User (Admin dan kasir)</i> mengubah data pengguna inisial dan kata sandi		√			√		√		
11.	Mengijinkan <i>User (Admin)</i> mengubah dan menghapus data barang		√				√		√	
12.	Mengijinkan <i>User (Admin dan kasir)</i> dapat melihat daftar barang yang ada dalam <i>inventory</i>	√			√			√		
13.	Menampilkan pesan peringatan pada saat menghapus data pengguna, data barang, dan data rak barang	√			√			√		
14.	Menampilkan pesan peringatan pada saat penambahan data									

	pengguna, data barang dan data rak barang								
15.	Mengijinkan <i>User (Admin)</i> dapat melihat daftar rak barang		√		√			√	
16.	Mengijinkan <i>User (Admin dan kasir)</i> dapat melihat daftar stok barang yang dilengkapi informasi kode barang, nama barang, jumlah, dan tanggal kadaluarsa		√			√			√
17.	Mengijinkan <i>User (Admin dan kasir)</i> melakukan transaksi penjualan		√			√			√
18.	Mengijinkan <i>User (Admin dan kasir)</i> melihat keseluruhan transaksi penjualan hari ini		√			√			√
19.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan kode barang	√			√			√	
20.	Menampilkan pesan peringatan pada saat <i>User</i> memasukkan jumlah barang beli lebih besar dari jumlah stok yang tersedia	√			√			√	
21.	Menampilkan pesan peringatan pada saat <i>User</i> salah memasukkan jumlah total bayar	√			√			√	

22.	Mengizinkan <i>User (Admin)</i> dapat melihat rincian laporan keluar – masuk barang		√			√		√		
23.	Mengizinkan <i>User (Admin)</i> dapat melihat rincian laporan keuangan		√			√		√		
24.	Mengizinkan <i>User (Admin)</i> dapat melihat laporan keseluruhan mencakup laporan barang terjual, laporan penambahan stok, laporan pengurangan stok dan laporan aset		√			√		√		
25.	Mengizinkan <i>User (Admin)</i> dapat melihat visualisasi barang		√			√			√	
26.	Mengizinkan <i>User (Admin dan kasir)</i> logout dari sistem	√			√			√		

Non Functional										
Feasibility		T			O			E		
		L	M	H	L	M	H	L	M	H
No.	Analisis Kebutuhan									
Saya ingin sistem dapat										
1.	Menyajikan tampilan <i>User interface</i> yang <i>User friendly</i> .		√		√				√	
2.	Mampu berjalan dengan berbasis web.		√		√				√	

3.	Menampilkan warna tampilan yang tegas dan jelas.		√			√		√		
----	--	--	---	--	--	---	--	---	--	--

Keterangan:

T = *Technical*

E = *Economic*

O = *Operational*

M = *Middle*

L = *Low*

H = *High*

Mengetahui,

()

Pemilik Minimarket Indri

Lampiran 4 : Wawancara

WAWANCARA

Transkrip wawancara dengan pemilik Minimarket:

Tempat : Jl. Kiay Modjo Lingkungan III No. 243, Kelurahan Kampung
Jawa, Kabupaten Minahasa, Sulawesi Utara

Waktu : 21 Juli 2015, Pukul 16:00 WITA s.d Selesai

Keterangan :P: Pewawancara (Penulis)

SS: *Informan* (Pemilik Minimarket XYZ)

Di bawah ini merupakan rangkuman wawancara yang penulis (selanjutnya disebut dengan P) lakukan dengan Ibu Suryati Suratinoyo umur (selanjutnya disebut dengan SS), wawancara dilakukan dengan menggunakan bahasa daerah dikarenakan keterbatasan akan penggunaan bahasa Indonesia yang baik. Transkrip wawancara berikut ini sudah diterjemahkan ke dalam bahasa Indonesia .

P : Selamat sore Ibu saya Chandra dari Universitas Bakrie, saya disini mau wawancara untuk penelitian saya yang berjudul aplikasi penjualan dan *inventory* untuk *minimarket*

SS : Iya, silahkan saja. Apa yang mau ditanyakan?

P : Jadi begini Ibu, saya ingin menanyakan tentang proses pembelian barang untuk minimarket, apakah dicatat ?

SS : Iya dicatat, apa saja yang dibeli supaya tahu apa saja yang dibeli

P : Bagaimana proses pencatatannya Bu ?

SS : Tulis manual di buku aja biasa, yang penting ada catatan saja, masih manual dek

P : Oh gitu, jadi masih manual sistemnya, tidak takut Bu apabila ada yang lupa dicatat dan bukunya rusak atau hilang ?

- SS : Takut sih dek, tapi harus teliti untuk catat semua barang, kalau bukunya rusak, kita ulang lagi proses pencatatannya.
- P : Bagaimana ibu mencari tahu informasi jumlah stok barang yang masih ada ?
- SS : Iya, pengecekan lewat buku catatan secara manual, jadi saya selalu melakukan pencatatan berapa banyak barang yang saya beli dan untuk barang yang terjual saya juga catat, jadi saya bisa tahu barang yang masih ada sesuai dengan catatan atau tidak.
- P : Seperti itu ya Bu, berarti ibu selalu melihat dari buku catatan saja ya , barang yang sudah terjual
- SS : iya iya
- P : Ibu pernah mengalami kerugian disebabkan karena barang kadaluarsa ?
- SS : Lumayan sering, tapi ada beberapa barang yang sudah kadaluarsa tidak mungkin kita jual, yah kalau usaha seperti ini pasti ada kerugian kalau barangnya kadaluarsa.
- P : Lanjut ya Bu, Bagaimana cara Ibu untuk mengetahui keuntungan per bulannya ?
- SS : Jadi iya dicatat secara manual semua, apa saja transaksi pembelian, transaksi penjualan, terus keuntungan berapa nanti tinggal dijumlahin aja semuanya. Berapa pengeluaran dan berapa banyak pembelian stok barang.
- P : Oke Bu, jadi disini saya mengajukan sebuah sistem untuk manajemen penjualan barang dan persediaan, jadi nanti tidak akan melakukan proses pencatatan secara manual. Apakah menurut ibu ini penting?
- SS : Oh, boleh saja, yang terpenting mudah dipakai dan dimengerti, asalkan aplikasi mudah digunakan.
- P : Yang sederhana ya Bu, dalam aplikasi ini juga nanti Ibu bisa melihat laporan stok barang, laporan keuangan dan laporan keluar-masuk barang, apakah menurut Ibu fitur ini penting dan mau menggunakannya ?
- SS : Kalau misalnya membuat bisnis saya menjadi lebih mudah saya rasa ini penting dan saya mau menggunakannya.

P : Oh begitu ya Bu, Oke kalau begitu bu, sepertinya sudah cukup wawancara saya untuk hari ini. Terimakasih ya bu atas informasinya

SS : Iya , sama sama

Manado, 21 Juli 2015

()
Pemilik Minimarket Indri

SURAT KETERANGAN WAWANCARA

Yang bertanda tangan di bawah ini:

Nama : Suryati Suratinoyo
Alamat : Jl. Kiay Modjo Lingkungan III No. 243, Kelurahan
Kampung Jawa, Kabupaten Minahasa, Sulawesi Utara
Pekerjaan : Wiraswasta

Menerangkan bahwa:

Nama : Chandra Setiawan Gimon
Nim : 1112001043
Jurusan : Informatika

Mahasiswa yang bersangkutan telah melakukan wawancara kebutuhan user dalam rangka penyusunan skripsi dengan judul “ *RANCANG BANGUN APLIKASI PENJUALAN DAN INVENTORY MINIMARKET DENGAN MENGGUNAKAN SHORTEST REMAINING TIME FIRST ALGORITHM*” sebagai penelitian tugas akhir perkuliahan.

Surat keterangan ini diberikan dan digunakan sebagaimana mestinya.

Manado, 21 Juli 2015

Mengetahui,

()

Pemilik Minimarket Indri

Lampiran 5 : Pengujian Aplikasi

SURAT KETERANGAN PENGUJIAN APLIKASI

Yang bertanda tangan di bawah ini:

Nama : Suryati Suratinoyo

Alamat : Jl. Kiay Modjo Lingkungan III No. 243, Kelurahan
Kampung Jawa, Kabupaten Minahasa, Sulawesi Utara

Menyatakan bahwa aplikasi **Penjualan dan Inventory** untuk keperluan tugas akhir berjudul **Rancang Bangun Aplikasi Penjualan dan Inventory Minimarket Dengan Menggunakan Shortest Remaining Time First Algorithm** oleh **Chandra Setiawan Gimon** jurusan **Informatika** angkatan **2011** sudah diuji oleh pemilik *minimarket* dan sudah berjalan dengan baik sesuai dengan tujuan akhir tugas akhir tersebut. Adapun beberapa hal yang perlu diperhatikan sebagai berikut:

1. Aplikasi diuji dengan menggunakan perangkat komputer Notebook Asus N46VJ Series; Windows 7 64-bit; Intel Core i5 2.50GHz; RAM 4GB.
2. Aplikasi *database* yang digunakan adalah MySQL yang dipasang pada *server* Apache di jaringan localhost.
3. Aplikasi *browser* yang digunakan adalah Google Chrome.

Demikian surat keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Manado, 14 Mei 2016

Tanda Tangan Penguji

()

Data Pengujian *Functionality* Aplikasi Penjualan dan *Inventory Minimarket*

Menu	Aksi	Hasil Yang Diharapkan	Berhasil
<i>Login</i>	Memasukkan <i>username</i> dan <i>password</i> , klik tombol <i>Login</i>	Masuk ke dalam halaman utama	
<i>Dashboard</i>	Klik Menu Pengguna	Menampilkan halaman untuk melihat daftar pengguna	
	Klik Menu Barang/Produk	Menampilkan halaman daftar barang / produk	
	Klik Menu Rak	Menampilkan halaman daftar rak	
	Klik Menu Kasir	Menampilkan halaman transaksi penjualan	
	Klik Menu Penjualan Hari Ini	Menampilkan halaman daftar penjualan hari ini	
	Klik Menu Informasi Stok	Menampilkan halaman daftar informasi stok	
	Klik Menu Tambah Barang	Menampilkan halaman tambah data produk	
	Klik Menu Catatan Barang	Menampilkan halaman daftar catatan barang	
	Klik Menu Keuangan	Menampilkan halaman daftar laporan keuangan	
	Pilih Menu Visualisasi	Menampilkan halaman visualisasi	
Data Utama	Pilih Menu Pengguna	Menampilkan halaman untuk melihat daftar pengguna	

	Klik tombol Tambah	Menampilkan halaman formulir tambah data pengguna	
	Klik <i>Save</i>	Menyimpan data pengguna yang ingin ditambahkan	
	Klik Menu Barang/Produk	Menampilkan halaman daftar barang / produk	
	Klik tombol Tambah	Menampilkan halaman formulir tambah data produk	
	Klik <i>Save</i>	Menyimpan data produk yang ingin ditambahkan	
	Klik Menu Rak	Menampilkan halaman daftar rak	
	Klik tombol Tambah	Menampilkan halaman formulir tambah data rak	
	Klik <i>Save</i>	Menyimpan data rak yang ingin ditambahkan	
Penjualan	Klik Menu Kasir	Menampilkan halaman transaksi penjualan	
	Klik tombol Tambah	Menampilkan halaman formulir tambah data penjualan	
	Klik <i>Save</i>	Menyimpan transaksi penjualan yang dilakukan	
	Klik Menu Penjualan Hari Ini	Menampilkan halaman daftar penjualan hari ini	
	Klik <i>Edit</i>	Menampilkan halaman untuk mengubah data transaksi penjualan yang dipilih	
	Klik <i>Delete</i>	Menampilkan halaman untuk menghapus data	

		transaksi penjualan yang dipilih	
Stok Produk	Klik Menu Informasi Stok	Menampilkan halaman daftar informasi stok	
	Klik <i>Edit</i>	Menampilkan halaman untuk mengubah data informasi stok yang dipilih	
	Klik <i>Delete</i>	Menampilkan halaman untuk menghapus data informasi stok yang dipilih	
	Klik Pembelian	Menampilkan halaman transaksi penjualan	
	Klik <i>Save</i>	Menyimpan transaksi pembelian yang dilakukan	
Laporan	Klik Menu Catatan Barang	Menampilkan halaman daftar catatan barang	
	Klik <i>Delete</i>	Menampilkan halaman untuk menghapus data laporan catatan barang yang dipilih	
	Klik Menu Keuangan	Menampilkan halaman daftar laporan keuangan	
	Klik <i>Delete</i>	Menampilkan halaman untuk menghapus data laporan keuangan yang dipilih	
	Pilih Menu Visualisasi	Menampilkan halaman visualisasi produk yang disimpan dalam rak	

Data Pengujian *Usability* Aplikasi Penjualan dan *Inventory Minimarket*

No	Pertanyaan	TS	KS	N	S	SS
	<i>Learnability</i>					
1.	Tulisan teks yang digunakan untuk halaman tersebut mudah dan jelas.					
2.	Kata – kata dalam menu yang ada cukup mudah untuk dipahami.					
3.	Penempatan menu-menu pada sistem mudah dijangkau.					
4.	Kata-kata yang dipakai dalam memberikan informasi sudah jelas.					
	<i>Efficiency</i>					
5.	Saat menu Anda klik, sistem dapat menampilkan dengan cepat.					
6.	Pencarian informasi tentang data barang mudah dan cepat.					
7.	Proses penjualan dapat dilakukan dengan mudah dan cepat.					
8.	Visualisai barang dapat membantu Anda dalam menentukan barang yang harus dijual terlebih dahulu.					
9.	Laporan – laporan yang tersedia dalam aplikasi dapat membantu dan mengontrol					

	proses jual beli serta membantu pemilik dalam mengembangkan bisnisnya					
	<i>Memorability</i>					
10.	Letak posisi menu atau tombol mudah diingat.					
11.	Pemilihan kata dalam aplikasi mudah diingat.					
	<i>Error</i>					
12.	Tidak ada informasi yang tumpang tindih dalam sistem					
13.	Tidak terdapat link yang tidak sesuai atau belum jadi dalam aplikasi ketika menu atau link di klik.					
14.	Tidak terdapat klik menu yang tidak memberikan respon atau apapun.					
	<i>Satisfaction</i>					
15.	Aplikasi memberikan layanan dan informasi mudah dipahami dan nyaman digunakan.					

16.	Secara keseluruhan Anda puas dengan informasi yang diberikan oleh aplikasi.					
17.	Aplikasi ini memberikan informasi yang akurat dan membantu Anda dalam mengembangkan bisnis Anda					
18.	Secara keseluruhan Anda puas dengan fitur, dan kemudahan yang diberikan oleh aplikasi.					
19.	Aplikasi memudahkan Anda dalam mengontrol proses bisnis Anda dari jarak jauh					
20.	Apabila telah diimplementasi, Anda akan memakai aplikasi ini dalam proses bisnis Anda					

Keterangan:

TS = Tidak Setuju

KS = Kurang Setuju

N = Netral

S = Setuju

SS = Sangat Setuju

Data Pengujian *Interface* Aplikasi Penjualan dan *Inventory Minimarket*

No	Pertanyaan	TS	KS	N	S	SS
	<i>Pleasure</i>					
1.	Aplikasi memberikan tampilan yang menyenangkan saat digunakan. (<i>happiness</i>)					
2.	Aplikasi menampilkan kemasan yang menarik. (<i>beautiful</i>)					
3.	Tampilan aplikasi disukai. (<i>like</i>)					
4.	Tampilan aplikasi memberikan rasa positif. (<i>positive</i>)					
	<i>Classic Aesthetics</i>					
5.	Menampilkan konten yang terstruktur dengan baik. (<i>well-organized</i>)					
6.	Menampilkan konten aplikasi dengan porsi yang simetris dan pembagian posisi yang sesuai dipandang. (<i>balance</i>)					
7.	Menampilkan desain yang jelas. (<i>clear design</i>)					
8.	Memberikan tampilan aplikasi secara responsive. (<i>responsive</i>)					
	<i>Expresif Aesthetics</i>					
9.	Tampilan secara keseluruhan memiliki kreasi yang baik. (<i>creative</i>)					

10.	Tampilan aplikasi menarik dan memikat. (<i>fascinating</i>)					
11.	Tampilan aplikasi tidak meniru web lainya. (<i>original</i>)					
12.	Memberikan kemasan tampilan yang canggih. (<i>sophisticated</i>)					

Keterangan:

TS = Tidak Setuju

KS = Kurang Setuju

N = Netral

S = Setuju

SS = Sangat Setuju

Lampiran 6 : Hasil Pengujian *White Box*

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
1	<i>getListData Visualisasi</i>	Array ([0] => Array ([0] => 14 [inventory_id] => 14 [1] => 2016-09-30 [expired_date] => 2016-09-30 [2] => 0 [stock] => 0 [3] => 22000.00 [pricesale] => 22000.00 [4] => 4 [product_id] => 4 [5] => 1 [rack_id] => 1 [6] => 4 [7] => Minuman Botol 600ml [name] => Rak ABC [8] => Aqua [brand] => Aqua [9] => liter [unit] => liter [10] => 1 [11] => Rak ABC [12] => Minuman Botol 600ml [product_name] => Minuman Botol 600ml [13] => Rak ABC [rack_name] => Rak ABC) [1] => Array ([0] => 15 [inventory_id] => 15 [1] => 2016-11-11 [expired_date] => 2016-11-11 [2] => 6 [stock] =>	Array ([0] => Array ([0] => 14 [inventory_id] => 14 [1] => 2016-09-30 [expired_date] => 2016-09-30 [2] => 0 [stock] => 0 [3] => 22000.00 [pricesale] => 22000.00 [4] => 4 [product_id] => 4 [5] => 1 [rack_id] => 1 [6] => 4 [7] => Minuman Botol 600ml [name] => Rak ABC [8] => Aqua [brand] => Aqua [9] => liter [unit] => liter [10] => 1 [11] => Rak ABC [12] => Minuman Botol 600ml [product_name] => Minuman Botol 600ml [13] => Rak ABC [rack_name] => Rak ABC) [1] => Array ([0] => 15 [inventory_id] => 15 [1] => 2016-11-11 [expired_date] => 2016-11-11 [2] => 6 [stock] =>	Array ([0] => Array ([0] => 14 [inventory_id] => 14 [1] => 2016-09-30 [expired_date] => 2016-09-30 [2] => 0 [stock] => 0 [3] => 22000.00 [pricesale] => 22000.00 [4] => 4 [product_id] => 4 [5] => 1 [rack_id] => 1 [6] => 4 [7] => Minuman Botol 600ml [name] => Rak ABC [8] => Aqua [brand] => Aqua [9] => liter [unit] => liter [10] => 1 [11] => Rak ABC [12] => Minuman Botol 600ml [product_name] => Minuman Botol 600ml [13] => Rak ABC [rack_name] => Rak ABC	<i>True</i>

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
		=> 2016-11-11 [2] => 6 [stock] => 6 [3] => 81500.00 [pricesale] => 81500.00 [4] => 6 [product_id] => 6 [5] => 1 [rack_id] => 1 [6] => 6 [7] => Susu Bubuk 800gram [name] => Rak ABC [8] => Dancow [brand] => Dancow [9] => pack [unit] => pack [10] => 1 [11] => Rak ABC [12] => Susu Bubuk 800gram [product_name] => Susu Bubuk 800gram [13] => Rak ABC [rack_name] => Rak ABC) [2] => Array ([0] => 16 [inventory_id] => 16 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00 [pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak	6 [3] => 81500.00 [pricesale] => 81500.00 [4] => 6 [product_id] => 6 [5] => 1 [rack_id] => 1 [6] => 6 [7] => Susu Bubuk 800gram [name] => Rak ABC [8] => Dancow [brand] => Dancow [9] => pack [unit] => pack [10] => 1 [11] => Rak ABC [12] => Susu Bubuk 800gram [product_name] => Susu Bubuk 800gram [13] => Rak ABC [rack_name] => Rak ABC) [2] => Array ([0] => 16 [inventory_id] => 16 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00 [pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] =>) [1] => Array ([0] => 15 [inventory_id] => 15 [1] => 2016-11-11 [expired_date] => 2016- 11-11 [2] => 6 [stock] => 6 [3] => 81500.00 [pricesale] => 81500.00 [4] => 6 [product_id] => 6 [5] => 1 [rack_id] => 1 [6] => 6 [7] => Susu Bubuk 800gram [name] => Rak ABC [8] => Dancow [brand] => Dancow [9] => pack [unit] => pack [10] => 1 [11] => Rak ABC [12] => Susu Bubuk 800gram [product_name] => Susu Bubuk 800gram [13] => Rak ABC [rack_name] => Rak ABC) [2] => Array ([0] => 16 [inventory_id] => 16 [1] => 2016-11-18	

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
		ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC) [3] => Array ([0] => 17 [inventory_id] => 17 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00 [pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan	bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC) [3] => Array ([0] => 17 [inventory_id] => 17 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00 [pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC) [3] => Array ([0] => 17 [inventory_id] => 17 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00	[expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00 [pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC) [3] => Array ([0] => 17 [inventory_id] => 17 [1] => 2016-11-18 [expired_date] => 2016-11-18 [2] => 30 [stock] => 30 [3] => 2500.00	

<i>Test Case</i>	<i>Function</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Status</i>
		[12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC) [4] => Array ([0] => 18 [inventory_id] => 18 [1] => 2016-09-09 [expired_date] => 2016-09-09 [2] => 2 [stock] => 2 [3] => 125000.00 [pricesale] => 125000.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC))		[pricesale] => 2500.00 [4] => 7 [product_id] => 7 [5] => 1 [rack_id] => 1 [6] => 7 [7] => Mi Instan [name] => Rak ABC [8] => Indomie Ayam Bawang [brand] => Indomie Ayam Bawang [9] => bungkus [unit] => bungkus [10] => 1 [11] => Rak ABC [12] => Mi Instan [product_name] => Mi Instan [13] => Rak ABC [rack_name] => Rak ABC))	

