

ANALISIS DAN PERANCANGAN SISTEM INFORMASI
PENJUALAN, PEMBELIAN NON-KONSINYASI DAN
PEMBELIAN KONSINYASI PADA “TOKO BARBIE”

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh :
Marta Stevani Lawat
2014110003



JURUSAN SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2021

ANALISIS DAN PERANCANGAN SISTEM INFORMASI
PENJUALAN, PEMBELIAN NON-KONSINYASI DAN
PEMBELIAN KONSINYASI PADA “TOKO BARBIE”

Oleh :
Marta Stevani Lawat
2014110003

Bandung, 10 Desember 2020
Menyetujui,

Budi Maryanto, S.Kom., M.T.
Pembimbing

Kezia Stefani, S.T., M.Kom.
Ketua Jurusan

JURUSAN SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2021

ABSTRAK

Sistem informasi dan teknologi saat ini memegang peranan yang sangat penting baik dalam bidang pendidikan, jasa maupun bisnis. Begitupun bagi Toko Barbie sebuah toko kelontong, berdiri pada tahun 2003 yang kini sedang berkembang dibidang penjualan, pembelian konsinyasi yang mana merupakan sistem pembelian titip jual, dan pembelian non-konsinyasi. Namun sayangnya, saat ini Toko Barbie masih menggunakan sistem manual baik dalam pendataan maupun pembuatan laporannya. Sehingga, Toko Barbie membutuhkan waktu yang cukup lama dalam pembuatan laporan serta pencarian, pengelompokan dan pembuatan data.

Sehubungan dengan itu menanggapi masalah yang dihadapi oleh Toko Barbie dalam pendataan dan pembuatan laporan maka dapat disimpulkan bahwa Toko Barbie membutuhkan sistem informasi yang terkomputerisasi agar dapat membantu dalam menyelesaikan masalah yang sedang dihadapi saat ini. Untuk itu, dilakukanlah analisis dan merancang sebuah sistem informasi sederhana untuk penjualan, pembelian konsinyasi, dan pembelian non-konsinyasi sesuai dengan kebutuhan Toko Barbie juga sebagai syarat tugas akhir.

Metode pengembangan sistem yang digunakan dalam penyusunan tugas akhir ini adalah metode prototipe. Sedangkan untuk metode perancangan sistem informasi, menggunakan metode perancangan berorientasi objek, yang meliputi pembuatan *usecase diagram*, *activity diagram*, *class diagram*, *sequence diagram*, *component diagram*, dan *deployment diagram*. Perangkat lunak yang digunakan untuk membuat diagram-diagram UML adalah StarUML, serta Visual Studio 2012 untuk pembuatan program/aplikasi. Sedangkan untuk DBMS digunakan Microsoft SQL Server.

Perancangan serta pengadaan sistem baru pada Toko Barbie ini, dapat mempermudah dan mempercepat dalam pembuatan laporan dan meningkatkan keakuratan data juga mengurangi kesalahan serta kehilangan data yang selama ini sering dirasakan oleh Toko Barbie.

KATA PENGANTAR

Puji syukur dipanjatkan kepada Tuhan Yang Maha Esa atas segala berkat, kasih dan kemurahan-Nya sehingga akhirnya dapat menyelesaikan tugas akhir ini.

Terima kasih juga yang sebesar-besarnya kepada pihak-pihak yang telah membantu dalam menyelesaikan tugas akhir ini berupa bimbingan, doa, saran, dan dorongan. Terima kasih yang tulus disampaikan kepada:

1. Bapak Budi Maryanto, S.Kom., M.T. selaku dosen pembimbing yang telah bersedia dan senantiasa selalu memberikan perhatian serta arahan demi kelancaran tugas akhir ini
2. Toko Barbie yang telah mengizinkan untuk melakukan penelitian
3. Kepada Papa dan Agnes selaku keluarga yang tidak ada henti-hentinya memberikan dukungan, semangat, motivasi dan doa sehingga dapat menyelesaikan tugas akhir ini
4. Bapak Muji yang telah memberikan bantuan dalam menyelesaikan program untuk tugas akhir ini
5. Teman-teman yaitu; Ricky, ci Gina, Mely, Pinky, Leo, Chintia, dan Yonathan yang telah memberikan bantuan, dukungan serta semangat selama mengerjakan tugas akhir ini
6. Semua pihak yang tidak dapat disebutkan satu-persatu yang juga turut membantu dalam penyusunan tugas akhir ini

Menyadari akan adanya kekurangan pada tugas akhir ini untuk itu, kritik dan saran yang berguna dalam memperbaiki setiap kekurangan pada tugas akhir ini sangat diharapkan. Semoga skripsi ini dapat berguna bagi pembuat dan rekan-rekan pembaca pada umumnya.

Bandung, 10 Desember 2020

Penulis

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	xi
DAFTAR SIMBOL	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	2
1.3 Maksud dan Tujuan.....	2
1.4 Batasan Masalah	3
1.5 Metodologi Penelitian.....	3
1.6 Lokasi dan Waktu Penelitian.....	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Pembangunan Sistem Informasi	5
2.1.1 Pengertian Sistem	5
2.1.2 Pengertian Informasi	8
2.1.3 Pengertian Sistem Informasi	8
2.1.4 Komponen Sistem Informasi	9
2.1.5 Model Pengembangan Prototipe	10
2.2 Metode Pengembangan Berorientasi Objek	11

2.3	Tahapan Analisis dan Perancangan Berorientasi Objek	12
2.3.1	Analisis Sistem	12
2.3.2	Perancangan Sistem	13
2.3.3	Alat Bantu Pemodelan Sistem.....	13
2.4	Perangkat Lunak untuk Analisis dan Perancangan Sistem	16
2.5	Pengertian Penjualan dan Pembelian pada Toko Kelontong	17
2.5.1	Pengertian Toko Kelontong.....	17
2.5.2	Pengertian Penjualan	17
2.5.3	Pengertian Pembelian	18
2.5.4	Pengertian Konsinyasi.....	18
BAB III ANALISIS SISTEM INFORMASI		20
3.1	Gambaran Umum Perusahaan	20
3.1.1	Sejarah Perusahaan.....	20
3.1.2	Struktur Organisasi Perusahaan	20
3.1.3	Uraian Tugas	21
3.2	Analisis Prosedur Kerja.....	22
3.2.1	Deskripsi Prosedur Kerja.....	22
3.2.2	Business Use Case	24
3.3	Deskripsi Dokumen	24
3.4	Analisis dan Evaluasi Sistem	26
3.4.1	Tinjauan Sistem Komputer	26
3.4.2	Evaluasi Sistem	26

3.5	Analisis Kebutuhan Sistem	27
3.5.1	Kebutuhan Informasi	27
3.5.2	Kebutuhan Program Aplikasi	27
3.5.3	Kebutuhan Perangkat Keras	28
3.6	Pemodelan Berorientasi Objek	30
3.6.1	Sistem Use Case	30
3.6.2	Use Case Scenario.....	31
3.6.3	<i>Activity Diagram</i>	39
BAB IV PERANCANGAN SISTEM INFORMASI		47
4.1	Perancangan Berorientasi Obyek	47
4.1.1	Class Diagram	47
4.1.2	Sequence Diagram.....	48
4.1.3	Sistem Pengkodean	57
4.2	Perancangan Antar Muka	58
4.2.1	Struktur Menu	58
4.2.2	Tata Letak Layar.....	59
4.3	Perancangan Dokumen	65
4.4	Implementasi Sistem	67
4.4.2	<i>Component Diagram</i>	67
4.4.3	Persiapan Penerapan Sistem.....	68
4.4.4	<i>Deployment Diagram</i>	69
4.4.5	Perkiraan Kebutuhan Biaya.....	70

4.5	Pengujian Sistem	70
BAB V KESIMPULAN DAN SARAN		78
5. 1	Kesimpulan.....	78
5. 2	Saran	78
DAFTAR PUSTAKA.....		80
LAMPIRAN.....		82
LAMPIRAN 1 CONTOH DOKUMEN SISTEM LAMA		83
LAMPIRAN 2 CONTOH TAMPILAN PROGRAM		84
LAMPIRAN 3 LISTING PROGRAM.....		91
LAMPIRAN 4 RINCIAN KEBUTUHAN BIAYA.....		152
LAMPIRAN 5 MODEL KETERHUBUNGAN ANTAR TABEL		155

DAFTAR GAMBAR

Gambar 2.1 Pendekatan Prototipe	10
Gambar 3.1 Struktur organisasi Toko Barbie	20
Gambar 3.2 <i>Business Use Case</i>	24
Gambar 3.3 Topologi <i>Star</i>	29
Gambar 3.5 <i>Activity Diagram</i> Pembelian Non-Konsinyasi	39
Gambar 3.6 <i>Activity Diagram</i> Pembelian Konsinyasi	40
Gambar 3.7 <i>Activity Diagram</i> Kelola Data <i>Supplier</i>	41
Gambar 3.8 <i>Activity Diagram</i> Kelola Data Pelanggan	41
Gambar 3.9 <i>Activity Diagram</i> Kelola Data Barang	42
Gambar 3.10 <i>Activity Diagram</i> Penjualan	43
Gambar 3.11 <i>Activity Diagram</i> Cetak Laporan Pembelian Non-Konsinyasi	44
Gambar 3.12 <i>Activity Diagram</i> Cetak Laporan Pembelian Konsinyasi	44
Gambar 3.13 <i>Activity Diagram</i> Cetak Laporan Stok Konsinyasi	45
Gambar 3.14 <i>Activity Diagram</i> Cetak Laporan Stock Non-konsinyasi	45
Gambar 3.15 <i>Activity Diagram</i> Cetak Laporan Stock Penjualan	46
Gambar 4.1 <i>Class Diagram</i>	47
Gambar 4.2 <i>Sequence Diagram</i> Pembelian Non-Konsinyasi	48
Gambar 4.4 <i>Sequence Diagram</i> Kelola Data <i>Supplier</i>	50
Gambar 4.5 <i>Sequence Diagram</i> Kelola Data Pelanggan	51
Gambar 4.6 <i>Sequence Diagram</i> Kelola Data Barang	52
Gambar 4.7 <i>Sequence Diagram</i> Penjualan	53
Gambar 4.8 <i>Sequence Diagram</i> Cetak Laporan Pembelian Non-Konsinyasi	54
Gambar 4.9 <i>Sequence Diagram</i> Cetak Laporan Pembelian Konsinyasi	55
Gambar 4.10 <i>Sequence Diagram</i> Cetak Laporan Stok Konsinyasi	55
Gambar 4.11 <i>Sequence Diagram</i> Cetak Laporan Stok Non-Konsinyasi	56
Gambar 4.12 <i>Sequence Diagram</i> Cetak Laporan Penjualan	56
Gambar 4.13 Struktur Menu	58

Gambar 4.14 Tata Letak Form Login.....	59
Gambar 4.15 Tata Letak Master Data Barang	60
Gambar 4.16 Tata Letak Master Data <i>Supplier</i>	61
Gambar 4.17 Tata Letak Master Data Pelanggan	61
Gambar 4.18 Tata Letak Data User.....	62
Gambar 4.19 Tata Letak Pembelian NonKonsinyasi.....	62
Gambar 4.20 Tata Letak Pembelian Konsinyasi	63
Gambar 4.21 Tata Letak Retur Pembelian	64
Gambar 4.22 Tata Letak Penjualan.....	64
Gambar 4.23 Laporan Pembelian Konsinyasi	65
Gambar 4.24 Laporan Pembelian Nonkonsinyasi	65
Gambar 4.25 Laporan Penjualan.....	66
Gambar 4.26 Laporan Stok Barang Konsinyasi	66
Gambar 4.27 Laporan Stok Barang Nonkonsinyasi	67
Gambar 4.28 <i>Component Diagram</i>	68
Gambar 4.29 <i>Deployment Diagram</i>	69

DAFTAR TABEL

Tabel 3.1 Tabel Kebutuhan Informasi.....	27
Tabel 3.2 Tabel Kebutuhan Program Aplikasi	27
Tabel 3.3 Tabel Spesifikasi Komputer Server	29
Tabel 3.4 Tabel Spesifikasi Komputer Client.....	29
Tabel 3.5 Skenario Normal <i>Use Case</i> Pembelian Non-konsinyasi	31
Tabel 3.6 Skenario Alternatif <i>Use Case</i> Nama Barang Tidak Ada	31
Tabel 3.7 Skenario <i>Use Case</i> Pembelian Konsinyasi	32
Tabel 3.8 Skenario Alternatif <i>Use Case</i> Nama <i>Supplier</i> Tidak Ada	33
Tabel 3.9 Skenario <i>Use Case</i> Data <i>Supplier</i>	34
Tabel 3.10 Skenario <i>Use Case</i> Data Pelanggan	34
Tabel 3.11 Skenario <i>Use Case</i> Data Barang	35
Tabel 3.12 Skenario <i>Use Case</i> Kelola Data Penjualan	35
Tabel 3.13 Skenario Alternatif <i>Use Case</i> Kelola Data Penjualan.....	36
Tabel 3.14 Skenario <i>Use Case</i> Cetak Laporan Pembelian Non-Konsinyasi.....	37
Tabel 3.15 Skenario <i>Use Case</i> Cetak Laporan Pembelian Konsinyasi	37
Tabel 3.16 Skenario <i>Use Case</i> Cetak Laporan Stok Konsinyasi	38
Tabel 3.17 Skenario <i>Use Case</i> Cetak Laporan Stok Non-Konsinyasi	38
Tabel 3.18 Skenario <i>Use Case</i> Cetak Laporan Penjualan	39
Tabel 4.1 Rincian Perkiraan Kebutuhan Biaya Penerapan Sistem Baru	70
Tabel 4.2 Pengujian Proses Login	70
Tabel 4.3 Pengujian Proses Master Data Barang	71
Tabel 4.4 Pengujian Proses Master Data <i>Supplier</i>	71
Tabel 4.5 Pengujian Proses Master Data Pelanggan.....	72
Tabel 4.6 Pengujian Proses Master Data User	72
Tabel 4.7 Pengujian Proses Pembelian Nonkonsinyasi	73
Tabel 4.8 Pengujian Proses Pembelian Konsinyasi	74
Tabel 4.9 Pengujian Proses Retur Pembelian.....	74
Tabel 4.10 Pengujian Proses Penjualan.....	75

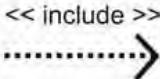
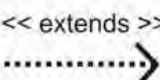
Tabel 4.11 Pengujian Proses Laporan Pembelian Nonkonsinyasi.....	75
Tabel 4.12 Pengujian Proses Laporan Pembelian Konsinyasi.....	76
Tabel 4.13 Pengujian Proses Laporan Penjualan	76
Tabel 4.14 Pengujian proses Laporan Pembelian Konsinyasi	77

DAFTAR LAMPIRAN

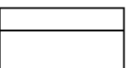
LAMPIRAN.....	82
LAMPIRAN 1 CONTOH DOKUMEN SISTEM LAMA	83
LAMPIRAN 2 CONTOH TAMPILAN PROGRAM	84
LAMPIRAN 3 LISTING PROGRAM.....	91
LAMPIRAN 4 RINCIAN KEBUTUHAN BIAYA.....	152
LAMPIRAN 5 MODEL KETERHUBUNGAN ANTAR TABEL	155

DAFTAR SIMBOL

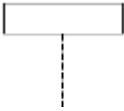
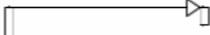
1. Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Association</i>	Garis yang menghubungkan antara objek satu dengan objek lainnya.
3		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

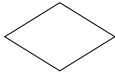
2. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		<i>Association</i>	Garis yang menghubungkan antara objek satu dengan objek lainnya.

3. Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

4. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
2		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
3		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
5		<i>Decision</i>	Suatu titik pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Diera globalisasi seperti saat ini teknologi sangat besar perannya bagi kehidupan manusia, bahkan keberadaannya tidak dapat dilepaskan dari kehidupan kita sehari-hari. Tidak dapat dipungkiri, bahwa kini teknologi telah menjadi kebutuhan primer bagi kita semua terutama teknologi informasi. Setiap aspek kehidupan manusia bergantung pada teknologi informasi. Dimulai dari memesan makanan, berkomunikasi, transportasi, hingga berbelanja semua dapat dilakukan dengan bantuan teknologi informasi.

Selain untuk menunjang kehidupan kita sehari-hari teknologi juga dapat membantu para pegiat bisnis dalam melancarkan, mempermudah serta mengontrol segala kegiatan bisnisnya. Namun sayangnya, beberapa pengusaha terutama para pengusaha kecil belum menyadari seberapa penting dan bergunanya teknologi informasi dalam membantu usahanya. Mereka lebih memilih untuk menggunakan sistem manual yang mana masih banyak kekurangannya. Hal ini disebabkan oleh kurangnya kesadaran para pengusaha kecil tentang teknologi informasi.

Dengan bantuan teknologi informasi berupa komputer pengusaha dapat mencatat, menyimpan, mengelola juga mengirim data. Dan hal tersebut dapat menekan resiko dari kesalahan manusia yang tidak jarang dapat mengakibatkan kerugian bagi pengusaha yang bersangkutan misalnya terjadinya penimbunan atau kehabisan stok barang.

Toko Barbie merupakan salah satu usaha kecil yang saat ini belum menggunakan teknologi informasi untuk menunjang usahanya dan jelas merasakan banyak kerugian kerennanya. Untuk itulah maka dirancang suatu sistem informasi yang dibutuhkan oleh Toko Barbie guna meminimalisir resiko yang dihasilkan dari sistem manual yang selama ini masih digunakan. Hasil penelitian dan analisis akan dituangkan dalam bentuk tugas akhir dengan judul **“ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN, PEMBELIAN NON-KONSINYASI DAN PEMBELIAN KONSINYASI PADA TOKO BARBIE”**

1.2 Identifikasi Masalah

Berdasarkan hasil penelitian dari studi lapangan, disimpulkanlah beberapa permasalahan yang terdapat di Toko Barbie, yaitu :

1. Terjadinya penimbunan atau kekurangan stok barang tertentu. Dimana penimbunan barang dapat mengurangi kualitas barang tertentu serta membuat perputaran uang untuk usaha jadi terhambat. Sedangkan, kekurangan stok barang membuat Toko tidak dapat memenuhi kebutuhan pelanggan yang mana jika dibiarkan cepat atau lambat Toko akan kehilangan pelanggannya karena kecewa.
2. Pencatatan data pembelian sering terjadi kekeliruan. Hal ini mengakibatkan Toko sering kali membeli barang yang tidak sesuai kebutuhan dan melakukan pengeluaran yang tidak perlu. Contohnya saat berbelanja karena sebuah kesalahan pencatatan, Toko jadi membeli barang yang sebenarnya tidak laku dijual.
3. Kesulitan atau bahkan terjadinya kehilangan berbagai data atau dokumen penting yang dibutuhkan oleh Toko karena penyimpanan yang masih berantakan.
4. Sistem yang masih manual membuat pengontrolan persediaan barang kurang efisien. Pengontrolan harus dilakukan secara fisik di gudang sehingga harus memakan waktu dan tenaga selain itu juga tidak menutup kemungkinan adanya kesalahan dalam pengontrolan.

1.3 Maksud dan Tujuan

Maksud dari dibuatnya tulisan ini adalah untuk memenuhi syarat utama kelulusan. Untuk itulah dilakukan penelitian dan perancangan guna melengkapi tulisan ini.

Tujuan dari penelitian ini sesungguhnya adalah untuk membantu pengusaha agar dapat mengontrol serta mempermudah pengelolaan data dalam usahanya. Hal itu dibantu dengan adanya sistem sederhana yang akan dirancangkan.

1.4 Batasan Masalah

Ruang lingkup permasalahan yang akan dibahas pada penelitian ini diberi batasan masalah. Adapun batasan masalah yang batasi, yaitu:

1. Membahas proses pembelian meliputi pembelian non-konsinyasi serta pembelian berupa konsinyasi.
2. Membahas proses penjualan secara langsung dengan pembayaran tunai.

1.5 Metodologi Penelitian

Untuk memperoleh informasi serta data guna melengkapi tulisan ini, digunakan metode penelitian sebagai berikut:

1. Studi Kepustakaan

Memperoleh dan mengumpulkan informasi dari berbagai sumber pustaka yang akan dijadikan referensi dalam penulisan.

2. Studi Lapangan

Mencari dan mendapatkan informasi dengan terjun langsung ke lapangan melalui 2 cara yaitu:

- a. Wawancara, pengumpulan informasi melalui tanya jawab secara langsung dengan pemilik toko dan beberapa orang yang memiliki pengetahuan tentang objek yang diteliti.
- b. Observasi, pengumpulan data yang diambil langsung dari toko yang menjadi tempat penelitian.

1.6 Lokasi dan Waktu Penelitian

Guna mendapatkan informasi serta melengkapi data-data untuk keperluan tugas akhir ini maka dilakukanlah penelitiannya di Toko Barbie di Jalan Nyomplong 29 Sukabumi, Jawa Barat, Indonesia yang dilakukan mulai bulan September 2018 hingga Januari 2019.

1.7 Sistematika Penulisan

Sistematika penulisan ini dibuat agar tulisan ini terarah dan sesuai dengan ketentuan yang berlaku. Sehingga diharapkan tujuan dari dibuatnya tulisan ini dapat tercapai. Adapun urutan penulisan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, waktu dan lokasi penelitian yang semuanya adalah dasar dalam pembuatan laporan tugas akhir yang dituangkan oleh penulis berdasarkan hasil penelitian dari objek yang diteliti oleh penulis.

BAB II LANDASAN TEORI

Pada bab ini dikemukakan tentang teori-teori serta pengertian mengenai sistem informasi, metode pengembangan yang digunakan, tahapan analisis dan perancangan sistem informasi.

BAB III ANALISIS SISTEM INFORMASI

Dalam bab ini diterangkan tentang gambaran umum mengenai perusahaan yang diamati, menganalisis prosedur kerja, menganalisis sistem informasi, serta mengevaluasi sistem.

BAB IV PERANCANGAN SISTEM INFORMASI

Bab ini berisi tentang metode perancangan sistem yang digunakan, implementasi sistem serta pengujian sistem.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan bagian penutup dari tugas akhir yang berisikan tentang kesimpulan dari seluruh isi laporan yang dibuat oleh penulis serta saran dari penulis.

BAB II

LANDASAN TEORI

2.1 Pembangunan Sistem Informasi

Pembangunan sistem informasi adalah sebuah usaha untuk mengubah sistem saat ini atau membangun sistem baru untuk memperbaiki bagian dari sistem sebelumnya agar dapat berjalan lebih baik, lebih efisien dan lebih mudah untuk digunakan.

2.1.1 Pengertian Sistem

Kata sistem memiliki berbagai pengertian, berikut ini adalah beberapa pengertian menurut para ahli:

Susanto pada buku *“Sistem Informasi Akuntansi”* mengatakan *“Sistem merupakan gabungan dua atau lebih subsistem baik itu fisik maupun bukan yang saling berkaitan untuk mencapai sasaran tertentu dengan bekerja sama.”* (Susanto, 2013:22)

Harry J. Rosenblatt mengatakan *“Sistem adalah set dari komponen-komponen yang berkaitan dan menghasilkan sesuatu.”* (2013:6)

Definisi sistem menurut McLeod, Jr dalam Prasjo (2011:152), *“Sistem ialah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan”*.

Berdasarkan dari pendapat para ahli di atas maka, dapat disimpulkan bahwa sistem merupakan kumpulan elemen yang terintegrasi, saling berhubungan dan bekerjasama untuk mencapai suatu tujuan tertentu.

2.1.1.1 Elemen Sistem

Menurut McLeod yang ditulis oleh Yakub (2012:3) bahwasannya semua sistem memiliki susunan dasar yang sama, tetapi tidak semua sistem memiliki komponen yang sama. Tanda dari elemen yang terdapat dalam sistem yaitu:

1. Tujuan

Tujuan merupakan motivasi yang mengarah pada sistem dan menjadikan sistem tetap terarah dan terkendali.

2. Masukan

Masukan adalah setiap yang masuk dalam sistem dan menjadi bahan untuk diproses, baik itu berwujud maupun yang tidak berwujud. Masukan berwujud sendiri adalah bahan mentah, sedangkan yang tidak berwujud merupakan informasi.

3. Proses

Proses adalah elemen yang bertugas untuk mengubah masukan menjadi keluaran yang memiliki nilai dan berguna.

4. Keluaran

Keluaran ialah hasil dari masukan yang telah diproses oleh sistem dan keluaran dapat menjadi masukan untuk subsistem lainnya.

5. Batasan

Batasan merupakan pemisah antara sistem dan daerah diluarnya. Selain itu juga sebagai batasan dari tujuan yang ingin dicapai. Batas sistem menentukan ruang lingkup, kemampuan sistem, atau konfigurasi.

6. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian diwujudkan dengan menggunakan *feed back*, sedangkan *feed back* ini digunakan untuk mengendalikan masukan maupun proses agar dapat mencapai tujuan.

7. Lingkungan

Lingkungan ialah segala hal yang berada diluar sistem.

2.1.1.2 Klasifikasi Sistem

Klasifikasi sistem dari beberapa sudut pandang menurut Rohmat Taufiq (2013:8) diantaranya :

1. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*)

Sistem abstrak adalah segala sesuatu yang tidak berbentuk, baik itu berupa hasil pemikiran atau ide-ide yang tidak memiliki fisik, misalnya sistem teologi. Sistem fisik merupakan sistem yang dapat disentuh dan berbentuk, misalnya sistem komputer.

2. Sistem Dapat Dipastikan dan Sistem Tidak Dapat Dipastikan

Sistem dapat dipastikan merupakan sistem yang sudah direncanakan baik dalam penginputan maupun *output*. Sistem tidak dapat dipastikan ialah sistem yang belum direncanakan *input-proses-outputnya*.

3. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*)

Sistem tertutup adalah sistem yang tidak terpengaruh dan terhubung dengan lingkungan diluar sistem. Sistem terbuka adalah sistem yang terhubung dan dipengaruhi oleh lingkungan luar sistem.

4. Sistem Manusia dan Sistem Mesin

Sistem manusia merupakan sistem yang dikerjakan oleh manusia contohnya pelaku transaksi jual beli. Sedangkan sistem mesin merupakan sistem yang proses kerjanya dilakukan oleh mesin, contohnya sistem motor, mesin industri dan lain-lain.

5. Sistem Sederhana dan Sistem Kompleks

Sistem sederhana merupakan sistem yang subsistem dan komponen tidak banyak atau sedikit. Sedangkan sistem kompleks merupakan sistem yang memiliki dua atau lebih sub sistem sehingga prosesnya yang sangat rumit dan panjang.

6. Sistem Bisa Beradaptasi dan Sistem Tidak Bisa Beradaptasi

Sistem yang bisa beradaptasi dengan lingkungan merupakan sistem yang mampu bertahan dan dapat mengikuti perubahan lingkungan tersebut. Sistem yang tidak bisa beradaptasi pada lingkungan adalah sistem yang tidak mampu bertahan jika terjadi perubahan pada lingkungan tersebut.

7. Sistem Alamiah dan Sistem Buatan

Sistem alamiah adalah sistem yang terjadi murni karena proses alam dan tidak ada campur tangan manusia, contohnya sistem perputaran bumi. Sistem buatan merupakan

sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine system*, misalnya sistem informasi komputer.

2.1.2 Pengertian Informasi

Pengertian informasi menurut Gordon B. Davis adalah sebagai berikut :

“Informasi merupakan sekumpulan data yang diolah agar menjadi sesuatu yang lebih berguna dan memiliki nilai bagi penerima informasi yang nantinya dapat menjadi pertimbangan dalam pengambilan keputusan saat ini atau yang akan datang. (Hutahaeen, 2015:9)

McLeod mengatakan bahwa *“Informasi ialah data yang diolah dan diproses agar menjadi bentuk yang lebih berguna dan memiliki arti bagi penerimanya” (Yakub,2012:1)*

Sutarman dalam buku yang berjudul *"Buku Pengantar Teknologi Informasi"* berpendapat bahwa *“Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.* (Sutarman,2012:14)

Berdasarkan berbagai pandangan para ahli diatas dapat disimpulkan bahwa informasi adalah sekumpulan data yang diolah sedemikian rupa hingga menjadi bentuk yang memiliki nilai dan berguna bagi penerimanya.

2.1.3 Pengertian Sistem Informasi

Sistem informasi menurut Tata Sutabri (2012:46), *“Sebuah sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang bersifat manajerial dengan kegiatan strategi untuk dapat menyediakan informasi kepada pihak tertentu dengan laporan yang diperlukan.”*

“Sistem informasi merupakan sistem yang dapat diartikan sebagai mengumpulkan, mengolah, menyimpan, menganalisa, dan menyebarkan luaskan informasi untuk maksud tertentu. Seperti sistem yang lainnya, sistem informasi terdiri atas masukan (data, instruksi) dan keluaran (laporan, kalkulasi).” Sutarman (2012:13).

Bersumber dari kedua ahli diatas maka dapat disimpulkan bahwa sistem informasi adalah suatu sistem dalam organisasi yang terdiri dari masukan dan keluaran guna mendukung fungsi operasi organisasi untuk mencapai tujuan tertentu.

2.1.4 Komponen Sistem Informasi

Tata Sutabri (2012:47) berpendapat bahwa:

“Sistem informasi terdiri dari beberapa komponen yang disebut sebagai blok bangunan (Building Block), yang terdiri dari masukan, model, keluaran, teknologi, basis data, dan kendali. Sebagai suatu sistem, keenam blok bangunan tersebut saling berinteraksi antara satu dengan yang lainnya membentuk sebuah kesatuan guna mencapai tujuan”.

Blok bangunan itu terdiri dari:

1. Komponen Masukan (*Input Block*)

Komponen masukan mewakili setiap data yang masuk dalam sistem informasi.

Masukan yang dimaksud merupakan segala metode dan media guna menangkap data yang telah diinput yang juga dapat berupa dokumen dasar.

2. Komponen Model (*Model Block*)

Komponen model terdiri atas kombinasi prosedur, logika dan model matematik yang selanjutnya akan memanipulasi data yang telah disimpan dibasis data dengan cara tertentu untuk mengeluarkan *output* yang diinginkan.

3. Komponen Keluaran (*Output Block*)

Produk yang dihasilkan sistem informasi merupakan keluaran berupa informasi serta dokumentasi yang berkualitas dan berguna bagi setiap tingkatan manajemen serta semua pengguna sistem.

4. Komponen Teknologi (*Technology Block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan serta mengakses data, menghasilkan juga mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Komponen teknologi terdiri dari teknisi (*Humanware* atau *Brainware*), perangkat lunak (*Software*) dan perangkat keras (*Hardware*).

5. Komponen Basis Data (*Database Block*)

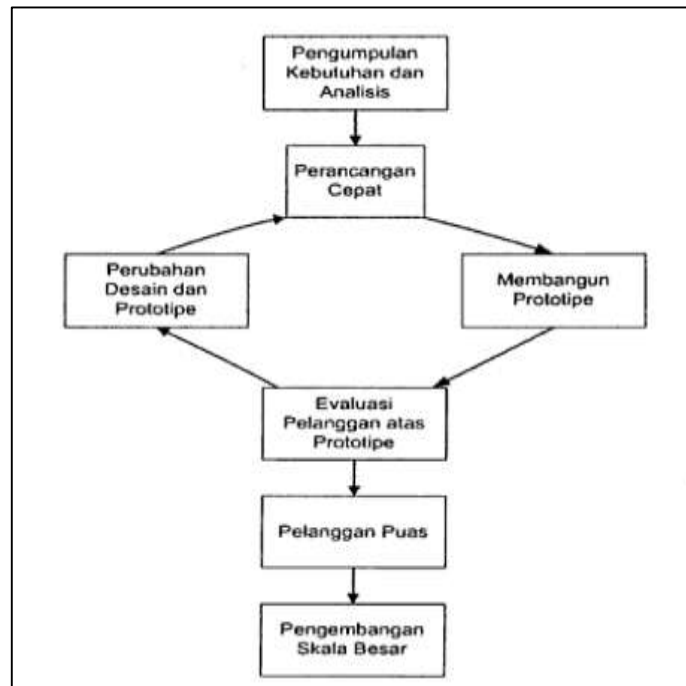
Basis data (*Database*) merupakan kumpulan dari data yang saling berkaitan antara satu dengan yang lainnya, tersimpan dalam *hardware* dan digunakan *software* untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan aplikasi *software* yang disebut dengan DBMS (*Database Management Systems*).

6. Komponen Kendali (*Controls Block*)

Beberapa pengendalian perlu dirancang dan diterapkan. Gunanya adalah sebagai tindakan preventif atas hal-hal yang dapat merusak sistem dan ataupun bila terlanjur terjadi kesalahan-kesalahan tersebut dapat segera diatasi.

2.1.5 Model Pengembangan Prototipe

Model pengembangan sistem yang akan digunakan adalah model pendekatan prototipe. Model pendekatan prototipe memungkinkan untuk menguji masukan, keluaran serta antarmuka pengguna sebelum keputusan akhir dibuat. Dengan menggunakan prototipe ini, konsumen dan tim pengembang dapat mengklarifikasi kebutuhan dan interpretasi mereka.



Gambar 2.1
Pendekatan Prototipe
Sumber : (Simarmata, 2010:63)

Langkah-langkah pendekatan prototipe :

1. Mengumpulkan dan menganalisis kebutuhan
2. Melakukan perancangan cepat
3. Membangun sebuah prototipe

4. Evaluasi dilakukan oleh konsumen atas prototipe
5. Perubahan rencana dan prototipe
6. Apabila pelanggan kecewa dengan prototipe yang telah dibuat, maka ulangi langkah ke 5
7. Apabila pelanggan merasa puas atas prototipe yang telah dibangun sebelumnya pengembangan produk berskala besar baru dapat dimulai

2.2 Metode Pengembangan Berorientasi Objek

Metode pengembangan yang digunakan dalam penelitian ini adalah metodologi pengembangan berorientasi objek yang memiliki arti sebagai berikut:

Menurut Rosa A.S. dan M. Shalahuddin dalam bukunya *“Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”* (2013:103);

Sistem berorientasi objek dibangun berdasarkan metode berorientasi objek. Metode berorientasi objek sendiri ialah sebuah sistem yang komponennya dienkapsulasi menjadi kelompok data dan fungsi. Setiap komponen dalam sistem dapat mewarisi sifat dari atribut dan komponen lainnya juga dapat berinteraksi satu sama lainnya.

Menurut Sutarman (2012:165), *“Pemograman Berorientasi Objek adalah suatu program komputer dipandang sebagai kumpulan dari unit tunggal atau objek yang dapat melakukan aksi atau tindakan satu sama lain.”*

Karakteristik Metodologi Berorientasi Objek - Terdapat 4 karakteristik utama dalam metodologi pengembangan sistem berorientasi objek , yakni :

1. Abstraksi

Prinsip objek untuk mempresentasikan kebutuhan sistem yang kompleks menjadi sebuah bentuk yang lebih sederhana.

2. Enkapsulasi

Setiap objek dibungkus (dikemas) untuk menyembunyikan implementasi dari objek sehingga objek lain tidak dapat mengetahui cara kerjanya.

3. Polimorfisme

Kemampuan suatu objek untuk dipakai dibanyak tujuan atau proses yang berbeda dengan nama yang sama sehingga menghemat baris program.

4. Inheritance

Mekanisme yang memungkinkan suatu objek atau kelas mewarisi sebagian atau seluruh definisi dari objek lain sebagai bagian dari dirinya.

2.3 Tahapan Analisis dan Perancangan Berorientasi Objek

2.3.1 Analisis Sistem

“Kegiatan analisis sistem merupakan kegiatan menganalisa sistem yang telah berjalan, serta menganalisa bagian mana saja yang bagus dan bagian mana yang tidak, lalu kebutuhan yang akan dipenuhi tersebut didokumentasikan pada sistem yang baru.”

(Rosa,2013:18)

Pendapat dari Rohmat Taufiq (2013:155) bahwa, *“Analisis sistem ialah suatu kegiatan dalam mempelajari sistem secara keseluruhan mulai dari menganalisis sistem, analisa masalah, design logic, dan memberikan keputusan atas hasil analisa tersebut”*.

Menurut Sutabri (2012:220), proses analisis sistem dalam pengembangan sistem informasi adalah suatu prosedur yang dilakukan sebagai pemeriksaan masalah dan penyusunan alternatif pemecahannya serta membuat spesifikasi sistem yang baru atau sistem yang akan diusulkan dan dimodifikasi. Adapun tujuan utama dari tahap analisis sistem ini yaitu:

1. Melayani kebutuhan informasi kepada fungsi manajerial dalam pengambilan keputasan dan pengendalian pelaksanaan usaha operasional perusahaan.
2. Memberikan bahan perbandingan bagi para pengambil keputusan sebagai pertimbangan dan tolak ukur terhadap hasil yang telah dicapainya. Mengevaluasi setiap sistem yang telah ada dan berjalan hingga saat ini, baik itu pengolahan data ataupun pembuatan laporannya.
3. Merumuskan setiap tujuan yang ingin diraih berupa pengolahan data dan pembuatan laporan yang baru.
4. Menyusun sebuah tahapan rencana pengembangan sistem serta penerapan juga perumusan langkahnya.

Berdasarkan pendapat para ahli di atas dapat disimpulkan bahwa analisis sistem adalah suatu kegiatan untuk melihat dan mempelajari sistem yang telah berjalan secara keseluruhan yang kemudian didokumentasikan serta hasilnya digunakan untuk mengambil keputusan.

2.3.2 Perancangan Sistem

Verzello/John Reuter III (2013:227) mengatakan bahwa, *“Perancangan Sistem merupakan sebuah tahapan setelah analisis dan merupakan pendefinisian dari kebutuhan fungsional serta persiapan untuk rancangan membangun implementasi yang juga menggambarkan bagaimana suatu sistem dibentuk.”*

Menurut Al-Jufri dalam bukunya yang berjudul *Sistem Infomasi Manajemen Pendidikan* (2011:141), *“Rancangan Sistem ialah penentuan atas proses dan data yang dibutuhkan oleh sistem baru.”*

Sugiyono (2013:18) mengatakan bahwa, *“Perancangan sistem merupakan sebuah kegiatan untuk membuat desain teknis berdasarkan evaluasi yang dilakukan pada waktu proses analisis.”*

Berdasarkan beberapa sumber di atas dapat disimpulkan bahwa perancangan sistem adalah tahapan setelah analisis sistem yang mana adalah sebuah kegiatan penentuan proses dan data yang diperlukan oleh sistem yang baru.

2.3.3 Alat Bantu Pemodelan Sistem

Pembuatan pemodelan sistem dalam penelitian ini menggunakan alat bantu Unified Modeling Language (UML). Berikut ini adalah beberapa pengertian UML menurut beberapa ahli:

Nugroho (2010:6) berpendapat bahwa, *“UML merupakan bahasa pemodelan yang berorientasi objek baik pada sistem ataupun perangkat lunak.”*

Berdasarkan buku *Analisis Serta perancangan Sistem Informasi Melalui Pendekatan UML* Widodo mengatakan bahwa (2011:6), *“Unified Modeling Languages yang merupakan bahasa pemodelan standar yang memiliki sintaks dan semantik.”* Yang mana elemen pada model-model yang telah dibuat berhubungan satu dengan yang lainnya itu harus mengikuti standar yang ada. UML bukan hanya sekedar

diagram tetapi juga menceritakan konteksnya. UML dapat diaplikasikan untuk tujuan tertentu, antara lain yaitu:

1. Merancang *software*.
2. Sarana komunikasi antara perangkat lunak.
3. Menjabarkan sistem dengan rinci untuk analisa dan mencari kebutuhan sistem.
4. Mendokumentasikan sistem yang ada baik proses dan organisasinya.

UML sendiri terdiri dari beberapa diagram, yaitu:

1. *Use Case Diagram*

Use Case Diagram adalah sebuah diagram yang merupakan ringkasan dari beberapa *use case* yang saling berhubungan dalam sistem atau sub sistem.

a. *Actor*

Merupakan siapa pun atau apa saja yang merupakan entitas dari luar yang berinteraksi dengan sistem. *Actor* dapat didefinisikan sebagai berikut :

- (1) *Actor* hanya memberikan informasi kepada sistem.
- (2) *Actor* hanya menerima informasi dari sistem.
- (3) *Actor* memberikan dan menerima informasi ke dan dari sistem.

b. *Use Case*

Merupakan fungsi sistem (proses otomatis atau manual). Setiap aktor harus terhubung dengan *use case* sementara beberapa *use case* mungkin tidak terhubung dengan aktor.

c. *Association / Directed Association*

Merupakan hubungan statis antar elemen. Umumnya asosiasi menggambarkan *class* yang memiliki atribut seperti *class* lain.

d. *Generalization / Pewarisan*

Pewarisan memungkinkan sebuah objek yang disebut anak untuk memperoleh satu atau lebih atribut dari objek lain yang disebut induk.

2. *Activity Diagram*

Activity diagram merupakan sebuah diagram yang menggambarkan rangkaian tindakan dan kejadian secara runtun sesuai dengan kejadian. Notasi-notasi yang terdapat dalam *activity diagram*, diantaranya yaitu :

a. *Activity*

Menggambarkan rangkaian tindakan pada *activity diagram*.

b. *Initial State*

Menggambarkan awal dari serangkaian tindakan atau aktivitas.

c. *Control Flow*

Menunjukkan urutan dalam pelaksanaan

d. *Final State*

Bagian akhir dari serangkaian tindakan atau aktivitas

3. *Sequence Diagram*

Sequence Diagram merupakan sebuah diagram interaksi yang menggambarkan secara detail bagaimana sebuah proses berjalan dan untuk menunjukkan rangkaian pesan dan kapan pesan tersebut dikirimkan. *Sequence diagram* dibuat berurutan berdasarkan urutan waktu. Singkatnya, *sequence diagram* merupakan gambaran langkah demi langkah yang semestinya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram. Notasi yang terdapat pada *sequence diagram*, yaitu :

a. *Actor*

Menggambarkan pengguna, perangkat keras eksternal ataupun subjek lain yang berinteraksi dengan subjek.

b. *Lifeline*

Mewakili individu yang berpartisipasi dalam interaksi

c. *Activations*

Persegi panjang tipis pada *lifeline* yang mewakili periode selama elemen melakukan operasi

d. *Call Message*

Mendefinisikan komunikasi tertentu antar *lifeline* dari sebuah interaksi.

4. *Class Diagram*

Class Diagram merupakan sebuah gambaran struktur dan deskripsi *class*, *package*, dan objek beserta hubungan antara satu dengan yang lainnya seperti *containment*, pewarisan, asosiasi, dan lainnya. Notasi pada *class diagram* diantaranya adalah:

a. *Class Name*

Nama *Class* yang muncul pada baris pertama gunanya sebagai pembeda antar *class*.

b. *Class Attributes*

Atribut ditampilkan pada baris kedua fungsinya untuk menjelaskan kualitas *class*.

c. *Methods*

Method ditampilkan pada baris terakhir menggambarkan bagaimana suatu kelas berinteraksi dengan data.

2.4 Perangkat Lunak untuk Analisis dan Perancangan Sistem

Beberapa perangkat lunak yang digunakan untuk perancangan sistem dalam penelitian ini adalah sebagai berikut:

1. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak dari Microsoft. Perangkat lunak ini umumnya digunakan untuk melakukan pengembangan berbagai aplikasi atau program komputer seperti *website* dan *mobile app*. Microsoft Visual Studio juga dapat digunakan sebagai pengembang aplikasi dalam *native code* atau *managed code*.

Visual Studio mendukung 36 bahasa pemrograman yang berbeda. Sedangkan untuk bahasa bawaannya termasuk C, C++, JavaScript, Visual Basic.NET dan lain-lain.

2. Microsoft SQL Server

Microsoft SQL Server merupakan sebuah sistem manajemen bisnis data relasional dari Microsoft. Sebagai *database* SQL Server memiliki fungsi utama untuk menyimpan dan mengambil data yang diminta oleh aplikasi perangkat lunak lain baik yang berjalan dalam komputer yang sama atau pun berbeda. Microsoft SQL juga mendukung

beragam tipe data termasuk tipe dasar seperti *Integer*, *Float*, *Decimal*, *Char*, *Varchar*, binari, *Text* dan yang lainnya.

3. Star UML

Star UML pada mulanya dikenal dengan sebutan "*Plastic*" dan "*Agora Plastic*". Star UML adalah *platform* pemodelan perangkat lunak yang mendukung diagram UML (*Unified Modeling Language*) seperti (Triandini, 2012:1-2):

- a. *Use Case Diagram*
- b. *Class Diagram*
- c. *Sequence Diagram*
- d. *Collaboration Diagram*
- e. *Statechart Diagram*
- f. *Activity Diagram*
- g. *Component Diagram*
- h. *Deployment Diagram*

2.5 Pengertian Penjualan dan Pembelian pada Toko Kelontong

2.5.1 Pengertian Toko Kelontong

Toko kelontong merupakan toko lokal sederhana yang normalnya mudah untuk diakses secara umum. Toko semacam ini umumnya berlokasi di jalan yang ramai, pemukiman penduduk, atau tempat perhentian transportasi umum. Umumnya, toko kelontong masih bersifat tradisional, dimana penjual melayani pembeli sehingga pembeli tidak dapat mengambil barang yang diinginkannya sendiri, dikarenakan tatanan rak toko yang belum modern sehingga menjadi pembatas antara penjual dan pembeli.

2.5.2 Pengertian Penjualan

Azhar Susanto, (2013:30) mengatakan bahwa "*Penjualan ialah sebuah kegiatan untuk mengembangkan rencana strategis yang diarahkan pada usaha untuk memuaskan kebutuhan serta keinginan pelanggan dengan tujuan mendapatkan laba.*" Sedangkan Hery (2011:123) mengatakan "*Penjualan adalah jumlah yang dibebankan kepada pembeli terhadap barang dagangan yang dijual baik itu penjualan secara tunai ataupun kredit*".

Berdasarkan pendapat kedua ahli tersebut maka dapat disimpulkan bahwa penjualan adalah suatu usaha terpadu guna mendapatkan laba yang dibebankan kepada pembeli atas terpenuhinya kebutuhan pembeli.

2.5.3 Pengertian Pembelian

Pembelian memiliki berbagai arti dan menurut (Susan Irawati, 2010:64) pembelian merupakan:

Suatu kegiatan untuk memperoleh sejumlah harta maupun jasa dari satu pihak untuk kelangsungan usaha atau kebutuhan yang mendasar, sehingga dilakukan pembayaran atas sejumlah uang atau jasa tersebut, untuk kelangsungan operasional perusahaan.

Pembelian merupakan sebuah proses bisnis yang dilakukan untuk mendapatkan barang atau jasa untuk mencapai tujuan. Meskipun ada beberapa pihak atau bagian yang mencoba menetapkan standar dalam proses pembelian, namun pembelian tetaplah sangat bervariasi. Contohnya dalam pembelian dapat terjadi tawar-menawar antara pembeli dan penjual hingga mendapatkan harga yang dikira sesuai oleh kedua belah pihak yang kemudian akan dilakukan transaksi penukaran barang atau jasa dengan alat tukar yang sah dan disepakati kedua belah pihak. Adapun manfaat dari pembelian adalah sebagai berikut:

1. Mendapatkan barang hasil pembelian.
2. Dapat memenuhi kebutuhan melalui pembelian.
3. Hasil Pembelian dapat bermanfaat.
4. Penjual mendapatkan hasil dari penjualannya.

2.5.4 Pengertian Konsinyasi

Menurut Yunus dan Harnanto (2013:141) "*Konsinyasi ialah suatu perjanjian dagang dimana pihak yang memiliki barang menyerahkan sejumlah barang sesuai dengan kesepakatan untuk dititipkan kepada pemilik usaha dengan memberikan komisi sebagai timbal baliknya*".

Karakteristik transaksi konsinyasi adalah sebagai berikut:

1. Hak milik barang yang dititipkan masih berada pada pihak penitip barang, maka barang-barang konsinyasi harus dilaporkan sebagai persediaan oleh pihak penitip barang. Dan

sebaliknya barang tersebut tidak boleh diperhitungkan sebagai persediaan oleh pihak penjual atau penerima barang.

2. Pihak yang menitipkan barang sebagai pemilik harus tetap bertanggung jawab sepenuhnya atas setiap biaya yang berhubungan dengan barang-barang konsinyasi sejak saat pengiriman sampai dengan saat penerima atau penjual berhasil menjualnya kepada pihak ketiga, kecuali terdapat ketentuan dan syarat lain dalam perjanjian diantara kedua belah pihak yang bersangkutan.

3. Pihak penerima barang juga memiliki kewajiban untuk menjaga keamanan dan kualitas dari barang-barang konsinyasi yang diterimanya guna keuntungan bersama.

BAB III

ANALISIS SISTEM INFORMASI

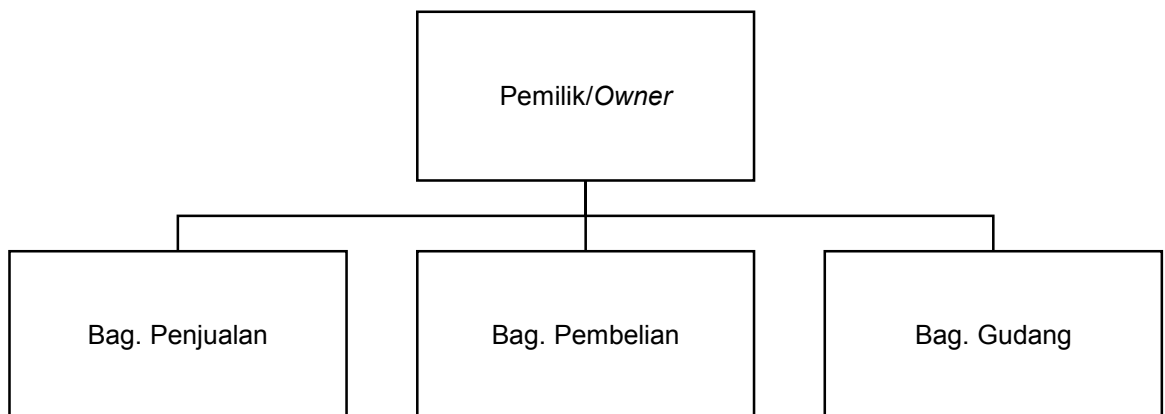
3.1 Gambaran Umum Perusahaan

3.1.1 Sejarah Perusahaan

Toko Barbie merupakan toko kelontong yang berdiri sejak tahun 2003 berlokasi di Jl. Nyomplong No 29. Nama Barbie sendiri diambil berdasarkan pendapat dari para pelanggan dikarenakan anak dari pemilik toko yang memiliki mata berwarna biru terang menyerupai boneka Barbie. Pada mulanya Toko Barbie merupakan toko kecil yang menjual barang dagangannya secara satuan. Namun, lambat laun Toko Barbie mulai mengembangkan usahanya dan mulai menjual barang dalam skala yang lebih besar yaitu secara grosir.

3.1.2 Struktur Organisasi Perusahaan

Struktur organisasi di Toko Barbie adalah sebagai berikut:



Gambar 3.1
Struktur organisasi Toko Barbie

3.1.3 Uraian Tugas

Uraian tugas menjelaskan tugas dari setiap anggota organisasi yang berbeda antara yang satu dengan yang lainnya. Berikut ini adalah uraian tugas dari Toko Barbie:

1. *Owner*
 - a. Menentukan dan mengambil keputusan pada toko
 - b. Mengangkat dan memberhentikan karyawan
 - c. Menentukan isi toko
2. Bag. Penjualan
 - a. Memberi informasi pada pelanggan ketika barang yang dipesan tidak tersedia
 - b. Melakukan promosi barang dagang pada pelanggan
 - c. Menghitung total harga belanjaan pelanggan
 - d. Mengeluarkan bon bila pelanggan memintanya atau penjualan dengan jumlah besar
 - e. Memastikan ketersediaan barang di *display*
 - f. Menghitung dan mencatat jumlah barang konsinyasi yang habis terjual
3. Bag. Pembelian
 - a. Melakukan pemesanan barang pada agen melalui telepon
 - b. Membeli barang dagangan langsung pada agen
 - c. Melakukan negosiasi harga
 - d. Melakukan pembayaran pada *supplier* konsinyasi
 - e. Meminta persetujuan *owner* untuk melakukan pembelian dan pembayaran barang
4. Bag. Gudang
 - a. Menghubungi bagian pembelian ketika ada barang yang kosong
 - b. Memasukkan barang yang baru dibeli oleh bagian pembelian ke dalam gudang
 - c. Mengatur barang yang ada di gudang
 - d. Menyediakan barang yang dibutuhkan bagian penjualan
 - e. Mengantarkan barang dagangan ke tempat pembeli berdasarkan permintaan

3.2 Analisis Prosedur Kerja

Dalam menjalankan aktivitas bisnisnya, terdapat beberapa prosedur kerja pada Toko Barbie. Prosedur kerja tersebut akan dijelaskan dalam bentuk deskripsi dan *business use case diagram* sebagai berikut:

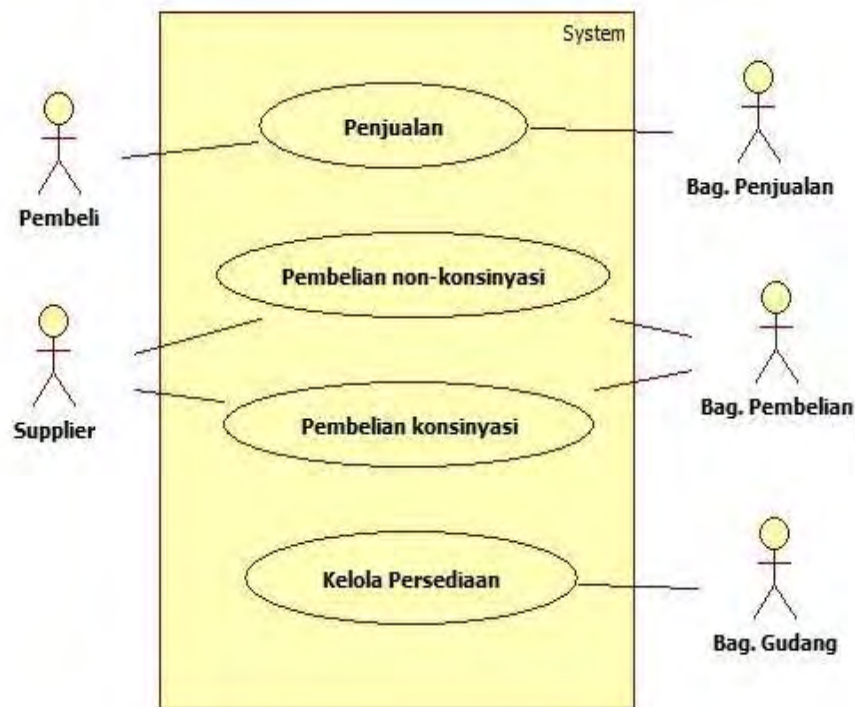
3.2.1 Deskripsi Prosedur Kerja

1. Prosedur Penjualan
 - a. Pelanggan datang ke toko dan memesan barang
 - b. Bagian penjualan menyediakan barang yang dipesan
 - c. Bagian penjualan menawarkan produk lain yang kira-kira akan dibutuhkan pelanggan
 - d. Bagian penjualan akan menghubungi bagian gudang ketika barang yang diminta pelanggan tidak ada atau telah kosong di *display*
 - e. Bagian penjualan akan mengeluarkan bon untuk pembelian jumlah besar atau sesuai dengan permintaan pelanggan
 - f. Pelanggan dan bagian penjualan melakukan transaksi pembayaran
 - g. Jika pembeli membeli barang berat atau melakukan pembelian dalam jumlah besar barang yang dibeli dapat diantar oleh bagian gudang berdasarkan permintaan pembeli
2. Prosedur pembelian non-konsinyasi
 - a. Bagian gudang menghubungi bagian pembelian ketika stok habis
 - b. Bagian pembelian akan melakukan pencatatan mengenai barang yang harus dibeli
 - c. Bagian pembelian akan meminta persetujuan dari owner untuk melakukan pembelian
 - d. Setelah mendapat persetujuan bagian pembelian akan memilah mana barang yang harus dibeli melalui pemesanan dan mana yang secara langsung
 - e. Bagian pembelian mendatangi langsung agen untuk melakukan transaksi jual beli

- f. Bagian pembelian menghubungi agen melalui telepon guna memesan barang yang harus melalui pemesanan
 - g. Bagian pembelian menegosiasikan harga bila ternyata harga tidak sesuai
3. Prosedur pembelian konsinyasi
- a. *Supplier* konsinyasi akan menitipkan barang dagangannya dan menentukan waktu pengambilan barang dagangannya
 - b. Bagian pembelian akan melakukan negosiasi harga bila harga tidak sesuai
 - c. Bagian pembelian mencatat jumlah dan tanggal pengambilan barang
 - d. Bagian penjualan akan menghitung berapa banyak jumlah barang yang laku terjual, ketika waktu yang ditentukan tiba
 - e. Bagian penjualan menginformasikan bagian pembelian jumlah barang yang habis terjual
 - f. Bagian pembelian akan menghitung total harga yang harus dibayarkan pada *supplier*
 - g. Bagian pembelian melakukan transaksi dengan *supplier*
4. Prosedur persediaan
- a. Bagian gudang akan menghitung persediaan barang yang ada di gudang setiap toko tutup
 - b. Bagian gudang akan mengajukan pembelian barang pada bagian pembelian jika ada barang yang habis
 - c. Bagian gudang akan mencatat setiap barang yang masuk
 - d. Bagian gudang akan menyediakan barang yang dibutuhkan oleh bagian penjualan
 - e. Bagian gudang akan mengantar barang yang diminta pelanggan jika diperlukan

3.2.2 Business Use Case

Business Use Case diagram Toko Barbie dapat dilihat pada Gambar 3.2 dimana bagian penjualan dan pembeli melakukan proses penjualan, bagian pembelian dan *supplier* melakukan proses pembelian non-konsinyasi dan konsinyasi, sedangkan bagian gudang melakukan proses kelola persediaan.



Gambar 3.2
Business Use Case

3.3 Deskripsi Dokumen

Deskripsi dokumen berfungsi untuk menjelaskan dokumen-dokumen yang digunakan oleh Toko Barbie dalam menjalankan seluruh proses bisnis pada toko tersebut.

- Nama dokumen : Bon penjualan

Fungsi : Mencatat barang dan total harga penjualan

Sumber : Bagian Penjualan

Frekuensi : Setiap ada penjualan dengan jumlah besar atau berdasarkan keinginan pembeli

- Rangkap : Satu (1)
- Distribusi : Pembeli
- Item-item data : Tanggal Transaksi, Nama Pembeli, Nama Barang, Jumlah Barang, Total Harga, Mengetahui
2. Nama dokumen : Bukti Konsinyasi
- Fungsi : Mencatat barang dan jumlahnya
- Sumber : Bagian Penjualan
- Frekuensi : Setiap *supplier* konsinyasi menitipkan dagangannya
- Rangkap : Satu (1)
- Distribusi : *Supplier* Konsinyasi
- Item-item data : Tanggal Penitipan, Nama Barang, Jumlah Barang, Harga Barang
3. Nama dokumen : Data Barang
- Fungsi : Mancatat barang yang harus dibeli bagian pembelian
- Sumber : Bagian Gudang
- Frekuensi : Setiap ada barang yang habis
- Rangkap : Satu (1)
- Distribusi : Bagian Pembelian
- Item-item data : Nama Barang, Jumlah Barang
4. Nama dokumen : Data *Supplier*
- Fungsi : Mancatat nama dan kontak *supplier*
- Sumber : *Supplier*
- Frekuensi : Setiap ada *supplier* atau tempat pemesanan barang yang baru
- Rangkap : Satu (1)
- Distribusi : Bagian Pembelian
- Item-item data : Nama *Supplier*, No Telepon, Alamat

3.4 Analisis dan Evaluasi Sistem

3.4.1 Tinjauan Sistem Komputer

Berdasarkan analisis dan evaluasi terhadap sistem lama, Toko Barbie masih menggunakan sistem manual dalam setiap prosedur-prosedurnya. Belum ada perangkat komputer di toko ini sehingga, diperlukan waktu yang relatif lebih lama dalam pendataan maupun dalam pengecekan dibandingkan dengan menggunakan sistem yang telah terkomputerisasi.

3.4.2 Evaluasi Sistem

Berdasarkan penelitian pada Toko Barbie ditemukan beberapa kelebihan dan kekurangan dari sistem yang telah berjalan, yaitu sebagai berikut:

1. Kelebihan:
 - a. Sistem manual lebih sederhana sehingga lebih mudah dimengerti oleh setiap kalangan
 - b. Tidak membutuhkan perangkat komputer sehingga lebih hemat biaya
2. Kekurangan:
 - a. Proses pencatatan dokumen dan pembuatan laporan akan lebih memakan waktu
 - b. Penggunaan sistem manual sangat boros kertas karena segala hal harus dicatat pada buku atau kertas
 - c. Sistem manual membutuhkan lebih banyak tempat dikarenakan banyaknya perlengkapan yang digunakan seperti kertas, map, dan alat tulis

3.5 Analisis Kebutuhan Sistem

3.5.1 Kebutuhan Informasi

Informasi yang dibutuhkan dalam sistem informasi penjualan, pembelian, dan pembelian konsinyasi pada Toko Barbie dengan sistem baru adalah sebagai berikut :

Tabel 3.1
Tabel Kebutuhan Informasi

No.	Informasi yang Dibutuhkan	Tujuan	Frekuensi
1.	Data master barang	Bagian penjualan, Bagian pembelian, Bagian gudang	Setiap kali ada barang baru
2.	Data master <i>supplier</i>	Bagian pembelian	Setiap kali ada <i>supplier</i> baru
3.	Data master pelanggan	Bagian penjualan	Setiap kali transaksi pertama bagi pelanggan
4.	Transaksi penjualan	Bagian penjualan	Setiap transaksi penjualan
5.	Transaksi pembelian non-konsinyasi	Bagian pembelian	Setiap transaksi pembelian barang non-konsinyasi
6.	Transaksi pembelian konsinyasi	Bagian pembelian	Setiap transaksi pembelian konsinyasi
7.	Stok konsinyasi	<i>Owner</i>	Setiap akhir bulan
8.	Stok Non-konsinyasi	<i>Owner</i>	Setiap akhir bulan

3.5.2 Kebutuhan Program Aplikasi

Program aplikasi yang dibutuhkan dalam sistem informasi penjualan, pembelian, dan pembelian konsinyasi pada Toko Barbie dengan sistem baru adalah sebagai berikut :

Tabel 3.2
Tabel Kebutuhan Program Aplikasi

No.	Deskripsi Kebutuhan	Keterangan
1.	Mengelola data pelanggan	Entri, tambah, hapus, cari, dan simpan
2.	Mengelola data <i>supplier</i>	Entri, tambah, hapus, cari, dan simpan
3.	Mengelola data barang	Entri, tambah, hapus, cari, dan simpan
4.	Mengelola data retur dan pembelian konsinyasi	Entri, tambah, dan simpan
5.	Mengelola data pembelian non-konsinyasi	Entri, tambah, cari, dan simpan
6.	Mengelola data penjualan	Entri, tambah, cari, dan simpan

No.	Deskripsi Kebutuhan	Keterangan
7.	Mengelola laporan penjualan	Lihat dan cetak
8.	Mengelola laporan pembelian konsinyasi	Lihat dan cetak
9.	Mengelola laporan pembelian non-konsinyasi	Lihat dan cetak
10.	Mengelola laporan stok konsinyasi	Lihat dan cetak
11.	Mengelola laporan stok non-konsinyasi	Lihat dan cetak

3.5.3 Kebutuhan Perangkat Keras

Dalam menunjang kinerja sistem yang baru, Toko Barbie membutuhkan beberapa perangkat keras. Beberapa perangkat keras yang dibutuhkan antara lain :

1. *Server*

Penghubung komputer *client* untuk saling berhubungan agar dapat melakukan sejumlah akses data. *Server* juga bertugas untuk mengatur lalu lintas data yang ada pada jaringan.

2. *Client/Workstation*

Komputer yang melakukan pengolahan data yang terdapat pada server.

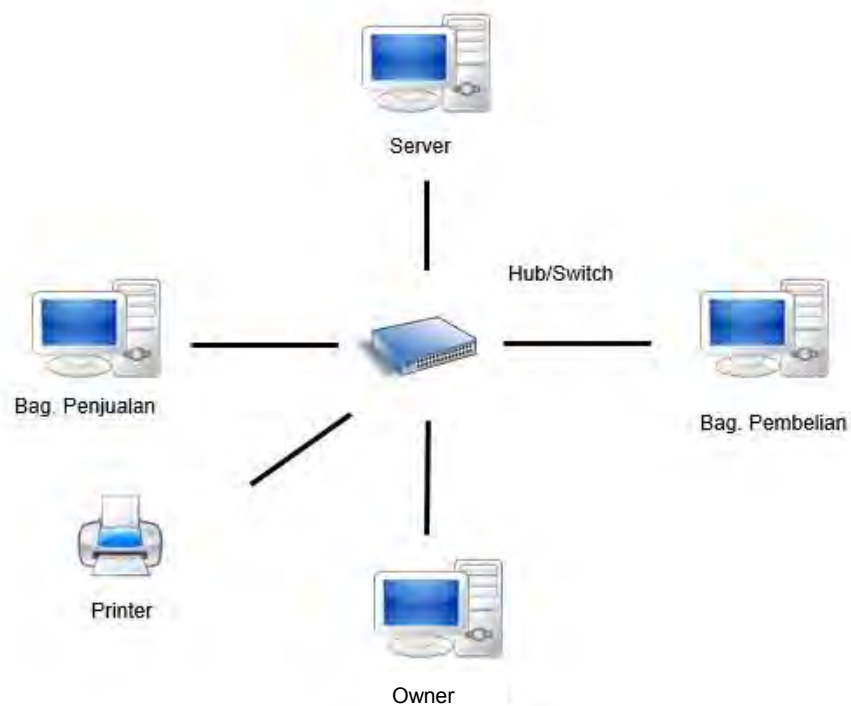
3. *Switch/Hub*

Perangkat yang menerima sinyal dan membagi sinyal tersebut ke beberapa komputer lain. *Switch/Hub* juga berfungsi sebagai penguat sinyal.

4. *Printer*

Perangkat untuk mencetak data dari komputer.

Topologi jaringan yang disarankan untuk mendukung sistem yang akan dibuat adalah topologi star. Topologi ini dipilih karena pemasangannya tidak sulit dan tidak memakan biaya yang besar. Selain itu, bila ada jaringan yang terganggu maka tidak akan mengganggu jaringan antar komputer yang lain.



Gambar 3.3
Topologi Star

Spesifikasi komputer yang diperlukan untuk sistem baru di Toko Barbie dijabarkan pada Tabel 3.3 dan Tabel 3.4:

Tabel 3.3
Tabel Spesifikasi Komputer Server

No.	Nama Perangkat	Spesifikasi
1.	<i>Processor</i>	Intel Core I3 8100 3.3 GHz
2.	<i>Memory</i>	4GB
4.	<i>Harddisk</i>	1TB SATA 2.5
5.	<i>Monitor</i>	LCD 15.6"
6.	<i>Operating System</i>	Windows server 2016

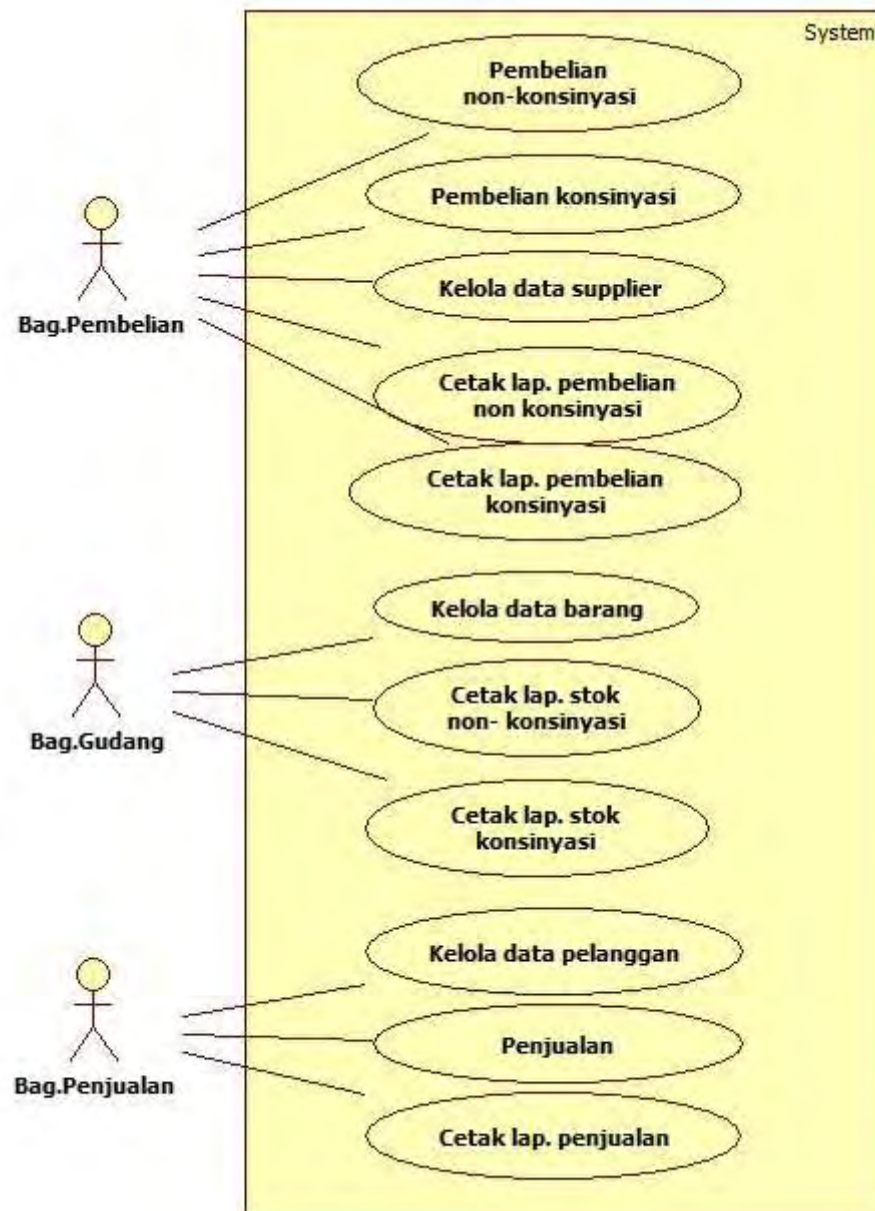
Tabel 3.4
Tabel Spesifikasi Komputer Client

No.	Nama Perangkat	Spesifikasi
1.	<i>Processor</i>	Intel Pentium G3220 3.0GHz
2.	<i>Memory</i>	2GB
3.	<i>Harddisk</i>	320GB
4.	<i>Monitor</i>	LCD 15.6"
5.	<i>Operating System</i>	Windows 10

3.6 Pemodelan Berorientasi Objek

3.6.1 Sistem Use Case

System Use Case merupakan diagram yang menggambarkan hubungan antar *use case* dan aktor di dalam sistem. Use case yang terdapat di dalam sistem penjualan, pembelian, pembelian konsinyasi dan stok digambarkan pada Gambar 3.4:



Gambar 3.4
Use Case Sistem

3.6.2 Use Case Scenario

3.6.2.1 Use Case Scenario Pembelian Non-Konsinyasi

Nama *use case* : Pembelian Non-Konsinyasi

Aktor : Bagian pembelian

Pre-condition : Sistem aktif dan berhasil login

Post-condition : Pembelian barang telah tersimpan

Tabel 3.5
Skenario Normal *Use Case* Pembelian Non-konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu pembelian	
2. Pilih sub menu non-konsinyasi	
	3. Menampilkan halaman pembelian non-konsinyasi
4. Pilih nama <i>supplier</i>	
5. <i>Input</i> nama barang	
	6. Menampilkan data barang (kode barang, nama barang, dan harga beli)
7. <i>Input</i> jumlah barang	
8. Klik button simpan	
	9. Menyimpan data pembelian non-konsinyasi

Skenario alternatif I : *Scenario Use Case* Nama Barang Tidak Ada

Tabel 3.6
Skenario Alternatif *Use Case* Nama Barang Tidak Ada

Aksi Aktor	Reaksi Sistem
Skenario Alternatif	
1. Pilih menu pembelian	
2. Pilih sub menu non-konsinyasi	
	3. Menampilkan halaman pembelian non-konsinyasi
4. Pilih nama <i>supplier</i>	
5. <i>Input</i> nama barang	
	6. Tidak menampilkan data barang
7. Pilih menu master	
8. Pilih sub menu data barang	
	9. Menampilkan halaman master data barang
10. <i>Input</i> data barang	

Aksi Aktor	Reaksi Sistem
Skenario Alternatif	
11. Klik button simpan	
	12. Menyimpan data barang
13. Pilih menu pembelian	
14. Pilih sub menu non-konsinyasi	
	15. Menampilkan halaman pembelian non-konsinyasi
16. Pilih nama <i>supplier</i>	
17. <i>Input</i> nama barang	
	18. Menampilkan nama barang
19. <i>Input</i> jumlah barang	
20. Klik button simpan	
	21. Menyimpan data pembelian ke database

3.6.2.2 Use Case Scenario Pembelian Konsinyasi

Nama *use case* : Pembelian Konsinyasi

Aktor : Bagian pembelian, bagian gudang

Pre-condition : Sistem aktif dan berhasil login

Post-condition : Pembelian barang telah tersimpan

Tabel 3.7
Skenario *Use Case* Pembelian Konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu pembelian	
2. Pilih sub menu konsinyasi	
	3. Menampilkan halaman pembelian konsinyasi
4. Pilih nama <i>supplier</i>	
5. <i>Input</i> tanggal titip dan tanggal kembali	
6. <i>Input</i> nama barang	
	7. Menampilkan nama barang
8. <i>Input</i> jumlah barang	
9. Klik button simpan	
	10. Menyimpan data pembelian ke database

Skenario alternatif I : *Scenario Use Case* Nama *Supplier* Tidak Ada

Tabel 3.8
Skenario Alternatif *Use Case* Nama *Supplier* Tidak Ada

Aksi Aktor	Reaksi Sistem
Skenario Alternatif	
1. Pilih menu pembelian	
2. Pilih sub menu konsinyasi	
	3. Menampilkan halaman pembelian non-konsinyasi
4. Pilih nama <i>supplier</i>	
	5. Tidak menampilkan nama <i>supplier</i>
7. Pilih menu master	
8. Pilih sub menu data <i>supplier</i>	
	9. Menampilkan halaman master data <i>supplier</i>
10. <i>Input</i> data <i>supplier</i>	
11. Klik button simpan	
	12. Menyimpan data <i>supplier</i>
13. Pilih menu pembelian	
14. Pilih sub menu konsinyasi	
	15. Menampilkan halaman pembelian non-konsinyasi
16. Pilih nama <i>supplier</i>	
17. <i>Input</i> nama barang	
	18. Menampilkan nama barang
19. <i>Input</i> jumlah barang	
20. Klik button simpan	
	21. Menyimpan data pembelian ke database

3.6.2.3 Use Case Scenario Kelola Data *Supplier*

Nama *use case* : Data *Supplier*

Aktor : Bagian pembelian

Pre-condition : Sistem aktif dan berhasil login

Post-condition : Data *supplier* telah tersimpan

Tabel 3.9
Skenario *Use Case Data Supplier*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu master	
2. Pilih submenu data <i>supplier</i>	
	3. Menampilkan halaman master data <i>supplier</i>
4. <i>Input data supplier</i>	
5. Klik tombol simpan	
	6. Menyimpan data <i>supplier</i>

3.6.2.4 *Use Case Scenario* Kelola Data Pelanggan

Nama *use case* : Kelola data pelanggan
Aktor : Bagian penjualan
Pre-condition : Sistem aktif dan berhasil login
Post-condition : Data pelanggan telah tersimpan

Tabel 3.10
Skenario *Use Case Data Pelanggan*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu master	
2. Pilih submenu data pelanggan	
	3. Menampilkan halaman master data pelanggan
4. <i>Input data pelanggan</i>	
5. Klik tombol simpan	
	6. Menyimpan data pelanggan

3.6.2.5 *Use Case Scenario* Kelola Data Barang

Nama *use case* : Kelola Data Barang
Aktor : Bagian gudang
Pre-condition : Sistem aktif dan berhasil login
Post-condition : Data barang telah tersimpan

Tabel 3.11
Skenario Use Case Data Barang

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu master	
2. Pilih submenu data barang	
	3. Menampilkan halaman master data barang
4. <i>Input</i> data barang	
5. Pilih status barang	
6. Klik tombol simpan	
	7. Menyimpan data barang

3.6.2.6 Use Case Scenario Penjualan

Nama *use case* : Penjualan

Aktor : Bagian penjualan, bagian gudang, owner

Pre-condition : Sistem aktif dan berhasil login

Post-condition : Data penjualan telah tersimpan

Tabel 3.12
Skenario Use Case Kelola Data Penjualan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu penjualan	
2. Pilih submenu sales order	
	3. Menampilkan halaman sales order
4. Pilih nama pelanggan	
5. <i>Input</i> nama barang	
	6. Menampilkan barang
7. <i>Input</i> jumlah barang	
8. <i>Input</i> jumlah uang yang dibayarkan oleh pelanggan	
	9. Menampilkan jumlah kembalian
10. Klik button simpan	
	11. Menyimpan data penjualan

Skenario alternatif I : *Scenario Use Case* nama pelanggan belum terdaftar pada sistem

Tabel 3.13
Skenario Alternatif *Use Case* Kelola Data Penjualan

Aksi Aktor	Reaksi Sistem
Skenario Akternatif	
1. Pilih menu penjualan	
2. Pilih submenu sales order	
	3. Menampilkan halaman sales order
4. Pilih nama pelanggan	
	5. Tidak menampilkan nama pelanggan
6. Pilih menu master	
7. Pilih menu data pelanggan	
	8. Menampilkan halaman data pelanggan
9. <i>Input</i> data pelanggan	
10. Klik button simpan	
	11. Data pelanggan tersimpan
12. Pilih menu penjualan	
13. Pilih submenu sales order	
	14. Menampilkan halaman sales order
15. Pilih nama pelanggan	
16. Cari nama barang	
	17. Menampilkan barang
18. <i>Input</i> jumlah barang	
19. <i>Input</i> jumlah uang yang dibayarkan oleh pelanggan	
	20. Menampilkan jumlah kembalian
21. Klik button simpan	
	22. Menyimpan data penjualan

3.6.2.7 Use Case Scenario Cetak Laporan Pembelian Non-Konsinyasi

Nama *use case* : Cetak Laporan Pembelian Non-Konsinyasi

Aktor : Bagian pembelian

Pre-condition : Aktor telah melakukan pembelian non-konsinyasi

Post-condition : Laporan pembelian non-konsinyasi dicetak

Tabel 3.14
Skenario *Use Case* Cetak Laporan Pembelian Non-Konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu laporan	
2. Pilih sub menu lap pembelian non-konsinyasi	
	3. Menampilkan halaman laporan non-konsinyasi
4. Memilih periode laporan	
5. Klik button tampilkan	
	6. Menampilkan laporan pembelian non-konsinyasi

3.6.2.8 *Use Case Scenario* Cetak Laporan Pembelian Konsinyasi

Nama *use case* : Cetak Laporan Pembelian Konsinyasi

Aktor : Bagian pembelian

Pre-condition : Aktor telah melakukan pembelian konsinyasi

Post-condition : Laporan pembelian konsinyasi dicetak

Tabel 3.15
Skenario *Use Case* Cetak Laporan Pembelian Konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu laporan	
2. Pilih submenu laporan pembelian konsinyasi	
	3. Menampilkan halaman laporan konsinyasi
4. Memilih periode laporan	
5. Klik button tampilkan	
	6. Menampilkan laporan pembelian konsinyasi

3.6.2.9 *Use Case Scenario* Cetak Laporan Stok Konsinyasi

Nama *use case* : Cetak Laporan Stok Konsinyasi

Aktor : Bagian gudang

Pre-condition : Bagian gudang telah mengupdate data barang terbaru

Post-condition : Laporan stok konsinyasi dicetak

Tabel 3.16
Skenario Use Case Cetak Laporan Stok Konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu laporan	
2. Pilih submenu laporan stock	
	3. Menampilkan halaman laporan stock
4. Pilih status konsinyasi	
5. Klik button tampilkan	
	6. Menampilkan laporan stock konsinyasi

3.6.2.10 Use Case Scenario Cetak Laporan Stok Non-Konsinyasi

Nama *use case* : Cetak Laporan Stok Non-Konsinyasi

Aktor : Bagian penjualan

Pre-condition : Bagian penjualan telah melakukan penjualan barang kepada beberapa pelanggan

Post-condition : Laporan stok non-konsinyasi dicetak

Tabel 3.17
Skenario Use Case Cetak Laporan Stok Non-Konsinyasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu laporan	
2. Pilih submenu laporan stock	
	3. Menampilkan halaman laporan stock
4. Pilih status non-konsinyasi	
5. Klik button tampilan	
	6. Menampilkan laporan stock non-konsinyasi

3.6.2.11 Use Case Scenario Cetak Laporan Penjualan

Nama *use case* : Cetak Laporan Penjualan

Aktor : Bagian penjualan

Pre-condition : Bagian penjualan telah melakukan penjualan barang

Post-condition : Laporan penjualan dicetak

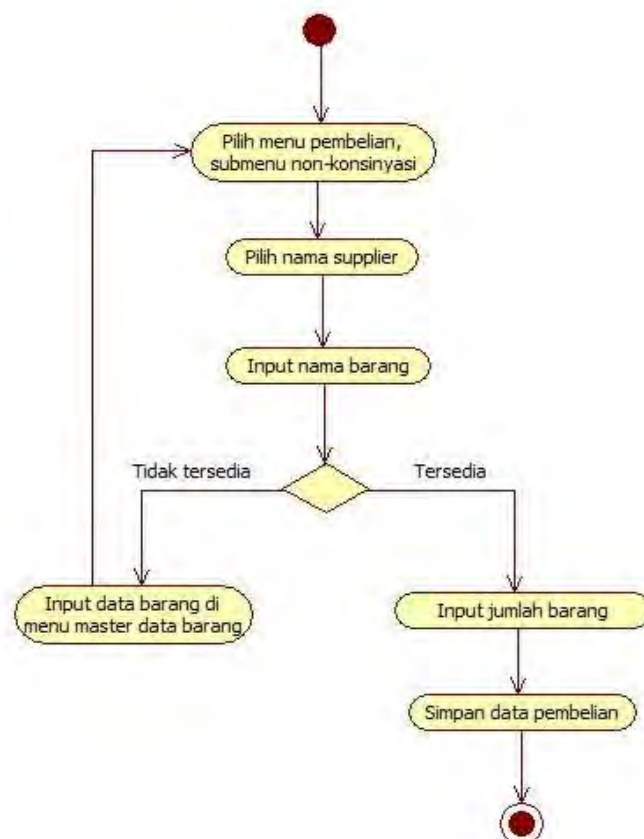
Tabel 3.18
Skenario Use Case Cetak Laporan Penjualan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih menu laporan	
2. Pilih submenu laporan penjualan	
	3. Menampilkan halaman laporan penjualan
4. Memilih periode laporan	
5. Klik button tampilkan	
	6. Menampilkan laporan penjualan

3.6.3 Activity Diagram

3.6.3.1 Activity Diagram Pembelian Non-Konsinyasi

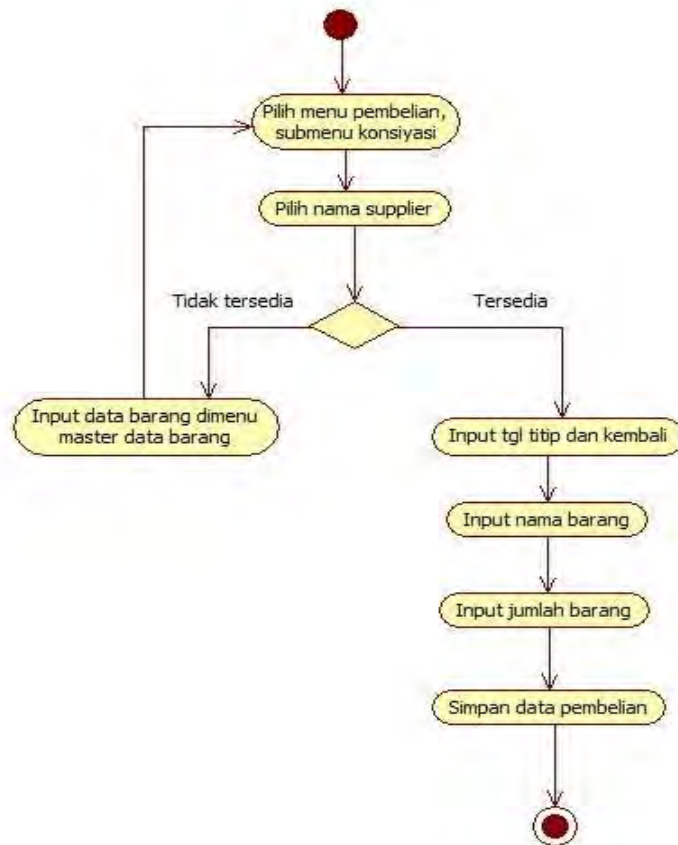
Activity Diagram untuk proses pembelian non-konsinyasi dapat dilihat pada Gambar 3.5:



Gambar 3.5
Activity Diagram Pembelian Non-Konsinyasi

3.6.3.2 Activity Diagram Pembelian Konsinyasi

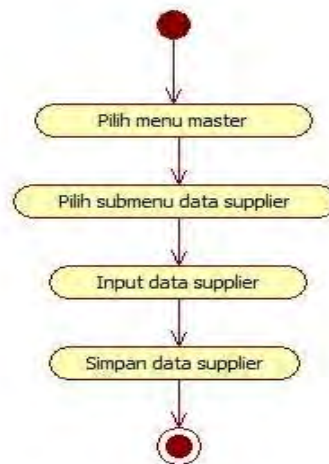
Activity Diagram untuk proses pembelian konsinyasi dapat dilihat pada Gambar 3.6:



Gambar 3.6
Activity Diagram Pembelian Konsinyasi

3.6.3.3 Activity Diagram Kelola Data *Supplier*

Activity Diagram untuk proses kelola data *supplier* dapat dilihat pada Gambar 3.7:



Gambar 3.7
Activity Diagram Kelola Data Supplier

3.6.3.4 Activity Diagram Kelola Data Pelanggan

Activity Diagram untuk proses kelola data pelanggan dapat dilihat pada Gambar 3.8:



Gambar 3.8
Activity Diagram Kelola Data Pelanggan

3.6.3.5 Activity Diagram Kelola Data Barang

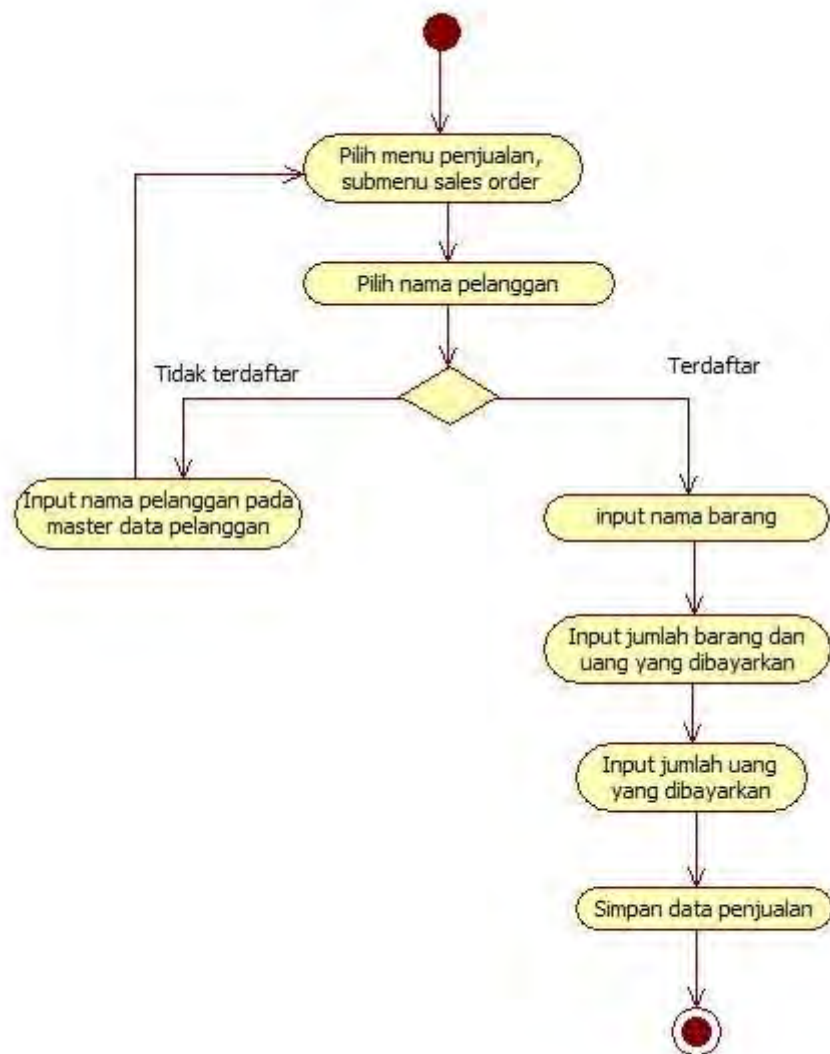
Activity Diagram untuk proses kelola data barang dapat dilihat pada Gambar 3.9:



Gambar 3.9
Activity Diagram Kelola Data Barang

3.6.3.6 Activity Diagram Penjualan

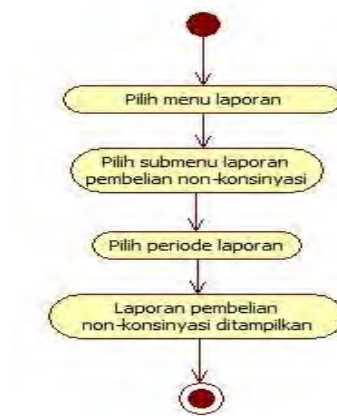
Activity Diagram untuk proses penjualan dapat dilihat pada Gambar 3.10:



Gambar 3.10
Activity Diagram Penjualan

3.6.3.7 Activity Diagram Cetak Laporan Pembelian Non-Konsinyasi

Activity Diagram untuk cetak laporan pembelian non-konsinyasi dapat dilihat pada Gambar 3.11:



Gambar 3.11
Activity Diagram Cetak Laporan Pembelian Non-Konsinyasi

3.6.3.8 Activity Diagram Cetak Laporan Pembelian Konsinyasi

Activity Diagram untuk cetak laporan pembelian konsinyasi dapat dilihat pada Gambar 3.12:



Gambar 3.12
Activity Diagram Cetak Laporan Pembelian Konsinyasi

3.6.3.9 Activity Diagram Cetak Laporan Stok Konsinyasi

Activity Diagram untuk cetak laporan stok konsinyasi dapat dilihat pada Gambar 3.13:



Gambar 3.13
Activity Diagram Cetak Laporan Stok Konsinyasi

3.6.3.10 Activity Diagram Cetak Laporan Stock Non-konsinyasi

Activity Diagram untuk cetak laporan stok non-konsinyasi dapat dilihat pada Gambar 3.14:



Gambar 3.14
Activity Diagram Cetak Laporan Stock Non-konsinyasi

3.6.3.11 Activity Diagram Cetak Laporan Penjualan

Activity Diagram untuk cetak laporan penjualan dapat dilihat pada Gambar 3.15:



Gambar 3.15
Activity Diagram Cetak Laporan Stock Penjualan

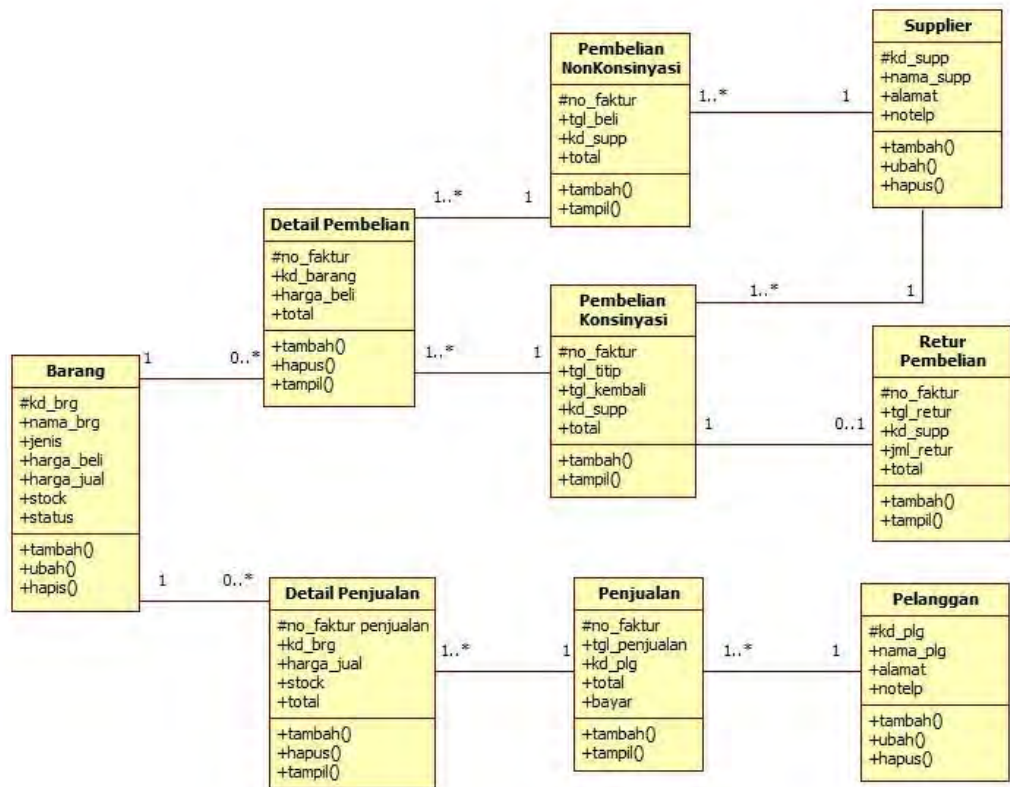
BAB IV

PERANCANGAN SISTEM INFORMASI

4.1 Perancangan Berorientasi Obyek

4.1.1 Class Diagram

Class diagram pada sistem yang baru akan tunjukkan pada gambar 4.1, yaitu sebagai berikut :

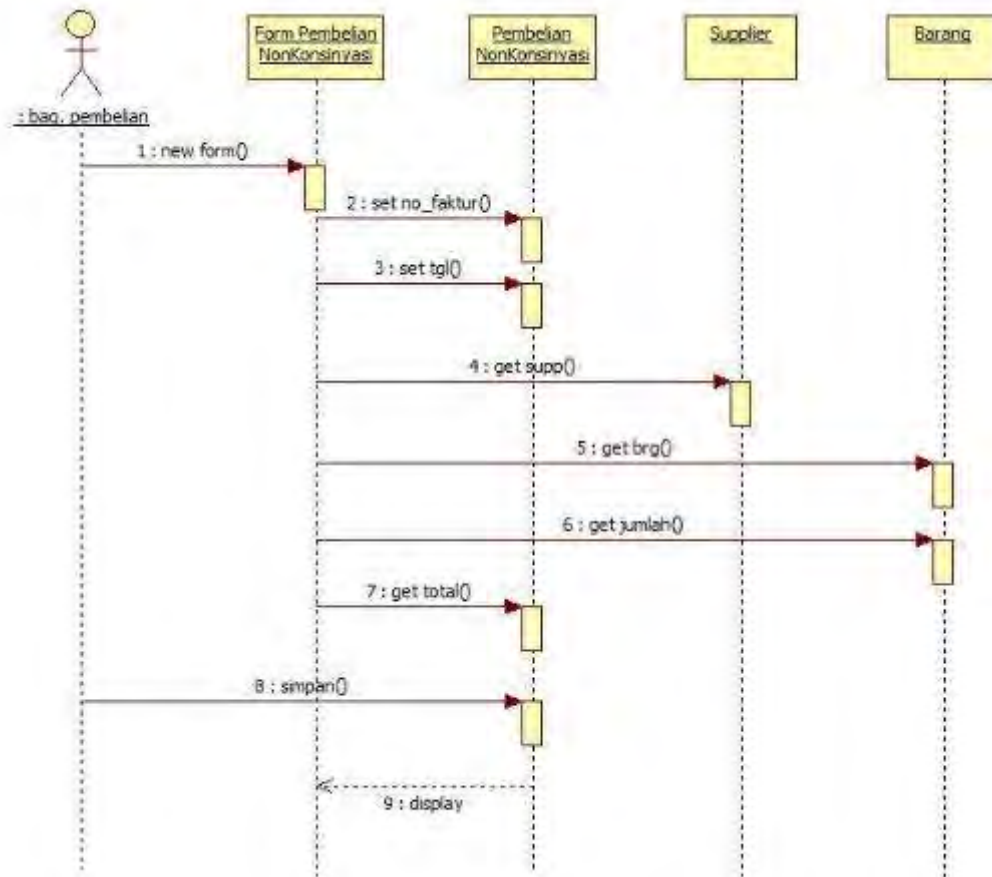


Gambar 4.1
Class Diagram

4.1.2 Sequence Diagram

4.1.2.1 Sequence Diagram Pembelian Non-Konsinyasi

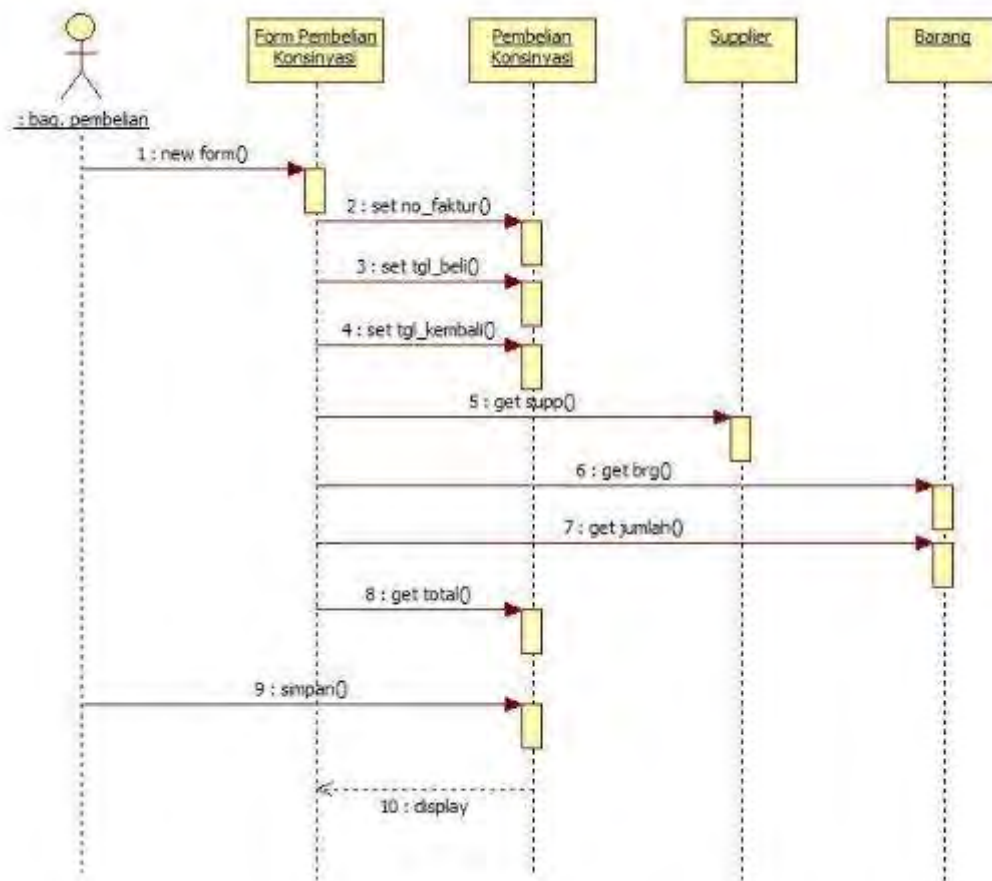
Sequence diagram Pembelian Non-Konsinyasi dapat dilihat pada gambar 4.2, dimana ini adalah proses pembelian non-konsinyasi yang dimasukkan kedalam sistem.



Gambar 4.2
Sequence Diagram Pembelian Non-Konsinyasi

4.1.2.2 Sequence Diagram Pembelian Konsinyasi

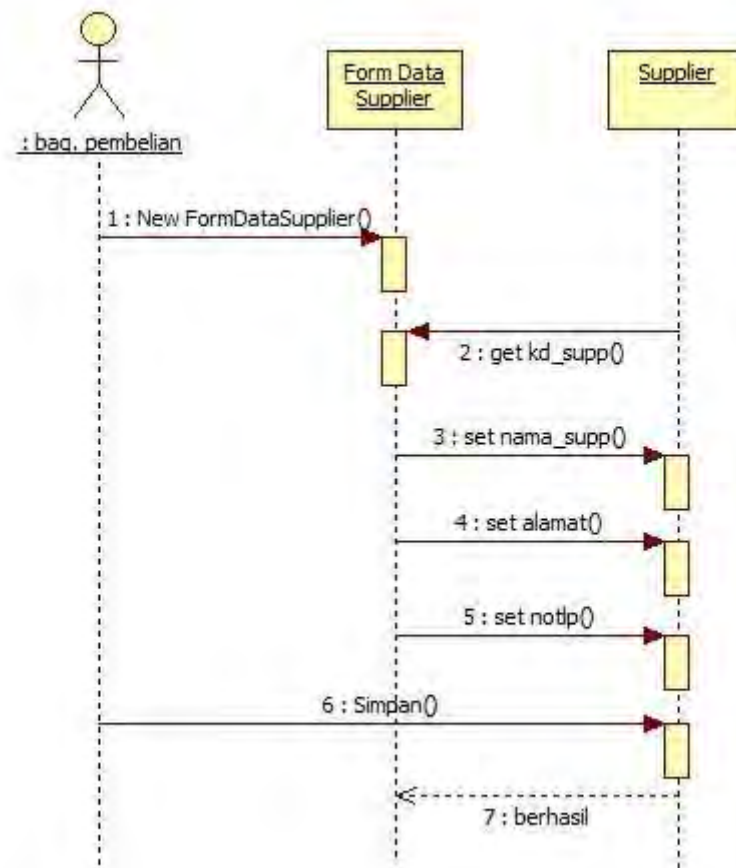
Gambar 4.3 merupakan *sequence diagram* pembelian konsinyasi pada sistem baru yang menggambarkan proses pembelian dengan sistem konsinyasi.



Gambar 4.3
Sequence Diagram Pembelian Konsinyasi

4.1.2.3 Sequence Diagram Kelola Data Supplier

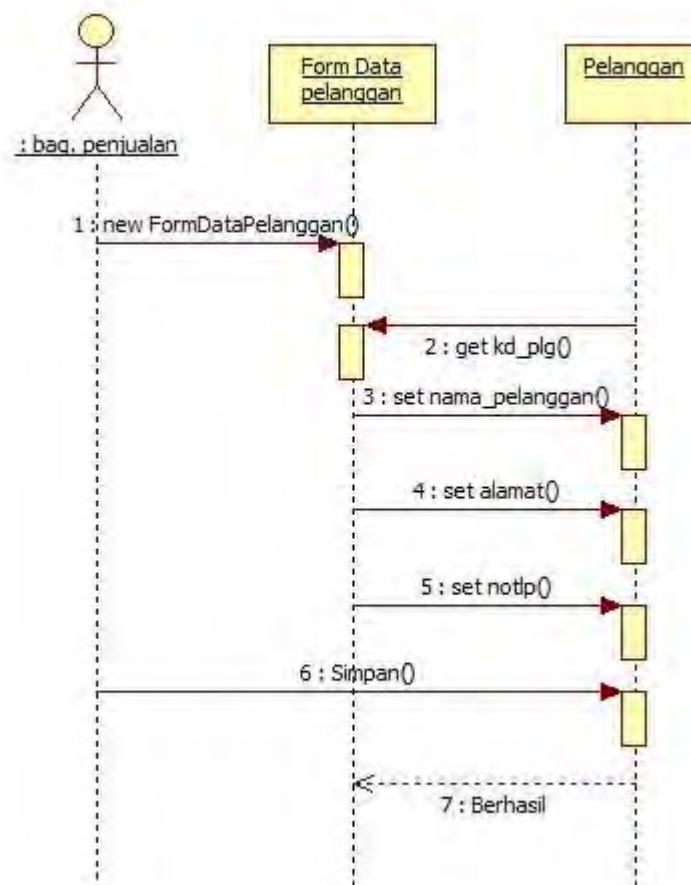
Gambar 4.4 merupakan *sequence diagram* kelola data *supplier* pada sistem baru yang menggambarkan proses peng-*input-an* data *supplier* sampai menyimpan data *supplier*.



Gambar 4.4
Sequence Diagram Kelola Data Supplier

4.1.2.4 Sequence Diagram Kelola Data Pelanggan

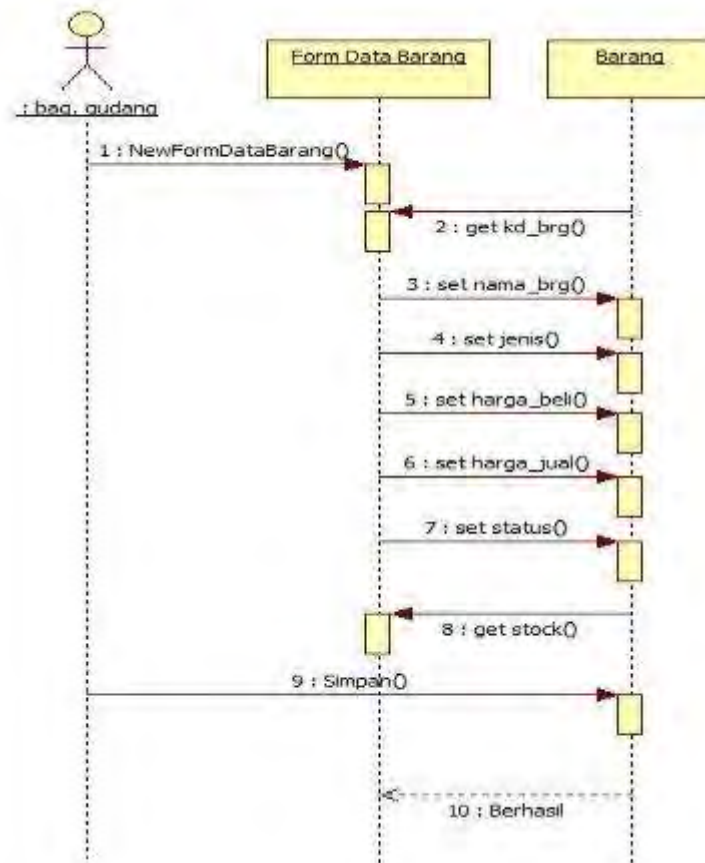
Gambar 4.5 merupakan *sequence diagram* kelola data pelanggan pada sistem baru yang menggambarkan proses peng-*input-an* data pelanggan sampai menyimpan data pelanggan.



Gambar 4.5
Sequence Diagram Kelola Data Pelanggan

4.1.2.5 Sequence Diagram Kelola Data Barang

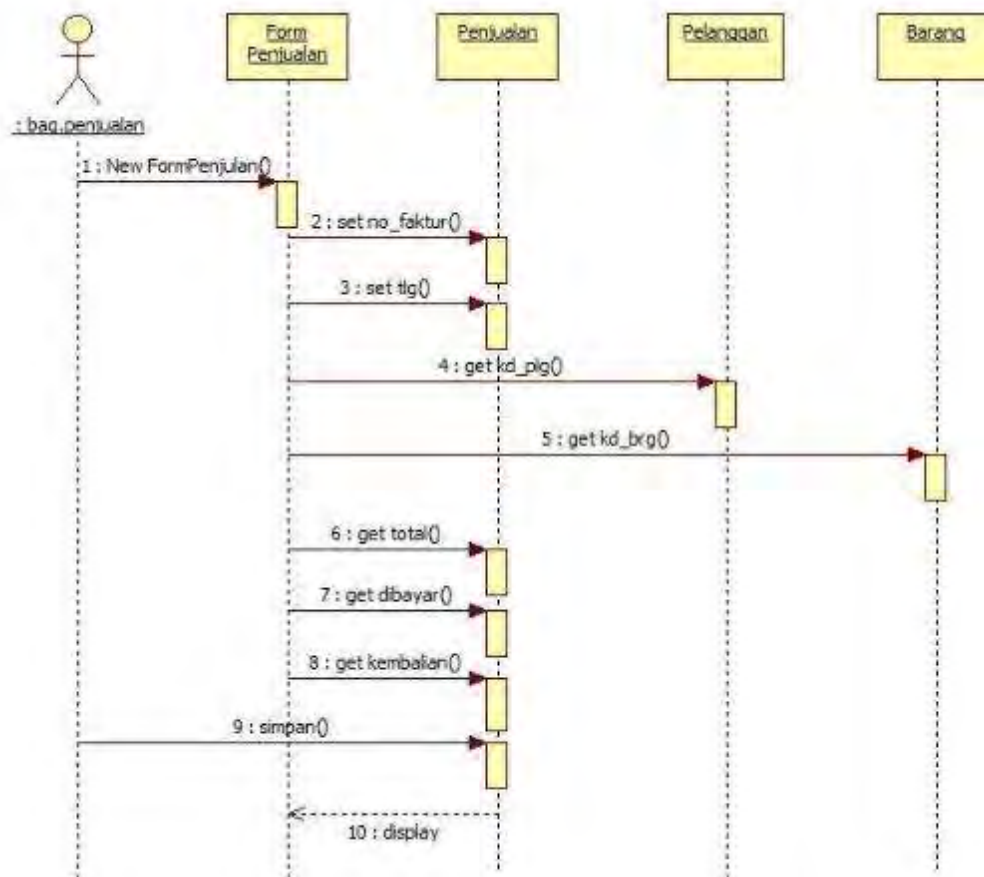
Gambar 4.6 merupakan *sequence diagram* kelola data barang pada sistem baru yang menggambarkan proses peng-*input-an* data barang sampai menyimpan data barang.



Gambar 4.6
Sequence Diagram Kelola Data Barang

4.1.2.6 Sequence Diagram Penjualan

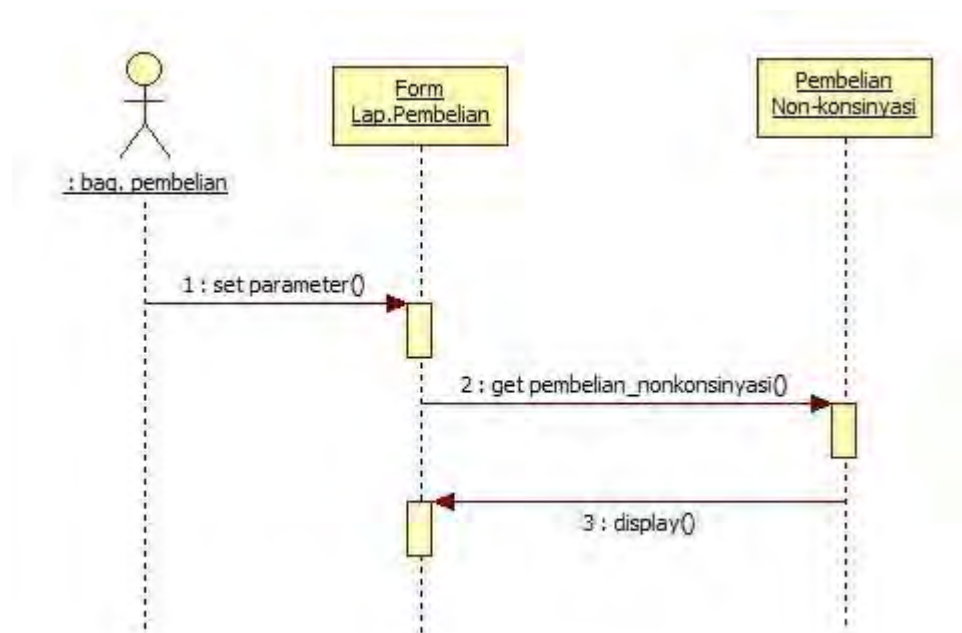
Gambar 4.7 merupakan *sequence diagram* penjualan pada sistem baru yang menggambarkan proses penjualan yang mana bagian penjualan menginputkan data-data penjualan.



Gambar 4.7
Sequence Diagram Penjualan

4.1.2.7 Sequence Diagram Cetak Laporan Pembelian Non-Konsinyasi

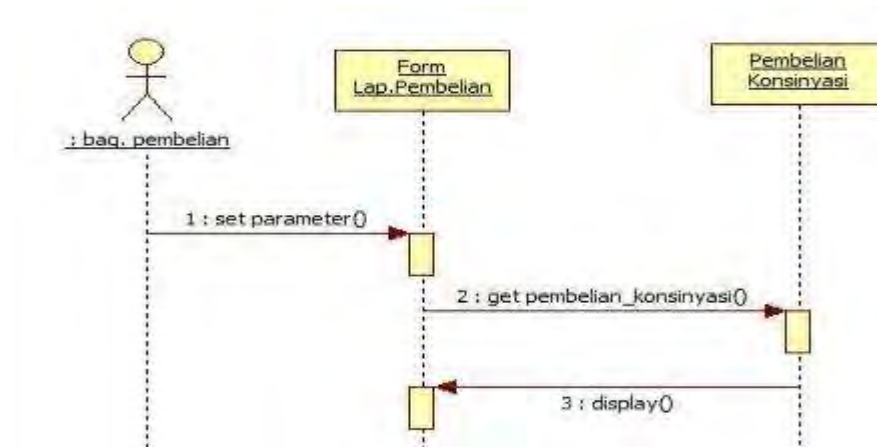
Gambar 4.8 merupakan *sequence diagram* cetak laporan pembelian non konsinyasi pada sistem baru yang menunjukkan laporan data pembelian nonkonsinyasi sesuai dengan periode yang ditentukan.



Gambar 4.8
Sequence Diagram Cetak Laporan Pembelian Non-Konsinyasi

4.1.2.8 Sequence Diagram Cetak Laporan Pembelian Konsinyasi

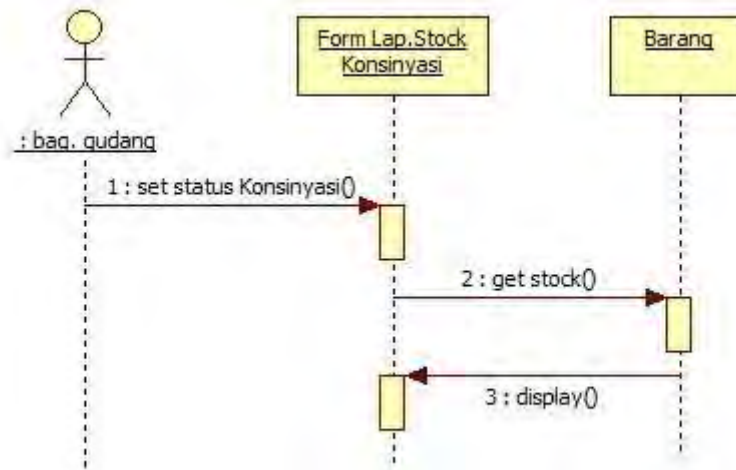
Gambar 4.9 merupakan *sequence diagram* cetak laporan pembelian konsinyasi pada sistem baru yang menunjukkan laporan data pembelian konsinyasi sesuai dengan periode yang ditentukan.



Gambar 4.9
Sequence Diagram Cetak Laporan Pembelian Konsinyasi

4.1.2.9 Sequence Diagram Cetak Laporan Stock Konsinyasi

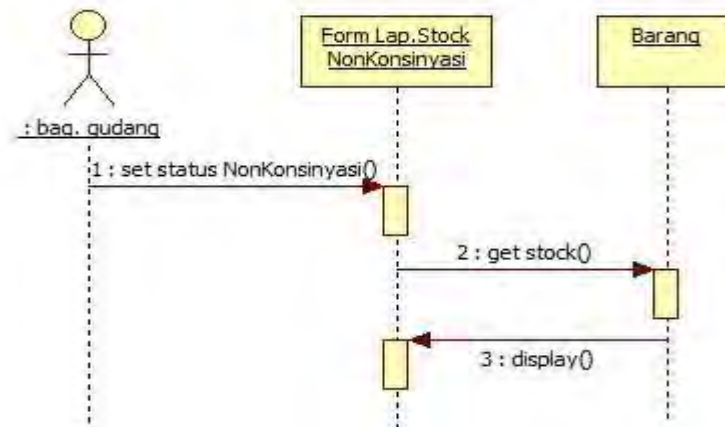
Gambar 4.10 merupakan *sequence diagram* cetak laporan pembelian konsinyasi pada sistem baru yang menunjukkan laporan data stock pembelian konsinyasi.



Gambar 4.10
Sequence Diagram Cetak Laporan Stok Konsinyasi

4.1.2.10 Sequence Diagram Cetak Laporan Stock Non-Konsinyasi

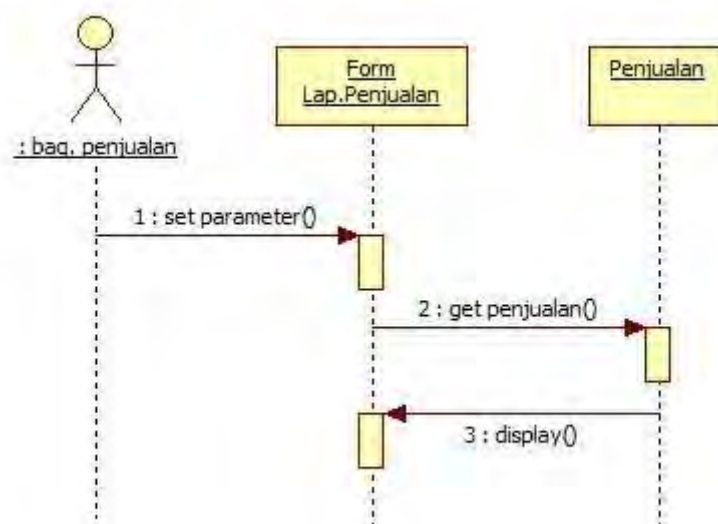
Gambar 4.11 merupakan *sequence diagram* cetak laporan pembelian konsinyasi pada sistem baru yang menunjukkan laporan data pembelian konsinyasi.



Gambar 4.11
Sequence Diagram Cetak Laporan Stok Non-Konsinyasi

4.1.2.11 Sequence Diagram Cetak Laporan Penjualan

Gambar 4.12 merupakan *sequence diagram* cetak laporan penjualan pada sistem baru yang menunjukkan laporan data penjualan pada periode yang telah ditentukan.



Gambar 4.12
Sequence Diagram Cetak Laporan Penjualan

4.1.3 Sistem Pengkodean

Pada sistem baru digunakan sistem pengkodean untuk memberikan keunikan pada masing-masing objek. Berikut ini dijelaskan rancangan sistem pengkodean, yaitu :

1. Kd_Supplier

Kd_Tamu terdiri dari 2 digit, dijelaskan sebagai berikut :

Contoh : 01

01 : Menunjukkan nomor urut

2. Kd_Pelanggan

Kd_Pelanggan terdiri dari 3 digit, dijelaskan sebagai berikut :

Contoh : 001

001 : Menunjukkan nomor urut

3. Kd_Barang

Kd_Barang terdiri dari 2 digit, dijelaskan sebagai berikut :

Contoh : 01

01 : Menunjukkan nomor urut

4. No_Pembelian Non-Konsinyasi

No_Pembelian Non-Konsinyasi terdiri dari 6 digit, dijelaskan sebagai berikut :

Contoh : 191101

19 : Menunjukkan tahun

11 : Menunjukkan bulan

01 : Menunjukkan nomor urut

5. No_Pembelian Konsinyasi

No_Pembelian Konsinyasi terdiri dari 6 digit, dijelaskan sebagai berikut :

Contoh : 191101

PK : Menunjukkan kode Pembelian Konsinyasi

19 : Menunjukkan tahun

11 : Menunjukkan bulan

01 : Menunjukkan nomor urut

6. No_Retur Pembelian Konsinyasi

No_Retur Pembelian Konsinyasi terdiri dari 6 digit, dijelaskan sebagai berikut :

Contoh : 191101

19 : Menunjukkan tahun

11 : Menunjukkan bulan

01 : Menunjukkan nomor urut

7. No_Penjualan

No_Penjualan terdiri dari 6 digit, dijelaskan sebagai berikut :

Contoh : 191101

19 : Menunjukkan tahun

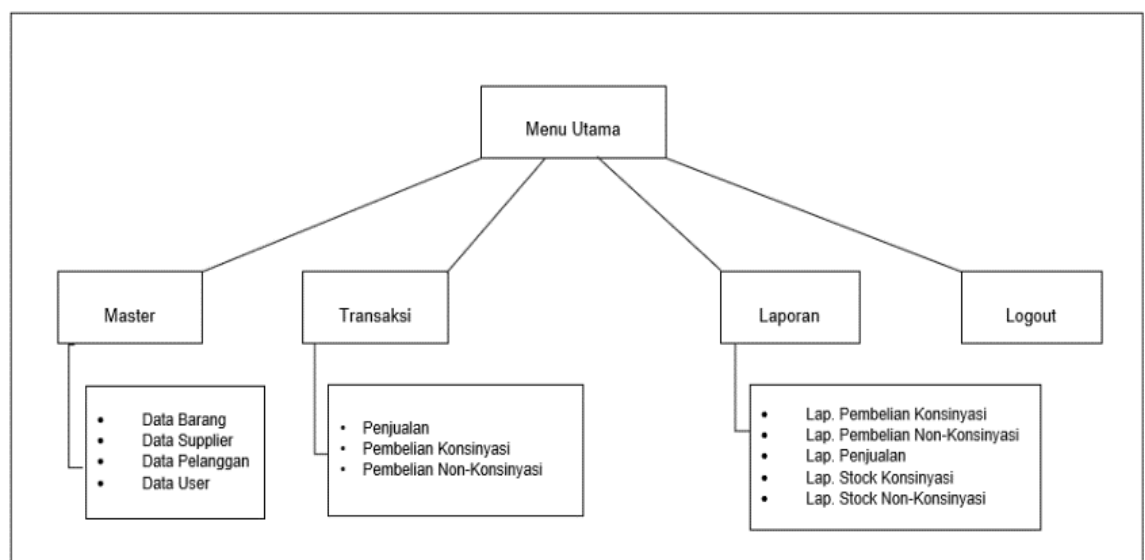
11 : Menunjukkan bulan

01 : Menunjukkan nomor urut

4.2 Perancangan Antar Muka

4.2.1 Struktur Menu

Struktur menu merupakan gambaran menu-menu serta submenunya yang terdapat pada sistem baru. Struktur menu tersebut digambarkan sebagai berikut :



Gambar 4.13
Struktur Menu

4.2.2 Tata Letak Layar

Tata letak layar merupakan tampilan pada *form-form* yang ada pada sistem baru.

Tata letak layar tersebut akan digambarkan sebagai berikut :

1. Tata Letak Form Login

Form login ditampilkan saat *user* menjalankan program. *User* akan diminta untuk menginputkan *username* dan *password* yang apabila *valid* akan langsung ditampilkan *form* menu utama.



The image shows a screenshot of a login window. The window has a title bar with the text 'Login' and a close button 'X'. Inside the window, there is a large rectangular box with the word 'LOGIN' in bold. Below this box, there are two input fields. The first is labeled 'User ID' and the second is labeled 'Password'. At the bottom of the window, there are two buttons: 'Login' and 'Batal'.

Gambar 4.14
Tata Letak Form Login

2. Tata Letak Master Data Barang

Form master data barang merupakan *form* yang digunakan untuk mengelola data barang. Mulai dari penambahan data, pengubahan data, penghapusan data, serta pencarian data. Tata letak layar data barang dapat dilihat pada gambar 4.15.

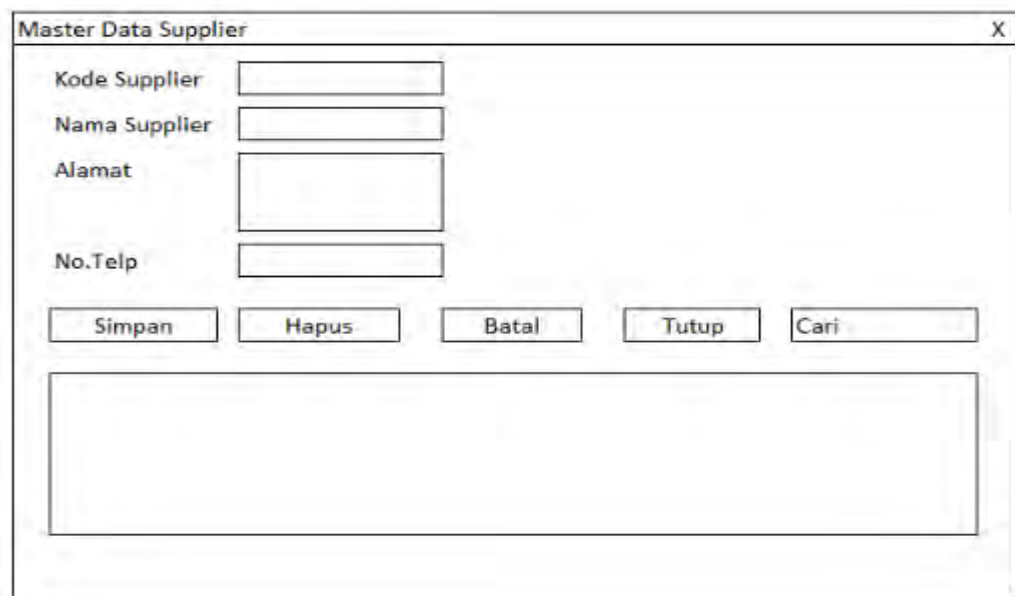
Master Data Barang X

Kode Barang				
Nama Barang				
Jenis	▼	Status	▼	
Harga Beli		Stock		
Harga Jual				

Gambar 4.15
Tata Letak Master Data Barang

3. Tata letak Master Data *Supplier*

Form master data *supplier* merupakan *form* yang digunakan untuk mengelola data *supplier*. Mulai dari penambahan data, pengubahan data, penghapusan data, serta pencarian data. Tata letak layar data tamu dapat dilihat pada gambar 4.16.

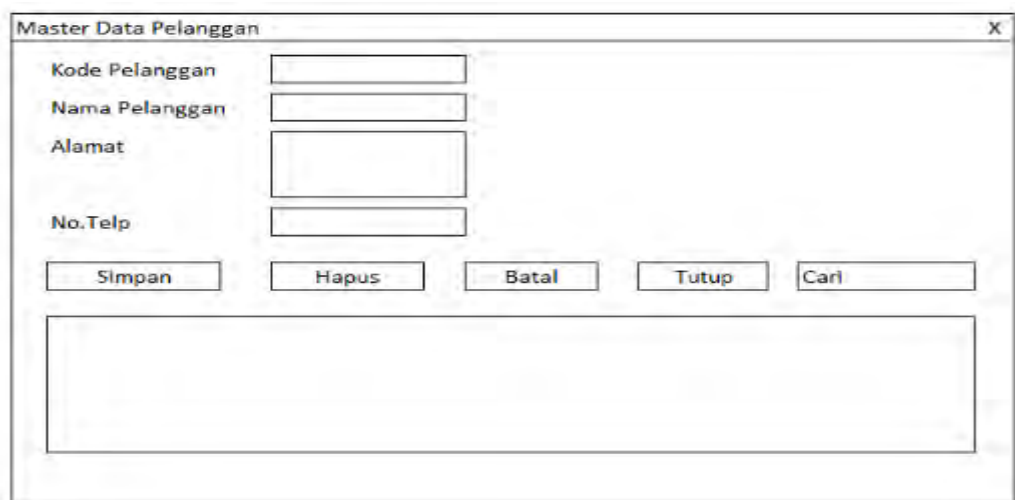


The screenshot shows a window titled "Master Data Supplier" with a close button (X) in the top right corner. The form contains four input fields: "Kode Supplier", "Nama Supplier", "Alamat", and "No.Telp". Below these fields is a row of five buttons: "Simpan", "Hapus", "Batal", "Tutup", and "Cari". At the bottom of the window is a large, empty rectangular box, likely for displaying a list of suppliers.

Gambar 4.16
Tata Letak Master Data Supplier

4. Tata Letak Master Data Pelanggan

Form master data pelanggan merupakan *form* yang digunakan untuk mengelola data pelanggan. Mulai dari penambahan data, pengubahan data, penghapusan data, serta pencarian data. Tata letak layar data tamu dapat dilihat pada gambar 4.17.



The screenshot shows a window titled "Master Data Pelanggan" with a close button (X) in the top right corner. The form contains four input fields: "Kode Pelanggan", "Nama Pelanggan", "Alamat", and "No.Telp". Below these fields is a row of five buttons: "Simpan", "Hapus", "Batal", "Tutup", and "Cari". At the bottom of the window is a large, empty rectangular box, likely for displaying a list of customers.

Gambar 4.17
Tata Letak Master Data Pelanggan

5. Tata Letak Data User

Form master data user merupakan *form* yang digunakan untuk membuat akun *user* baru yang nantinya akan digunakan untuk *login*. Tata letak layar dapat dilihat pada gambar 4.18.

The screenshot shows a window titled "Master Data User". Inside, there are three input fields labeled "User ID", "User Name", and "Password". Below these fields are five buttons: "Simpan", "Hapus", "Batal", "Tutup", and "Cari". At the bottom of the window is a large, empty rectangular box, likely for displaying a list of users.

Gambar 4.18
Tata Letak Data User

6. Tata Letak Pembelian NonKonsinyasi

Form pembelian nonkonsinyasi merupakan *form* yang digunakan oleh bagian pembelian untuk menginput data pembelian nonkonsinyasi. Tata letak form pembelian nonkonsinyasi dapat dilihat pada gambar 4.19.

The screenshot shows a window titled "Pembelian Non Konsinyasi". It contains several input fields: "No Faktur", "Supplier", "Tanggal", and "Kode Supplier". There is a section labeled "Cari Nama Barang" with a search input field. Below this is a table with 5 columns: "Kode Barang", "Nama Barang", "Harga Beli", "Jumlah", and "Total". At the bottom of the window are buttons for "Simpan", "Batal", and "Tutup", along with a "Total" label and an input field.

Gambar 4.19
Tata Letak Pembelian NonKonsinyasi

7. Tata Letak Pembelian Konsinyasi

Form pembelian konsinyasi merupakan *form* yang digunakan oleh bagian pembelian untuk menginput data pembelian konsinyasi. Tata letak form pembelian konsinyasi dapat dilihat pada gambar 4.20.

The form is titled "Pembelian Konsinyasi" and contains the following elements:

- Input fields for "No Faktur", "Supplier", "Tanggal Titip", "Kode Supplier", and "Tanggal Kembali".
- A search field labeled "Cari Nama Barang".
- Two tables for item details:

Kode Barang	Nama Barang	Harga Beli	Jumlah	Total

Kode Barang	Nama Barang
- A summary section at the bottom with buttons "Simpan", "Batal", and "Tutup", and a "Total" field.

Gambar 4.20
Tata Letak Pembelian Konsinyasi

8. Tata Letak Retur Pembelian

Form retur pembelian merupakan *form* yang digunakan oleh bagian pembelian untuk menginput data retur pembelian. Tata letak form retur pembelian dapat dilihat pada gambar 4.21.

Retur Pembelian Konsinyasi

No Faktur Faktur Pembelian Cari

Tanggal

Supplier

Daftar Pembelian Barang

Kode Barang	Nama Barang	Harga Beli	Jumlah	Total

Tambahkan ke Retur Hapus

Daftar Retur Pembelian

Kode Barang	Nama Barang	Harga Beli	Jumlah Retur	Total

Jumlah Barang

Simpan Batal Tutup Total

Gambar 4.21
Tata Letak Retur Pembelian

9. Tata Letak Penjualan

Form penjualan merupakan *form* yang digunakan oleh bagian penjualan untuk menginput data penjualan. Tata letak form penjualan dapat dilihat pada gambar 4.22.

Penjualan

No Faktur Pelanggan Cari Nama Barang

Tanggal Kode Pelanggan

Kode Barang	Nama Barang	Harga Beli	Jumlah	Total

Kode Barang	Nama Barang

Total

Dibayar

Kembali

Simpan Batal Tutup

Gambar 4.22
Tata Letak Penjualan

5. Laporan Stok Barang Nonkonsinyasi

Laporan stok barang konsinyasi merupakan dokumen berupa laporan yang digunakan untuk mengetahui mengenai jumlah stok barang konsinyasi.

Toko Barbie
Jl. Nyomplong No. 29

Laporan Stock Barang

No	Kode Barang	Nama Barang	Status	Jenis	Stock
				Total	

Gambar 4.27
Laporan Stok Barang Nonkonsinyasi

4.4 Implementasi Sistem

4.4.1 Pemilihan Perangkat Lunak

Beberapa perangkat lunak yang dibutuhkan oleh komputer server agar dapat menjalankan program aplikasi sistem baru pada Toko Barbie, yaitu :

1. Windows server 2016
2. Microsoft Visual Studio 2012
3. SQL Server management studio

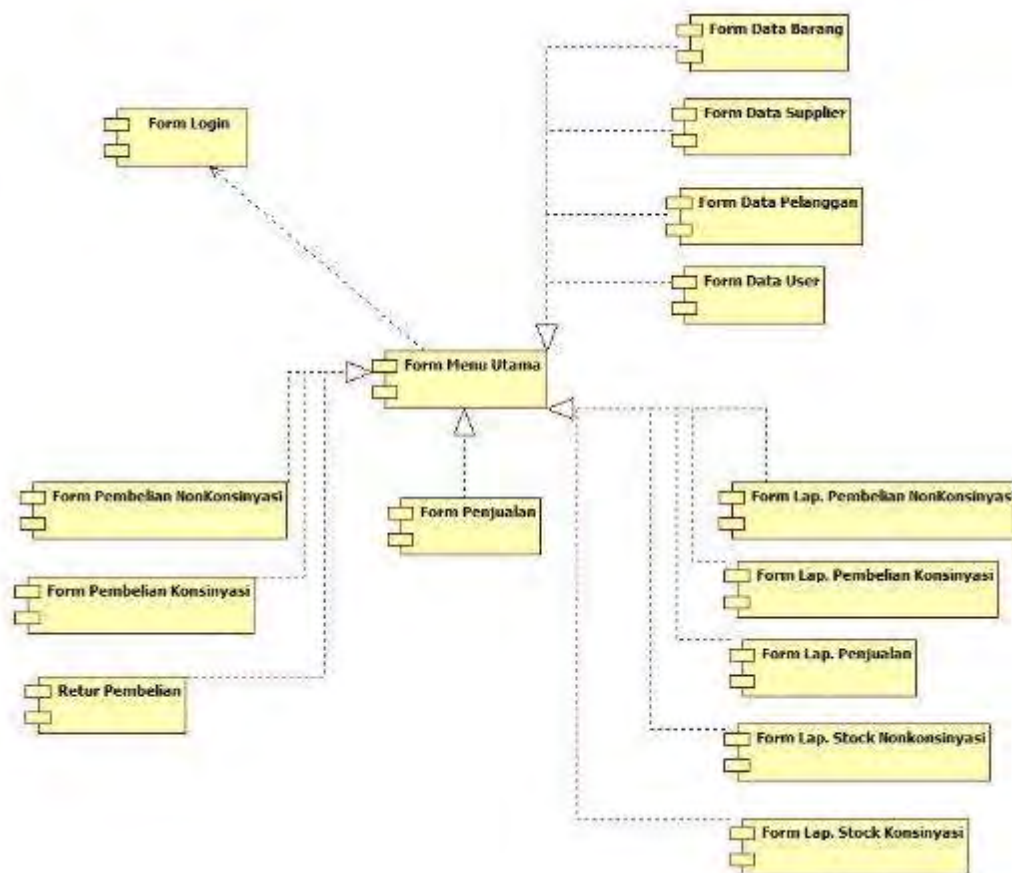
Pada komputer client sendiri diperlukan perangkat lunak sebagai berikut:

1. Windows 10
2. Microsoft Visual Studio 2012

4.4.2 Component Diagram

Component diagram pada Gambar 4.28 menjelaskan hubungan antar komponen yang terdapat pada sistem baru, dimana terdapat dua *package*, yaitu *package form* dan *package class*. *Package form* meliputi *interface-interface* yang terdapat pada sistem,

sedangkan *package class* meliputi kelas-kelas yang berisikan *atribut* atau *method*. Kelas-kelas tersebut tidak dapat langsung berinteraksi dengan *user*, maka dari itu kelas tersebut dihubungkan dengan *form* melalui *interface*. Terdapat pula komponen yang saling bergantung satu dengan yang lainnya, sehingga apabila komponen awalnya tidak dapat di-*compile* dengan benar, maka komponen yang berhubungan dengan komponen awalnya tidak dapat dijalankan.



Gambar 4.28
Component Diagram

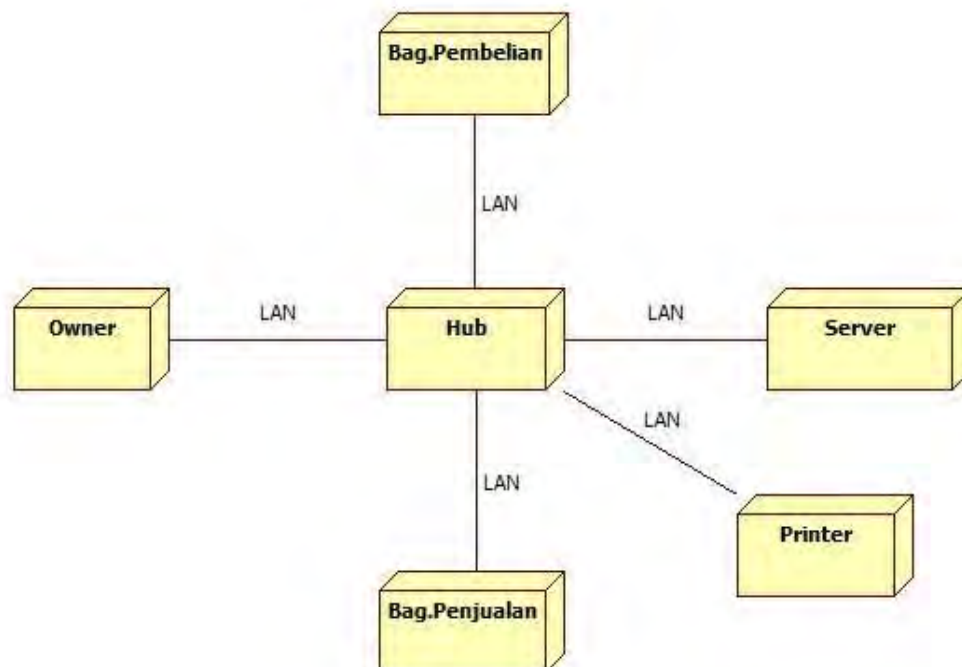
4.4.3 Persiapan Penerapan Sistem

Persiapan penerapan sistem dilakukan agar sistem baru yang telah dibangun dapat berjalan dengan baik sesuai dengan tujuannya salah satunya adalah dengan diadakannya pelatihan sumber daya manusia untuk memperkenalkan sistem baru dan cara bekerjanya.

Selain sumber daya manusia yang perlu disiapkan, evaluasi dan pengujian terhadap sistem baru juga perlu dilakukan agar sistem yang baru berjalan secara optimal sehingga hasil yang didapatkan memuaskan semua pihak.

4.4.4 Deployment Diagram

Deployment diagram pada gambar 4.29 menjelaskan mengenai perangkat keras yang saling berhubungan pada sistem baru. Pada sistem baru tersebut terdapat satu buah *server* yang terhubung dengan tiga *workstation* dan satu buah *printer* menggunakan *hub*.



Gambar 4.29
Deployment Diagram

4.4.5 Perkiraan Kebutuhan Biaya

Rincian perkiraan kebutuhan biaya untuk kebutuhan sistem baru adalah sebagai berikut:

Tabel 4.1
Rincian Perkiraan Kebutuhan Biaya
Penerapan Sistem Baru

No	Kebutuhan Biaya	Rincian	Harga
1.	Pengadaan perangkat keras	1 buah komputer server	Rp 10.908.199,00
		3 buah komputer client	Rp 7.200.000,00
		1 buah <i>hub</i>	Rp 72.000,00
		1 buah <i>printer</i>	Rp 1.725.000,00
2.	Pengadaan <i>software</i>	1 buah software windows 10 64bit	Rp 1.025.000,00
3.	Biaya pelatihan	Melatih pengguna dan dokumentasi	Rp 1.200.000,00
Total			Rp 22.130.199,00

4.5 Pengujian Sistem

Berikut ini merupakan daftar pengujian sistem terhadap sistem baru:

1. Pengujian Proses Login

Tabel 4.2
Pengujian Proses Login

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	UserId dan password sesuai, klik tombol Login	Muncul pesan berhasil dan form menu utama	Muncul pesan berhasil dan form menu utama	OK
2.	UserId atau password tidak sesuai klik Login	Muncul pesan error	Muncul pesan error "Login gagal silahkan cek kembali UserId & Password"	OK
3.	UserId kosong, password benar	Muncul pesan error	Muncul pesan error "UserId masih kosong"	OK
4.	UserId benar, password kosong	Muncul pesan error	Muncul pesan error "Password masih kosong"	OK
5.	UserID dan password benar klik batal	Form login tertutup	Form login tertutup	OK

2. Pengujian Proses Master Data Barang

Tabel 4.3
Pengujian Proses Master Data Barang

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Isi Nama barang, jenis, harga beli, harga jual, status, dan <i>stock</i> lalu tekan tombol simpan	Muncul pesan berhasil, data barang ditambahkan, tampil pada tabel barang	Muncul pesan berhasil, data barang ditambahkan, tampil pada tabel barang	OK
2.	Klik 2x pada data barang yang dipilih lalu ubah data barang dan tekan tombol simpan	Muncul pesan berhasil, data barang diubah, tampil pada tabel barang	Muncul pesan berhasil, data barang diubah, tampil pada tabel barang	OK
3.	Isi teks box cari dengan nama barang	Menampilkan data barang berdasarkan nama barang	Menampilkan data barang berdasarkan nama barang	OK
4.	Klik data barang lalu tekan hapus	Muncul pesan "Data berhasil dihapus" data barang terhapus dari tabel	Muncul pesan "Data berhasil dihapus" data barang terhapus dari tabel	OK
5.	Klik tombol tutup	Menutup form Master Data Barang	Menutup form Master Data Barang	OK

3. Pengujian Proses Master Data *Supplier*

Tabel 4.4
Pengujian Proses Master Data *Supplier*

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Isi nama, alamat, dan no. Telp <i>supplier</i> lalu tekan tombol simpan	Muncul pesan berhasil, data <i>supplier</i> ditambahkan, tampil pada tabel <i>supplier</i>	Muncul pesan berhasil, data <i>supplier</i> ditambahkan, tampil pada tabel <i>supplier</i>	OK
2.	Klik 2x pada data <i>supplier</i> yang dipilih lalu ubah data <i>supplier</i> dan tekan tombol simpan	Muncul pesan berhasil, data <i>supplier</i> diubah, tampil pada tabel <i>supplier</i>	Muncul pesan berhasil, data <i>supplier</i> diubah, tampil pada tabel <i>supplier</i>	OK
3.	Isi teks box cari dengan nama <i>supplier</i>	Menampilkan data <i>supplier</i> berdasarkan nama <i>supplier</i>	Menampilkan data <i>supplier</i> berdasarkan nama <i>supplier</i>	OK

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
4.	Klik data <i>supplier</i> lalu tekan hapus	Muncul pesan "Data berhasil dihapus" data <i>supplier</i> terhapus dari tabel	Muncul pesan "Data berhasil dihapus" data <i>supplier</i> terhapus dari tabel	OK
5.	Klik tombol tutup	Menutup form Master Data <i>supplier</i>	Menutup form Master Data <i>supplier</i>	OK

4. Pengujian Proses Master Data Pelanggan

Tabel 4.5
Pengujian Proses Master Data Pelanggan

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Isi nama, alamat, dan no. Telp pelanggan lalu tekan tombol simpan	Muncul pesan berhasil, data pelanggan ditambahkan, tampil pada tabel pelanggan	Muncul pesan berhasil, data pelanggan ditambahkan, tampil pada tabel pelanggan	OK
2.	Klik 2x pada data <i>supplier</i> yang dipilih lalu ubah data pelanggan dan tekan tombol simpan	Muncul pesan berhasil, data pelanggan diubah, tampil pada tabel pelanggan	Muncul pesan berhasil, data pelanggan diubah, tampil pada tabel pelanggan	OK
3.	Isi teks box cari dengan nama pelanggan	Menampilkan data pelanggan berdasarkan nama pelanggan	Menampilkan data pelanggan berdasarkan nama pelanggan	OK
4.	Klik data pelanggan lalu tekan hapus	Muncul pesan "Data berhasil dihapus" data pelanggan terhapus dari tabel	Muncul pesan "Data berhasil dihapus" data pelanggan terhapus dari tabel	OK
5.	Klik tombol tutup	Menutup form Master Data pelanggan	Menutup form Master Data pelanggan	OK

5. Pengujian Proses Master Data User

Tabel 4.6
Pengujian Proses Master Data User

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Isi user id, user name, dan password lalu tekan simpan	Muncul pesan berhasil, data user ditambahkan,	Muncul pesan berhasil, data user ditambahkan,	OK

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
		tampil pada tabel user	tampil pada tabel user	
2.	Klik 2x pada data user yang dipilih lalu ubah data user dan tekan tombol simpan	Muncul pesan berhasil, data user diubah, tampil pada tabel user	Muncul pesan berhasil, data user diubah, tampil pada tabel user	OK
3.	Isi teks box cari dengan nama user	Menampilkan data user berdasarkan nama user	Menampilkan data user berdasarkan nama user	OK
4.	Klik data user lalu tekan hapus	Muncul pesan "Data berhasil dihapus" data user terhapus dari tabel	Muncul pesan "Data berhasil dihapus" data user terhapus dari tabel	OK
5.	Klik tombol tutup	Menutup form Master Data user	Menutup form Master Data user	OK

6. Pengujian Proses Pembelian Nonkonsinyasi

Tabel 4.7
Pengujian Proses Pembelian Nonkonsinyasi

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih nama <i>supplier</i> , nama barang, dan jumlah barang, lalu tekan simpan	Muncul pesan berhasil, data pembelian ditambahkan	Muncul pesan berhasil, data pembelian ditambahkan	OK
2.	Klik tombol batal	Seluruh isi tabel akan terhapus	Seluruh isi tabel akan terhapus	OK
3.	Jumlah tidak <i>diinput</i> lalu klik simpan	Muncul pesan "Pastikan diisi angka"	Muncul pesan "Pastikan diisi angka"	OK
4.	Nama <i>supllier</i> kosong lalu klik simpan	Muncul pesan " <i>Supplier</i> masih kosong"	Muncul pesan " <i>Supplier</i> masih kosong"	OK
5.	Klik tombol tutup	Menutup form Pembelian Nonkonsinyasi	Menutup form Pembelian Nonkonsinyasi	OK

7. Pengujian Proses Pembelian Konsinyasi

Tabel 4.8
Pengujian Proses Pembelian Konsinyasi

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih nama <i>supplier</i> , nama barang, tanggal kembali dan jumlah barang, lalu tekan simpan	Muncul pesan berhasil, data pembelian ditambahkan	Muncul pesan berhasil, data pembelian ditambahkan	OK
2.	Klik tombol batal	Seluruh isi tabel akan terhapus	Seluruh isi tabel akan terhapus	OK
3.	Jumlah tidak <i>diinput</i> lalu klik simpan	Muncul pesan "Pastikan diisi angka"	Muncul pesan "Pastikan diisi angka"	OK
4.	Nama suplier kosong lalu klik simpan	Muncul pesan " <i>Supplier</i> masih kosong"	Muncul pesan " <i>Supplier</i> masih kosong"	OK
5.	Klik tombol tutup	Menutup form Pembelian Konsinyasi	Menutup form Pembelian Konsinyasi	OK

8. Pengujian Proses Retur Pembelian

Tabel 4.9
Pengujian Proses Retur Pembelian

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih cari faktur pembelian	Menampilkan faktur pembelian	Menampilkan faktur pembelian	OK
2.	Menambahkan data pembelian ke retur	Data barang ditambahkan ke table retur pembelian	Data barang ditambahkan ke table retur pembelian	OK
3.	Mengisi jumlah barang retur lebih besar dari faktur pembelian	Menampilkan pesan "Quantity retur lebih besar dari pembelian"	Menampilkan pesan "Quantity retur lebih besar dari pembelian"	OK
4.	Jumlah tidak <i>diinput</i> lalu klik simpan	Muncul pesan "Pastikan diisi angka"	Muncul pesan "Pastikan diisi angka"	OK
5.	Klik tombol tutup	Menutup form Retur Pembelian	Menutup form Retur Pembelian	OK

9. Pengujian Proses Penjualan

Tabel 4.10
Pengujian Proses Penjualan

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih nama pelanggan, nama barang, dan jumlah barang, lalu tekan simpan	Muncul pesan berhasil, data penjualan ditambahkan	Muncul pesan berhasil, data penjualan ditambahkan	OK
2.	Klik tombol batal	Seluruh isi tabel akan terhapus	Seluruh isi tabel akan terhapus	OK
3.	Jumlah yang <i>diinput</i> lebih besar dari jumlah stock	Muncul pesan "Stock tidak mencukupi" dan memberitahu jumlah stock saat ini	Muncul pesan "Stock tidak mencukupi" dan memberitahu jumlah stock saat ini	OK
4.	Menginput jumlah uang yang dibayarkan	Field kembali akan memunculkan jumlah kembalian yang seharusnya	Field kembali akan memunculkan jumlah kembalian yang seharusnya	OK
5.	Klik tombol tutup	Menutup form Penjualan	Menutup form Penjualan	OK

10. Pengujian Proses Laporan Pembelian Nonkonsinyasi

Tabel 4.11
Pengujian Proses Laporan
Pembelian Nonkonsinyasi

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih periode laporan	Menampilkan pembelian pada sepanjang periode yang telah dipilih	Menampilkan pembelian pada sepanjang periode yang telah dipilih	OK
2.	Pilih export laporan ke Excel, PDF, atau Word	Menampilkan laporan pada aplikasi yang telah dipilih	Menampilkan laporan pada aplikasi yang telah dipilih	OK
3.	Menginput kata yang ingin dicari pada laporan	Menampilkan kata yang ditemukan dalam laporan	Menampilkan kata yang ditemukan dalam laporan	OK
4.	Klik close	Menutup Form Laporan Pembelian Nonkonsinyasi	Menutup Form Laporan Pembelian Nonkonsinyasi	OK

11. Pengujian Proses Laporan Pembelian Konsinyasi

Tabel 4.12
Pengujian Proses Laporan
Pembelian Konsinyasi

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih periode laporan	Menampilkan pembelian pada sepanjang periode yang telah dipilih	Menampilkan pembelian pada sepanjang periode yang telah dipilih	OK
2.	Pilih export laporan ke Excel, PDF, atau Word	Menampilkan laporan pada aplikasi yang telah dipilih	Menampilkan laporan pada aplikasi yang telah dipilih	OK
3.	Menginput kata yang ingin dicari pada laporan	Menampilkan kata yang ditemukan dalam laporan	Menampilkan kata yang ditemukan dalam laporan	OK
4.	Klik close	Menutup Form Laporan Pembelian Nonkonsinyasi	Menutup Form Laporan Pembelian Nonkonsinyasi	OK

12. Pengujian Proses Laporan Penjualan

Tabel 4.13
Pengujian Proses Laporan Penjualan

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih periode laporan	Menampilkan penjualan pada sepanjang periode yang telah dipilih	Menampilkan penjualan pada sepanjang periode yang telah dipilih	OK
2.	Pilih export laporan ke Excel, PDF, atau Word	Menampilkan laporan pada aplikasi yang telah dipilih	Menampilkan laporan pada aplikasi yang telah dipilih	OK
3.	Menginput kata yang ingin dicari pada laporan	Menampilkan kata yang ditemukan dalam laporan	Menampilkan kata yang ditemukan dalam laporan	OK
4.	Klik close	Menutup Form Laporan Penjualan	Menutup Form Laporan Penjualan	OK

13. Pengujian proses Laporan Stock Barang

Tabel 4.14
Pengujian proses Laporan
Pembelian Konsinyasi

No	Data Uji	Hasil yang Diharapkan	Output Test	Kesimpulan
1.	Pilih status barang yang ingin ditampilkan konsinyasi atau nonkonsinyasi	Menampilkan stock barang dari status yang dipilih	Menampilkan stock barang dari status yang dipilih	OK
2.	Pilih export laporan ke Excel, PDF, atau Word	Menampilkan laporan pada aplikasi yang telah dipilih	Menampilkan laporan pada aplikasi yang telah dipilih	OK
3.	Menginput kata yang ingin dicari pada laporan	Menampilkan kata yang ditemukan dalam laporan	Menampilkan kata yang ditemukan dalam laporan	OK
4.	Klik close	Menutup Form Laporan Pembelian Nonkonsinyasi	Menutup Form Laporan Pembelian Nonkonsinyasi	OK

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian analisis dan perancangan sistem informasi penjualan, pembelian non-konsinyasi dan pembelian konsinyasi pada Toko Barbie yang telah dilakukan, maka dapat disimpulkan beberapa kesimpulan, yaitu:

1. Pada sistem manual yang masih diterapkan oleh Toko Barbie terjadi redundansi data *supplier* dan pelanggan yang menyebabkan pencatatan data *supplier* dan pelanggan dapat berulang. Maka dengan sistem baru ini diharapkan dapat memperkecil kemungkinan pencatatan data berulang.
2. Pada sistem manual membutuhkan lebih banyak tempat dikarenakan banyaknya perlengkapan yang digunakan seperti kertas, map, dan alat tulis, sedangkan dengan sistem baru pencatatan data dilakukan secara sistem dan lebih hemat tempat.
3. Adanya sistem baru membuat proses penginputan menjadi lebih cepat dikarenakan tidak perlu lagi dicatat secara manual.
4. Pencarian data lebih cepat dan mudah karena data disimpan dalam bentuk file pada komputer, bukan lagi berupa dokumen fisik
5. Proses pembuatan laporan menjadi lebih cepat karena sudah ada fitur cetak laporan dalam sistem.

5.2 Saran

Dari hasil penelitian yang telah dilakukan, maka diusulkanlah saran bagi Toko Barbie. Beberapa hal yang disarankan antara lain sebagai berikut :

1. Diharapkan sistem informasi yang dibuat dapat diterapkan dan data-data yang diperlukan diharapkan untuk di-*input* agar memudahkan dalam bertransaksi.
2. Disarankan untuk dilakukan pelatihan kepada karyawan yang ada untuk meminimalisir kesalahan dalam meng-*input* data atau *error* pada aplikasi.
3. Melakukan pemeliharaan terhadap sistem dan melakukan pengembangan sehingga dapat menyesuaikan dengan kebutuhan toko yang makin berkembang.

4. Untuk semakin mengurangi resiko kehilangan data akibat bencana alam atau kesalahan *user*, disarankan untuk melengkapi sistem informasi dengan penyimpanan data berbasis *online*, sehingga resiko kehilangan data dapat semakin dikurangi.

DAFTAR PUSTAKA

- Al-Jufri, Hamid. 2011. *Sistem Infomasi Manajemen Pendidikan*. Jakarta: PT. Smart Grafika.
- Darmawan, Deni. 2013. *Sistem Informasi Manajemen*. Bandung: PT. Remaja Rosdakarya Offset.
- Djahir, Yulia, dan Dewi Pratita. 2015. *Bahan Ajar Sistem Informasi Manajemen*. Yogyakarta: Deepublish Publisher.
- Hadori Yunus dan Harnanto. 2013. *Akuntansi Keuangan Lanjutan*. Yogyakarta : BPFE.
- Hartono, Bambang. 2013. *Sistem Informasi Manajemen Berbasis Komputer*. Jakarta: PT Rineka Cipta.
- Hery. 2011. *Teori Akuntansi*. Jakarta: Kencana.
- Hutahaeen, Jeperson. 2015. *Konsep Sistem Informasi*. Yogyakarta: Deepublish Publisher.
- Irawati, Susan. 2010. *Manajemen Keuangan*. Bandung : Pustaka
- Moekijat, Prasjo. 2011. *Pengantar Sistem Informasi Manajemen* Bandung: CV.Remadja
- Nofriadi. 2015. *Java Fundamental Dengan Netbeans 8.0.2*. Yogyakarta: Deepublish Publisher.
- Nugroho, Adi. 2010. *Rekayasa Perangkat Lunak Berorientasi Objek dengan metode USDP*. Yogyakarta: Andi Offset.
- Nugroho, Bunafit. 2013. *Dasar Pemograman Web PHP – MySQL dengan Dreamweaver*. Yogyakarta : Gava Media.
- Rosenblatt, Harry J. 2013. *Systems Analysis and Design, Tenth Edition*. Boston: Cengage Learning.
- Simarmata, Janner. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: CV Andi Offset.
- Sugiyono. 2013. *Metode Penelitian Manajemen*. Bandung : Alfabeta.
- Sukanto, Rosa A. dan M. Salahuddin. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Susanto, Azhar. 2013. *Sistem Informasi Akuntansi* : Bandung : Lingga Jaya.
- Sutabri, Tata. 2012. *Konsep Sistem Informasi*.Yogyakarta: CV. Andi Offset.
- Sutarman. 2012. *Buku Pengantar Teknologi Informasi*. Jakarta: Bumi Aksara.
- Taufiq, Rohmat. 2013. *Sistem Informasi Manajemen, Konsep Dasar, Analisa Dan Metode Pengembangan*. Yogyakarta : Graha Ilmu.
- Tohari, Hamim. 2014. *Analisis Serta perancangan Sistem Informasi Melalui Pendekatan UML*. Yogyakarta: CV Andi.

Triandini, Evi dan I Gede Suardika. 2012. *Step by Step Desain Proyek Menggunakan UML*. Yogyakarta: CV. Andi Offset.

Widodo, Prabowo Pudjo. 2011. *Menggunakan UML*. Bandung: Informatika.

Yakub. 2012. *Pengantar Sistem Informasi*, Yogyakarta: Graha Ilmu.

https://id.wikipedia.org/wiki/Toko_kelontong 6 Nov 2019.

LAMPIRAN

LAMPIRAN 1 CONTOH DOKUMEN SISTEM LAMA

1. Buku Telepon

KIM LAY	0852 5336 1740
Kiky	0857 4608 1421
Mukli	0850 5561 1434
Iwan	0848 5417 417
Jamie	0858 6225 4322
Andi	0858 6225 4322
Samudra	0858 6021 8740
Sheld	0815 1397 5222
Sri	0817 9532 2629
Mia	0857 2114 2334
Lina	0857 2000 9244
Aspin	0857 2283 3334
Aque	0877 4824 0745
Yuli	0877 7015 0750
Aming	0896 5977 0804
Daya	0817 6484 2415

2. Daftar Harga Barang

Pada ini barang	2.500
Lada ini barang	3.000
Sedap saja	2.500
Kecep Gani	25.000
Kecep Bontang A	17.500
Kecep Bontang B	12.500
Gambel ADE Smt	500
Kentor Ndt	23.000
Kentor Rnde	2.000
Sat B	155.000
Sat K	21.000
Aque Gdang	20.000
Aque Botal Gdang/Botal	3.000
Ate Gdang/Kecep	500
Ate Gdang/Kecep	16.500
Poka Jie Gdang	10.500
Jasum Super	20.000
Pokok Mangrove	22.000
Pokok Mangrove	15.000
Pokok Mangrove	21.000
Pokok Mangrove	21.000
Pokok Mangrove	25.000
Pokok Mangrove	20.000
Pokok Mangrove	10.000/lt.
Pokok Mangrove	2.000
Pokok Mangrove	2.000
Pokok Mangrove	3.500
Pokok Mangrove	5.000
Pokok Mangrove	3.500
Pokok Mangrove	3.000

LAMPIRAN 2 CONTOH TAMPILAN PROGRAM

1. Form Login



The screenshot shows a window titled "Login" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a dark blue header bar with the word "LOGIN" in white capital letters and a small yellow padlock icon on the right. Below the header, there are two text input fields: "User ID" and "Password". The "User ID" field has a cursor in it. Below the input fields, there are two buttons: a green button with a white checkmark and the text "Login", and a grey button with a red 'X' icon and the text "Batal".

2. Form Menu Utama



3. Form Master Data Barang

Master Data Barang

Kode Barang: 06

Nama Barang:

Jenis:

Status:

Harga Beli: 0

Harga Jual: 0

Stock: 0

Simpan Hapus Batal Tutup

Kode Barang	Nama Barang	Jenis	Harga Beli	Harga Jual	Stock	Status
01	Sammi	Makanan	2200.00	2500.00	10.00	Non Konsumsi
02	Aqua Galon	Minuman	18000.00	20000.00	30.00	Non Konsumsi
03	Lux	Perlengkapan	3300.00	3500.00	50.00	Non Konsumsi
04	Gas Besar	Perlengkapan	152000.00	155000.00	15.00	Non Konsumsi
05	Kerupuk	Makanan	4500.00	5000.00	100.00	Konsumsi

4. Form Master Data Supplier

Master Data Supplier

Kode Supplier: 04

Nama Supplier:

Alamat:

No. Telp:

Simpan Hapus Batal Tutup

Kode Supplier	Nama Supplier	Alamat	No. Telp
01	Mukti	Jl. Abc no. 32	08128272721
02	Ikhlan	Jl. Garuda No. 23	089272727212
03	Jimmy	Jl. Baros Kencana No. 412	0266214167

5. Form Master Data Pelanggan

Master Data Pelanggan

Kode Pelanggan: 004

Nama Pelanggan:

Alamat:

No. Telp:

Simpan Hapus Batal Tutup

	Kode Pelanggan	Nama Pelanggan	Alamat	No. Telp
▶	001	Apri	Jl. Pajagalan Dalam No. 21	098561218790
	002	Sn	Jl. Nyomplong No. 192	0266223364
	003	Deeh	Jl. Sriwedari No. 12	089781123176
*				

6. Form Master Data User

Master Data User

User Id: Admin

User Name: Administrator

Password: 123

Simpan Hapus Batal Tutup

	User Id	User Name	Password
▶	Admin	Administrator	123
	Gudang	Dinda	Dindacutie1
*			

7. Form Pembelian Non Konsinyasi

Pembelian Non Konsinyasi

No. Faktur: 2012002 Supplier / Toko: Jimmy
 Tanggal: 12/ 1/2020 Kode Supplier: 03

Cari Nama Barang: a

Kode Barang	Nama Barang	Harga Beli	Jumlah	Total
02	Aqua Galon	18000.00	10	180.000

Total: 180000

Simpan Batal Tutup

8. Form Pembelian Konsinyasi

Pembelian Konsinyasi

No. Faktur: 2012001 Supplier / Toko: Ikhsan
 Tanggal Titip: 12/ 1/2020 Kode Supplier: 02
 Tanggal Kembali: 12/ 8/2020

Cari Nama Barang: a

Kode Barang	Nama Barang	Harga Beli	Jumlah	Total
05	Kerupuk	4500.00	100	450.000

Total: 450000

Simpan Batal Tutup

9. Form Penjualan

Penjualan

No. Faktur: 2012001 Pelanggan: Rpn
Tanggal: 12/1/2020 Kode Pelanggan: 001

Kode Barang	Nama Barang	Harga	Jumlah	Total
04	Gas Besar	152,000.00	15	2,280,000

Car Nama Barang: gas

Total: 775,000
Dibayar: 780,000
Kembali: 5,000

Simpan Batal Tutup

10. Form Laporan Pembelian Non Konsinyasi

Laporan Pembelian Non Konsinyasi

Dari: 01 Dec 2020 Sampai: 01 Dec 2020 [Tampilkan](#)

12/1/2020 9:17:06 PM

Toko Barbie
Jl. Nyomplong No. 29

Laporan Pembelian Non Konsinyasi
01-Dec-2020 s.d 01-Dec-2020

No Faktur / Supplier	No	Tanggal	Kode Barang	Nama Barang	Harga Beli	Jumlah	Total
2012001 / Mukti	1	1 Dec 2020	04	Gas Besar	152,000	15	2,280,000
				Total		15	2,280,000
2012002 / Jimmy	2	1 Dec 2020	02	Aqua Galon	18,000	10	180,000
				Total		10	180,000
Grand Total						25	2,460,000

1 of 1

11. Form Laporan Pembelian Konsinyasi

Laporan Pembelian Konsinyasi

Dari (tanggal titip) 01 Dec 2020 Sampai 01 Dec 2020 [Tampilkan](#)

12/1/2020 9:17:53 PM

Toko Barbie
Jl. Nyomplong No. 29

Laporan Pembelian Konsinyasi
 01-Dec-2020 s.d 01-Dec-2020

No Faktur / Supplier	No	Tgl Titip	Tgl Ambil	Kode Barang	Nama Barang	Harga Beli	Jumlah	Retur	Total
2012001 / Ikhsan	1	1 Dec 2020	8 Dec 2020	05	Kerupuk	4,500	100	20	450,000
					Total		100	20.00	450,000
Total							100.00	20.00	450,000

1 of 1

12. Form Laporan Penjualan

Laporan Penjualan

Dari 01 Dec 2020 Sampai 31 Dec 2020 [Tampilkan](#)

12/1/2020 9:20:05 PM

Toko Barbie
Jl. Nyomplong No. 29

Laporan Penjualan
 01-Dec-2020 s.d 31-Dec-2020

No Faktur / Pelanggan	No	Tanggal	Kode Barang	Nama Barang	Harga	Jumlah	Total
2012001 / Apin	1	1 Dec 2020	04	Gas Besar	155,000	5	775,000
				Total	155,000	5	775,000
Grand Total					155,000	5	775,000

1 of 1

13. Form Laporan Stock Barang Non Konsinyasi

Laporan Stock Barang Non Konsiyasi

Tampilkan

1 of 1 100% Find Next

12/1/2020 9:20:19 PM

Toko Barbie
Jl. Nyomplong No. 29

Laporan Stock Barang Non Konsiyasi

No	Kode Barang	Nama Barang	Status	Jenis	Stock
1	01	Sarimi	Non Konsiyasi	Makanan	10.00
2	02	Aqua Galon	Non Konsiyasi	Minuman	40.00
3	03	Lux	Non Konsiyasi	Perlengkapan	50.00
4	04	Gas Besar	Non Konsiyasi	Perlengkapan	25.00

1 of 1

14. Form Laporan Stock Barang Konsinyasi

Laporan Stock Barang Konsiyasi

Tampilkan

1 of 1 100% Find Next

12/1/2020 9:20:42 PM

Toko Barbie
Jl. Nyomplong No. 29

Laporan Stock Barang Konsiyasi

No	Kode Barang	Nama Barang	Status	Jenis	Stock
1	05	Kerupuk	Konsiyasi	Makanan	220.00

1 of 1

LAMPIRAN 3 LISTING PROGRAM

1. Form Login

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmLogin : Form
    {
        ConnectionClass clscon;
        public FrmLogin()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void FrmLogin_Load(object sender, EventArgs e)
        {
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            if (txtUserID.Text == "")
            {
                MessageBox.Show("User Id masih kosong", "Informasi",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                txtUserID.Focus();
                return;
            }
            if (txtPassword.Text == "")
            {
                MessageBox.Show("Password masih kosong", "Informasi",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                txtPassword.Focus();
                return;
            }
            Login();
        }

        private void Login()
        {
            DataTable DtRow = clscon.GetDataTable("Select * From Master_User Where
            User_Id ='" + txtUserID.Text + "' And Password = '" + txtPassword.Text + "'");
            if (DtRow.Rows.Count > 0)
            {

```

```

        MessageBox.Show("Login Berhasil", "Informasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        UserClass.Instance.UserId = DataRow.Rows[0]["User_Id"].ToString();
        UserClass.Instance.UserName = DataRow.Rows[0]["User_Name"].ToString();
        this.Hide();
        FrmMain frmMain = new FrmMain();

        frmMain.ShowDialog();
        this.Close();
    }
    else
    {
        MessageBox.Show("Login Gagal. Silahkan Cek Kembali UserId dan
        Password", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnBatal_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

2. Form LookupPembelian

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmLookupPembelian : Form
    {
        ConnectionClass clscon;
        string CmdType = "Tambah";
        public string IsOke = "";
        public FrmLookupPembelian()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void LoadDataPembelian(string Filter)
        {
        }
    }
}

```

```

        DGVPembelian.Rows.Clear();
        string SQLQuery = "Select no_faktur,tgl,nama_supp,kd_supp,SUM(total_beli)
As total_beli From v_pembelian_for_retur " + Filter + " Group by
no_faktur,tgl,nama_supp,kd_supp Order By no_faktur";
        DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
        if (dataTable1.Rows.Count > 0)
        {
            foreach (DataRow dtr in dataTable1.Rows)
            {
                DataGridViewRow row =
                (DataGridViewRow)DGVPembelian.Rows[0].Clone();

                row.Cells[0].Value = dtr["no_faktur"].ToString();
                row.Cells[1].Value = dtr["tgl"].ToString();
                row.Cells[2].Value = dtr["nama_supp"].ToString();
                row.Cells[3].Value = dtr["kd_supp"].ToString();
                row.Cells[4].Value = dtr["total_beli"].ToString();
                DGVPembelian.Rows.Add(row);
            }
        }

        private void bntTampilkan_Click(object sender, EventArgs e)
        {
            string filter = " Where tgl >= " + StartDate.Value.ToString("yyyy-MM-dd") + "
And tgl <= " + EndDate.Value.ToString("yyyy-MM-dd") + """";
            LoadDataPembelian(filter);
        }

        private void btnPilih_Click(object sender, EventArgs e)
        {
            if (DGVPembelian.Rows.Count > 1)
            {
                IsOke = "Selected";
                this.Close();
            }
        }

        private void FrmLookupPembelian_FormClosed(object sender,
        FormClosedEventArgs e)
        {
            if (IsOke == null) IsOke = "NoSelected";
        }

        private void DGVPembelian_DoubleClick(object sender, EventArgs e)
        {
            if (DGVPembelian.Rows.Count > 1)
            {
                IsOke = "Selected";
                this.Close();
            }
        }
    }
}

```

3. Form Main

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RetailProgram
{
    public partial class FrmMain : Form
    {
        public FrmMain()
        {
            InitializeComponent();
        }

        private void dataBarangToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FrmMasterBarang frm1 = new FrmMasterBarang();
            foreach (Form form in Application.OpenForms)
            {
                if (form.GetType() == typeof(FrmMasterBarang))
                {
                    form.Activate();
                    return;
                }
            }
            frm1.Show();
        }

        private void dataSupplierToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FrmMasterSupplier frm1 = new FrmMasterSupplier();
            foreach (Form form in Application.OpenForms)
            {
                if (form.GetType() == typeof(FrmMasterSupplier))
                {
                    form.Activate();
                    return;
                }
            }
            frm1.Show();
        }

        private void dataPelangganToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FrmMasterPelanggan frm1 = new FrmMasterPelanggan();
            foreach (Form form in Application.OpenForms)
            {
                if (form.GetType() == typeof(FrmMasterPelanggan))
                {
                    form.Activate();
                    return;
                }
            }
        }
    }
}

```

```

    }
    }
    frm1.Show();
}

private void dataUserToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmMasterUser frm1 = new FrmMasterUser();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmMasterUser))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void FrmMain_Load(object sender, EventArgs e)
{
    this.Width = SystemInformation.VirtualScreen.Width - 100;
    this.Height = SystemInformation.VirtualScreen.Height - 200;
    // this.Location = new
    System.Drawing.Point((SystemInformation.VirtualScreen.Width - this.Width) / 2,
    (SystemInformation.VirtualScreen.Height - this.Height - 10) / 2);

    this.Location = new
    System.Drawing.Point((SystemInformation.VirtualScreen.Width - this.Width) / 2, 10);
}

private void konsiyasiToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmPembelianKonsiyasi frm1 = new FrmPembelianKonsiyasi();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPembelianKonsiyasi))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void orderToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmPenjualan frm1 = new FrmPenjualan();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPenjualan))
        {
            form.Activate();
            return;
        }
    }
}

```

```

    }
    frm1.Show();
}

private void lapPembelianNoKonsiyasiToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmPrintPembelianNon frm1 = new FrmPrintPembelianNon();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPrintPembelianNon))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void lapPembelianKonsiyasiToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmPrintPembelian frm1 = new FrmPrintPembelian();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPrintPembelian))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void lapPenjualanToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmPrintPenjualan frm1 = new FrmPrintPenjualan();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPrintPenjualan))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void lapStockToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmPrintStockNon frm1 = new FrmPrintStockNon();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPrintStockNon))
        {
            form.Activate();
            return;
        }
    }
}

```

```

    }
    frm1.Show();
}

private void lapBonusToolStripMenuItem_Click(object sender, EventArgs e)
{

}

private void logoutToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dResult = MessageBox.Show("Yakin ingin keluar ?",
"Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dResult == DialogResult.Yes)
    {
        this.Close();
    }
}

e) private void returPembelianToolStripMenuItem_Click(object sender, EventArgs
{

    FrmReturPembelian frm1 = new FrmReturPembelian();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmReturPembelian))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void penjualanToolStripMenuItem1_Click(object sender, EventArgs e)
{
    FrmPenjualan frm1 = new FrmPenjualan();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPenjualan))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

e) private void returPembelianToolStripMenuItem1_Click(object sender, EventArgs
{
    FrmReturPembelian frm1 = new FrmReturPembelian();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmReturPembelian))
        {
            form.Activate();

```

```

        return;
    }
}
frm1.Show();
}

private void lapStokKonsinyasiToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmPrintStockKonsiyasi frm1 = new FrmPrintStockKonsiyasi();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPrintStockKonsiyasi))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void returPembelToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmReturPembelian frm1 = new FrmReturPembelian();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmReturPembelian))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void pembelianNonKonsiyasiToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmPembelian frm1 = new FrmPembelian();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPembelian))
        {
            form.Activate();
            return;
        }
    }
    frm1.Show();
}

private void pembelianKonsiyasiToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmPembelianKonsiyasi frm1 = new FrmPembelianKonsiyasi();
    foreach (Form form in Application.OpenForms)
    {
        if (form.GetType() == typeof(FrmPembelianKonsiyasi))
        {

```

```

        form.Activate();
        return;
    }
}
frm1.Show();
}
}
}

```

4. Form Master Barang

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmMasterBarang : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsGen;
        string CmdType = "Tambah";
        public FrmMasterBarang()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsGen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadDataBarang("");
            txtKodeBarang.Text = clsGen.RunNumberRec("Kode_Barang");
        }

        private void LoadDataBarang(string Filter)
        {
            DGVBarang.Rows.Clear();
            string SQLQuery = "Select *, Isnull(stock,0) As stock1 From Master_Barang "
+ Filter + " Order By nama_brg,kd_brg";
            DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
            if (dataTable1.Rows.Count > 0)
            {
                foreach (DataRow dtr in dataTable1.Rows)
                {
                    DataGridViewRow row =
                    (DataGridViewRow)DGVBarang.Rows[0].Clone();

                    row.Cells[0].Value = dtr["kd_brg"].ToString();
                    row.Cells[1].Value = dtr["nama_brg"].ToString();
                }
            }
        }
    }
}

```

```

        row.Cells[2].Value = dtr["jenis"].ToString();
        row.Cells[3].Value = dtr["harga_beli"].ToString();
        row.Cells[4].Value = dtr["harga_jual"].ToString();
        row.Cells[5].Value = dtr["stock1"].ToString();
        row.Cells[6].Value = dtr["status"].ToString();
        DGVBarang.Rows.Add(row);
    }
}

private void ClearText()
{
    CmdType = "Tambah";
    txtKodeBarang.Text = clsGen.RunNumberRec("Kode_Barang");
    txtKodeBarang.ReadOnly = false;
    txtNamaBarang.Text = "";
    cmbJenis.ResetText();
    cmbJenis.SelectedIndex = -1;
    cmbStatusBarang.ResetText();
    cmbStatusBarang.SelectedIndex = -1;
    txtHargaBeli.Value = "0";
    txtHargaJual.Value = "0";
    StockAvailable.Value = "0";
}

private bool Validasi()
{
    if (txtKodeBarang.Text == "")
    {
        MessageBox.Show("Kode barang masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtKodeBarang.Focus();
        return false;
    }
    if (txtNamaBarang.Text == "")
    {
        MessageBox.Show("Nama barang masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtNamaBarang.Focus();
        return false;
    }
    if (cmbJenis.Text == "")
    {
        MessageBox.Show("Jenis Masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        cmbJenis.Focus();
        return false;
    }
    if (cmbStatusBarang.Text == "")
    {
        MessageBox.Show("Status barang kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        cmbJenis.Focus();
        return false;
    }
    if (txtHargaBeli.Text == "0")
    {

```

```

        MessageBox.Show("Harga beli masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtHargaBeli.Focus();
        return false;
    }
    if (txtHargaJual.Text == "0")
    {
        MessageBox.Show("Harga jual masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtHargaJual.Focus();
        return false;
    }
    return true;
}

private void SaveData()
{
    try
    {
        string SqlSave = "";
        switch (CmdType)
        {
            case "Tambah":
                SqlSave = "Insert Into Master_Barang (kd_brg, jenis, nama_brg,
                harga_beli, harga_jual,status,stock) Values(" +
                " " + txtKodeBarang.Text + "," + cmbJenis.Text + "," +
                txtNamaBarang.Text + "," +
                "" + Convert.ToDecimal(txtHargaBeli.Value) + "," +
                Convert.ToDecimal(txtHargaJual.Value) + "," + cmbStatusBarang.Text + "," +
                Convert.ToDecimal(StockAvailable.Value) + ")";

                break;

            case "Ubah":
                SqlSave = "Update Master_Barang Set jenis = " + cmbJenis.Text +
                " " +
                " nama_brg = " + txtNamaBarang.Text + " " +
                " harga_beli = " +
                Convert.ToDecimal(txtHargaBeli.Value) + " " +
                " harga_jual = " +
                Convert.ToDecimal(txtHargaJual.Value) + " " +
                " status = " + cmbStatusBarang.Text + " " +
                " stock= " +
                Convert.ToDecimal(StockAvailable.Value) + " " +
                " Where kd_brg = " + txtKodeBarang.Text + """;

                break;
        }
        if (clscon.Esekusi(SqlSave) == true)
        {
            if (CmdType == "Tambah")
            {
                clsGen.UpdateRunNumber("Kode_Barang", txtKodeBarang.Text);
            }
        }
    }
}

```

```

        MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR
        nama_brg like'" + txtPencarian.Text + "%'");

        ClearText();

    }

}
catch (Exception e)
{
    MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}
}

private void HapusDataBarang(string kdbrg)
{
    try
    {
        string SqlHapus = "Delete from Master_Barang where kd_brg = '" + kdbrg +
        "' ";

        if (clscon.Esekusi(SqlHapus) == true)
        {
            MessageBox.Show("Data berhasil dihapus ", "Konfirmasi",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR
            nama_brg like'" + txtPencarian.Text + "%'");
            ClearText();
        }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnHapus_Click(object sender, EventArgs e)
{
    if (DGVBarang.Rows.Count > 1)
    {

```

```

        string kdbrg =
DGVBarang.Rows[DGVBarang.CurrentCell.RowIndex].Cells[0].FormattedValue.ToString();
        DialogResult dResult = MessageBox.Show("yakin ingin hapus " + kdbrg + "
?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dResult == DialogResult.Yes)
        {
            HapusDataBarang(kdbrg);
        }
    }

    private void btnBatal_Click(object sender, EventArgs e)
    {
        ClearText();
    }

    private void btnTutup_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtPencarian_KeyPress(object sender, KeyPressEventArgs e)
    {

    }

    private void DGVBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVBarang.Rows.Count > 1)
        {
            int indx = DGVBarang.CurrentCell.RowIndex;

            txtKodeBarang.Text =
DGVBarang.Rows[indx].Cells[0].FormattedValue.ToString();
            txtNamaBarang.Text =
DGVBarang.Rows[indx].Cells[1].FormattedValue.ToString();
            cmbJenis.Text =
DGVBarang.Rows[indx].Cells[2].FormattedValue.ToString();
            txtHargaBeli.Value =
Convert.ToDecimal(DGVBarang.Rows[indx].Cells[3].Value).ToString();
            txtHargaJual.Value =
Convert.ToDecimal(DGVBarang.Rows[indx].Cells[4].Value).ToString();
            cmbStatusBarang.Text =
DGVBarang.Rows[indx].Cells[6].FormattedValue.ToString();
            StockAvailable.Value =
Convert.ToDecimal(DGVBarang.Rows[indx].Cells[5].Value).ToString();
            CmdType = "Ubah";
            txtKodeBarang.ReadOnly = true;
        }
    }

    private void txtPencarian_TextChanged(object sender, EventArgs e)

```

```

        {
            LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR
nama_brg like'" + txtPencarian.Text + "%'");
        }

        private void DGVBarang_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
        }
    }
}

```

5. Form Pelanggan

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmMasterPelanggan : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsGen;
        string CmdType = "Tambah";
        public FrmMasterPelanggan()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsGen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadDataPelanggan("");
            txtKodePelanggan.Text = clsGen.RunNumberRec("Kode_Pelanggan");
        }

        private void LoadDataPelanggan(string Filter)
        {
            DGVBarang.Rows.Clear();
            string SQLQuery = "Select * From Master_Pelanggan " + Filter + " Order By
nama_pelanggan,kd_pelanggan";
            DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
            if (dataTable1.Rows.Count > 0)
            {
                int i = 1;
            }
        }
    }
}

```

```

        foreach (DataRow dtr in dataTable1.Rows)
        {
            DataGridViewRow row =
            (DataGridViewRow)DGVBarang.Rows[0].Clone();

            row.Cells[0].Value = dtr["kd_pelanggan"].ToString();
            row.Cells[1].Value = dtr["nama_pelanggan"].ToString();
            row.Cells[2].Value = dtr["alamat"].ToString();
            row.Cells[3].Value = dtr["notelp"].ToString();
            DGVBarang.Rows.Add(row);
            i += 1;
        }
    }
}

private void ClearText()
{
    CmdType = "Tambah";
    txtKodePelanggan.Text = clsGen.RunNumberRec("Kode_Pelanggan");
    txtNamaPelanggan.Text = "";
    txtAlamat.Text = "";
    txtNoTelp.Text = "";
}

private bool Validasi()
{
    if (txtKodePelanggan.Text == "")
    {
        MessageBox.Show("Kode pelanggan masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtKodePelanggan.Focus();
        return false;
    }
    if (txtNamaPelanggan.Text == "")
    {
        MessageBox.Show("Nama pelanggan masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtNamaPelanggan.Focus();
        return false;
    }
    if (txtAlamat.Text == "")
    {
        MessageBox.Show("Alamat masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtAlamat.Focus();
        return false;
    }
    if (txtNoTelp.Text == "")
    {
        MessageBox.Show("No Telp masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtNoTelp.Focus();
        return false;
    }

    return true;
}

```

```

private void SaveData()
{
    try
    {
        string SqlSave = "";
        switch (CmdType)
        {
            case "Tambah":
                SqlSave = "Insert Into Master_Pelanggan (kd_pelanggan,
nama_pelanggan, alamat, notelp) Values(" +
                " " + txtKodePelanggan.Text + " , " + txtNamaPelanggan.Text +
                " , " + txtAlamat.Text + " , " + txtNoTelp.Text + ")";
                break;

            case "Ubah":
                SqlSave = "Update Master_Pelanggan Set nama_pelanggan = " +
txtNamaPelanggan.Text + " , " +
                " alamat = " + txtAlamat.Text + " , " +
                " notelp = " + txtNoTelp.Text + " " +
                " Where kd_pelanggan = " +
txtKodePelanggan.Text + " ";
                break;
        }
        if (clscon.Esekusi(SqlSave) == true)
        {
            MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            if (CmdType == "Tambah")
            {
                clsGen.UpdateRunNumber("Kode_Pelanggan",
txtKodePelanggan.Text);
            }
            LoadDataPelanggan("Where kd_pelanggan like " + txtPencarian.Text +
"% ' OR nama_pelanggan like " + txtPencarian.Text + "% '");
            ClearText();
        }
    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void HapusDataBarang(string kd_pelanggan)
{
    try
    {
        string SqlHapus = "Delete from Master_Pelanggan where kd_pelanggan =
" + kd_pelanggan + " ";
    }
}

```

```

        if (clscon.Esekusi(SqlHapus) == true)
        {
            MessageBox.Show("Data berhasil dihapus ", "Konfirmasi",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            LoadDataPelanggan("Where kd_pelanggan like '" + txtPencarian.Text +
            "%' OR nama_pelanggan like'" + txtPencarian.Text + "%'");
            ClearText();
        }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnHapus_Click(object sender, EventArgs e)
{
    if (DGVBarang.Rows.Count > 1)
    {
        string kd_supp =
        DGVBarang.Rows[DGVBarang.CurrentCell.RowIndex].Cells[0].FormattedValue.ToString();
        DialogResult dResult = MessageBox.Show("yakin ingin hapus " + kd_supp
        + " ?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dResult == DialogResult.Yes)
        {
            HapusDataBarang(kd_supp);
        }
    }
}

private void btnBatal_Click(object sender, EventArgs e)
{
    ClearText();
}

private void btnTutup_Click(object sender, EventArgs e)
{
    this.Close();
}

private void txtPencarian_KeyPress(object sender, KeyPressEventArgs e)

```

```

    {

    }

    private void DGVBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVBarang.Rows.Count > 1)
        {
            int indx = DGVBarang.CurrentCell.RowIndex;

            txtKodePelanggan.Text =
            DGVBarang.Rows[indx].Cells[0].FormattedValue.ToString();
            txtNamaPelanggan.Text =
            DGVBarang.Rows[indx].Cells[1].FormattedValue.ToString();
            txtAlamat.Text =
            DGVBarang.Rows[indx].Cells[2].FormattedValue.ToString();
            txtNoTelp.Text =
            DGVBarang.Rows[indx].Cells[3].FormattedValue.ToString();
            CmdType = "Ubah";
            txtKodePelanggan.ReadOnly = true;

        }
    }

    private void txtPencarian_TextChanged(object sender, EventArgs e)
    {
        LoadDataPelanggan("Where kd_pelanggan like '" + txtPencarian.Text + "%'
OR nama_pelanggan like'" + txtPencarian.Text + "%'");
    }

    private void DGVBarang_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

}

```

6. Form Master Supplier

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmMasterSupplier : Form

```

```

{
    ConnectionClass clscon;
    ClassGeneral clsGen;
    string CmdType = "Tambah";
    public FrmMasterSupplier()
    {
        InitializeComponent();
        clscon = new ConnectionClass();
        clsGen = new ClassGeneral();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        LoadDataSupplier("");
        txtKodeSupplier.Text = clsGen.RunNumberRec("Kode_Supplier");
    }

    private void LoadDataSupplier(string Filter)
    {
        DGVBarang.Rows.Clear();
        string SQLQuery = "Select * From Master_Supplier " + Filter + " Order By
nama_supp,kd_supp";
        DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
        if (dataTable1.Rows.Count > 0)
        {
            int i = 1;
            foreach (DataRow dtr in dataTable1.Rows)
            {
                DataGridViewRow row =
(DataGridViewRow)DGVBarang.Rows[0].Clone();

                row.Cells[0].Value = dtr["kd_supp"].ToString();
                row.Cells[1].Value = dtr["nama_supp"].ToString();
                row.Cells[2].Value = dtr["alamat"].ToString();
                row.Cells[3].Value = dtr["notelp"].ToString();
                DGVBarang.Rows.Add(row);
                i += 1;
            }
        }
    }

    private void ClearText()
    {
        CmdType = "Tambah";
        //txtKodeSupplier.Text = "";
        //txtKodeSupplier.ReadOnly = false;
        txtKodeSupplier.Text = clsGen.RunNumberRec("Kode_Supplier");
        txtNamaSupplier.Text = "";
        txtAlamat.Text = "";
        txtNoTelp.Text = "";
    }

    private bool Validasi()
    {
        if (txtKodeSupplier.Text == "")
        {

```

```

        MessageBox.Show("Kode supplier masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtKodeSupplier.Focus();
        return false;
    }
    if (txtNamaSupplier.Text == "")
    {
        MessageBox.Show("Nama supplier masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtNamaSupplier.Focus();
        return false;
    }
    if (txtAlamat.Text == "")
    {
        MessageBox.Show("Alamat masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtAlamat.Focus();
        return false;
    }
    if (txtNoTelp.Text == "0")
    {
        MessageBox.Show("No Telp masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtNoTelp.Focus();
        return false;
    }

    return true;
}

```

```

private void SaveData()
{
    try
    {
        string SqlSave = "";
        switch (CmdType)
        {
            case "Tambah":
                SqlSave = "Insert Into Master_Supplier (kd_supp, nama_supp,
alamat, notelp) Values(" +
                " " + txtKodeSupplier.Text + "," + txtNamaSupplier.Text + "," +
txtAlamat.Text + "," + txtNoTelp.Text + ")";
                break;

            case "Ubah":
                SqlSave = "Update Master_Supplier Set nama_supp = " +
txtNamaSupplier.Text + "," +
                " alamat = " + txtAlamat.Text + "," +
                " notelp = " + txtNoTelp.Text + "" +
                " Where kd_supp = " + txtKodeSupplier.Text + """;

                break;
        }
        if (clscon.Esekusi(SqlSave) == true)
        {

```

```

        MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        if (CmdType == "Tambah")
        {
            clsGen.UpdateRunNumber("Kode_Supplier", txtKodeSupplier.Text);
        }
        LoadDataSupplier("Where kd_supp like '" + txtPencarian.Text + "%' OR
nama_supp like'" + txtPencarian.Text + "%'");
        ClearText();
    }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void HapusDataBarang(string kd_supp)
{
    try
    {
        string SqlHapus = "Delete from Master_Supplier where kd_supp = '" +
kd_supp + "'";

        if (clscon.Esekusi(SqlHapus) == true)
        {
            MessageBox.Show("Data berhasil dihapus ", "Konfirmasi",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            LoadDataSupplier("Where kd_supp like '" + txtPencarian.Text + "%' OR
nama_supp like'" + txtPencarian.Text + "%'");
            ClearText();
        }
    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnHapus_Click(object sender, EventArgs e)
{
    if (DGVBarang.Rows.Count > 1)

```

```

        {
            string kd_supp =
DGVBarang.Rows[DGVBarang.CurrentCell.RowIndex].Cells[0].FormattedValue.ToString();
            DialogResult dResult = MessageBox.Show("yakin ingin hapus " + kd_supp
+ " ?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (dResult == DialogResult.Yes)
            {
                HapusDataBarang(kd_supp);
            }
        }
    }

    private void btnBatal_Click(object sender, EventArgs e)
    {
        ClearText();
    }

    private void btnTutup_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtPencarian_KeyPress(object sender, KeyPressEventArgs e)
    {
    }

    private void DGVBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVBarang.Rows.Count > 1)
        {
            int indx = DGVBarang.CurrentCell.RowIndex;

            txtKodeSupplier.Text =
DGVBarang.Rows[indx].Cells[0].FormattedValue.ToString();
            txtNamaSupplier.Text =
DGVBarang.Rows[indx].Cells[1].FormattedValue.ToString();
            txtAlamat.Text =
DGVBarang.Rows[indx].Cells[2].FormattedValue.ToString();
            txtNoTelp.Text =
DGVBarang.Rows[indx].Cells[3].FormattedValue.ToString();
            CmdType = "Ubah";
            // txtKodeSupplier.ReadOnly = true;
        }
    }

    private void txtPencarian_TextChanged(object sender, EventArgs e)
    {
        LoadDataSupplier("Where kd_supp like '" + txtPencarian.Text + "%' OR
nama_supp like '" + txtPencarian.Text + "%'");
    }
}

```

```

        private void DGVBarang_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

    }
}

```

7. Form Mater User

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmMasterUser : Form
    {
        ConnectionClass clscon;
        string CmdType = "Tambah";
        public FrmMasterUser()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadDataUser("");
        }

        private void LoadDataUser(string Filter)
        {
            DGVBarang.Rows.Clear();
            string SQLQuery = "Select * From Master_User " + Filter + " Order By
user_name,user_id";
            DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
            if (dataTable1.Rows.Count > 0)
            {
                int i = 1;
                foreach (DataRow dtr in dataTable1.Rows)
                {
                    DataGridViewRow row =
(DataGridViewRow)DGVBarang.Rows[0].Clone();

                    row.Cells[0].Value = dtr["user_id"].ToString();
                    row.Cells[1].Value = dtr["user_name"].ToString();
                    row.Cells[2].Value = dtr["password"].ToString();
                }
            }
        }
    }
}

```

```

        DGVBarang.Rows.Add(row);
        i += 1;
    }
}

private void ClearText()
{
    CmdType = "Tambah";
    txtUserId.Text = "";
    txtUserId.ReadOnly = false;
    txtUserName.Text = "";
    txtPassword.Text = "";
}

private bool Validasi()
{
    if (txtUserId.Text == "")
    {
        MessageBox.Show("User Id masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtUserId.Focus();
        return false;
    }
    if (txtUserName.Text == "")
    {
        MessageBox.Show("User name masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtUserName.Focus();
        return false;
    }
    if (txtPassword.Text == "")
    {
        MessageBox.Show("Password masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        txtPassword.Focus();
        return false;
    }

    return true;
}

private void SaveData()
{
    try
    {
        string SqlSave = "";
        switch (CmdType)
        {
            case "Tambah":
                SqlSave = "Insert Into Master_User (user_id, user_name, password)
                Values(" +
                " " + txtUserId.Text + ", " + txtUserName.Text + ", " +
                txtPassword.Text + ")";
                break;

```

```

        case "Ubah":
            SqlSave = "Update Master_User Set user_name = '" +
txtUserName.Text + "','" +
                                " password = '" + txtPassword.Text + "'" +
                                " Where user_id = '" + txtUserId.Text + "'";

            break;
        }
        if (clscon.Esekusi(SqlSave) == true)
        {
            MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            LoadDataUser("Where user_id like '" + txtPencarian.Text + "%' OR
user_name like'" + txtPencarian.Text + "%'");
            ClearText();
        }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void HapusDataUser(string user_id)
{
    try
    {
        string SqlHapus = "Delete from Master_User where user_id = '" + user_id +
"" ",
                                "

        if (clscon.Esekusi(SqlHapus) == true)
        {
            MessageBox.Show("Data berhasil dihapus ", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            LoadDataUser("Where user_id like '" + txtPencarian.Text + "%' OR
user_name like'" + txtPencarian.Text + "%'");
            ClearText();
        }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)

```

```

        {
            SaveData();
        }
    }

    private void btnHapus_Click(object sender, EventArgs e)
    {
        if (DGVBarang.Rows.Count > 1)
        {
            string user_id =
DGVBarang.Rows[DGVBarang.CurrentCell.RowIndex].Cells[0].FormattedValue.ToString();
            DialogResult dResult = MessageBox.Show("yakin ingin hapus " + user_id +
" ?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (dResult == DialogResult.Yes)
            {
                HapusDataUser(user_id);
            }
        }
    }

    private void btnBatal_Click(object sender, EventArgs e)
    {
        ClearText();
    }

    private void btnTutup_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtPencarian_KeyPress(object sender, KeyPressEventArgs e)
    {
    }

    private void DGVBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVBarang.Rows.Count > 1)
        {
            int indx = DGVBarang.CurrentCell.RowIndex;

            txtUserId.Text =
DGVBarang.Rows[indx].Cells[0].FormattedValue.ToString();
            txtUserName.Text =
DGVBarang.Rows[indx].Cells[1].FormattedValue.ToString();
            txtPassword.Text =
DGVBarang.Rows[indx].Cells[2].FormattedValue.ToString();
            CmdType = "Ubah";
            txtUserId.ReadOnly = true;
        }
    }
}

```

```

        private void txtPencarian_TextChanged(object sender, EventArgs e)
        {
            LoadDataUser("Where user_id like '" + txtPencarian.Text + "%' OR
user_name like'" + txtPencarian.Text + "%'");
        }

        private void DGVBarang_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
        }
    }
}

```

8. Form Pembelian

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmPembelian : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsgen;

        string CmdType = "Tambah";
        public FrmPembelian()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsgen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadDataSupplier();

            txtFaktur.Text = clsgen.RunNumberRec("PembelianNonKonsiyasi");
        }

        private void LoadDataSupplier()
        {
            string SQLQuery = "Select * From Master_Supplier Order By
nama_supp,kd_supp";

```

```

DataTable table1 = clscon.GetDataTable(SQLQuery);
if (table1.Rows.Count > 0)
{
    cmbSupplier.DataSource = table1;
    cmbSupplier.ValueMember = "kd_supp";
    cmbSupplier.DisplayMember = "nama_supp";
}
cmbSupplier.SelectedIndex = -1;
}

private void ClearText()
{
    CmdType = "Tambah";
    txtFaktur.Text = "";
    txtFaktur.ReadOnly = false;
    dtTanggalBeli.Value = DateTime.Today;
    cmbSupplier.SelectedIndex = -1;
    txtJumlahBarang.Value = "0";
    txtGrandTotal.Value = "0";
    DGVListPembelian.Rows.Clear();
    txtFaktur.Text = clsgen.RunNumberRec("PembelianNonKonsiyasi");
}

private bool Validasi()
{
    {
        if (cmbSupplier.Text == "")
        {
            MessageBox.Show("Supplier masih kosong", "Konfirmasi",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            cmbSupplier.Focus();
            return false;
        }
        if (DGVListPembelian.Rows.Count < 1)
        {
            MessageBox.Show("Data barang masih kosong", "Konfirmasi",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            DGVListPembelian.Focus();
            return false;
        }
    }

    return true;
}

private void SaveData()
{
    try
    {
        string SqlSave = "";
        bool val = false;
        switch (CmdType)
        {
            case "Tambah":

                for (int i = 0; i < DGVListPembelian.RowCount; i++)
                {

```

```

string kd_brg =
DGVListPembelian.Rows[i].Cells["ColKodeBarang"].FormattedValue.ToString();
decimal Qty =
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].Value);
decimal Harga =
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColHargaBeli"].Value);
decimal Subtotal =
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].Value);

```

```

SqlSave = " Insert Into PembelianNonKonsiyasi (no_faktur,tgl_beli, kd_brg,
kd_supp, jml, harga_beli,total_beli,user_id) Values(" +
" "" + txtFaktur.Text + "," + dtTanggalBeli.Value.ToString("yyyy-MM-dd") + "," +
kd_brg + "," + txtKodeSupplier.Text + "," +
"" + Qty + "," + Harga + "," + Subtotal + "," + UserClass.Instance.UserId + ") ";

```

```

val = clscon.Esekusi(SqlSave);
if (val == true)
{
    clsgen.updateStock(kd_brg, Qty);
}
}

```

```

break;
}
if (val == true)
{
    MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    clsgen.UpdateRunNumber("PembelianNonKonsiyasi", txtFaktur.Text);
    ClearText();
}

```

```

}
catch (Exception e)
{
    MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
}

```

```

private void LoadDataBarang(string Filter)
{
    DGVListBarang.Rows.Clear();
    string SQLQuery = "Select * From Master_Barang " + Filter + " Order By
nama_brg,kd_brg";
    DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
    if (dataTable1.Rows.Count > 0)
    {
        int i = 1;
        foreach (DataRow dtr in dataTable1.Rows)
        {
            DataGridViewRow row = (DataGridViewRow)DGVListBarang.Rows[0].Clone();

```

```

row.Cells[0].Value = dtr["kd_brg"].ToString();
row.Cells[1].Value = dtr["nama_brg"].ToString();
row.Cells[2].Value = dtr["harga_beli"].ToString();
DGVListBarang.Rows.Add(row);
i += 1;
}
}
}

```

```

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

```

```

private void btnBatal_Click(object sender, EventArgs e)
{
    ClearText();
}

```

```

private void btnTutup_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```

private void txtPencarian_TextChanged(object sender, EventArgs e)
{
    DGVListBarang.Rows.Clear();
    if (txtPencarian.Text != "")
    {
        LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR nama_brg like '" + txtPencarian.Text + "%' And Isnull([status], 'Non Konsiyasi') = 'Non Konsiyasi'");
    }
}

```

```

}

```

```

private void cmbSupplier_SelectedIndexChanged(object sender, EventArgs e)
{
    txtKodeSupplier.Text = cmbSupplier.GetItemText(cmbSupplier.SelectedValue);
}

```

```

private void DGVListBarang_DoubleClick(object sender, EventArgs e)
{
    if (DGVListBarang.Rows.Count > 1)
    {

```

```

int indx = DGVListBarang.CurrentRow.Index;

for (int vloop = 0; vloop < DGVListPembelian.Rows.Count; vloop++)
{
    string KodeItems =
    DGVListPembelian.Rows[vloop].Cells["ColKodeBarang"].FormattedValue.ToString();
    if (KodeItems == DGVListBarang.Rows[indx].Cells[0].FormattedValue.ToString())
    {
        MessageBox.Show("Kode barang sudah ada", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

        return;
    }
}

DGVListPembelian.Rows.Add();
DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColKodeBarang"].Value =
DGVListBarang.Rows[indx].Cells[0].FormattedValue.ToString();
DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColNamaBarang"].Value =
DGVListBarang.Rows[indx].Cells[1].FormattedValue.ToString();
DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColHargaBeli"].Value =
DGVListBarang.Rows[indx].Cells[2].FormattedValue.ToString();
}

private void DGVListPembelian_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 3)
    {
        decimal Qty =
        Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColJumlah"].Value);
        decimal HargaBeli =
        Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColHargaBeli"].Value);
        decimal Subtotal = Qty * HargaBeli;

        this.DGVListPembelian.Rows[e.RowIndex].Cells["ColTotal"].Value =
        string.Format("{0:##,###0}", Subtotal);
        GrandTotalQty();
    }
}

private void DGVListPembelian_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
{
    if (e.ColumnIndex == 3) // 1 should be your column index
    {
        int i;

```

```

if (!int.TryParse(Convert.ToString(e.FormattedValue), out i))
{
    e.Cancel = true;
    MessageBox.Show("Pastikan diisi angka", "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
else
{
    // the input is numeric
}
}
}

private void GrandTotalQty()
{
    try
    {
        decimal Total = 0;
        decimal TotalItems = 0;
        if (DGVListPembelian.RowCount > 0)
        {
            for (int i = 0; i < DGVListPembelian.RowCount; i++)
            {
                Total +=
                Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].FormattedValue
                );
                TotalItems +=
                Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].FormattedValue
                );
            }
        }
        txtJumlahBarang.Text = TotalItems.ToString();
        txtGrandTotal.Text = Total.ToString();
    }
    catch { }

}

}
}

```

9. Form Pembelian Konsinyasi

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

```

```

namespace RetailProgram

```

```

{
    public partial class FrmPembelianKonsiyasi : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsgen;

        string CmdType = "Tambah";
        public FrmPembelianKonsiyasi()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsgen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadDataSupplier();

            txtFaktur.Text = clsgen.RunNumberRec("PembelianKonsiyasi");
        }

        private void LoadDataSupplier()
        {
            string SQLQuery = "Select * From Master_Supplier Order By
nama_supp,kd_supp";
            DataTable table1 = clscon.GetDataTable(SQLQuery);
            if (table1.Rows.Count > 0)
            {
                cmbSupplier.DataSource = table1;
                cmbSupplier.ValueMember = "kd_supp";
                cmbSupplier.DisplayMember = "nama_supp";
            }
            cmbSupplier.SelectedIndex = -1;
        }

        private void ClearText()
        {
            CmdType = "Tambah";
            txtFaktur.Text = "";
            txtFaktur.ReadOnly = false;
            dtTanggalTitip.Value = DateTime.Today;
            cmbSupplier.SelectedIndex = -1;
            txtJumlahBarang.Value = "0";
            txtGrandTotal.Value = "0";
            DGVLstPembelian.Rows.Clear();
            txtFaktur.Text = clsgen.RunNumberRec("PembelianKonsiyasi");
        }

        private bool Validasi()
        {
            if (cmbSupplier.Text == "")
            {
                MessageBox.Show("Supplier masih kosong", "Konfirmasi",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                cmbSupplier.Focus();
                return false;
            }
        }
    }
}

```

```

    }
    if (DGVListPembelian.Rows.Count < 1)
    {
        MessageBox.Show("Data barang masih kosong", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        DGVListPembelian.Focus();
        return false;
    }

    return true;
}

private void SaveData()
{
    try
    {
        string SqlSave = "";
        bool val = false;
        switch (CmdType)
        {
            case "Tambah":

                for (int i = 0; i < DGVListPembelian.RowCount; i++)
                {

                    string kd_brg =
                    DGVListPembelian.Rows[i].Cells["ColKodeBarang"].FormattedValue.ToString();
                    decimal Qty =
                    Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].Value);
                    decimal Harga =
                    Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColHargaBeli"].Value);
                    decimal Subtotal =
                    Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].Value);

                    SqlSave = " Insert Into PembelianKonsiyasi
                    (no_faktur,tgl_titip,tgl_ambil, kd_brg, kd_supp, jml, harga_beli,total_beli,user_id)
                    Values(" +
                        " " + txtFaktur.Text + "," + dtTanggalTitip.Value.ToString("yyyy-
                    MM-dd") + "," + dtTanggalKembali.Value.ToString("yyyy-MM-dd") + "," + kd_brg +
                    "," + txtKodeSupplier.Text + "," +
                        "" + Qty + "," + Harga + "," + Subtotal + "," +
                    UClass.Instance.UserId + ") ";

                    val = clscon.Esekusi(SqlSave);
                    if (val == true)
                    {
                        clsgen.updateStock(kd_brg, Qty);
                    }
                }

                break;
            }
        if (val == true)
        {

```

```

        MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        clsgen.UpdateRunNumber("PembelianKonsiyasi", txtFaktur.Text);
        ClearText();
    }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void LoadDataBarang(string Filter)
{
    DGVListBarang.Rows.Clear();
    string SQLQuery = "Select * From Master_Barang " + Filter + " Order By
nama_brg,kd_brg";
    DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
    if (dataTable1.Rows.Count > 0)
    {
        int i = 1;
        foreach (DataRow dtr in dataTable1.Rows)
        {
            DataGridViewRow row =
(DataGridViewRow)DGVListBarang.Rows[0].Clone();

            row.Cells[0].Value = dtr["kd_brg"].ToString();
            row.Cells[1].Value = dtr["nama_brg"].ToString();
            row.Cells[2].Value = dtr["harga_beli"].ToString();
            DGVListBarang.Rows.Add(row);
            i += 1;
        }
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnBatal_Click(object sender, EventArgs e)
{
    ClearText();
}

private void btnTutup_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```

    }

    private void txtPencarian_TextChanged(object sender, EventArgs e)
    {
        DGVListBarang.Rows.Clear();
        if (txtPencarian.Text != "")
        {
            LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR
nama_brg like'" + txtPencarian.Text + "%' And Isnull([status],'Non Konsiyasi') =
'Konsiyasi' ");
        }
    }

    private void cmbSupplier_SelectedIndexChanged(object sender, EventArgs e)
    {
        txtKodeSupplier.Text =
cmbSupplier.GetItemText(cmbSupplier.SelectedValue);
    }

    private void DGVListBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVListBarang.Rows.Count > 1)
        {
            int indx = DGVListBarang.CurrentCell.RowIndex;

            for (int vloop = 0; vloop < DGVListPembelian.Rows.Count; vloop++)
            {
                string KodeItems =
DGVListPembelian.Rows[vloop].Cells["ColKodeBarang"].FormattedValue.ToString();
                if (KodeItems ==
DGVListBarang.Rows[indx].Cells[0].FormattedValue.ToString())
                {
                    MessageBox.Show("Kode barang sudah ada", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                }

                return;
            }
        }

        DGVListPembelian.Rows.Add();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColKodeBarang"].Value =
DGVListBarang.Rows[indx].Cells[0].FormattedValue.ToString();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColNamaBarang"].Value =
DGVListBarang.Rows[indx].Cells[1].FormattedValue.ToString();
    }

```

```

        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColHargaBeli"].Value =
DGVListBarang.Rows[indx].Cells[2].FormattedValue.ToString();
    }
}

private void DGVListPembelian_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 3)
    {
        decimal Qty =
Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColJumlah"].Value)
;
        decimal HargaBeli =
Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColHargaBeli"].Val
ue);
        decimal Subtotal = Qty * HargaBeli;

        this.DGVListPembelian.Rows[e.RowIndex].Cells["ColTotal"].Value =
string.Format("{0:#,##0}", Subtotal);
        GrandTotalQty();
    }
}

private void DGVListPembelian_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
{
    if (e.ColumnIndex == 3) // 1 should be your column index
    {
        int i;

        if (!int.TryParse(Convert.ToString(e.FormattedValue), out i))
        {
            e.Cancel = true;
            MessageBox.Show("Pastikan diisi angka", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
        else
        {
            // the input is numeric
        }
    }
}

private void GrandTotalQty()
{
    try
    {
        decimal Total = 0;
        decimal TotalItems = 0;
        if (DGVListPembelian.RowCount > 0)
        {
            for (int i = 0; i < DGVListPembelian.RowCount; i++)
            {

```

```

        Total +=
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].FormattedValue);
        TotalItems +=
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].FormattedValue);
    }
    }
    txtJumlahBarang.Text = TotalItems.ToString();
    txtGrandTotal.Text = Total.ToString();
}
catch { }

}

private void DGVListBarang_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{

}

private void textBox6_TextChanged(object sender, EventArgs e)
{

}

}
}

```

10. Form Penjualan

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmPenjualan : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsgen;

        string CmdType = "Tambah";
        public FrmPenjualan()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsgen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)

```

```

    {
        LoadDataSupplier();

        txtFaktur.Text = clsgen.RunNumberRec("Penjualan");
    }

    private void LoadDataSupplier()
    {
        string SQLQuery = "Select * From Master_Pelanggan Order By
nama_pelanggan,kd_pelanggan";
        DataTable table1 = clscon.GetDataTable(SQLQuery);
        if (table1.Rows.Count > 0)
        {
            cmbPelanggan.DataSource = table1;
            cmbPelanggan.ValueMember = "kd_pelanggan";
            cmbPelanggan.DisplayMember = "nama_pelanggan";
        }
        cmbPelanggan.SelectedIndex = -1;
    }

    private void ClearText()
    {
        CmdType = "Tambah";
        txtFaktur.Text = "";
        txtFaktur.ReadOnly = false;
        dtTanggalJual.Value = DateTime.Today;
        cmbPelanggan.SelectedIndex = -1;
        txtJumlahBarang.Value = "0";
        txtGrandTotal.Value = "0";
        txtDibayar.Value = "0";
        txtKembali.Value = "0";
        DGVListPenjualan.Rows.Clear();
        txtFaktur.Text = clsgen.RunNumberRec("Penjualan");
    }

    private bool Validasi()
    {
        if (cmbPelanggan.Text == "")
        {
            MessageBox.Show("Pelanggan masih kosong", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            cmbPelanggan.Focus();
            return false;
        }
        if (DGVListPenjualan.Rows.Count < 1)
        {
            MessageBox.Show("Data barang masih kosong", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            DGVListPenjualan.Focus();
            return false;
        }
        if (Convert.ToDecimal(txtDibayar.Value) < 0)
        {
            MessageBox.Show("Pembayaran masih kosong", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            txtDibayar.Focus();
        }
    }

```

```

        return false;
    }

    return true;
}

private void SaveData()
{
    try
    {
        string SqlSave = "";
        bool val = false;
        switch (CmdType)
        {
            case "Tambah":

                SqlSave = "Insert Penjualan
(no_faktur,kd_pelanggan,tgl,total_jual,total_qty,total_bayar,kembalian,user_id)
Values (" +
                " " + txtFaktur.Text + "," + txtKodePelanggan.Text + "," + " " +
dtTanggalJual.Value.ToString("yyyy-MM-dd") + " " +
                " " + Convert.ToDecimal(txtGrandTotal.Value) + "," + " " +
Convert.ToDecimal(txtJumlahBarang.Value) + " " +
                " " + Convert.ToDecimal(txtDibayar.Value) + "," + " " +
Convert.ToDecimal(txtKembali.Value) + " " + UserClass.Instance.UserId + ")";
                val = clscon.Esekusi(SqlSave);
                if (val == true)
                {
                    SqlSave = "";
                    for (int i = 0; i < DGVListPenjualan.RowCount; i++)
                    {

                        string kd_brg =
DGVListPenjualan.Rows[i].Cells["ColKodeBarang"].FormattedValue.ToString();
                        decimal Qty =
Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColJumlah"].Value);
                        decimal HargaBeli =
Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColHargaBeli"].Value);
                        decimal HargaJual =
Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColHargaJual"].Value);
                        decimal Subtotal =
Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColTotal"].Value);

                        SqlSave = " Insert Into PenjualanDetail (no_faktur,kd_brg,
harga_beli, harga_jual, jumlah, total) Values(" +
                        " " + txtFaktur.Text + "," + kd_brg + "," + HargaBeli + "," +
HargaJual + "," + Qty + "," + Subtotal + ")";

                        val = clscon.Esekusi(SqlSave);
                        if (val == true)
                        {
                            clsgen.updateStock(kd_brg, Qty * -1);
                        }
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
    if (val == true)
    {
        MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        clsgen.UpdateRunNumber("Penjualan", txtFaktur.Text);
        ClearText();
    }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void LoadDataBarang(string Filter)
{
    DGVListBarang.Rows.Clear();
    string SQLQuery = "Select * From Master_Barang " + Filter + " Order By
nama_brg,kd_brg";
    DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
    if (dataTable1.Rows.Count > 0)
    {
        int i = 1;
        foreach (DataRow dtr in dataTable1.Rows)
        {
            DataGridViewRow row =
(DataGridViewRow)DGVListBarang.Rows[0].Clone();

            row.Cells[0].Value = dtr["kd_brg"].ToString();
            row.Cells[1].Value = dtr["nama_brg"].ToString();
            row.Cells[2].Value = dtr["harga_beli"].ToString();
            row.Cells[3].Value = dtr["harga_jual"].ToString();
            DGVListBarang.Rows.Add(row);
            i += 1;
        }
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnBatal_Click(object sender, EventArgs e)
{
    ClearText();
}

```

```

    }

    private void btnTutup_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void txtPencarian_TextChanged(object sender, EventArgs e)
    {
        DGVLisBarang.Rows.Clear();
        if (txtPencarian.Text != "")
        {
            LoadDataBarang("Where kd_brg like '" + txtPencarian.Text + "%' OR
nama_brg like'" + txtPencarian.Text + "%'");
        }
    }

    private void cmbSupplier_SelectedIndexChanged(object sender, EventArgs e)
    {
        txtKodePelanggan.Text =
cmbPelanggan.GetItemText(cmbPelanggan.SelectedValue);
    }

    private void DGVLisBarang_DoubleClick(object sender, EventArgs e)
    {
        if (DGVLisBarang.Rows.Count > 1)
        {
            int indx = DGVLisBarang.CurrentCell.RowIndex;

            for (int vloop = 0; vloop < DGVLisPenjualan.Rows.Count; vloop++)
            {
                string KodeItems =
DGVLisPenjualan.Rows[vloop].Cells["ColKodeBarang"].FormattedValue.ToString();
                if (KodeItems ==
DGVLisBarang.Rows[indx].Cells[0].FormattedValue.ToString())
                {
                    MessageBox.Show("Kode barang sudah ada", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

                    return;
                }
            }

            DGVLisPenjualan.Rows.Add();
            DGVLisPenjualan.Rows[DGVLisPenjualan.Rows.Count -
1].Cells["ColKodeBarang"].Value =
DGVLisBarang.Rows[indx].Cells[0].FormattedValue.ToString();
            DGVLisPenjualan.Rows[DGVLisPenjualan.Rows.Count -
1].Cells["ColNamaBarang"].Value =
DGVLisBarang.Rows[indx].Cells[1].FormattedValue.ToString();

```

```

        DGVListPenjualan.Rows[DGVListPenjualan.Rows.Count -
1].Cells["ColHargaBeli"].Value =
DGVListBarang.Rows[indx].Cells[2].FormattedValue.ToString();
        DGVListPenjualan.Rows[DGVListPenjualan.Rows.Count -
1].Cells["ColHargaJual"].Value =
DGVListBarang.Rows[indx].Cells[3].FormattedValue.ToString();
    }
}

private void DGVListPembelian_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 4)
    {
        string Kodebrg = DGVListPenjualan.Rows[DGVListPenjualan.Rows.Count -
1].Cells["ColKodeBarang"].FormattedValue.ToString();
        decimal sisstock = clsgen.LastStock(Kodebrg);
        decimal Qty =
Convert.ToDecimal(DGVListPenjualan.Rows[e.RowIndex].Cells["ColJumlah"].Value);
        if (Qty > sisstock)
        {
            MessageBox.Show("Stock tidak mencukupi. Sisa stock " + sisstock,
"Konfirmasi", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            if (sisstock < 1)
            {
                this.DGVListPenjualan.Rows[e.RowIndex].Cells["ColJumlah"].Value =
0;
            }
            else if (sisstock > 0)
            {
                this.DGVListPenjualan.Rows[e.RowIndex].Cells["ColJumlah"].Value =
string.Format("{0:##0}", sisstock);
            }
            return;
        }
        decimal HargaBeli =
Convert.ToDecimal(DGVListPenjualan.Rows[e.RowIndex].Cells["ColHargaJual"].Valu
e);
        decimal Subtotal = Qty * HargaBeli;

        this.DGVListPenjualan.Rows[e.RowIndex].Cells["ColTotal"].Value =
string.Format("{0:##0}", Subtotal);
        GrandTotalQty();
    }
}

private void DGVListPembelian_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
{
    if (e.ColumnIndex == 3) // 1 should be your column index
    {
        int i;

        if (!int.TryParse(Convert.ToString(e.FormattedValue), out i))
        {
            e.Cancel = true;
        }
    }
}

```

```

        MessageBox.Show("Pastikan diisi angka", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        // the input is numeric
    }
}

private void GrandTotalQty()
{
    try
    {
        decimal Total = 0;
        decimal TotalItems = 0;
        if (DGVListPenjualan.RowCount > 0)
        {
            for (int i = 0; i < DGVListPenjualan.RowCount; i++)
            {
                Total +=
                Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColTotal"].FormattedValue);
                TotalItems +=
                Convert.ToDecimal(DGVListPenjualan.Rows[i].Cells["ColJumlah"].FormattedValue);
            }
            txtJumlahBarang.Text = TotalItems.ToString("N0");
            txtGrandTotal.Text = Total.ToString("N0");
        }
        catch { }
    }

    private void txtDibayar_TextChanged(object sender, EventArgs e)
    {
        if (txtDibayar.Value == "") txtDibayar.Value = "0";
        txtKembali.Value = "0";
        if ( Convert.ToDecimal(txtDibayar.Value) >
        Convert.ToDecimal(txtGrandTotal.Value))
        {
            decimal totalKembali = (Convert.ToDecimal(txtDibayar.Value)-
            Convert.ToDecimal(txtGrandTotal.Value));
            if (totalKembali < 0 ) totalKembali = totalKembali * -1;
            txtKembali.Value = totalKembali.ToString("N0");
        }
    }

    private void DGVListBarang_CellContentClick(object sender,
    DataGridViewCellEventArgs e)

    {
    }
}

```

11. Form Pembelian

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmReturPembelian : Form
    {
        ConnectionClass clscon;
        ClassGeneral clsgen;

        string CmdType = "Tambah";
        public FrmReturPembelian()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
            clsgen = new ClassGeneral();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            txtFaktur.Text = clsgen.RunNumberRec("ReturPembelian");
        }

        private void ClearText()
        {
            CmdType = "Tambah";
            txtFaktur.Text = "";
            txtFaktur.ReadOnly = false;
            dtTanggal.Value = DateTime.Today;
            txtFakturPembelian.Text = "";

            txtSupplier.Text = "";
            txtKodeSupplier.Text = "";
            txtJumlahBarang.Value = "0";
            txtGrandTotal.Value = "0";
            DGVLstBarangPembelian.Rows.Clear();
            DGVLstPembelian.Rows.Clear();
            txtFaktur.Text = clsgen.RunNumberRec("ReturPembelian");
        }

        private bool Validasi()
        {
            if (DGVLstPembelian.Rows.Count < 1)

```

```

        {
            MessageBox.Show("Data barang masih kosong", "Konfirmasi",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            DGVListPembelian.Focus();
            return false;
        }

        return true;
    }

    private void SaveData()
    {
        try
        {
            string SqlSave = "";
            bool val = false;
            switch (CmdType)
            {
                case "Tambah":

                    for (int i = 0; i < DGVListPembelian.RowCount; i++)
                    {

                        string kd_brg =
                            DGVListPembelian.Rows[i].Cells["ColKodeBarang"].FormattedValue.ToString();
                        decimal Qty =
                            Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].Value);
                        decimal Harga =
                            Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColHargaBeli"].Value);
                        decimal Subtotal =
                            Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].Value);

                        SqlSave = " Insert Into ReturPembelian
(no_faktur,no_faktur_pemb,tgl,kd_brg, kd_supp,jml,
harga_retur,total,user_id,retur_type) Values(" +
                            " " + txtFaktur.Text + "," + txtFakturPembelian.Text + "," +
                            dtTanggal.Value.ToString("yyyy-MM-dd") + "," + kd_brg + "," + txtKodeSupplier.Text
                            + "," +
                            "" + Qty + "," + Harga + "," + Subtotal + "," +
                            UserClass.Instance.UserId + "','Konsiyasi' " );

                        val = clscon.Esekusi(SqlSave);
                        if (val == true)
                        {
                            clsgen.updateStock(kd_brg, Qty);

                            SqlSave = "Update PembelianKonsiyasi Set jmlretur =
IsNull(jmlretur,0) + " + Qty + " Where no_faktur = " + txtFakturPembelian.Text + " And
kd_brg = " + kd_brg + " ";
                            clscon.Esekusi(SqlSave);
                        }
                    }

                break;
            }
        }
    }

```

```

    }
    if (val == true)
    {
        MessageBox.Show("Data berhasil disimpan ", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        clsgen.UpdateRunNumber("ReturPembelian", txtFaktur.Text);
        ClearText();
    }

    }
    catch (Exception e)
    {
        MessageBox.Show("Gagal simpan " + e.Message, "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (Validasi() == true)
    {
        SaveData();
    }
}

private void btnBatal_Click(object sender, EventArgs e)
{
    ClearText();
}

private void btnTutup_Click(object sender, EventArgs e)
{
    this.Close();
}

private void DGVListPembelian_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 3)
    {
        decimal Qty =
        Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColJumlah"].Value)
        ;
        decimal QtyPembelian =
        Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColJmlPembelian"].
        Value);

        if (Qty > QtyPembelian )
        {

```

```

        MessageBox.Show("Quantity retur lebih besa dari pembelian",
"Konfirmasi", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        DGVListPembelian.Rows[e.RowIndex].Cells["ColJumlah"].Value = "";
        DGVListPembelian.Rows[e.RowIndex].Cells["ColTotal"].Value = "";
        GrandTotalQty();
        return;
    }

    decimal HargaBeli =
Convert.ToDecimal(DGVListPembelian.Rows[e.RowIndex].Cells["ColHargaBeli"].Value);
    decimal Subtotal = Qty * HargaBeli;

    this.DGVListPembelian.Rows[e.RowIndex].Cells["ColTotal"].Value =
string.Format("{0:#,###0}", Subtotal);
    GrandTotalQty();
}

private void DGVListPembelian_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
{
    if (e.ColumnIndex == 3) // 1 should be your column index
    {
        int i;

        if (!int.TryParse(Convert.ToString(e.FormattedValue), out i))
        {
            e.Cancel = true;
            MessageBox.Show("Pastikan diisi angka", "Konfirmasi",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
        else
        {
            // the input is numeric
        }
    }
}

private void GrandTotalQty()
{
    try
    {
        decimal Total = 0;
        decimal TotalItems = 0;
        if (DGVListPembelian.RowCount > 0)
        {
            for (int i = 0; i < DGVListPembelian.RowCount; i++)
            {
                Total +=
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColTotal"].FormattedValue);
            }
        }
    }
    catch { }
}

```

```

        TotalItems +=
Convert.ToDecimal(DGVListPembelian.Rows[i].Cells["ColJumlah"].FormattedValue);
    }
    }
    txtJumlahBarang.Text = TotalItems.ToString();
    txtGrandTotal.Text = Total.ToString();
}
catch { }

}

private void txtFakturPembelian_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyData == Keys.Enter && txtFakturPembelian.Text != "")
    {
        LoadDataPembelian(txtFakturPembelian.Text);
    }
}

private void txtFakturPembelian_TextChanged(object sender, EventArgs e)
{
    if (txtFakturPembelian.Text != "")
    {
        LoadDataPembelian(txtFakturPembelian.Text);
    }
}

private void LoadDataPembelian(string FakturPembelian)
{
    DGVListBarangPembelian.Rows.Clear();
    //ClearText();
    string SQLQuery = "Select * From v_pembelian_for_retur Where no_faktur = "
    + FakturPembelian + " Order By nama_brg,kd_brg";
    DataTable dataTable1 = clscon.GetDataTable(SQLQuery);
    if (dataTable1.Rows.Count > 0)
    {
        txtKodeSupplier.Text = dataTable1.Rows[0]["kd_supp"].ToString();
        txtSupplier.Text = dataTable1.Rows[0]["nama_supp"].ToString();
        int i = 1;
        foreach (DataRow dtr in dataTable1.Rows)
        {
            DataGridViewRow row =
(DataGridViewRow)DGVListBarangPembelian.Rows[0].Clone();

            row.Cells[0].Value = dtr["kd_brg"].ToString();
            row.Cells[1].Value = dtr["nama_brg"].ToString();
            row.Cells[2].Value = dtr["harga_beli"].ToString();
            row.Cells[3].Value = dtr["jml"].ToString();
            row.Cells[4].Value = dtr["total_beli"].ToString();
            DGVListBarangPembelian.Rows.Add(row);
            i += 1;
        }
    }
}

```

```

private void btnTambahRetur_Click(object sender, EventArgs e)
{
    tambahkankeDaftarRetur();
}

private void btnHapus_Click(object sender, EventArgs e)
{
    if (DGVListPembelian.Rows.Count > 0)
    {
        DGVListPembelian.Rows.RemoveAt(DGVListPembelian.CurrentCell.RowIndex);
        GrandTotalQty();
    }
}

private void tambahkankeDaftarRetur()
{
    if (DGVListBarangPembelian.Rows.Count > 1)
    {
        int indx = DGVListBarangPembelian.CurrentCell.RowIndex;

        for (int vloop = 0; vloop < DGVListPembelian.Rows.Count; vloop++)
        {
            string KodeItems =
DGVListPembelian.Rows[vloop].Cells["ColKodeBarang"].FormattedValue.ToString();
            if (KodeItems ==
DGVListBarangPembelian.Rows[indx].Cells[0].FormattedValue.ToString())
            {
                MessageBox.Show("Kode barang sudah ada", "Konfirmasi",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

                return;
            }
        }

        DGVListPembelian.Rows.Add();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColKodeBarang"].Value =
DGVListBarangPembelian.Rows[indx].Cells[0].FormattedValue.ToString();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColNamaBarang"].Value =
DGVListBarangPembelian.Rows[indx].Cells[1].FormattedValue.ToString();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColHargaBeli"].Value =
DGVListBarangPembelian.Rows[indx].Cells[2].FormattedValue.ToString();
        DGVListPembelian.Rows[DGVListPembelian.Rows.Count -
1].Cells["ColJmlPembelian"].Value =
DGVListBarangPembelian.Rows[indx].Cells[3].FormattedValue.ToString();
    }
}

private void btnCariFakturPembelian_Click(object sender, EventArgs e)
{
    FrmLookupPembelian frm1 = new FrmLookupPembelian();
    frm1.ShowDialog();
    if (frm1.IsOke == "Selected" && frm1.DGVPEmbelian.RowCount > 1)

```

```

        {
            string NoFakturPembelian =
frm1.DGVPembelian.Rows[frm1.DGVPembelian.CurrentRowIndex].Cells["ColNo
Faktur"].FormattedValue.ToString();
            txtFakurPembelian.Text = NoFakturPembelian;
            LoadDataPembelian(txtFakurPembelian.Text);
        }
    }
}

```

12. Class Koneksi

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data;
using System.Windows.Forms;

namespace RetailProgram.Classes
{
    public class ConnectionClass
    {

        public static string ServerName = "DESKTOP-MNTCCD1\\SQLEXPRESS2012";
        public static string Database = "db_retail_x";
        public static string UserName = "sa";
        public static string Password = "Admin123";

        //public static string ServerName = "DESKTOP-1TFQLQJ\\MSSQLSERVER2012";
        //public static string Database = "db_retail_x";
        //public static string UserName = "sa";
        //public static string Password = "ifmuji";

        public SqlConnection con = new SqlConnection("Data Source=" + ServerName +
";Initial Catalog=" + Database + ";User ID=" + UserName + ";Password=" + Password +
""");

        public Boolean Esekusi(string SqlCommand)
        {
            Koneksi();
            if (con.State == ConnectionState.Closed) con.Open();
            SqlCommand sqlCmnd = new SqlCommand(SqlCommand, con);
            var val = sqlCmnd.ExecuteNonQuery();
            sqlCmnd.Dispose();
            con.Close();
            if (val > 0)
            {
                return true;
            }
            else
            {
                return false
            }
        }
    }
}

```

```

    }
    }

    public void Koneksi()
    {
        if (con.State == ConnectionState.Open)
        {
            con.Close();
        }
        con.Open();
    }
    public DataTable GetDataTable(string SqlSelect)
    {
        Koneksi();
        SqlDataAdapter dataAd = new SqlDataAdapter(SqlSelect, con);
        DataTable dt = new DataTable();
        dataAd.Fill(dt);
        return dt;
    }
    public DataSet GetDataDataset(string SqlSelect)
    {
        Koneksi();
        SqlDataAdapter dataAd = new SqlDataAdapter(SqlSelect, con);

        DataSet dt = new DataSet();
        dataAd.Fill(dt);
        return dt;
    }
}
}

```

13. Class General

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using RetailProgram.Classes;
using System.Data;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram.Classes
{

    public class ClassGeneral
    {
        SqlCommand Cmd;

        ConnectionClass clsConn = new ConnectionClass();
    }
}

```

```

public string RunNumberRec(string Trans)
{
    clsConn.Koneksi();
    string LastNumber = "";
    Cmd = new SqlCommand("Select notrans From NoTransTemp Where Jenis = 
" + Trans + " And tahun = " + DateTime.Now.Year + " And Bulan = " + 
DateTime.Now.ToString("MM") + "", clsConn.con);
    SqlDataReader dtreader = Cmd.ExecuteReader();
    dtreader.Read();
    if (dtreader.HasRows)
    {
        if (Trans == "PembelianNonKonsiyasi")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(4, 3)) + 
1).ToString();
        }
        else if (Trans == "PembelianKonsiyasi")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(4, 3)) + 
1).ToString();
        }
        else if (Trans == "Penjualan")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(4, 3)) + 
1).ToString();
        }
        else if (Trans == "ReturPembelian")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(4, 3)) + 
1).ToString();
        }
        else if (Trans == "Kode_Supplier")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(0, 2)) + 
1).ToString();
        }
        else if (Trans == "Kode_Pelanggan")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(0, 3)) + 
1).ToString();
        }
        else if (Trans == "Kode_Barang")
        {
            LastNumber = (Convert.ToInt32(dtreader[0].ToString().Substring(0, 2)) + 
1).ToString();
        }
        if (Trans == "Kode_Supplier" || Trans == "Kode_Barang")
        {
            if (LastNumber.Length == 1)
            {
                LastNumber = "0" + LastNumber;
            }
        }
        else
        {
            if (LastNumber.Length == 1)

```

```

        {
            LastNumber = "00" + LastNumber;
        }
        else if (LastNumber.Length == 2)
        {
            LastNumber = "0" + LastNumber;
        }

        else if (LastNumber.Length > 3)
        {
            LastNumber = "Over load Run Number";
            return LastNumber;
        }
    }

}
else
{
    if (Trans == "Kode_Supplier" || Trans == "Kode_Barang")
    {
        LastNumber = "01";
    }
    else
    {
        LastNumber = "001";
    }
}

if (Trans == "PembelianNonKonsiyasi")
{
    LastNumber = DateTime.Now.ToString("yy") +
DateTime.Now.ToString("MM") + LastNumber;
}
else if (Trans == "PembelianKonsiyasi")
{
    LastNumber = DateTime.Now.ToString("yy") +
DateTime.Now.ToString("MM") + LastNumber;
}
else if (Trans == "Penjualan")
{
    LastNumber = DateTime.Now.ToString("yy") +
DateTime.Now.ToString("MM") + LastNumber;
}
else if (Trans == "ReturPembelian")
{
    LastNumber = DateTime.Now.ToString("yy") +
DateTime.Now.ToString("MM") + LastNumber;
}

dtreader.Close();
return LastNumber;
}

public void UpdateRunNumber(string Trans, string LastNumber)
{

```

```

        try
        {
            string Sql = "Select * From NoTransTemp Where Jenis='" + Trans + "' And
tahun = " + DateTime.Now.Year + " And bulan = '" + DateTime.Now.ToString("MM") +
"'"',

            clsConn.Koneksi();
            Cmd = new SqlCommand(Sql, clsConn.con);
            SqlDataReader dtreader = Cmd.ExecuteReader();
            dtreader.Read();
            Cmd.Dispose();
            if (dtreader.HasRows)
            {
                dtreader.Close();
                Sql = "Update NoTransTemp Set notrans = '" + LastNumber + "' Where
Jenis='" + Trans + "' And tahun = " + DateTime.Now.Year + " And bulan = '" +
DateTime.Now.ToString("MM") + "'";
                clsConn.Esekusi(Sql);
            }
            else
            {
                dtreader.Close();
                Sql = "Insert Into NoTransTemp (Jenis,notrans,tahun,bulan) Values(" +
"'" + Trans + "','" + LastNumber + "', " + DateTime.Now.Year + ", '" +
DateTime.Now.ToString("MM") + "')";
                clsConn.Esekusi(Sql);
            }
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message + ". Data gagal disimpan", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }

    public void updateStock(string KodeBarang,decimal Qty)
    {
        string Sql = "Update Master_Barang Set stock = isnull(stock,0) +" + Qty + "
Where Kd_brg = '" + KodeBarang + "'";
        clsConn.Esekusi(Sql);
    }

    public decimal LastStock(string KodeBarang)
    {
        decimal stock = 0;
        string Sql = "Select Isnull(stock,0) As Stock From Master_Barang Where
Kd_brg = '" + KodeBarang + "'";
        DataTable dt1 = clsConn.GetDataTable(Sql);
        if (dt1.Rows.Count > 0)
        {
            stock = Convert.ToDecimal(dt1.Rows[0]["Stock"]);
        }
        return stock;
    }
}
}

```

14. Class User

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using RetailProgram.Classes;
using System.Data;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram.Classes
{

    public class UserClass
    {
        SqlCommand Cmd;

        ConnectionClass clsConn = new ConnectionClass();

        public string UserId = "";
        public string UserName = "";
        private UserClass() { }
        private static UserClass instance = null;
        public static UserClass Instance
        {
            get
            {
                if (instance == null)
                {
                    instance = new UserClass();
                }

                return instance;
            }
        }
    }
}

```

15. Form Print Pembelian Konsinyasi

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

```

```

namespace RetailProgram
{
    public partial class FrmPrintPembelian : Form
    {
        ConnectionClass clscon;
        public FrmPrintPembelian()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void FrmPrintPembelian_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
            'db_retail_xDataSet.v_pembelian_konsiyasi' table. You can move, or remove it, as
            needed.

        }

        private void bntPrint_Click(object sender, EventArgs e)
        {
            Cursor.Current = Cursors.WaitCursor;

            v_pembelian_konsiyasiBindingSource.Filter = " tgl_titip >= " +
            StartDate.Value.ToString("yyyy-MM-dd") + " And tgl_titip <= " +
            EndDate.Value.ToString("yyyy-MM-dd") + """";

            this.v_pembelian_konsiyasiTableAdapter.Fill(this.db_retail_xDataSet.v_pembelian_k
            onsiyasi);
            ReportParameter[] parameters = new ReportParameter[2];
            parameters[0] = new ReportParameter("StartDate",
            StartDate.Value.ToString("dd-MMM-yyyy"), false);
            parameters[1] = new ReportParameter("EndDate",
            EndDate.Value.ToString("dd-MMM-yyyy"), false);
            this.reportViewer1.LocalReport.SetParameters(parameters);
            this.reportViewer1.RefreshReport();
            Cursor.Current = Cursors.Default;
        }
    }
}

```

16. Form Print Pembelian Nonkonsinyasi

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

```

```

namespace RetailProgram
{
    public partial class FrmPrintPembelianNon : Form
    {
        ConnectionClass clscon;
        public FrmPrintPembelianNon()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void bntPrint_Click(object sender, EventArgs e)
        {
            Cursor.Current = Cursors.WaitCursor;

            vpembeliannonkonsiyasiBindingSource.Filter = " tgl_beli >= " +
            StartDate.Value.ToString("yyyy-MM-dd") + " And tgl_beli <= " +
            EndDate.Value.ToString("yyyy-MM-dd") + """";

            this.v_pembelian_non_konsiyasiTableAdapter.Fill(this.db_retail_xDataSet.v_pembeli
            an_non_konsiyasi);
            ReportParameter[] parameters = new ReportParameter[2];
            parameters[0] = new ReportParameter("StartDate",
            StartDate.Value.ToString("dd-MMM-yyyy"), false);
            parameters[1] = new ReportParameter("EndDate",
            EndDate.Value.ToString("dd-MMM-yyyy"), false);
            this.reportViewer1.LocalReport.SetParameters(parameters);
            this.reportViewer1.RefreshReport();
            Cursor.Current = Cursors.Default;
        }
    }
}

```

17. Form Print Penjualan

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmPrintPenjualan : Form
    {
        ConnectionClass clscon;
        public FrmPrintPenjualan()
        {
            InitializeComponent();

```

```

        clscon = new ConnectionClass();
    }

    private void FrmPrintPenjualan_Load(object sender, EventArgs e)
    {

    }

    private void bntPrint_Click(object sender, EventArgs e)
    {
        Cursor.Current = Cursors.WaitCursor;
        v_penjualanBindingSource.Filter = " tgl >= " +
        StartDate.Value.ToString("yyyy-MM-dd") + " And tgl <= " +
        EndDate.Value.ToString("yyyy-MM-dd") + """";
        db_retail_xDataSet.EnforceConstraints = false;
        this.v_penjualanTableAdapter.Fill(this.db_retail_xDataSet.v_penjualan);

        ReportParameter[] parameters = new ReportParameter[2];
        parameters[0] = new ReportParameter("StartDate",
        StartDate.Value.ToString("dd-MMM-yyyy"), false);
        parameters[1] = new ReportParameter("EndDate",
        EndDate.Value.ToString("dd-MMM-yyyy"), false);
        this.reportViewer1.LocalReport.SetParameters(parameters);
        this.reportViewer1.RefreshReport();
        Cursor.Current = Cursors.Default;
    }

    }
}

```

18. Form Print Stock Konsinyasi

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmPrintStockKonsiyasi : Form
    {
        ConnectionClass clscon;
        public FrmPrintStockKonsiyasi()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }
    }
}

```

```

private void FrmPrintStockKonsiyasi_Load(object sender, EventArgs e)
{
    this.bntPrint_Click(sender, e);

}

private void bntPrint_Click(object sender, EventArgs e)
{
    Cursor.Current = Cursors.WaitCursor;
    Master_BarangBindingSource.Filter = " Isnull([status], 'Non Konsiyasi') =
'Konsiyasi'";

    this.Master_BarangTableAdapter.Fill(this.db_retail_xDataSet.Master_Barang);

    ReportParameter[] parameters = new ReportParameter[1];
    parameters[0] = new ReportParameter("Status", "Konsiyasi", false);
    this.reportViewer1.LocalReport.SetParameters(parameters);

    this.reportViewer1.RefreshReport();
    Cursor.Current = Cursors.Default;
}
}
}

```

19. Form Print Stock Non Konsinyasi

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using RetailProgram.Classes;

namespace RetailProgram
{
    public partial class FrmPrintStockNon : Form
    {
        ConnectionClass clscon;
        public FrmPrintStockNon()
        {
            InitializeComponent();
            clscon = new ConnectionClass();
        }

        private void FrmPrintStockNon_Load(object sender, EventArgs e)
        {
            this.bntPrint_Click(sender, e);
        }
    }
}

```

```
private void bntPrint_Click(object sender, EventArgs e)
{
    Cursor.Current = Cursors.WaitCursor;
    Master_BarangBindingSource.Filter = " Isnull([status], 'Non Konsiyasi') = 'Non  
Konsiyasi'";

    this.Master_BarangTableAdapter.Fill(this.db_retail_xDataSet.Master_Barang);

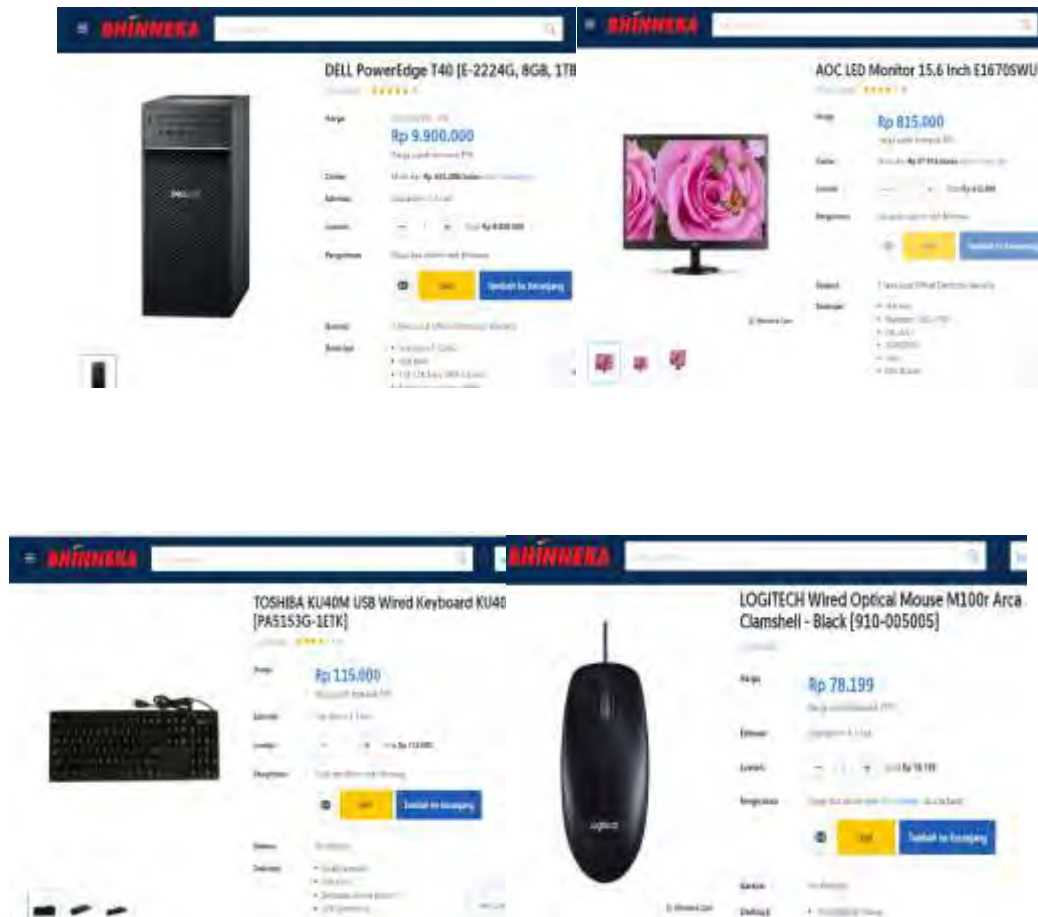
    ReportParameter[] parameters = new ReportParameter[1];
    parameters[0] = new ReportParameter("Status", "Non Konsiyasi", false);
    this.reportViewer1.LocalReport.SetParameters(parameters);

    this.reportViewer1.RefreshReport();
    Cursor.Current = Cursors.Default;
}

}
```

LAMPIRAN 4 RINCIAN KEBUTUHAN BIAYA

1. Komputer Server



2. Printer



3. Komputer Client

m.tokopedia.com

✕ Deskripsi produk

Deskripsi



**PC KOMPUTER CORE I3 RAM 4GB
FULL SET LED 19 INCI KASIR PPOB
DESAIN**
Rp 2.400.000

PAKET FULLSET 2 LED 19 IN

- _Proci Core i3 2xxx 3.1ghz/Ram 4GB/Hardisk 500GB/Case Optimax
- _LED Monitor 19 inci BARU
- _Keyboard Mouse PowerUP USB
- _Wifi Adapter

HARGA PROMO : 2,650.000

4. Hub

m.tokopedia.com

✕ Deskripsi produk

Deskripsi



Switch HUB TP LINK 8 Port TL-SF1008D 10/100Mbps Network Lan Internet
Rp 72.000

8-Port 10/100Mbps Desktop Switch
TL-SF1008D

- * 8 port RJ45 10/100Mbps Auto-Negotiation, Mendukung Auto MDI / MDIX
- * Green Ethernet teknologi menghemat daya hingga 70%
- * IEEE 802.3x flow control menyediakan transfer data yang handal
- * Casing plastik, didesain untuk pemasangan desktop / pada dinding
- * Plug and play, tidak ada konfigurasi yang diperlukan

5. Microsoft Windows 10



LAMPIRAN 5 MODEL KETERHUBUNGAN ANTAR TABEL

