

TUGAS AKHIR
RANCANG BANGUN SISTEM INFORMASI PENJUALAN BERBASIS
CLOUD BAGI USAHA KECIL DAN MENENGAH DI INDONESIA
(STUDI KASUS PADA TOKO “TOKO BESAR”)



Nama : Stevanus Christopel M. A.H

NIM : 09.41010.0260

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

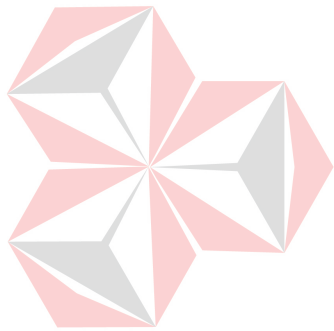
SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA

2013

**RANCANG BANGUN SISTEM INFORMASI PENJUALAN BERBASIS
CLOUD BAGI USAHA KECIL DAN MENENGAH DI INDONESIA
(STUDI KASUS PADA TOKO “TOKO BESAR”)**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer**



UNIVERSITAS
Dinamika

Oleh:

Nama : Stevanus Christopel M. A. H

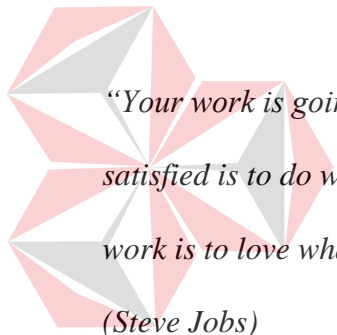
NIM : 09.41010.0260

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

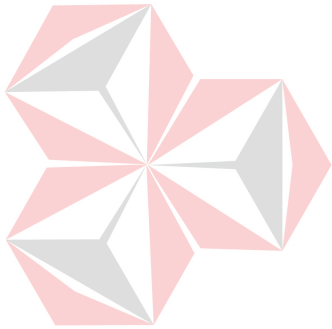
2013



“Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do.”

(Steve Jobs)

UNIVERSITAS
Dinamika



Kupersembahkan kepada

Tuhan Yang Maha Esa

Ayah, Ibu dan semua keluarga tercinta

Semua orang yang telah menginspirasi

UNIVERSITAS
Dinamika

TUGAS AKHIR
RANCANG BANGUN SISTEM INFORMASI PENJUALAN BERBASIS
CLOUD BAGI USAHA KECIL DAN MENENGAH DI INDONESIA
(STUDI KASUS PADA TOKO “TOKO BESAR”)

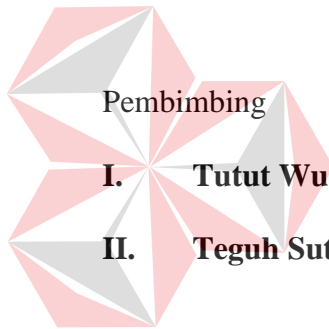
dipersiapkan dan disusun oleh

Stevanus Christopel M. A. H

NIM : 09.41010.0260

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji
pada : Maret 2013

Susunan Dewan Penguji



Pembimbing

I. Tutut Wuriyanto, M.Kom

II. Teguh Sutanto, M. Kom., MCP

Penguji

I. Dr. Jusak

II. Arifin Puji Widodo, S.E., M.SA.

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom., M.Eng, OCA
Pembantu Ketua Bidang Akademik

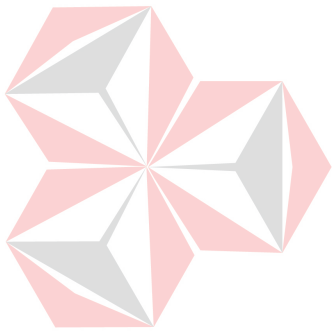
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN TEKNIK KOMPUTER SURABAYA

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam tugas akhir ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya.

Apabila dikemudian hari ditemukan adanya tindak plagiat pada karya tugas akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 27 Maret 2013



UNIVERSITAS
Dinamika
Stevanus Christopel M. A. H.
NIM : 09.41010.0260

ABSTRAK

Usaha Kecil dan Menengah (UKM) memiliki pengaruh yang besar bagi perekonomian dan pemberdayaan tenaga kerja di Indonesia karena menghasilkan PDB sebesar 56,5% dan menyerap 97% tenaga kerja (Yoga; 2011). Walaupun demikian, perkembangan UKM masih terhambat oleh minimnya pengetahuan dan informasi pasar. Saat ini, informasi yang diperoleh UKM hanya jumlah penjualan dan pendapatan berdasarkan rekap nota transaksi. Padahal UKM memerlukan informasi tren penjualan untuk mengetahui produk yang diminati, informasi stok terkini untuk menentukan waktu order ke *supplier* serta *history* transaksi pelanggan dan *supplier*. Informasi tersebut diperlukan pihak manajemen untuk mendukung pengambilan keputusan.

Dari permasalahan di atas dibuat sistem informasi penjualan untuk mencatat transaksi penjualan dan menghasilkan laporan penjualan yang dibutuhkan manajemen. Pengguna juga dapat memilih metode persediaan *FIFO*, *LIFO* atau *Average Costing*. Sistem informasi penjualan dibangun dengan teknologi *cloud computing* sehingga UKM tidak perlu investasi besar untuk infrastruktur dan pengelolaannya. Berdasarkan hasil uji coba didapatkan bahwa sistem dapat menghasilkan laporan tren penjualan, stok terkini, *history* transaksi *supplier* dan pelanggan sesuai dengan kebutuhan manajemen serta dapat diakses kapanpun dan dimanapun melalui jaringan internet. Sistem juga dapat digunakan sebagai media promosi barang di jejaring sosial Facebook berdasarkan data terkini.

Kata kunci: Sistem Informasi Penjualan, Usaha Kecil dan Menengah, Komputasi Awan

KATA PENGANTAR

Pertama-tama penulis panjatkan puji syukur ke hadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya akhirnya penulis dapat menyelesaikan laporan Tugas Akhir ini dengan sebaik-baiknya. Penulis membuat laporan Tugas Akhir yang berjudul Rancang Bangun Sistem Informasi Penjualan Berbasis *Cloud* bagi Usaha Kecil dan Menengah di Indonesia untuk membantu semua usaha kecil dan menengah agar memiliki pengetahuan dan informasi yang lebih banyak mengenai pasar mereka. Informasi-informasi ini diharapkan dapat membantu pihak manajemen UKM sebagai pendukung pengambilan keputusan-keputusan bisnis yang akan dilakukan. Usaha Kecil dan Menengah merupakan pelaku ekonomi utama di Indonesia dan memiliki pengaruh yang besar terhadap kemajuan dan pemberdayaan tenaga kerja di Indonesia. Dengan semakin majunya UKM yang ada di Indonesia, diharapkan dapat sejalan dengan perkembangan bangsa ke arah yang lebih baik lagi.

Dalam pelaksanaan Tugas Akhir maupun pembuatan laporan Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak. Oleh karena itu, penulis juga ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu melimpahkan berkat dan rahmat sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Orang tua beserta semua keluarga tercinta yang selalu memberikan motivasi, inspirasi dan kisah-kisah kehidupan yang selalu memberi semangat kepada penulis.

3. STIKOM Surabaya beserta semua *civitas academica* yang telah mendukung dan menginspirasi penulis dalam menyelesaikan tugas akhir ini.
4. Bapak Tutut Wuriyanto, M.Kom dan bapak Teguh Sutanto, M.Kom., MCP selaku dosen pembimbing I dan II yang telah membimbing dan mengarahkan penulis dalam menyelesaikan tugas akhir ini.
5. Bapak Dr. Jusak dan bapak Arifin Puji Widodo, S.E., M.SA. selaku dosen penguji I dan II yang telah memberi banyak masukan kritik dan saran yang membangun bagi tugas akhir ini.
6. Saudara Afiful Hashif dan Zainul Mas'adi sebagai kolega yang banyak membantu penulis dalam pemahaman teknologi-teknologi yang digunakan dalam tugas akhir ini.
7. Semua kolega yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam laporan Tugas Akhir ini masih banyak kekurangan. Oleh karena itu penulis memohon maaf apabila terdapat kesalahan dan selalu menerima kritik, saran dan masukan dari pembaca. Semoga karya penulis dapat memberikan manfaat positif bagi para pembaca.

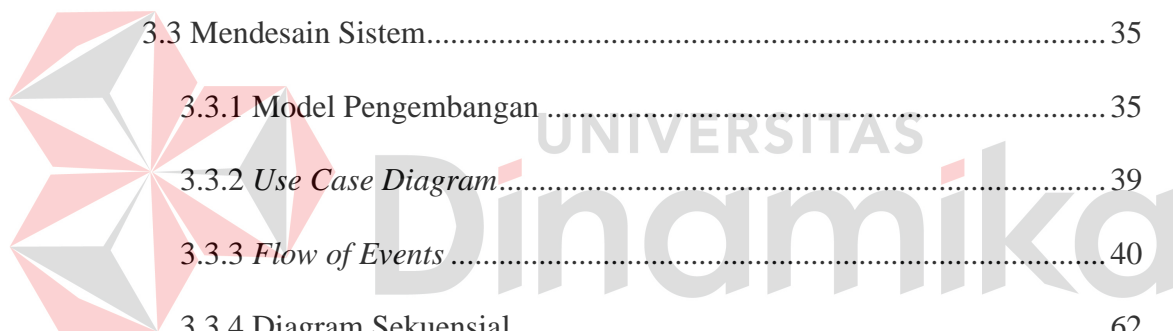
Surabaya, Maret 2013

Penulis

DAFTAR ISI

ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	4
1.3 Batasan Masalah.....	5
1.4 Tujuan.....	5
1.5 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Penjualan	8
2.2 Sistem Informasi.....	10
2.3 Sistem Informasi Penjualan.....	11
2.4 Usaha Kecil dan Menengah (UKM).....	12
2.5 Metode Pencatatan Persediaan Perpetual	12
2.5.1 Sistem Perpetual – Rata-rata bergerak.....	13
2.5.2 Sistem Perpetual – FIFO.....	14
2.5.3 Sistem Perpetual – LIFO	15
2.6 Metode Perhitungan Bunga Kredit <i>Flat Rate</i>	16
2.7 Komputasi Awan (<i>Cloud Computing</i>).....	17

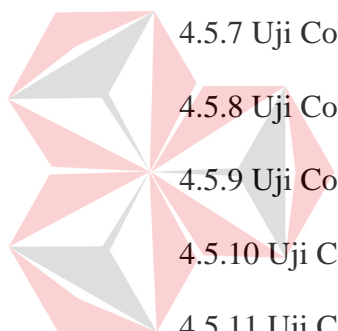
2.8 Windows Azure	20
2.9 <i>Virtual Private Server (VPS)</i>	20
2.10 ASP.NET MVC (<i>Model, View, Controller</i>)	21
2.11 Azure SQL Database	22
2.12 <i>Black Box Testing</i>	23
BAB III ANALISIS DAN PERANCANGAN SISTEM	24
3.1 Identifikasi Masalah	24
3.2 Analisis Sistem	26
3.2.1 <i>Activity Diagram</i>	27
3.2.2 Hasil Analisis	33
3.3 Mendesain Sistem	35
3.3.1 Model Pengembangan	35
3.3.2 <i>Use Case Diagram</i>	39
3.3.3 <i>Flow of Events</i>	40
3.3.4 Diagram Sekuensial	62
3.3.5 <i>Class Diagram</i>	77
3.3.6 <i>Component Diagram</i>	80
3.3.7 <i>Deployment Diagram</i>	80
3.3.8 Rencana Penggunaan Infrastruktur <i>Cloud</i>	81
3.3.9 <i>Physical Data Model</i>	83
3.3.10 Desain <i>User Interface</i>	85
3.3.11 Desain <i>Input/Output</i>	98
3.3.12 Rancangan Pengujian dan Evaluasi Sistem	109
BAB IV IMPLEMENTASI DAN EVALUASI	121



4.1	Kebutuhan Sistem.....	121
4.1.1	Kebutuhan <i>Server</i>	121
4.1.2	Kebutuhan Perangkat Lunak	122
4.2	Pembuatan Program	122
4.3	Implementasi Sistem	123
4.3.1	Registrasi Akun Windows Azure	123
4.3.2	Konfigurasi <i>Virtual Machine</i>	126
4.3.3	Konfigurasi Azure SQL <i>Database</i>	128
4.4	Proses Bisnis.....	132
4.4.1	Registrasi Pengguna	132
4.4.2	<i>Login</i>	135
4.4.3	<i>Maintenance</i> Data Karyawan Beserta Hak Akses	137
4.4.4	<i>Maintenance</i> Data Barang	139
4.4.5	Menjalankan Fitur Promosi di Facebook.....	141
4.4.6	<i>Maintenance</i> Data Kategori Barang	142
4.4.7	Mengkategorikan Barang	145
4.4.8	<i>Maintenance</i> Data Pelanggan	146
4.4.9	<i>Maintenance</i> Data <i>Supplier</i>	148
4.4.10	Mencatat Transaksi Penjualan	150
4.4.11	Mencatat Retur Penjualan.....	153
4.4.12	Mencatat Pembayaran Kredit	154
4.4.13	Mencatat Transaksi Penerimaan	155
4.4.14	Menampilkan Laporan Stok Terkini.....	156
4.4.15	Menampilkan Laporan Tren Penjualan	157



4.4.16 Menampilkan Laporan <i>History</i> Pelanggan	159
4.4.17 Menampilkan Laporan <i>History Supplier</i>	161
4.4.18 Mengubah Layanan	162
4.5 Uji Coba dan Evaluasi Sistem	164
4.5.1 Uji Coba Pendaftaran Toko	164
4.5.2 Uji Coba <i>Login</i>	165
4.5.3 Uji Coba Data Barang.....	166
4.5.4 Uji Coba Data Karyawan.....	167
4.5.5 Uji Coba Data Pelanggan	168
4.5.6 Uji Coba Data <i>Supplier</i>	168
4.5.7 Uji Coba Data Penjualan	169
4.5.8 Uji Coba Data Penerimaan	170
4.5.9 Uji Coba Laporan	171
4.5.10 Uji Coba Ubah Layanan	172
4.5.11 Uji Coba <i>Cloud Performance</i>	172
4.5.12 Uji Coba Proses Bisnis	173
4.5.13 Evaluasi Kesesuaian Hasil Akhir Sistem.....	184
4.6 <i>Service Level Agreement (SLA)</i>	185
BAB V PENUTUP.....	186
5.1 Kesimpulan.....	186
5.2 Saran.....	187
DAFTAR PUSTAKA	188



DAFTAR TABEL

Tabel 1.1 Rencana Harga Sistem Informasi Penjualan.....	4
Tabel 2.1 Kriteria <i>Cloud Computing</i>	19
Tabel 3.1 Model Pengguna	37
Tabel 3.2 <i>Flow of Events</i> Mendaftarkan Toko.....	41
Tabel 3.3 <i>Flow of Events Login</i>	43
Tabel 3.4 <i>Flow of Events</i> Mengelola Data Pelanggan	44
Tabel 3.5 <i>Flow of Events</i> Mencatat Transaksi Penjualan	46
Tabel 3.6 <i>Flow of Events</i> Menerima Retur Penjualan.....	48
Tabel 3.7 <i>Flow of Events</i> Menerima Pembayaran Kredit	50
Tabel 3.8 <i>Flow of Events</i> Mencatat Transaksi Penerimaan.....	52
Tabel 3.9 <i>Flow of Events</i> Mengelola Data <i>Supplier</i>	54
Tabel 3.10 <i>Flow of Events</i> Melihat Laporan.....	56
Tabel 3.11 <i>Flow of Events</i> Mengatur Hak Akses Karyawan	57
Tabel 3.12 <i>Flow of Events</i> Mengubah Paket Layanan.....	59
Tabel 3.13 <i>Flow of Events</i> Promosi di Facebook.....	61
Tabel 3.14 Rencana Pengujian SI Penjualan Berbasis <i>Cloud</i>	109
Tabel 3.15 Desain Uji Coba Pendaftaran Toko	110
Tabel 3.16 Desain Uji Coba <i>Login</i>	111
Tabel 3.17 Desain Uji Coba Data Barang.....	111
Tabel 3.18 Desain Uji Coba Data Karyawan	113
Tabel 3.19 Desain Uji Coba Data Pelanggan	113
Tabel 3.20 Desain Uji Coba Data <i>Supplier</i>	114

Tabel 3.21 Desain Uji Coba Data Penjualan	115
Tabel 3.22 Desain Uji Coba Data Penerimaan	115
Tabel 3.23 Desain Uji Coba Laporan	116
Tabel 3.24 Desain Uji Coba Ubah Layanan	117
Tabel 3.25 Desain Uji Coba <i>Cloud Performance</i>	117
Tabel 3.26 Desain Uji Coba Proses Bisnis	118
Tabel 3.27 Desain Evaluasi Hasil Akhir Sistem	120
Tabel 4.1 Uji Coba Pendaftaran Toko	165
Tabel 4.2 Uji Coba <i>Login</i>	165
Tabel 4.3 Uji Coba Data Barang	166
Tabel 4.4 Uji Coba Data Karyawan	167
Tabel 4.5 Uji Coba Data Pelanggan.....	168
Tabel 4.6 Uji Coba Data <i>Supplier</i>	169
Tabel 4.7 Uji Coba Data Penjualan.....	170
Tabel 4.8 Uji Coba Data Penerimaan.....	171
Tabel 4.9 Uji Coba Laporan.....	171
Tabel 4.10 Uji Coba Ubah Layanan.....	172
Tabel 4.11 Uji Coba <i>Cloud Performance</i>	173
Tabel 4.12 Uji Coba Proses Bisnis.....	183
Tabel 4.13 Evaluasi Kesesuaian Hasil Akhir Sistem.....	184
Tabel 4.14 <i>Service Level Agreement</i>	185

DAFTAR GAMBAR

Gambar 2.1 Rumus Perhitungan Keuntungan Kotor	9
Gambar 2.2 Rumus Perhitungan Keuntungan Kotor Per Buah	9
Gambar 2.3 Contoh Sistem Perpetual – Rata-rata bergerak	14
Gambar 2.4 Contoh Sistem Perpetual – FIFO	15
Gambar 2.5 Contoh Sistem Perpetual – LIFO	16
Gambar 2.6 Rumus Perhitungan Bunga Kredit Dengan Metode <i>Flat Rate</i>	16
Gambar 2.7 <i>MVC Pattern</i>	21
Gambar 3.1 Langkah-Langkah Analisis Sistem	24
Gambar 3.2 <i>Business Use Case</i>	26
Gambar 3.3 <i>Activity Diagram</i> Mencatat Transaksi Penjualan	27
Gambar 3.4 <i>Activity Diagram</i> Menerima Retur Penjualan	29
Gambar 3.5 <i>Activity Diagram</i> Membuat Laporan Penjualan & Pendapatan	30
Gambar 3.6 <i>Activity Diagram</i> Mencatat Transaksi Penerimaan Barang	31
Gambar 3.7 <i>Activity Diagram</i> Membuat Laporan Penerimaan Barang	32
Gambar 3.8 <i>Block Diagram</i>	35
Gambar 3.9 Model Arsitektur Sistem Informasi Penjualan Berbasis <i>Cloud</i>	36
Gambar 3.10 Interaksi Pengguna	38
Gambar 3.11 <i>Use Case Diagram</i>	39
Gambar 3.12 Diagram Sekuensial Mendaftarkan Toko.....	63
Gambar 3.13 Diagram Sekuensial <i>Login</i>	64
Gambar 3.14 Diagram Sekuensial Mengelola Data Pelanggan	65
Gambar 3.15 Diagram Sekuensial Mencatat Transaksi Penjualan	67

Gambar 3.16 Diagram Sekuensial Menerima Retur Penjualan	68
Gambar 3.17 Diagram Sekuensial Menerima Pembayaran Kredit	69
Gambar 3.18 Diagram Sekuensial Mencatat Transaksi Penerimaan	71
Gambar 3.19 Diagram Sekuensial Mengelola Data <i>Supplier</i>	72
Gambar 3.20 Diagram Sekuensial Melihat Laporan.....	73
Gambar 3.21 Diagram Sekuensial Mengatur Hak Akses Karyawan	74
Gambar 3.22 Diagram Sekuensial Mengubah Layanan.....	75
Gambar 3.23 Diagram Sekuensial Promosi di Facebook.....	76
Gambar 3.24 <i>Class Diagram Model dan Controller</i>	78
Gambar 3.25 <i>Class Diagram Controller dan View</i>	79
Gambar 3.26 <i>Component Diagram</i>	80
Gambar 3.27 <i>Deployment Diagram</i>	81
Gambar 3.28 <i>Physical Data Model</i>	84
Gambar 3.29 Desain <i>User Interface</i> Halaman Pendaftaran Toko.....	85
Gambar 3.30 Desain <i>User Interface Form Login</i>	85
Gambar 3.31 Desain <i>User Interface</i> Halaman Penjualan (Tunai)	86
Gambar 3.32 Desain <i>User Interface Form</i> Penjualan Kredit.....	86
Gambar 3.33 Desain <i>User Interface</i> Halaman <i>History</i> Penjualan	87
Gambar 3.34 Desain <i>User Interface History</i> Penjualan – <i>Detail</i> Barang	88
Gambar 3.35 Desain <i>User Interface Form Input</i> Retur Penjualan.....	88
Gambar 3.36 Desain <i>User Interface</i> Halaman Retur Penjualan	88
Gambar 3.37 Desain <i>User Interface Form</i> Pembayaran Kredit.....	89
Gambar 3.38 Desain <i>User Interface History</i> Pembayaran Kredit.....	89
Gambar 3.39 Desain <i>User Interface</i> Halaman Penerimaan	90

Gambar 3.40 Desain <i>User Interface</i> Halaman <i>History</i> Penerimaan	90
Gambar 3.41 Desain <i>User Interface</i> <i>History</i> Penerimaan – <i>Detail</i> Barang	91
Gambar 3.42 Desain <i>User Interface</i> Halaman Pelanggan	91
Gambar 3.43 Desain <i>User Interface</i> Halaman Karyawan.....	92
Gambar 3.44 Desain <i>User Interface</i> Halaman <i>Form</i> Karyawan.....	92
Gambar 3.45 Desain <i>User Interface</i> Halaman Barang.....	93
Gambar 3.46 Desain <i>User Interface</i> <i>Form</i> Promo Barang di Facebook.....	93
Gambar 3.47 Desain <i>User Interface</i> Halaman Kategori	94
Gambar 3.48 Desain <i>User Interface</i> Halaman Pengaturan Kategori	94
Gambar 3.49 Desain <i>User Interface</i> Halaman <i>Supplier</i>	95
Gambar 3.50 Desain <i>User Interface</i> <i>Parameter</i> Laporan Stok Terkini	95
Gambar 3.51 Desain <i>User Interface</i> <i>Parameter</i> Laporan Tren Penjualan.....	96
Gambar 3.52 Desain <i>User Interface</i> <i>Parameter</i> <i>History</i> Pelanggan.....	96
Gambar 3.53 Desain <i>User Interface</i> <i>Parameter</i> <i>History</i> <i>Supplier</i>	97
Gambar 3.54 Desain <i>User Interface</i> Halaman Ubah Layanan.....	97
Gambar 3.55 Desain <i>User Interface</i> Halaman Konfirmasi Pembayaran	98
Gambar 3.56 Desain <i>Output</i> Nota Transaksi Penjualan Tunai	99
Gambar 3.57 Desain <i>Output</i> Nota Transaksi Penjualan Kredit	99
Gambar 3.58 Desain <i>Output</i> Nota Retur Transaksi Penjualan.....	100
Gambar 3.59 Desain <i>Output</i> Nota Transaksi Pembayaran Kredit	101
Gambar 3.60 Desain <i>Output</i> Nota Transaksi Penerimaan Barang.....	101
Gambar 3.61 Desain <i>Output</i> Laporan Tabular Stok Terkini.....	102
Gambar 3.62 Desain <i>Output</i> Laporan Grafis Stok Terkini	102
Gambar 3.63 Desain <i>Output</i> Laporan Tabular Tren Penjualan.....	104

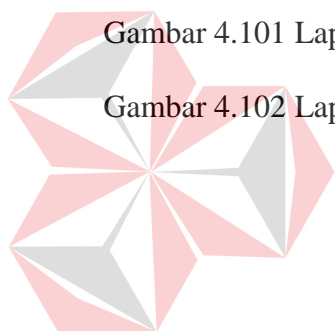
Gambar 3.64 Desain <i>Output</i> Laporan Grafis Tren Penjualan	104
Gambar 3.65 Desain <i>Output</i> Laporan Tabular <i>History</i> Pelanggan	105
Gambar 3.66 Desain <i>Output</i> Laporan Grafis <i>History</i> Pelanggan.....	106
Gambar 3.67 Desain <i>Output</i> Laporan Tabular <i>History Supplier</i>	107
Gambar 3.68 Desain <i>Output</i> Laporan Grafis <i>History Supplier</i>	107
Gambar 3.69 Desain <i>Output</i> Nota Tagihan Perubahan Layanan	108
Gambar 4.1 Registrasi <i>Microsoft Account</i>	124
Gambar 4.2 Registrasi <i>Windows Azure – Nomer Telepon</i>	125
Gambar 4.3 Registrasi <i>Windows Azure – Credit Card</i>	125
Gambar 4.4 Pembuatan <i>Virtual Machine</i>	126
Gambar 4.5 Pembuatan <i>folder website</i> pada <i>Internet Information Service</i>	127
Gambar 4.6 Konfigurasi <i>Application Pool</i>	128
Gambar 4.7 Pembuatan <i>Azure SQL Database</i>	129
Gambar 4.8 Konfigurasi <i>IP Address</i>	129
Gambar 4.9 <i>Menu Deploy Database</i> ke <i>Azure SQL Database</i>	130
Gambar 4.10 <i>Deploy Database</i> ke <i>Azure SQL Database</i>	131
Gambar 4.11 Konfigurasi <i>Connection String</i>	131
Gambar 4.12 Halaman Awal.....	132
Gambar 4.13 Halaman Registrasi Akun	133
Gambar 4.14 Halaman Reistrasi Akun Telah Berhasil Dilakukan	133
Gambar 4.15 <i>Email</i> Aktivasi Akun.....	134
Gambar 4.16 Halaman Akun Belum Diaktivasi	134
Gambar 4.17 Halaman Pengiriman Ulang <i>Email</i> Aktivasi Berhasil.....	135
Gambar 4.18 Halaman <i>Login</i>	135

Gambar 4.19 Pesan Kesalahan <i>Username</i> dan <i>Password</i>	136
Gambar 4.22 Pengiriman <i>Password</i> ke <i>Email</i> Berhasil	136
Gambar 4.21 <i>Email</i> Informasi <i>Password</i>	136
Gambar 4.22 Halaman Karyawan	137
Gambar 4.23 <i>Form</i> Karyawan Baru.....	138
Gambar 4.24 <i>Form</i> Ubah Karyawan.....	138
Gambar 4.25 Penghapusan Data Karyawan.....	139
Gambar 4.26 Penghapusan Data Karyawan Berhasil	139
Gambar 4.27 Halaman Barang.....	140
Gambar 4.28 Pengisian Data Barang	140
Gambar 4.29 Pengisian Data Barang Berhasil.....	140
Gambar 4.30 Pengubahan Data Barang	141
Gambar 4.31 Pengubahan Data Barang Berhasil.....	141
Gambar 4.32 <i>Form</i> Promosi Barang di Facebook	142
Gambar 4.33 Tampilan Promosi di Facebook.....	142
Gambar 4.34 Halaman Kategori Barang.....	143
Gambar 4.35 Pengisian Data Kategori Barang	143
Gambar 4.36 Pengisian Data Kategori Barang Berhasil.....	143
Gambar 4.37 Pengubahan Data Kategori Barang	144
Gambar 4.38 Pengubahan Data Kategori Barang Berhasil.....	144
Gambar 4.39 Penghapusan Data Kategori Barang.....	144
Gambar 4.40 Penghapusan Data Kategori Barang Berhasil	145
Gambar 4.41 Pengkategorian Barang	145
Gambar 4.42 Pengkategorian Barang Berhasil	145

Gambar 4.43 Halaman Data Pelanggan	146
Gambar 4.44 Pengisian Data Pelanggan	146
Gambar 4.45 Pengisian Data Pelanggan Berhasil.....	147
Gambar 4.46 Pengubahan Data Pelanggan	147
Gambar 4.47 Pengubahan Data Pelanggan Berhasil.....	147
Gambar 4.48 Penghapusan Data Pelanggan.....	148
Gambar 4.49 Penghapusan Data Pelanggan Berhasil	148
Gambar 4.50 Halaman <i>Supplier</i>	148
Gambar 4.51 Pengisian Data <i>Supplier</i>	149
Gambar 4.52 Pengisian Data <i>Supplier</i> Berhasil.....	149
Gambar 4.53 Pengubahan Data <i>Supplier</i>	149
Gambar 4.54 Pengubahan Data <i>Supplier</i> Berhasil.....	150
Gambar 4.55 Penghapusan Data <i>Supplier</i>	150
Gambar 4.56 Penghapusan Data <i>Supplier</i> Berhasil	150
Gambar 4.57 Pencatatan Transaksi Penjualan Tunai	151
Gambar 4.58 Nota Penjualan Tunai	151
Gambar 4.59 <i>Form</i> Penjualan Kredit	152
Gambar 4.60 Nota Penjualan Kredit	153
Gambar 4.61 <i>Form</i> Pencatatan Retur Penjualan	153
Gambar 4.62 Nota Retur Penjualan	154
Gambar 4.63 <i>Form</i> Pencatatan Pembayaran Kredit.....	154
Gambar 4.64 Nota Pembayaran Kredit	155
Gambar 4.65 Pencatatan Transaksi Penerimaan	155
Gambar 4.66 Nota Penerimaan Barang.....	156

Gambar 4.67 <i>Form</i> Laporan Stok Terkini.....	156
Gambar 4.68 Laporan Tabular Stok Terkini	157
Gambar 4.69 Laporan Grafis Stok Terkini	157
Gambar 4.70 <i>Form</i> Laporan Tren Penjualan	158
Gambar 4.71 Laporan Tabular Tren Penjualan.....	158
Gambar 4.72 Laporan Grafis Tren Penjualan	158
Gambar 4.73 <i>Form</i> Laporan <i>History</i> Pelanggan	159
Gambar 4.74 Laporan Tabular <i>History</i> Pelanggan	160
Gambar 4.75 Laporan Grafis <i>History</i> Pelanggan.....	160
Gambar 4.76 <i>Form</i> Laporan <i>History Supplier</i>	161
Gambar 4.77 Laporan Tabular <i>History Supplier</i>	161
Gambar 4.78 Laporan Grafis <i>History Supplier</i>	162
Gambar 4.79 <i>Form</i> Perubahan Layanan.....	162
Gambar 4.80 Nota Tagihan Perubahan Layanan.....	163
Gambar 4.81 <i>Form</i> Konfirmasi <i>Transfer</i>	163
Gambar 4.82 Konfirmasi <i>Transfer</i> Berhasil	164
Gambar 4.83 Pengaturan Metode Penentuan HPP.....	174
Gambar 4.84 Stok Awal Persediaan Barang	174
Gambar 4.85 Transaksi Penerimaan Barang Pertama.....	175
Gambar 4.86 Nota Transaksi Penerimaan Barang Pertama	175
Gambar 4.87 Stok Barang Setelah Transaksi Penerimaan Barang Pertama	175
Gambar 4.88 Transaksi Penerimaan Barang Kedua	176
Gambar 4.89 Nota Transaksi Penerimaan Barang Kedua.....	176
Gambar 4.90 Stok Barang Setelah Transaksi Penerimaan Barang Kedua.....	177

Gambar 4.91 Transaksi Penjualan Barang	177
Gambar 4.92 Nota Transaksi Penjualan.....	178
Gambar 4.93 Stok Barang Setelah Transaksi Penjualan.....	178
Gambar 4.94 <i>Input Parameter</i> Laporan Barang Terlaris	179
Gambar 4.95 Laporan Grafis Barang Terlaris.....	179
Gambar 4.96 Laporan Tabular Barang Terlaris	180
Gambar 4.97 <i>Input Parameter</i> Laporan Barang Tidak Laris	180
Gambar 4.98 Laporan Grafis Barang Tidak Laris	181
Gambar 4.99 Laporan Tabular Barang Tidak Laris	181
Gambar 4.100 <i>Input Parameter</i> Laporan <i>Supplier</i>	182
Gambar 4.101 Laporan Grafis <i>Supplier</i>	182
Gambar 4.102 Laporan Tabular <i>Supplier</i>	182



BAB I

PENDAHULUAN

1.1 Latar Belakang

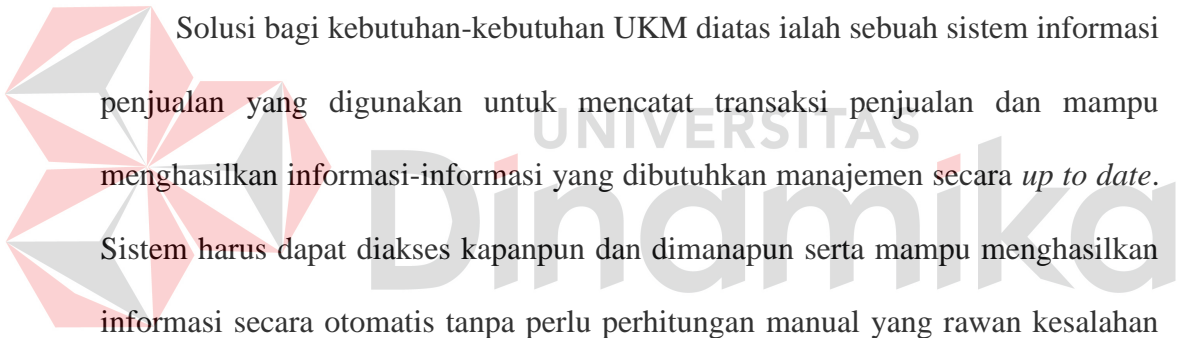
Jumlah Usaha Kecil dan Menengah (UKM) di Indonesia semakin meningkat. Saat ini, UKM yang tersebar di Indonesia berjumlah kurang lebih 52 juta (Rohmat; 2011). Bahkan berdasarkan data yang dimiliki Menteri Negara Koperasi dan UKM, UKM di Indonesia menyumbang 56,5% dari PDB dan menampung 97% tenaga kerja. Kemenkop UKM juga menyampaikan bahwa sampai saat ini 99% pelaku perekonomian Indonesia berasal dari koperasi dengan jumlah 177.483 unit dan UKM yang berjumlah 52,7 juta unit (Yoga; 2011). Berdasarkan data diatas, UKM memiliki pengaruh yang besar bagi perekonomian dan pemberdayaan tenaga kerja di Indonesia.

Walaupun berpengaruh sangat besar, UKM masih mengalami kesulitan untuk berkembang. Hanafi, Kepala Dinas Perindustrian, Perdagangan, Pertambangan dan Energi Karawang, mengatakan bahwa minimnya pengetahuan dan informasi pasar menjadi salah satu kelemahan yang menghambat perkembangan UKM (Kusmayadi; 2011). Padahal informasi tersebut sangat diperlukan pihak manajemen untuk mendukung pengambilan keputusan. Manajemen tidak dapat mengetahui tren penjualan yang diperlukan untuk mengetahui *lifetime* sebuah produk dan menentukan jumlah pemesanan kepada *supplier* pada periode berikutnya. Hal ini menyebabkan UKM membeli atau memproduksi barang yang sudah tidak laku lagi dalam jumlah banyak sehingga menimbulkan kerugian. Manajemen juga kesulitan mengetahui stok terkini karena

harus menghitung terlebih dahulu atau berdasarkan nota rekap. Hal ini menyebabkan manajemen tidak dapat memperoleh informasi stok secara *up to date* sehingga sering mengalami kehabisan stok atau penumpukan stok karena kesalahan perkiraan waktu saat *reorder* ke *supplier*. Jika kehabisan stok, UKM akan kehilangan peluang untuk memperoleh keuntungan lebih. Jika stok menumpuk, UKM terancam rugi karena barang kadaluarsa dari segi masa pemakaian atau minat pasar. Manajemen juga tidak memiliki data *history* transaksi pelanggan dan *supplier* untuk mengetahui barang yang paling sering dibeli oleh pelanggan tertentu serta harga beli barang tertentu dari *supplier* yang ada. Dengan semakin berkembangnya media internet dan jejaring sosial, UKM juga memerlukan data stok dan harga terkini untuk berpromosi di jejaring sosial. Hal ini bertujuan agar tidak terjadi kesalahan harga saat promo dan tidak mempromokan barang yang stoknya habis. Informasi-informasi diatas harus sebisa mungkin menggambarkan kondisi terkini di toko agar keputusan yang diambil oleh pihak manajemen lebih tepat. Informasi juga harus dapat diakses kapanpun dan dimanapun karena pihak manajemen cenderung bergerak *mobile* dan jarang terus menerus berada di tempat usaha, toko atau kantor mereka.

Beberapa UKM telah memanfaatkan peralatan dan teknologi untuk memenuhi kebutuhan informasi mereka. Namun sistem pencatatan transaksi penjualan mereka tidak mampu memenuhi kebutuhan pihak manajemen. Mayoritas UKM masih menggunakan kalkulator dan pencatatan manual sehingga data yang dicatat tidak lengkap dan kurang informatif karena bagian penjualan juga dituntut untuk melayani pelanggan dengan cepat. Mekanisme pencatatan tersebut juga mengharuskan bagian penjualan merekap dan mengolah

data secara manual sehingga tidak mampu menghasilkan informasi secara *up to date*. Beberapa UKM menggunakan mesin kasir atau *cash register*. Namun masih terdapat beberapa kekurangan seperti masih diperlukannya perekapan data sehingga tidak dapat menghasilkan informasi *up to date*, kurang *portable* bagi bisnis yang lokasi penjualannya berpindah-pindah serta informasi tidak dapat diakses kapanpun dan dimanapun. Beberapa UKM juga telah memanfaatkan program kasir *stand-alone*. Namun sistem ini memiliki kekurangan seperti biaya *maintenance* yang cukup besar untuk *server*, perangkat teknologi serta SDM yang melakukan *maintenance*. Informasi juga tidak dapat diakses kapanpun dan dimanapun serta banyak ancaman bagi keamanan data dan ketersediaan sistem.



Solusi bagi kebutuhan-kebutuhan UKM diatas ialah sebuah sistem informasi penjualan yang digunakan untuk mencatat transaksi penjualan dan mampu menghasilkan informasi-informasi yang dibutuhkan manajemen secara *up to date*. Sistem harus dapat diakses kapanpun dan dimanapun serta mampu menghasilkan informasi secara otomatis tanpa perlu perhitungan manual yang rawan kesalahan dan mengakibatkan informasi yang salah. Sistem juga harus memerlukan biaya *maintenance* yang murah serta memiliki tingkat kemanan dan ketersediaan sistem yang terpercaya sehingga solusi berbasis *cloud* sangat tepat karena sistem akan disimpan pada *data center* yang memiliki standar dan jaminan yang tinggi dan tercatat dalam *Service Level Agreement* (SLA). Sistem akan dibangun sebagai *software as a service* sehingga UKM tidak perlu melakukan *maintenance* infrastrukturnya sendiri melainkan hanya perlu membayar biaya layanan secara periodik dengan harga yang lebih murah seperti berikut ini.

Tabel 1. 1 Rencana Harga Sistem Informasi Penjualan

Nama paket	Harga (rupiah)	Jumlah maksimal transaksi penjualan (dalam 30 hari)
Gratis	0	250
Super	25.000	5.000
Premium	50.000	50.000

Data dan sistem yang tersimpan di *cloud* memungkinkan UKM untuk mengaksesnya kapanpun dan dimanapun dengan menggunakan komputer, laptop, *tablet* atau *smartphone* yang terhubung dengan internet. Sistem yang akan dibangun juga memiliki fitur promosi barang atau jasa di Facebook sesuai dengan data terkini yang dimiliki sehingga mampu memenuhi kebutuhan untuk berpromosi sesuai dengan kondisi terkini dan tidak terjadi kesalahan informasi promo yang disampaikan ke pelanggan.

Dengan adanya sistem informasi penjualan berbasis *cloud* ini, UKM dapat memperoleh informasi tren penjualan, stok terkini, *history* transaksi *supplier* dan pelanggan yang dapat digunakan pihak manajemen sebagai dasar dalam mengambil keputusan serta promosi barang di jejaring sosial Facebook sesuai stok dan harga terkini. Informasi tersebut dapat diakses kapanpun dan dimanapun melalui jaringan internet.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang dan membangun sistem informasi penjualan berbasis *cloud* yang menghasilkan informasi tren penjualan, stok terkini, *history* transaksi *supplier* dan pelanggan sesuai dengan kebutuhan manajemen serta dapat diakses kapanpun dan dimanapun melalui jaringan internet.

2. Bagaimana merancang dan membangun sistem informasi penjualan berbasis *cloud* yang dapat digunakan sebagai media promosi pada jejaring sosial Facebook berdasarkan data terkini yang dimiliki.

1.3 Batasan Masalah

Dalam pembuatan sistem informasi ini, agar tidak menyimpang dari tujuan yang ingin dicapai maka pembahasan dibatasi pada hal-hal berikut:

1. Sistem hanya dapat digunakan jika tersedia koneksi internet.
2. Sistem hanya menangani penjualan dengan satu harga jual, tidak menangani penjualan multi harga (*multi level*).
3. Promosi pada jejaring sosial Facebook dapat dipilih berdasarkan barang terlaris, persentase diskon terbesar dan pemilihan barang tertentu.
4. Jumlah barang yang digunakan dalam sistem adalah bilangan bulat.
5. Perhitungan bunga kredit dilakukan dengan metode *flat rate*.
6. Studi kasus dilakukan pada toko "Toko Besar".

1.4 Tujuan

Berdasarkan perumusan masalah di atas, maka tujuan yang hendak dicapai adalah sebagai berikut:

1. Merancang dan membangun sistem informasi penjualan berbasis *cloud* yang menghasilkan informasi tren penjualan, stok terkini, *history* transaksi *supplier* dan pelanggan sesuai dengan kebutuhan manajemen serta dapat diakses kapanpun dan dimanapun melalui jaringan internet.

2. Merancang dan membangun sistem informasi penjualan berbasis *cloud* yang dapat digunakan sebagai media promosi pada jejaring sosial Facebook berdasarkan data terkini yang dimiliki.

1.5 Sistematika Penulisan

Untuk memudahkan pemahaman persoalan dan pembahasannya, penulisan laporan Tugas Akhir ini dibuat dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dikemukakan hal-hal yang menjadi latar belakang, perumusan masalah, batasan masalah, tujuan yang ingin dicapai serta sistematika penulisan laporan tugas akhir ini.

BAB II LANDASAN TEORI

Pada bab ini dibahas secara singkat teori-teori yang berhubungan dan mendukung pembuatan tugas akhir ini. Teori-teori yang dibahas adalah penjualan, sistem informasi, sistem informasi penjualan, usaha kecil dan menengah (UKM), metode pencatatan persediaan perpetual, metode perhitungan bunga kredit *flat rate*, komputasi awan (*cloud computing*), Windows Azure, *virtual private server* (VPS), ASP.NET *Model View Controller* (MVC), *Azure Structured Query Language* (SQL) *Database* dan *blackbox testing*.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dibahas tentang gambaran umum, analisis sistem dan perancangan dari sistem yang dibuat. Perancangan sistem meliputi *use case diagram*, *flow of events*, *sequence diagram*, *class diagram*, *component diagram*, *deployment diagram*, rencana penggunaan

infrastruktur *cloud*, *physical data model*, desain *user interface*, desain *input output* dan rancangan pengujian dan evaluasi sistem.

BAB IV IMPLEMENTASI DAN EVALUASI

Pada bab ini dibahas tentang kebutuhan sistem, pembuatan program, langkah-langkah implementasi sistem di *cloud*, penggunaan sistem dalam proses bisnis serta pengujian untuk mengetahui apakah sistem yang dibangun dapat menyelesaikan permasalahan yang dihadapi sesuai dengan yang diharapkan. Pengujian sistem dilakukan pada validasi *input* dan *output*, uji coba proses bisnis dan evaluasi kesesuaian hasil akhir sistem. Pada bab ini juga dibahas tentang *Service Level Agreement* (SLA) dari sistem informasi penjualan yang dibangun dan mengacu pada SLA dari Windows Azure sebagai *cloud provider* yang digunakan.

BAB V PENUTUP

Pada bab ini dibahas tentang kesimpulan dan saran. Kesimpulan dan saran yang ada pada bab ini didapatkan dari hasil evaluasi sistem, sedangkan saran akan menjelaskan saran atau masukan untuk pengembangan aplikasi lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Penjualan

Menurut Marwan (Iskandar; 2009), penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana-rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli, guna mendapatkan penjualan yang menghasilkan laba. Penjualan merupakan sumber hidup suatu perusahaan, karena dari penjualan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan.

Menurut Winardi (Iskandar; 2009), penjualan adalah suatu *transfer* hak atas benda-benda. Dari penjelasan tersebut dalam memindahkan atau *transfer* barang dan jasa diperlukan orang-orang yang bekerja di bidang penjualan seperti pelaksanaan dagang, agen, wakil pelayanan dan wakil pemasaran.

Sistem penjualan adalah sistem yang melibatkan sumber daya dalam suatu organisasi, prosedur, data, serta sarana pendukung untuk mengoperasikan sistem penjualan, sehingga menghasilkan informasi yang bermanfaat bagi pihak manajemen dalam pengambilan keputusan (Karmatoli; 2012).

Salah satu proses penting dalam penjualan adalah menghitung keuntungan. Dalam penjualan, terdapat 2 buah keuntungan, yaitu keuntungan kotor dan keuntungan bersih. Keuntungan kotor adalah perhitungan keuntungan setiap hari dan bulannya dengan ketentuan harga jual dikurangi HPP atau modal pokok.

Sedangkan keuntungan bersih adalah jumlah total keuntungan kotor dikurangi dengan total biaya operasional usaha (Ayodya; 2010).

Dalam sistem informasi penjualan yang dibangun, sistem hanya menghitung keuntungan kotor karena perhitungan biaya operasional tidak termasuk dalam ruang lingkup sistem. Rumus perhitungan keuntungan kotor adalah seperti pada gambar 2.1.

$$\begin{aligned} \text{Perhitungan Keuntungan Kotor} &= \text{Total Penjualan} - \text{Total HPP} \\ \text{atau} \\ \text{Keuntungan Kotor} &= \text{Keuntungan Kotor/buah} \times \text{Jumlah Produk} \\ &\text{Terjual} \end{aligned}$$

Gambar 2. 1 Rumus Perhitungan Keuntungan Kotor

Dalam perhitungan keuntungan kotor, terdapat *variable* keuntungan kotor per buah. Rumus keuntungan kotor per buah secara umum ialah seperti pada gambar 2.2.

$$\text{Keuntungan kotor per buah} = \text{Harga jual} - \text{Harga beli (HPP)} - \text{Diskon}$$

Gambar 2. 2 Rumus Perhitungan Keuntungan Kotor Per Buah

Contoh kasus perhitungan transaksi penjualan adalah sebagai berikut. Pembeli membeli barang A sejumlah 10 buah dengan harga jual 10.000 rupiah per satuan serta mendapat diskon 5% untuk tiap satuan. Penjual membeli barang A tersebut dari *supplier* dengan harga 8.000 rupiah. Perhitungan nilai penjualan dan keuntungannya adalah sebagai berikut.

Harga jual per per satuan	= $10.000 - (5\% \times 10.000)$	= 9.500 rupiah
Jumlah barang terjual	= 10 buah	
Nilai penjualan	= 9.500×10	= 95.000 rupiah
Harga beli per satuan	= 8.000 rupiah	
Keuntungan per satuan	= $9.500 - 8.000$	= 1.500 rupiah
Total keuntungan	= 1.500×10	= 15.000 rupiah

2.2 Sistem Informasi

Sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang saling berfungsi dengan tujuan yang sama. Sistem informasi adalah serangkaian prosedur formal dimana data dikumpulkan, diproses menjadi informasi dan didistribusikan ke para pengguna (Hall; 2009).

Definisi umum sistem informasi adalah sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan (Kusrini; 2007). Dalam suatu sistem informasi, terdapat komponen-komponen sebagai berikut:

- Perangkat keras (*hardware*), mencakup berbagai piranti fisik seperti komputer dan *printer*.
- Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
- Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
- Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

- e. Basis data (*database*), yaitu sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.
- f. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resource*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2.3 Sistem Informasi Penjualan

Sistem informasi penjualan adalah sub sistem informasi bisnis yang mencakup kumpulan prosedur yang melaksanakan, mencatat, mengkalkulasi, membuat dokumen dan informasi penjualan untuk keperluan manajemen dan bagian lain yang berkepentingan, mulai dari diterimanya order penjualan sampai mencatat timbulnya tagihan atau piutang dagang (Marconi; 2011).

Berdasarkan pengertian di atas, dapat disimpulkan bahwa komponen-komponen dari sistem informasi penjualan secara umum terdiri dari :

- a. Pencatatan transaksi penjualan
- b. Pengecekan stok barang
- c. Kalkulasi jumlah dan harga
- d. Pembuatan dan pencetakan nota penjualan
- e. Pembuatan dokumen atau informasi penjualan untuk keperluan manajemen

Dalam sistem yang dibangun, komponen pencatatan transaksi penjualan, pengecekan stok barang, kalkulasi jumlah dan harga beserta pembuatan dan pencetakan nota penjualan masuk dalam sub sistem *point of sales*. Sedangkan dokumen atau informasi penjualan untuk keperluan manajemen akan menjadi *output* dari sistem informasi yang dibangun.

2.4 Usaha Kecil dan Menengah (UKM)

Departemen Perindustrian dan Perdagangan (Depperindag), Bank Indonesia, Departemen Keuangan, Depkop dan PKM mendefinisikan UKM berdasarkan aset (diluar tanah dan bangunan) dan nilai penjualan yang dihitung dalam rupiah (Tambunan; 2002). Sedangkan BPS mendefinisikan UKM berdasarkan jumlah tenaga kerja.

Menurut UU no. 9 tahun 1995, “Industri Kecil adalah kegiatan ekonomi yang dilakukan oleh perseorangan atau rumah tangga maupun suatu badan, yang bertujuan untuk diperniagakan secara komersial, yang mempunyai kekayaan bersih paling banyak Rp. 200 juta, dan mempunyai nilai penjualan per tahun sebesar satu milyar rupiah atau kurang. Industri Menengah adalah kegiatan ekonomi yang dilakukan oleh perseorangan atau badan, yang bertujuan untuk memproduksi barang ataupun jasa untuk diperniagakan secara komersial, yang mempunyai nilai penjualan per tahun lebih besar dari satu milyar rupiah namun kurang dari Rp. 50 milyar” (Sulistyastuti; 2004).

2.5 Metode Pencatatan Persediaan Perpetual

Dalam metode pencatatan persediaan perpetual, perkiraan persediaan digunakan untuk mencatat mutasi/perubahan persediaan yaitu dengan mencatat persediaan awal periode, pembelian selama periode, penjualan selama periode dan persediaan akhir pada periode tersebut (Sugiono; 2009).

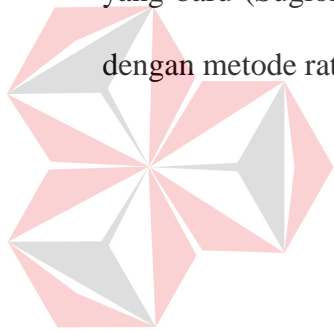
Setiap jenis persediaan dibuat kartu persediaan (*inventory card*) untuk mencatat mutasi persediaan baik kuantitas maupun harga pokok dari barang yang dibeli dan dijual. Dengan bantuan kartu persediaan tersebut, maka saldo

persediaan dan harga pokok penjualan dapat diketahui setiap saat tanpa harus melakukan perhitungan fisik terhadap persediaan.

Dalam metode perpetual, terhadap 3 metode untuk penentuan harga pokok, yaitu rata-rata bergerak (*average costing*), *First In First Out (FIFO)* dan *Last In First Out (LIFO)*.

2.5.1 Sistem Perpetual – Rata-rata bergerak

Sistem ini merupakan suatu metode penilaian persediaan dengan cara menentukan harga pokok rata-rata pada saat ada pembelian. Apabila terjadi penjualan harga pokok barang yang dijual akan dibebani harga pokok rata-rata yang baru (Sugiono; 2009). Gambar 2.3 menunjukkan contoh persediaan barang dengan metode rata-rata bergerak pada pencatatan perpetual.



UNIVERSITAS
Dinamika

1 Januari	Persediaan awal	400 unit	@ Rp. 240,-	Rp.	96.000,-
Pembelian :					
5 Maret		1.000 unit	@ Rp. 250,-	Rp.	250.000,-
15 Juli		2.280 unit	@ Rp. 260,-	Rp.	592.800,-
25 Nov		320 unit	@ Rp. 275,-	Rp.	88.000,-
		<u>3.600 unit</u>		Rp.	<u>930.800,-</u>
Penjualan :					
10 April		750 unit	@ Rp. 300,-	Rp.	225.000,-
20 Sept		1.750 unit	@ Rp. 315,-	Rp.	551.250,-
30 Des		1.000 unit	@ Rp. 325,-	Rp.	325.000,-
		<u>3.500 unit</u>		Rp.	<u>1.101.250,-</u>

Berdasarkan data persediaan diatas, dapat dibuat kartu persediaan :

KARTU PERSEDIAAN RATA – RATA BERGERAK (dg pembulatan)									
Tgl.	MASUK			KELUAR			SALDO		
	Q	Harga per unit	Total	Q	Harga perunit	Total	Q	Harga perunit	Total
1 Jan							400	240	96.000
5 Mar	1.000	250	250.000				1.400	247,14	346.000
10 Apr				750	247,14	188.355	650	247,14	156.645
15 Juli	2.280	260	592.800				2.930	255,78	749.445
20 Sept				1.750	255,78	447.615	1.180	255,78	301.830
25 Nov	320	275	88.000				1.500	260	389.830
30 Des				1.000	260	260.000	500	260	129.830
	<u>3.600</u>		<u>930.800</u>	<u>3.500</u>		<u>896.970</u>	<u>500</u>	<u>260</u>	<u>129.830</u>

Harga Pokok Penjualan adalah :

Persediaan awal	Rp.	96.000,-
Pembelian	Rp.	930.800,-
Persediaan akhir	- Rp.	<u>129.830,-</u>
Harga Pokok Penjualan	Rp.	896.970,-

Gambar 2. 3 Contoh Sistem Perpetual - Rata-rata bergerak

2.5.2 Sistem Perpetual – FIFO

Sistem ini melakukan perhitungan harga pokok pada setiap terjadi penjualan. Barang yang dijual pertama-tama dibebani dengan harga pokok dari barang yang berasal dari pembelian pertama yang ada pada saat penjualan. Apabila masih kurang, maka kekurangannya dibebani dengan harga pokok dari barang yang dibeli berikutnya (Sugiono; 2009). Gambar 2.4 menunjukkan contoh persediaan barang dengan metode FIFO pada pencatatan perpetual.

KARTU PERSEDIAAN FIFO									
Tgl.	MASUK			KELUAR			SALDO		
	Q	Harga per unit	Total	Q	Harga perunit	Total	Q	Harga perunit	Total
1 Jan							400	240	96.000
5 Mar	1.000	250	250.000				400	240	96.000
							1.000	250	250.000
10 Apr				400	240	96.000			
				350	250	87.500			
15 Juli	2.280	260	592.800				650	250	162.500
							650	250	162.500
20 Sept				650	250	162.500			
				1.100	260	286.000			
25 Nov	320	275	88.000				1.180	260	306.800
							1.180	260	306.800
30 Des				1.000	260	260.000	320	275	88.000
							180	260	46.800
							320	275	88.000
	3.600		930.800	3.500		892.000	500		134.800

Persediaan akhir adalah : 400 unit + 3.600 unit – 3.500 unit = 500 unit

Nilai Persediaan akhir adalah :

320 unit X Rp. 275,- = Rp. 88.000,-
 180 unit X Rp. 260,- = Rp. 46.800,-
Rp. 134.800,-



Harga Pokok Penjualan adalah :

Persediaan awal	Rp.	96.000,-
Pembelian	Rp.	930.800,-
Persediaan akhir	Rp.	<u>134.800,-</u>
Harga Pokok Penjualan	Rp.	892.000,-

Gambar 2. 4 Contoh Sistem Perpetual - FIFO

2.5.3 Sistem Perpetual – LIFO

Metode LIFO (*Last In First Out*) adalah metode penilaian persediaan yang menganggap barang yang terakhir kali masuk adalah barang yang pertama kali keluar. Berdasarkan metode ini, setiap kali ada barang yang keluar (terjadi penjualan), harga pokoknya diperhitungkan berdasarkan harga pokok per satuan barang yang terjadi paling akhir (Sugiono; 2009). Gambar 2.5 menunjukkan contoh persediaan barang dengan metode LIFO pada pencatatan perpetual.

KARTU PERSEDIAAN LIFO									
Tgl.	MASUK			KELUAR			SALDO		
	Q	Harga per unit	Total	Q	Harga per unit	Total	Q	Harga per unit	Total
1 Jan							400	240	96.000
5 Mar	1.000	250	250.000				400	240	96.000
							1.000	250	250.000
10 Apr				750	250	187.500	400	240	96.000
							250	250	62.500
15 Juli	2.280	260	592.800				400	240	96.000
							250	250	62.500
							2.280	260	592.800
20 Sept				1.750	260	455.000	400	240	96.000
							250	250	62.500
							530	260	137.800
25 Nov	320	275	88.000				400	240	96.000
							250	250	62.500
							530	260	137.800
							320	275	88.000
30 Des				320	275	88.000			
				530	260	137.800	400	240	96.000
				150	250	37.500	100	250	25.000
	3.600		930.800	3.500		905.800	500		121.000

Nilai Persediaan akhir adalah :

400 unit X Rp. 240,- = Rp. 96.000,-

100 unit X Rp. 250,- = Rp. 25.000,-

Rp. 121.000,-

Harga Pokok Penjualan adalah :

Persediaan awal Rp. 96.000,-

Pembelian Rp. 930.800,-

Persediaan akhir - Rp. 121.000,-

Harga Pokok Penjualan Rp. 905.800,-

Gambar 2. 5 Contoh Sistem Perpetual - LIFO

2.6 Metode Perhitungan Bunga Kredit *Flat Rate*

Perhitungan bunga kredit sistem *flat* atau model *constant payment mortgage* adalah perhitungan bunga kredit yang menghasilkan angsuran pinjaman yang tetap dari periode ke periode (Arifin; 2007). Hal ini karena bunga dihitung tetap dari pokok pinjaman awal, secara matematis dapat dihitung dengan rumus seperti pada gambar 2.6.

$$\text{BUNGA TETAP} = (\text{SALDO PINJAMAN AWAL} \times \text{BUNGA PER TAHUN}) / 12$$

Gambar 2. 6 Rumus Perhitungan Bunga Kredit Dengan Metode *Flat Rate*

Contoh kasus perhitungan bunga kredit dengan metode *flat rate* adalah sebagai berikut. Terdapat transaksi kredit dengan hutang 500.000 rupiah. Cicilan dilakukan 12 kali dengan periode tiap bulan (periode kredit 1 tahun). Bunga adalah 10% per tahun. Dengan demikian jumlah bunga kredit adalah:

$$\begin{aligned} \text{Bunga Tetap} &= (500.000 \times 0.1) / 12 \\ &= 50.000 / 12 \\ &= \mathbf{4166.67 \text{ rupiah}} \end{aligned}$$

$$\begin{aligned} \text{Jumlah Cicilan Bulanan} &= (500.000/12) + 4166.67 \\ &= \mathbf{45.832 \text{ rupiah}} \end{aligned}$$

2.7 Komputasi Awan (*Cloud Computing*)

Cloud computing atau komputasi awan ialah teknologi yang memanfaatkan layanan internet menggunakan pusat *server* yang bersifat *virtual* dengan tujuan pemeliharaan data dan aplikasi (Syaikh; 2010). Keberadaan komputasi awan jelas akan menimbulkan perubahan dalam cara kerja sistem teknologi informasi dalam sebuah organisasi. Hal ini karena komputasi awan melalui konsep virtualisasi, standarisasi dan fitur mendasar lainnya dapat mengurangi biaya Teknologi Informasi (TI), menyederhanakan pengelolaan layanan TI, dan mempercepat penghantaran layanan. Secara umum arsitektur komputasi awan terdiri dari *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) dan *Software as a Service* (SaaS).

National Institute of Standards and Technology (NIST) mendefinisikan tiga arsitektur komputasi awan di atas sebagai berikut :

- a. ***Cloud Software as a Service (SaaS)*** adalah kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia yang beroperasi

pada infrastruktur awan. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka seperti web *browser* (misalnya, email berbasis web). Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, *server*, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi individu, dengan kemungkinan pengecualian terbatas terhadap pengaturan konfigurasi aplikasi pengguna tertentu.

b. ***Cloud Platform as a Service (PaaS)*** adalah kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur komputasi awan menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, *server*, sistem operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi yang disebarkan dan memungkinkan aplikasi melakukan hosting konfigurasi.

c. ***Cloud Infrastructure as a Service (IaaS)*** adalah kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, berjejaring, dan komputasi sumberdaya lain yang penting, dimana konsumen dapat menyebarkan dan menjalankan perangkat lunak secara bebas, dapat mencakup sistem operasi dan aplikasi. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang disebarkan, dan mungkin

kontrol terbatas terhadap komponen jaringan yang dipilih (misalnya, *firewall host*).

NIST juga mengidentifikasi lima karakteristik penting dari komputasi awan sebagai berikut :

Tabel 2. 1 Kriteria *Cloud Computing*

Karakteristik	Deskripsi
<i>On-demand self-service</i>	Pengguna dapat memesan dan mengelola layanan tanpa interaksi manusia dengan penyedia layanan, misalnya dengan menggunakan, sebuah portal web dan manajemen antarmuka. Pengadaan dan perlengkapan layanan serta sumberdaya yang terkait terjadi secara otomatis pada penyedia.
<i>Broad network access</i>	Kemampuan yang tersedia melalui jaringan dan diakses melalui mekanisme standar, yang mengenalkan penggunaan berbagai platform (misalnya, telepon selular, laptop, dan PDA).
<i>Resource pooling</i>	Penyatuan sumberdaya komputasi yang dimiliki penyedia untuk melayani beberapa konsumen menggunakan model multi-penyewa, dengan sumberdaya fisik dan virtual yang berbeda, ditetapkan secara dinamis dan ditugaskan sesuai dengan permintaan konsumen. Ada rasa kemandirian lokasi bahwa pelanggan umumnya tidak memiliki kontrol atau pengetahuan atas keberadaan lokasi sumberdaya yang disediakan, tetapi ada kemungkinan dapat menentukan lokasi di tingkat yang lebih tinggi (misalnya, negara, negara bagian, atau <i>data center</i>). Contoh sumber daya termasuk penyimpanan, pemrosesan, memori, <i>bandwidth</i> jaringan, dan mesin virtual.
<i>Rapid elasticity</i>	Kemampuan dapat dengan cepat dan elastis ditetapkan.
<i>Measured Service</i>	Sistem komputasi awan secara otomatis mengawasi dan mengoptimalkan penggunaan sumberdaya dengan memanfaatkan kemampuan pengukuran (<i>metering</i>) pada beberapa tingkat yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, <i>bandwidth</i> , dan <i>account</i> pengguna aktif). Penggunaan sumberdaya dapat dipantau, dikendalikan, dan dilaporkan sebagai upaya memberikan transparansi bagi penyedia dan konsumen dari layanan yang digunakan.

2.8 Windows Azure

Berdasarkan informasi yang ada pada situs resmi www.windowsazure.com, Windows Azure adalah *cloud platform* yang terbuka dan fleksibel yang mempermudah pembangunan, penyebaran dan pengelolaan aplikasi melalui jaringan global *data center* milik Microsoft. Azure memungkinkan pembangunan aplikasi menggunakan berbagai bahasa, *tools* dan *framework* serta pengintegrasian aplikasi *cloud* pengguna dengan teknologi informasi yang dimiliki sebelumnya.

Windows Azure memberikan 99.5% SLA bulanan dan memungkinkan pembangunan aplikasi yang memiliki ketersediaan tinggi tanpa perlu berfokus pada infrastruktur. Windows Azure menyediakan layanan yang mempermudah pengguna untuk menambah atau mengurangi kebutuhan sumber daya komputasi saat diperlukan dan hanya perlu membayar sesuai dengan jumlah yang digunakan. Pengguna juga dapat memilih lokasi *data center* untuk menyimpan aplikasinya sehingga dapat lebih dekat dengan pelanggan.

2.9 Virtual Private Server (VPS)

Virtual Private Server atau lebih dikenal dengan singkatan VPS adalah sebuah metode untuk mempartisi sebuah komputer *server* menjadi beberapa *server* yang sepertinya *server-server* tersebut berdiri sendiri (Alamsyah; 2009). Ia seolah-olah sebagai *server* mandiri dan berlaku benar-benar seperti layaknya sebuah komputer. Dengan VPS, efisiensi penggunaan *server* fisik dapat meningkat karena dapat dibagi-bagi sesuai kebutuhan dan seolah-olah memiliki beberapa *server* fisik yang terpisah dan tidak saling mempengaruhi.

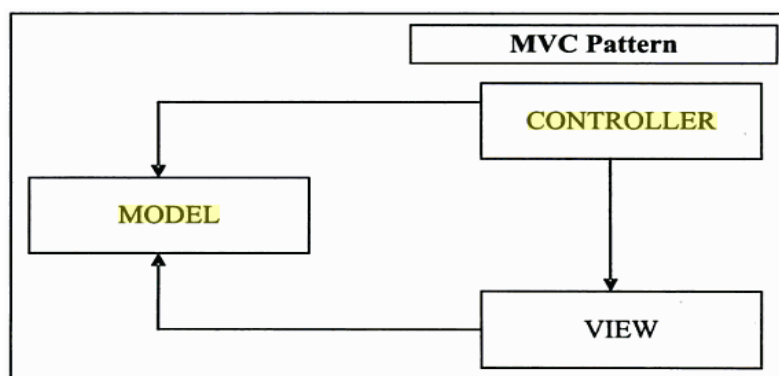
Saat ini, banyak penyedia layanan persewaan VPS di Indonesia. Dengan mekanisme sewa, penyewa tidak perlu melakukan *maintenance server* fisik

sekaligus menghilangkan resiko kerusakan fisik pada server karena hal tersebut merupakan tanggung jawab penyedia layanan. Penyewa hanya perlu melakukan *maintenance server virtual* dan dilakukan secara *remote*. Keamanan fisik server juga terjaga karena disimpan pada *data center* yang tidak diketahui letaknya secara persis. Yang diketahui hanyalah kota atau negara tempat server berada.

2.10 ASP.NET MVC (*Model, View, Controller*)

“ASP.NET merupakan teknologi Microsoft yang dikhususkan untuk pengembangan aplikasi berbasis web dinamis berbasis platform .NET Framework” (Kurniawan; 2010). ASP.NET didesain untuk memberikan kemudahan pada pengembang web untuk membuat aplikasi berbasis web dengan cepat, mudah dan efisien karena meminimalkan kode program dengan bantuan komponen-komponen yang tersedia, sehingga dapat meningkatkan produktivitas.

MVC *pattern* adalah sebuah *pattern* dalam teknik pemrograman yang memisahkan ketiga bagian program menjadi tiga buah objek yang terpisah (Azis; 2005). Interaksi antara ketiga bagian program tersebut digambarkan seperti di bawah ini.



Gambar 2. 7 MVC *Pattern*

Controller merupakan penerima tamu dari permintaan yang datang (*HTTP Request*). Fungsinya adalah untuk mengevaluasi permintaan yang datang, kemudian mengarahkannya ke objek yang tepat untuk memproses permintaan tersebut. *View* adalah hasil akhir yang dikirimkan ke pengguna aplikasi. *Format HTML* merupakan contoh umum hasil akhir yang dikirimkan dalam aplikasi web. *Model* merupakan representasi dari proses bisnis di dalam aplikasi perangkat lunak. Lebih mudahnya, *Model* adalah bagian yang bertugas untuk mengolah data mentah menjadi data yang mengandung arti yang diinginkan oleh pengguna. ASP.NET MVC merupakan komponen yang digunakan untuk mengimplementasikan *pattern MVC* ke dalam *framework ASP.NET*.

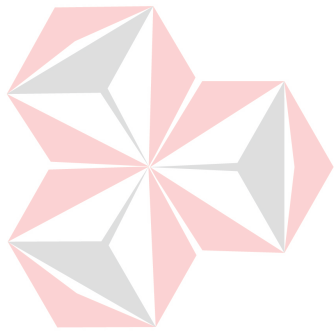
2.11 Azure SQL Database

SQL Server merupakan produk *Database Management System (DBMS)* yang dikeluarkan oleh Microsoft. “DBMS adalah perangkat lunak (*software*) yang berfungsi untuk mengelola *database* mulai dari membuat *database* itu sendiri, sampai dengan proses yang berlaku dalam *database* tersebut, baik berupa *entry*, *edit*, hapus dan lain sebagainya secara efektif dan efisien” (Yuhefizard; 2008).

Sedangkan berdasarkan informasi yang ada pada situs resmi www.windowsazure.com, Azure SQL Database merupakan solusi *cloud* untuk penyimpanan data di *data center* Microsoft sehingga memungkinkan organisasi untuk membuat, menambah dan mengurangi kapasitas serta mengelola basis data mereka secara cepat dengan menggunakan teknologi SQL Server. Layanan ini ditunjang dengan 99.9% SLA dari Microsoft.

2.12 *Black Box Testing*

Black box testing adalah cara pengujian yang terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi (Al Fatta; 2007). Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.



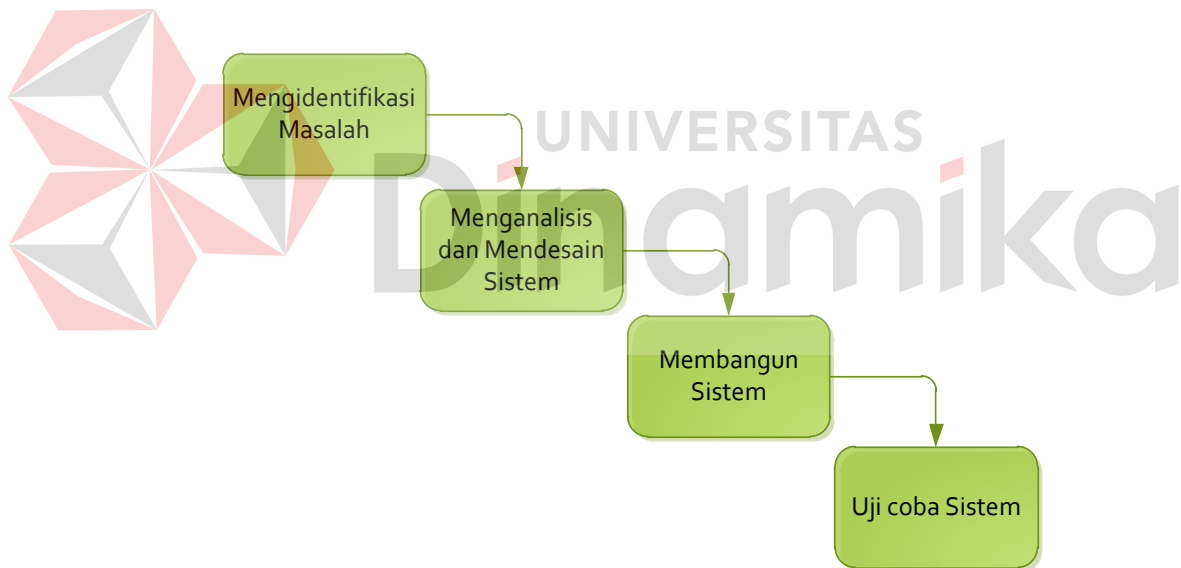
UNIVERSITAS
Dinamika

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Identifikasi Masalah

Sebelum memasuki tahap identifikasi masalah, perlu ditentukan tahap-tahap model pengembangan rancang bangun sistem informasi penjualan berbasis *cloud* ini. Model pengembangan yang digunakan adalah *waterfall* dengan tahapan-tahapan yang telah ditentukan, yaitu mengidentifikasi masalah, menganalisis dan mendesain sistem, membangun sistem dan uji coba sistem. Secara garis besar digambarkan seperti di bawah ini.



Gambar 3. 1 Langkah-Langkah Analisis Sistem

Untuk melakukan identifikasi masalah, maka dilakukan survey pada toko “Toko Besar” sebagai tempat studi kasus untuk mengumpulkan informasi. Toko “Toko Besar” memiliki tiga bagian, yaitu penjualan, pengelola stok persediaan dan pihak manajemen yang mengatur dan mengawasi semua kegiatan bisnis.

Bagian penjualan memiliki tanggung jawab melayani pelanggan dan mencatat semua transaksi penjualan, menerima retur penjualan serta membuat laporan pendapatan. Penjualan saat ini masih melayani transaksi tunai, namun pihak manajemen merencanakan penerimaan transaksi penjualan secara kredit atau cicilan. Bagian penjualan memiliki masalah yaitu sering hilang atau rusaknya nota penjualan yang akan direkap menjadi informasi penjualan. Hilang dan rusaknya nota-nota tersebut menjadikan informasi yang dihasilkan kurang tepat dan tidak bisa dijadikan dasar pengambilan keputusan bagi pihak manajemen. Selain itu, proses perhitungan dan perekapan data yang dilakukan secara *manual* memerlukan waktu yang lama dan sering terjadi kesalahan perhitungan yang disebabkan oleh *human error*.

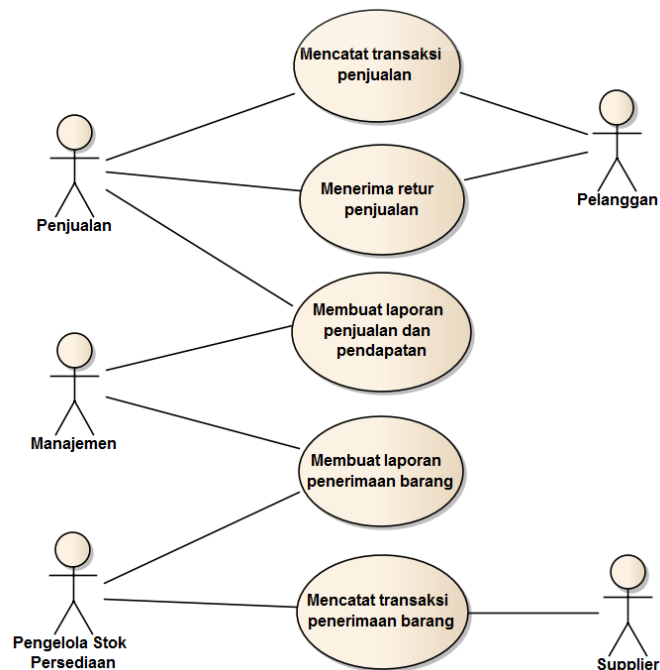
Bagian pengelola stok persediaan memiliki tanggung jawab mencatat transaksi penerimaan barang sekaligus meng-*update* stok dan membuat laporan penerimaan barang. Bagian manajemen memiliki tanggung jawab untuk mengelola dan mengawasi semua kegiatan operasional bisnis berdasarkan informasi yang diberikan oleh bagian penjualan dan pengelola stok persediaan dalam laporan mereka. Informasi-informasi tersebut membantu pihak manajemen untuk mengambil keputusan bisnis. Namun ada permasalahan yang dialami pihak manajemen yaitu tidak bisa mendapatkan informasi-informasi penjualan terkini sesuai yang mereka butuhkan, yaitu tren penjualan, stok terkini hingga *history* transaksi *supplier* dan pelanggan. Hal ini dikarenakan data penjualan dari rekap nota transaksi penjualan tidak diolah secara maksimal.

Saat ini, proses promosi barang dari toko “Toko Besar” hanya melalui *word of mouth*, yaitu rekomendasi dari pelanggan yang puas ke pelanggan lainnya. Hal

ini menyebabkan pelanggan dari toko “Toko Besar” terbatas pada lingkungan mereka saja. Selain itu, toko “Toko Besar” juga kesulitan menyampaikan program-program promosi barang mereka secara luas karena hanya disampaikan ke pelanggan-pelanggan terdekat yang dapat mereka hubungi melalui telepon atau SMS. Hal ini menyebabkan program promosi toko “Toko Besar” tidak diketahui oleh banyak orang yang bisa berpotensi menjadi pelanggan mereka.

3.2 Analisis Sistem

Setelah tahap identifikasi masalah selesai dilakukan, tahap selanjutnya adalah menganalisis sistem penjualan yang telah ada saat ini. Tahap analisis akan menghasilkan kelemahan-kelemahan yang ada pada sistem penjualan saat ini dan kebutuhan-kebutuhan yang perlu dipenuhi. Berikut adalah *business use case diagram* yang merupakan gambaran model bisnis dari hasil identifikasi yang telah dilakukan.



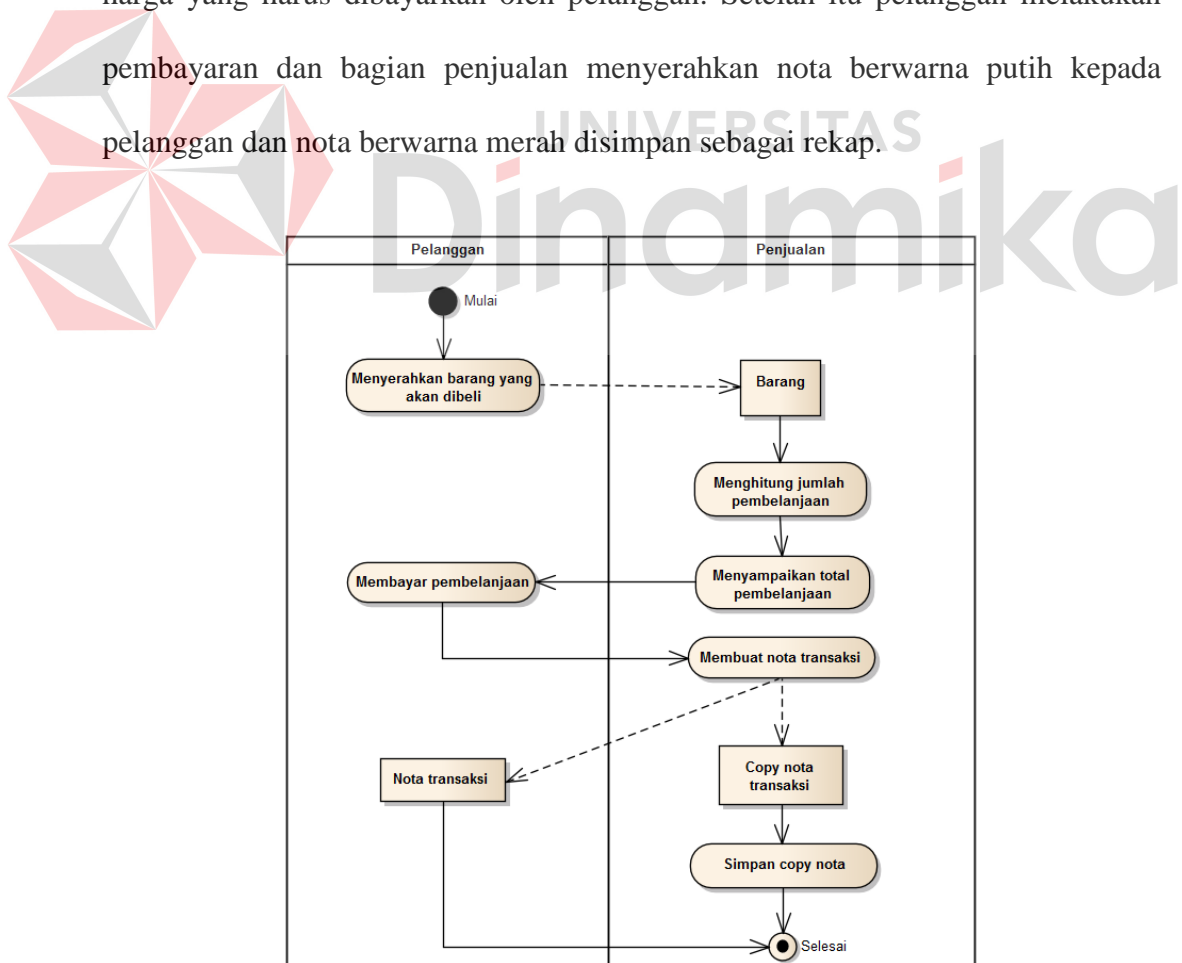
Gambar 3. 2 *Business Use Case*

Berikut ini adalah *activity diagram* yang memberikan gambaran detail tiap proses bisnis dari *business use case diagram* yang telah digambarkan di atas.

3.2.1 Activity Diagram

A. Mencatat Transaksi Penjualan

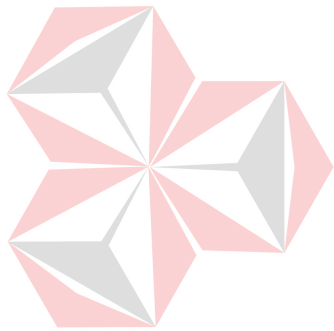
Proses pencatatan transaksi penjualan pada toko “Toko Besar” selama ini dapat dilihat pada gambar 3.3. Proses ini dimulai dari pelanggan memilih dan menyerahkan barang yang dibeli kepada bagian penjualan. Bagian penjualan menghitung barang yang dibeli pelanggan satu persatu dengan kalkulator. Setelah melakukan perhitungan seluruh barang, bagian penjualan akan menyampaikan harga yang harus dibayarkan oleh pelanggan. Setelah itu pelanggan melakukan pembayaran dan bagian penjualan menyerahkan nota berwarna putih kepada pelanggan dan nota berwarna merah disimpan sebagai rekap.



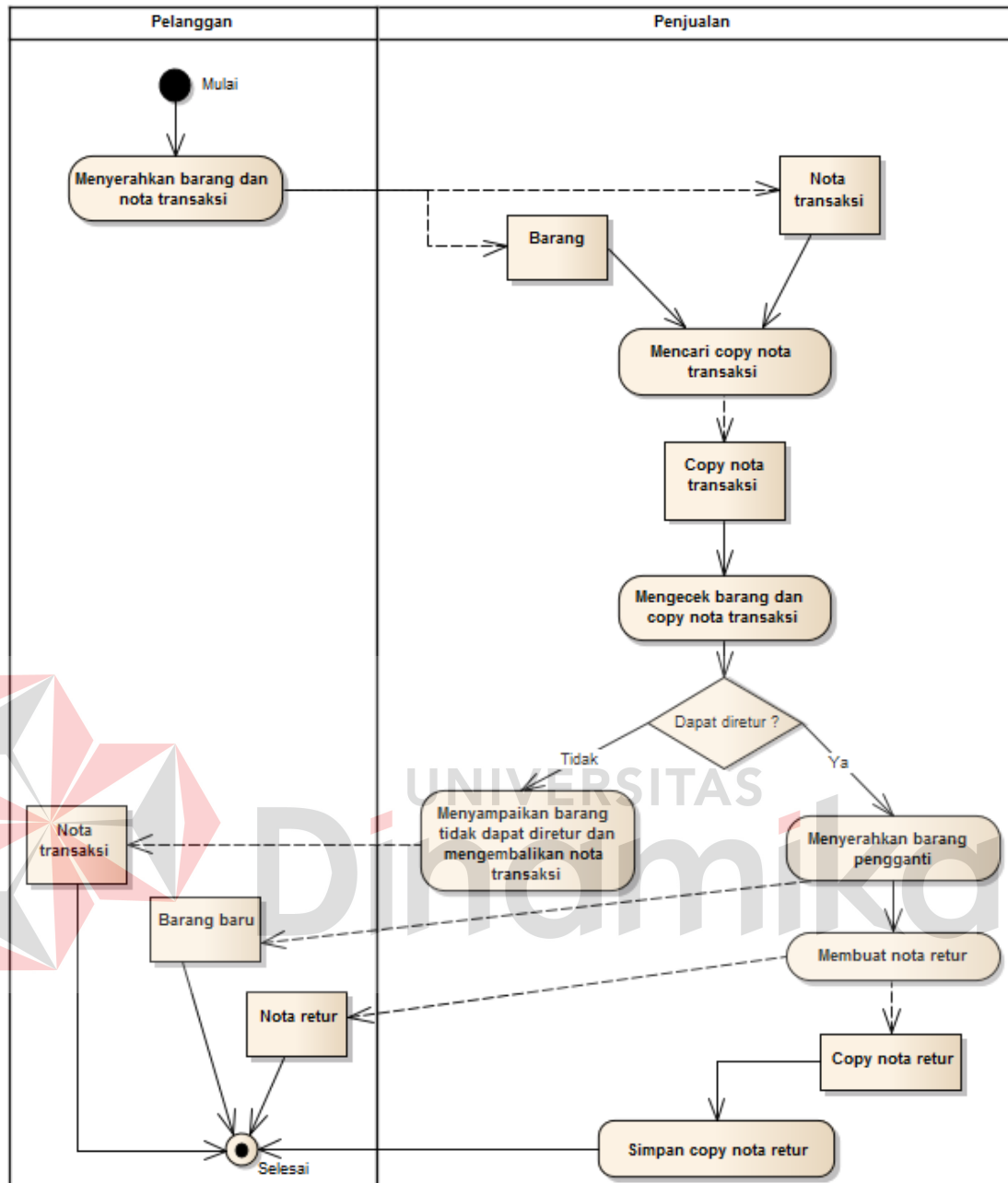
Gambar 3. 3 Activity Diagram Mencatat Transaksi Penjualan

B. Menerima Retur Penjualan

Proses menerima retur penjualan dapat dilihat pada gambar 3.4. Proses ini dimulai dari pelanggan menyerahkan nota transaksi yang akan diretur. Lalu bagian penjualan akan mencari *copy* nota transaksi tersebut. Setelah didapatkan, pelanggan menyerahkan barang yang akan diretur. Bagian penjualan mengecek barang dan *copy* nota transaksi untuk mengetahui apakah barang masih dapat diretur. Jika tidak, bagian penjualan menyampaikannya ke pelanggan dan mengembalikan nota transaksi. Jika ya, bagian penjualan menyerahkan barang pengganti lalu membuat nota retur. Bagian penjualan menyerahkan nota berwarna putih ke pelanggan dan nota berwarna merah disimpan sebagai rekap.



UNIVERSITAS
Dinamika

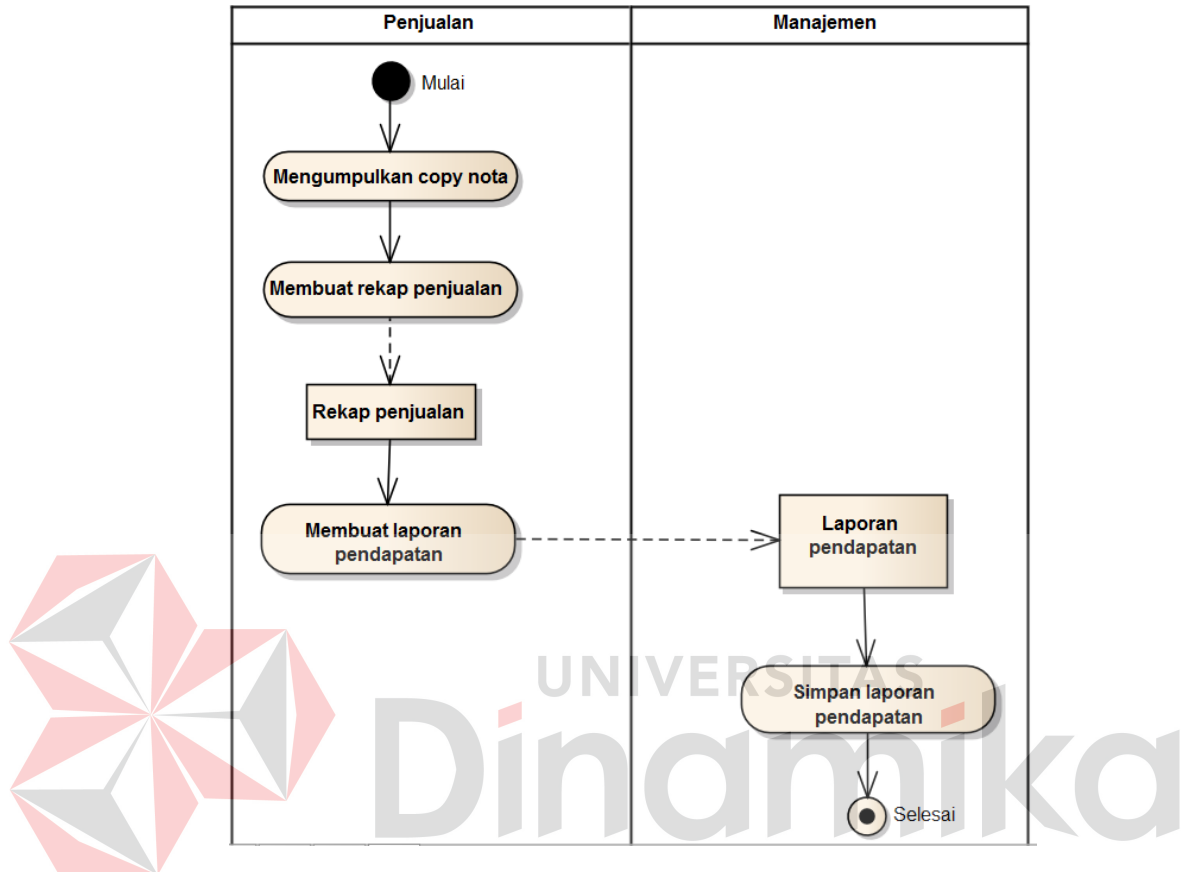


Gambar 3. 4 Activity Diagram Menerima Retur Penjualan

C. Membuat Laporan Penjualan dan Pendapatan

Proses pembuatan laporan penjualan dan pendapatan dapat dilihat pada gambar 3.5. Proses ini dimulai dari bagian penjualan mengumpulkan semua *copy* nota transaksi dan membuat rekap penjualan. Rekap penjualan akan diolah menjadi laporan pendapatan. Setelah itu, laporan pendapatan diserahkan kepada

pihak manajemen untuk disimpan dan dimanfaatkan sebagai pendukung keputusan bisnis yang akan diambil.

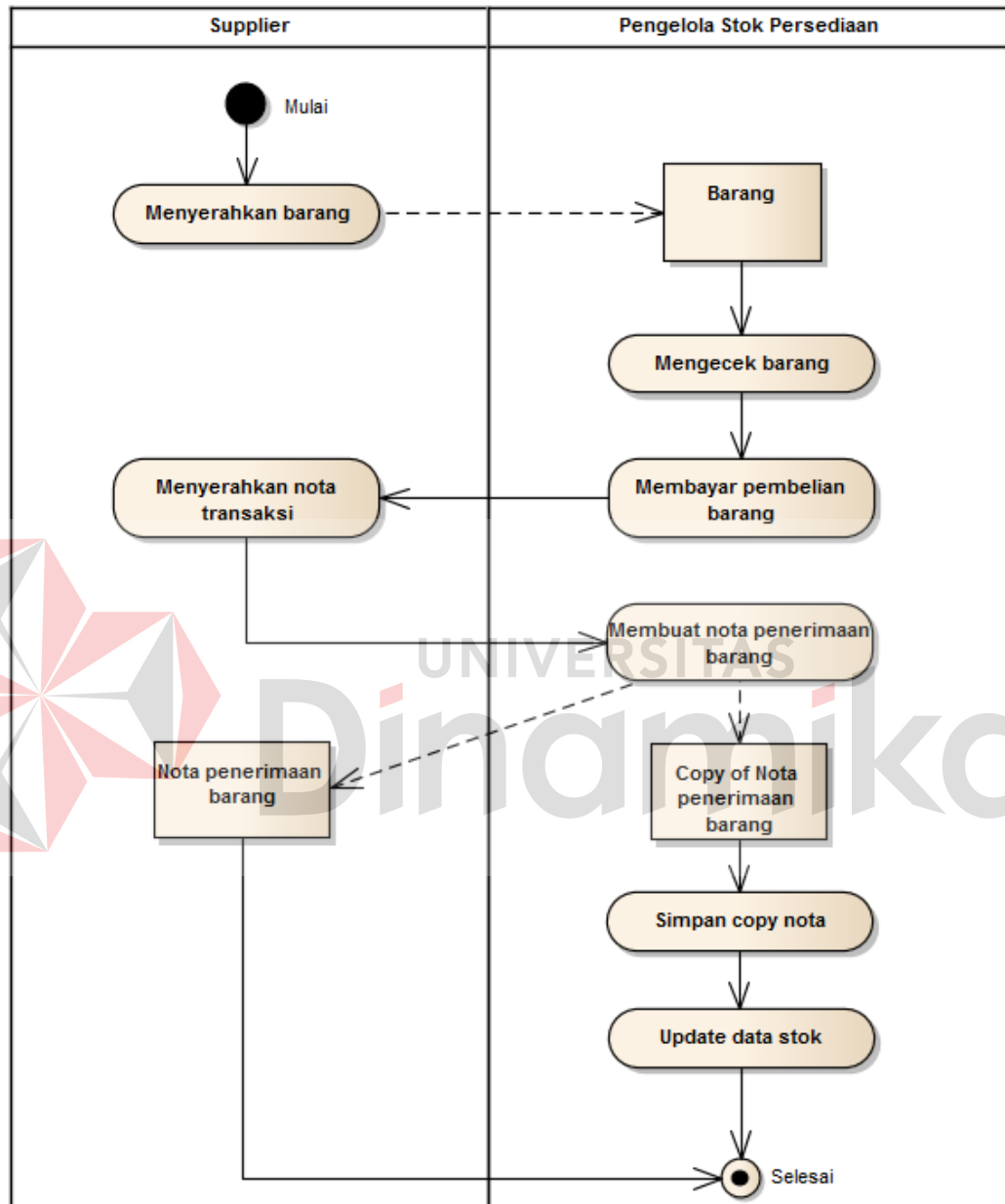


Gambar 3. 5 Activity Diagram Membuat Laporan Penjualan dan Pendapatan

D. Mencatat Transaksi Penerimaan Barang

Proses pencatatan transaksi penerimaan barang pada toko “Toko Besar” selama ini dapat dilihat pada gambar 3.6. Proses ini dimulai dari *supplier* menyerahkan barang ke bagian pengelola stok persediaan. Bagian stok persediaan mengecek kondisi barang yang diterima dan mencocokkan dengan pemesanan yang telah dilakukan. Setelah itu pengelola stok persediaan akan membayar dan menerima nota transaksi dari *supplier*. Bagian pengelolaan stok persediaan akan membuat nota penerimaan barang dua rangkap lalu menyerahkan nota berwarna

putih kepada *supplier* dan nota berwarna merah akan disimpan sebagai rekap. Bagian pengelola stok persediaan juga akan melakukan *update* data stok barang.

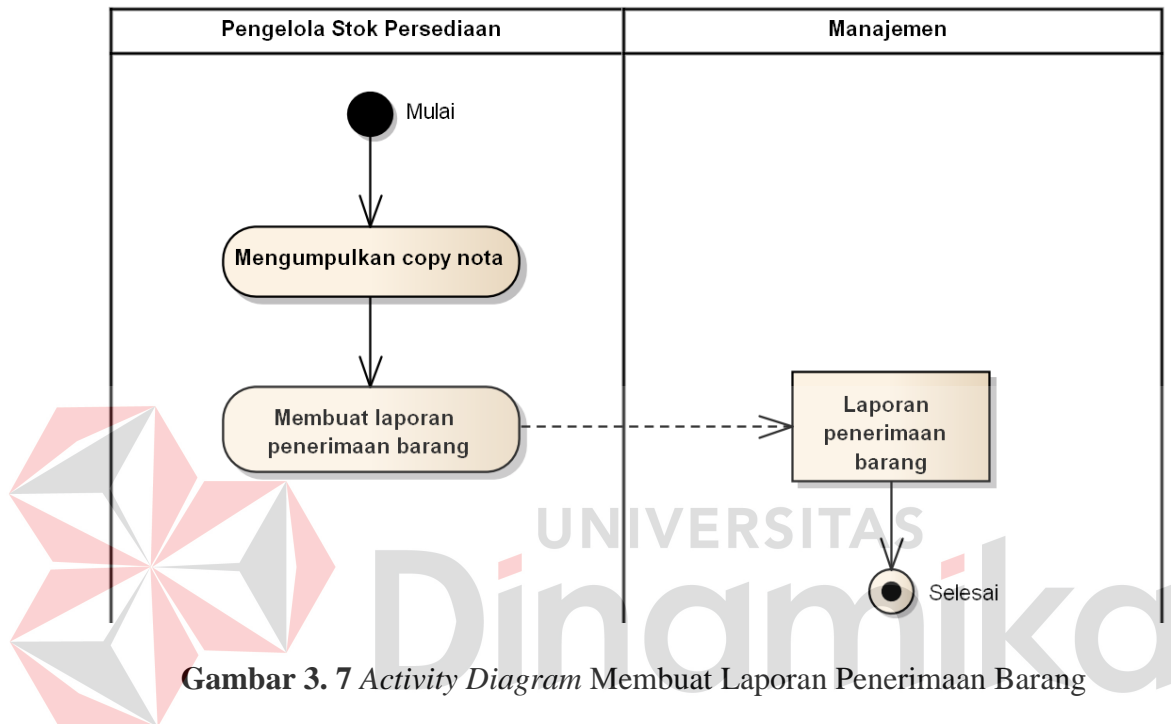


Gambar 3. 6 Activity Diagram Mencatat Transaksi Penerimaan Barang

E. Membuat Laporan Penerimaan Barang

Proses pembuatan laporan penjualan dan pendapatan dapat dilihat pada gambar 3.7. Proses ini dimulai dari bagian pengelola stok persediaan

mengumpulkan semua *copy* nota transaksi dan mengolahnya menjadi laporan penerimaan barang. Setelah itu, laporan penerimaan barang akan diserahkan kepada pihak manajemen untuk disimpan dan dimanfaatkan sebagai pendukung keputusan bisnis yang akan diambil.



Gambar 3. 7 Activity Diagram Membuat Laporan Penerimaan Barang

Berdasarkan identifikasi masalah di atas, dapat disimpulkan bahwa toko “Toko Besar” membutuhkan sistem informasi yang dapat mencatat kegiatan transaksi penjualan dan menghasilkan informasi penjualan untuk membantu pihak manajemen dalam mengambil keputusan. Informasi-informasi yang diperlukan adalah tren penjualan, stok terkini serta *history* transaksi *supplier* dan pelanggan. Sistem informasi yang dibangun juga harus memenuhi kebutuhan promosi barang di Facebook berdasarkan stok dan harga terkini.

3.2.2 Hasil Analisis

Dari proses analisis, didapatkan kelemahan-kelemahan sistem yang lama. Untuk mengatasi kelemahan-kelemahan sistem yang lama, perlu dibuat sebuah sistem informasi penjualan baru yang dapat memenuhi kebutuhan manajemen. Hasil identifikasi permasalahan pada toko “Toko Bagus” adalah sebagai berikut :

A. Kelemahan Sistem Penjualan Saat Ini

1. Sering terjadi kesalahan perhitungan jumlah barang yang dibeli beserta harga yang diberikan dalam kegiatan pencatatan transaksi penjualan yang masih konvensional dengan bantuan nota dan kalkulator, terutama ketika transaksi dengan jenis dan jumlah barang yang banyak.
2. Manajemen tidak dapat mengetahui informasi tren penjualan dan *history* transaksi dari pelanggan dan supplier. Informasi tren penjualan diperlukan untuk mengetahui *lifetime* sebuah produk dan menentukan jumlah pemesanan kepada *supplier* pada periode berikutnya. Informasi *history* transaksi dari pelanggan dan supplier untuk mengetahui siapa yang telah melakukan banyak transaksi. Informasi tersebut diperlukan sebagai *input* bagi strategi marketing dalam menjaga relasi dengan pelanggan atau *supplier* utama.
3. Manajemen kesulitan mengetahui stok terkini karena harus menghitung terlebih dahulu atau berdasarkan nota rekap. Hal ini menyebabkan manajemen tidak dapat memperoleh informasi stok secara *up to date* sehingga sering mengalami kehabisan stok atau penumpukan stok karena kesalahan perkiraan waktu saat *reorder* ke *supplier*. Jika kehabisan stok, UKM akan kehilangan peluang untuk memperoleh keuntungan lebih. Jika

stok menumpuk, UKM terancam rugi karena barang kadaluarsa dari segi masa pemakaian atau minat pasar

B. Kebutuhan Sistem Informasi Penjualan

Untuk mengatasi kelemahan-kelemahan yang ada dalam sistem penjualan saat ini, maka kebutuhan sistem informasi penjualan didefinisikan seperti di bawah ini.

1. Sistem informasi penjualan yang membantu pencatatan transaksi penjualan dan meminimalkan kesalahan perhitungan jumlah barang yang dibeli beserta harga yang diberikan, terutama ketika transaksi dengan jenis dan jumlah barang yang banyak.
2. Sistem informasi penjualan yang menghasilkan informasi tren penjualan dan *history* transaksi dari pelanggan dan supplier.
3. Sistem informasi penjualan yang menghasilkan informasi stok terkini untuk memperkirakan waktu *reorder* yang tepat ke *supplier*.

Selain untuk mengatasi kelemahan sistem penjualan saat ini dan memenuhi kebutuhan informasi manajemen, sistem yang dibangun juga memberikan keuntungan-keuntungan yang dapat dimanfaatkan oleh toko, yaitu :

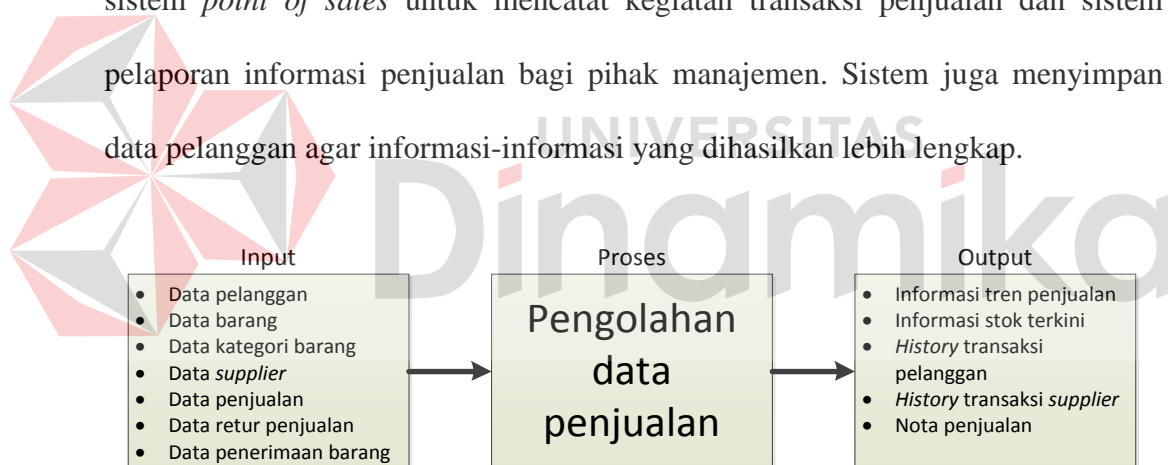
1. Sistem informasi penjualan yang dapat digunakan sebagai media promosi barang di jejaring sosial Facebook berdasarkan data terkini. Promosi dapat dilakukan untuk masing-masing barang, barang terlaris atau diskon terbesar.
2. Sistem informasi penjualan yang dapat diakses kapanpun dan dimanapun melalui internet serta disimpan di *data center* dengan SLA terjamin.

3.3 Mendesain Sistem

Desain atau perancangan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Sistem informasi penjualan yang dibangun akan terdiri dari sistem *point of sales* untuk mencatat kegiatan transaksi penjualan dan sistem pelaporan informasi penjualan bagi pihak manajemen. Sistem juga menyimpan data pelanggan agar informasi-informasi yang dihasilkan lebih lengkap.

3.3.1 Model Pengembangan

Desain atau perancangan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Sistem informasi penjualan yang dibangun terdiri dari sistem *point of sales* untuk mencatat kegiatan transaksi penjualan dan sistem pelaporan informasi penjualan bagi pihak manajemen. Sistem juga menyimpan data pelanggan agar informasi-informasi yang dihasilkan lebih lengkap.



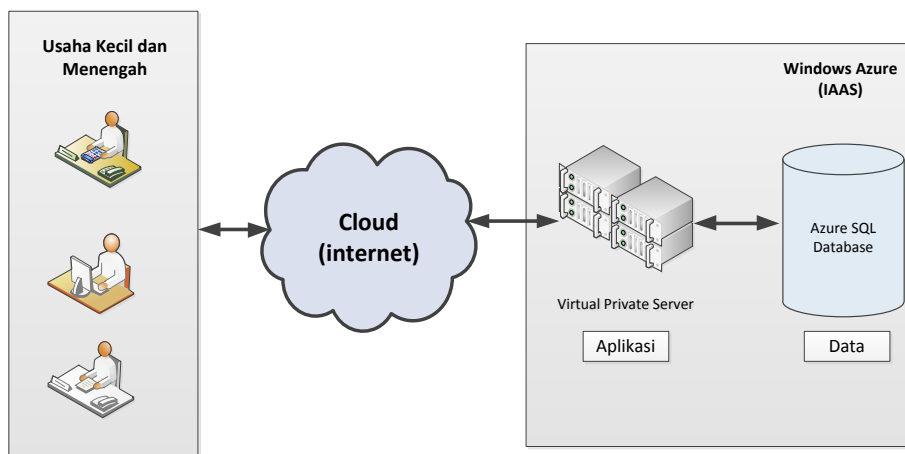
Gambar 3. 8 Block Diagram

Untuk *input* data penjualan, sistem akan memiliki modul kasir atau *point of sales* yang terintegrasi dengan *barcode reader* dan *printer* nota. Proses pengolahan data penjualan terdiri atas beberapa sub proses, yaitu :

- a. Menghitung keuntungan, jumlah penjualan dan nilai penjualan pada periode harian, bulanan dan tahunan.
- b. Menampilkan informasi tren penjualan barang berdasarkan transaksi penjualan yang telah dilakukan.

- c. Menghitung transaksi pelanggan dan *supplier* serta menentukan pelanggan dan *supplier* dengan transaksi terbanyak.
- d. Menganalisa transaksi dari pelanggan dan *supplier* serta menampilkan *history* barang-barang yang dibeli pada periode tertentu.

Sistem informasi penjualan yang dibangun merupakan sistem berbasis *cloud* sehingga sistem dan data disimpan di *data center* milik penyedia layanan infrastruktur *cloud*, Microsoft (Windows Azure). Arsitektur sistem yang akan dibangun adalah seperti pada gambar 3.9. Sistem disimpan pada *Virtual Private Server* atau *Virtual Machine* dan data akan disimpan pada *Azure SQL Database*. Lokasi *data center* yang dipilih adalah wilayah *Southeast Asia* karena merupakan pilihan lokasi terdekat dari pengguna yang berada di Indonesia. Penggunaan *Azure SQL Database* sebagai *database server* dilakukan karena ukuran *Azure SQL Database* dapat dinaikkan dan diturunkan sewaktu-waktu sesuai dengan kebutuhan sistem sehingga biaya yang dibayarkan untuk penyimpanan data di *cloud* benar-benar sesuai dengan yang digunakan.



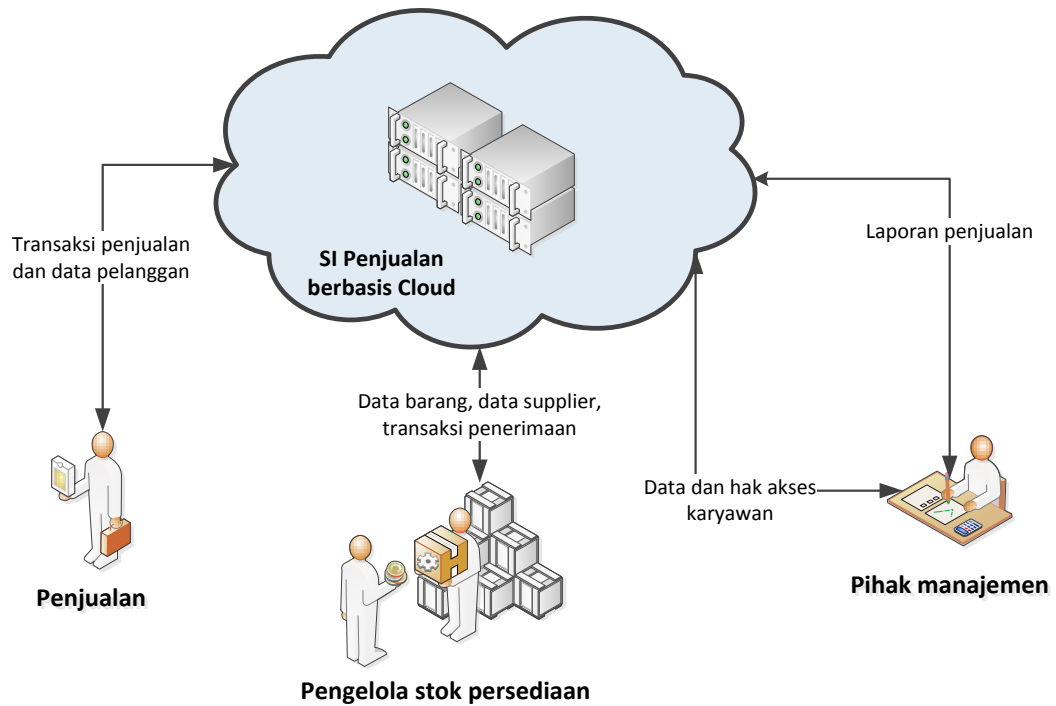
Gambar 3. 9 Model Arsitektur Sistem Informasi Penjualan berbasis *Cloud*

Dengan arsitektur sistem berbasis *cloud*, pengguna tidak perlu melakukan pengelolaan dan perawatan infrastruktur sistem karena hal tersebut akan dilakukan oleh penyedia layanan infrastruktur. Pengguna cukup menentukan layanan yang sesuai dengan kebutuhan mereka dan membayar biaya layanan. Untuk mengakses sistem, pengguna perlu menyediakan perangkat akses seperti komputer, laptop, *tablet* atau *smartphone* serta koneksi internet.

Model pengguna utama dari sistem didefinisikan sesuai fungsi dan tanggung jawabnya seperti di bawah ini.

Tabel 3. 1 Model Pengguna

Pengguna	Tanggung Jawab	Fungsi
Pihak manajemen	Memantau penjualan melalui laporan rutin.	Membaca informasi penjualan terkini melalui fitur <i>report</i> dan <i>history</i> .
	Mengelola data karyawan beserta hak aksesnya	Mengatur hak akses karyawan terhadap sistem
Bagian penjualan	Melayani penjualan dengan cepat dan mencatat semua transaksi penjualan yang dilakukan.	Mencatat transaksi penjualan dan mengelola data pelanggan.
Bagian pengelola stok persediaan	Mengawasi dan mengontrol persediaan stok barang.	Memperoleh informasi stok terkini, stok siap jual dan stok minimum.
	Mengupdate stok barang saat menerima barang dari supplier serta mencatat transaksi penerimaan barang	Mencatat transaksi penerimaan barang, mengubah data stok barang dan mengelola data <i>supplier</i> .

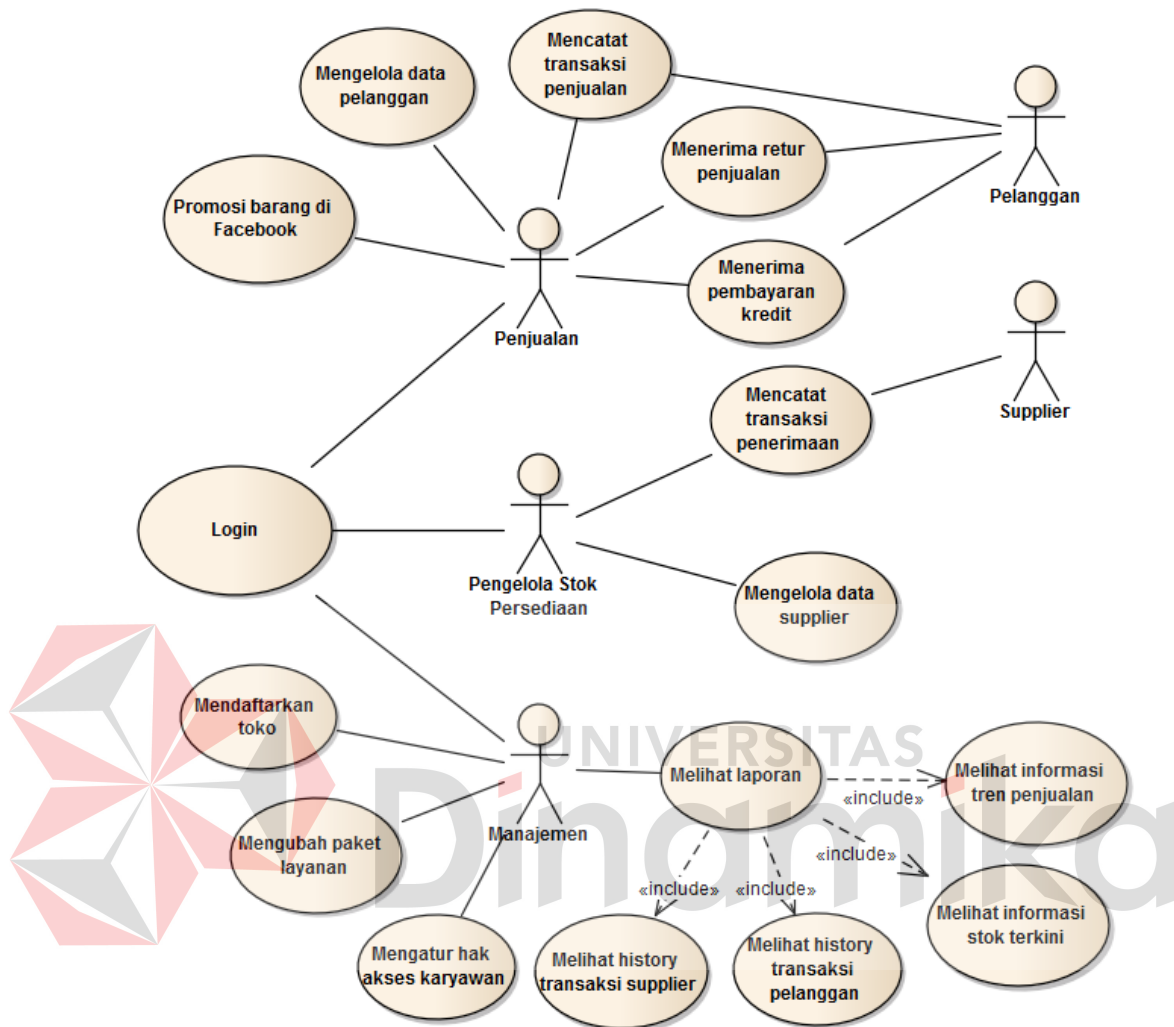


Gambar 3. 10 Interaksi Pengguna

Walaupun model pengguna telah didefinisikan di atas, UKM tetap dapat melakukan perubahan model pengguna sesuai kebutuhan mereka dengan mengatur hak akses pengguna terhadap sistem.

Model perancangan dan pembangunan sistem akan menggunakan konsep *object oriented*. Konsep ini dipilih karena lebih mudah dalam mengakomodasi perubahan, mengurangi kompleksitas perancangan dan desain serta kemampuan untuk bisa digunakan kembali (Hendri; 2012). *Pattern* yang digunakan ialah MVC agar pemisahan *layer* aplikasi semakin jelas dan mempermudah pengorganisasian kode program.

3.3.2 Use Case Diagram Sistem Informasi Penjualan



Gambar 3. 11 Use Case Diagram Sistem Informasi Penjualan

Pada *use case diagram* di gambar 3.11, sistem informasi penjualan yang akan dibangun memiliki 5 aktor. Aktor dalam *use case* merupakan semua yang ada di luar ruang lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut (Sholiq; 2010). Aktor penjualan, pengelola stok persediaan dan manajemen merupakan aktor internal dari toko “Toko Besar”. Ketiga aktor tersebut memberikan *input* dan menerima *output* dari sistem sesuai dengan peran dan tanggung jawab mereka. Aktor *supplier* dan *pelanggan*

merupakan aktor eksternal dari toko “Toko Besar”. Kedua aktor tersebut menerima *output* dari sistem berupa nota transaksi yang telah mereka lakukan.

3.3.3 Flow of Events

Setelah *use case diagram* digambarkan, hal selanjutnya yang dilakukan ialah membuat *flow of events* dari masing-masing *use case*. *Flow of events* bertujuan untuk mendokumentasikan alur logika dalam *use case* yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan (Sholiq; 2010).

A. Flow of Events Use Case Mendaftarkan Toko

Proses mendaftar toko dilakukan sebelum sebuah toko dapat menggunakan sistem informasi penjualan berbasis *cloud* yang akan dibangun. Pendaftaran dilakukan dengan mengisi *form* pendaftaran yang disediakan dan melakukan konfirmasi melalui *email* yang dikirimkan oleh sistem. Pengguna bisa mengakses halaman *login* jika proses pendaftaran telah dilalui. *Flow of events* proses mendaftar toko disajikan pada tabel 3.2.

Tabel 3. 2 *Flow of Events* Mendaftarkan Toko

Nama Use Case	Mendaftarkan toko
Kebutuhan terkait	Sebelum dapat menggunakan sistem informasi penjualan berbasis <i>cloud</i> , toko harus melakukan pendaftaran akun dengan mengisi data toko. Setelah semua data diisi, sistem akan mengirimkan <i>email</i> ke pendaftar untuk konfirmasi pendaftaran. Setelah konfirmasi pendaftaran dilakukan, akun toko telah aktif dan dapat diakses dengan menggunakan <i>id</i> toko.
Tujuan	Menangani proses pendaftaran toko.
Prasyarat	Tidak ada
Kondisi akhir sukses	Toko telah berhasil terdaftar.
Kondisi akhir gagal	Pengguna mendapatkan pesan pendaftaran toko gagal.

Aktor utama	Manajemen	
Aktor sekunder	Tidak ada	
Pemicu	Pengguna membuka halaman pendaftaran pada website sistem informasi penjualan.	
Primary Flow	Langkah	Aksi
	1	Pengguna membuka halaman pendaftaran pada website sistem informasi penjualan.
	2	Pengguna memasukkan <i>id</i> toko, nama toko, <i>password</i> dan <i>email</i> pada <i>form</i> pendaftaran yang disediakan.
	3	Sistem memeriksa <i>format</i> masukan dari pengguna. Jika ada <i>format</i> yang salah, masuk ke langkah 3.1 .
	4	Sistem melakukan koneksi ke <i>database</i> untuk memeriksa apakah <i>id</i> toko telah digunakan. Jika <i>id</i> toko telah digunakan, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem melakukan koneksi <i>database</i> untuk menyimpan data toko baru. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1 .
	6	Sistem mengirim <i>email</i> ke alamat <i>email</i> yang didaftarkan pengguna untuk melakukan konfirmasi pendaftaran. Jika pengiriman <i>email</i> gagal, masuk ke langkah 6.1 .
	7	Pengguna mengaktifkan akun toko melalui <i>email</i> konfirmasi yang dikirimkan sistem.
	8	Sistem melakukan koneksi <i>database</i> untuk mengaktifkan toko dan menampilkan halaman <i>login</i> ke pengguna sebagai tanda akun toko telah diaktifkan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 8.1 .
	9	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan <i>format</i> data pendaftaran toko salah dan meminta pengguna mengulangi langkah ke-2.
	4.1	Sistem menampilkan pesan <i>id</i> toko telah digunakan dan meminta pengguna mengulangi langkah ke-2.
Error Flow	Langkah	Aksi Percabangan
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk memeriksa <i>id</i> toko gagal.

		Pengguna kembali ke langkah 2.
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data toko baru gagal. Pengguna kembali ke langkah 2.
	6.1	Sistem menampilkan pesan pengiriman <i>email</i> gagal dan meminta pengguna mengulang langkah ke-2.
	8.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengaktifkan akun toko gagal. Pengguna kembali ke langkah 7.

B. *Flow of Events Use Case Login*

Proses *login* dilakukan sebelum pengguna masuk ke dalam sistem dan mengakses menu-menu yang ada. Proses *login* dilakukan untuk memastikan apakah orang yang mengakses sistem merupakan orang yang memiliki hak akses untuk membuka sistem tersebut. Sistem memberikan hak akses kepada pengguna berdasarkan *login* yang dilakukannya. *Flow of events* proses *login* disajikan pada tabel 3.3.

Tabel 3. 3 *Flow of Events Login*

Nama Use Case	Login
Kebutuhan terkait	Salah satu proses untuk masuk ke dalam sistem adalah <i>login</i> . Pengguna memiliki data berupa <i>id</i> toko, <i>id</i> karyawan dan <i>password</i> yang harus diverifikasi oleh sistem agar pengguna dapat melakukan proses bisnis pada sistem sesuai hak akses yang dimiliki.
Tujuan	Menangani proses <i>login</i> pengguna.
Prasyarat	Tidak ada
Kondisi akhir sukses	Pengguna berhasil <i>login</i> dan dapat mengakses menu di halaman utama berdasarkan hak akses yang telah dimiliki.
Kondisi akhir gagal	Pengguna mendapatkan pesan gagal <i>login</i> dan mengulang pengisian ulang pada <i>form login</i> .
Aktor utama	Pengguna (Bagian penjualan, pengelola stok persediaan dan manajemen)
Aktor sekunder	Tidak ada
Pemicu	Pengguna membuka halaman sistem melalui <i>web</i>

	<i>browser.</i>	
Primary Flow	Langkah	Aksi
	1	Pengguna membuka halaman sistem melalui web <i>browser</i> .
	2	Pengguna memasukkan <i>id</i> toko.
	3	Sistem melakukan koneksi ke <i>database</i> dan melakukan verifikasi <i>id</i> toko. Jika <i>id</i> toko belum terdaftar, masuk ke langkah 3.1 . Jika <i>id</i> toko belum diaktifkan melalui konfirmasi <i>email</i> saat pendaftaran, masuk ke langkah 3.2 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.3 .
	4	Sistem menampilkan halaman <i>login</i> toko.
	5	Pengguna mengisi <i>id</i> karyawan dan <i>password</i> .
	6	Sistem melakukan koneksi ke <i>database</i> dan melakukan verifikasi <i>id</i> karyawan dan <i>password</i> . Jika kombinasi <i>id</i> karyawan dan <i>password</i> salah, masuk ke langkah 6.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 6.2 .
	7	Sistem menampilkan halaman utama.
	8	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan <i>id</i> toko belum terdaftar dan meminta pengguna mengulangi langkah ke-2.
	3.2	Sistem menampilkan pesan <i>id</i> toko belum diaktifkan dan meminta pengguna melakukan konfirmasi melalui <i>email</i> . Sistem juga menawarkan pengiriman ulang <i>email</i> konfirmasi. Setelah itu kembali ke langkah ke-2.
	6.1	Sistem menampilkan pesan <i>id</i> karyawan atau <i>password</i> salah dan meminta pengguna mengulangi langkah ke-5.
Error Flow	Langkah	Aksi Percabangan
	3.3	Sistem menampilkan pesan koneksi <i>database</i> untuk melakukan verifikasi <i>id</i> toko gagal. Pengguna kembali ke langkah 2
	6.2	Sistem menampilkan pesan koneksi <i>database</i> untuk melakukan verifikasi <i>id</i> karyawan dan <i>password</i> gagal. Pengguna kembali ke langkah 5 .

C. *Flow of Events Use Case Mengelola Data Pelanggan*

Proses mengelola data pelanggan dilakukan untuk memastikan data pelanggan yang dimiliki ialah data terkini yang berhasil didapatkan dari pelanggan. Proses ini meliputi menambah, mengubah, menghapus dan mencari data pelanggan. Proses ini dilakukan ketika toko mendapatkan informasi terbaru tentang pelanggan mereka. *Flow of events* proses mengelola data pelanggan disajikan pada tabel 3.4.

Tabel 3. 4 *Flow of Events* Mengelola Data Pelanggan

Nama Use Case		Mengelola data pelanggan
Kebutuhan terkait	Proses mengelola data pelanggan meliputi proses menambah, mengubah, menghapus dan mencari data pelanggan.	
Tujuan	Memastikan bahwa data pelanggan yang dimiliki merupakan data terkini berdasarkan yang didapatkan dari pelanggan. Data ini dapat digunakan toko untuk berinteraksi dengan pelanggan.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola data pelanggan.	
Kondisi akhir sukses	Data pelanggan telah diperbaharui.	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Bagian Penjualan	
Aktor sekunder	Tidak ada	
Pemicu	Bagian penjualan memilih menu "Pelanggan".	
Primary Flow	Langkah	Aksi
	1	Bagian penjualan memilih menu "Pelanggan"
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data pelanggan dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1 .
	3	Bagian penjualan melakukan operasi mencari, menambah, mengubah atau menghapus data pelanggan.

	4	Sistem melakukan koneksi <i>database</i> untuk menyimpan semua perubahan yang dilakukan bagian penjualan terhadap data pelanggan. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data pelanggan berhasil.
	6	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data pelanggan yang dimasukkan salah dan meminta bagian penjualan kembali ke langkah 3 untuk mengisi ulang masukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data pelanggan gagal. Bagian penjualan kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mencari, menambah, mengubah atau menghapus data pelanggan gagal. Bagian penjualan kembali ke langkah 3 .

D. *Flow of Events Use Case Mencatat Transaksi Penjualan*

Proses mencatat transaksi penjualan dilakukan dengan cara yang sama dengan mesin kasir. Bagian penjualan memasukkan nama atau *id* barang yang dibeli beserta jumlahnya. Bagian penjualan juga dapat mencatat data pelanggan. Sistem akan menghitung jumlah yang harus dibayar, mencetak nota penjualan, mengurangi data stok terkini dan menyimpan data transaksi penjualan tersebut. Pembayaran dapat dilakukan dengan 2 cara, yaitu tunai dan cicilan. *Flow of events* proses mencatat transaksi penjualan disajikan pada tabel 3.5.

Tabel 3. 5 *Flow of Events* Mencatat Transaksi Penjualan

Nama Use Case	Mencatat transaksi penjualan	
Kebutuhan terkait	Penjualan dapat dilakukan secara tunai ataupun kredit. Bagian penjualan dapat memberikan diskon khusus pada masing-masing barang ataupun keseluruhan transaksi sesuai proses tawar-menawar dengan pelanggan. Bagian penjualan juga perlu mencatat data pelanggan yang membeli.	
Tujuan	Menangani penjualan tunai ataupun kredit.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat transaksi penjualan.	
Kondisi akhir sukses	Transaksi penjualan sukses dilakukan.	
Kondisi akhir gagal	Jika stok barang habis, maka transaksi gagal disimpan dan pesan "Stok habis" diberikan.	
Aktor utama	Bagian Penjualan	
Aktor sekunder	Pelanggan	
Pemicu	Bagian penjualan memilih menu "Penjualan".	
Primary Flow	Langkah	Aksi
	1	Bagian penjualan memilih menu "Penjualan".
	2	Bagian penjualan memasukkan nama atau <i>id</i> barang, jumlah yang dibeli dan diskon masing-masing barang (jika ada).
	3	Sistem melakukan koneksi <i>database</i> untuk memeriksa apakah jumlah yang dibeli tidak melebihi stok siap jual terkini. Jika melebihi, masuk ke langkah 3.1. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.2.
	4	Bagian penjualan memasukkan nama pelanggan. Jika nama pelanggan belum terdaftar, masuk ke langkah 4.1.
	5	Bagian penjualan memasukkan diskon keseluruhan transaksi (jika ada).
	6	Sistem menampilkan total harga.
	7	Bagian penjualan memasukkan jenis transaksi (tunai atau kredit).

	8	Bagian penjualan memasukkan jumlah uang yang dibayarkan pelanggan. Jika transaksi tunai, bagian penjualan memasukkan jumlah uang sesuai total harga. Jika jumlah uang yang dibayarkan pelanggan kurang, masuk ke langkah 8.1 . Jika transaksi kredit, bagian penjualan memasukkan uang muka, jumlah cicilan, periode pembayaran, bunga dan denda keterlambatan tiap periode. Jika <i>format</i> data masukan transaksi penjualan kredit salah, masuk ke langkah 8.2 .
	9	Sistem menampilkan jumlah kembalian.
	10	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan data transaksi dan <i>update</i> stok. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 9.1 .
	11	Sistem mencetak nota penjualan dan bagian penjualan menyerahkannya ke pelanggan.
	12	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan "stok barang tidak mencukupi". Bagian penjualan kembali ke langkah 2 untuk memasukkan nama atau <i>id</i> barang lainnya.
	4.1	Bagian penjualan memasukkan data pelanggan baru pada <i>form</i> yang disediakan dan menyimpannya. Jika ada <i>format</i> data yang salah, masuk ke langkah 4.2 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.3 .
	4.2	Sistem menampilkan pesan <i>format</i> data pelanggan baru yang dimasukkan salah dan meminta bagian penjualan kembali ke langkah 4.1 untuk mengisi ulang masukan dengan <i>format</i> yang benar.
	8.1	Sistem menampilkan pesan "uang yang dibayarkan kurang" dan meminta bagian penjualan mengulangi langkah ke- 8 .
	8.2	Sistem menampilkan pesan <i>format</i> data transaksi penjualan kredit yang dimasukkan salah dan meminta bagian penjualan kembali ke langkah 8 untuk mengisi ulang masukan dengan <i>format</i> yang benar.

<i>Error Flow</i>	Langkah	Aksi Percabangan
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk memeriksa jumlah stok barang gagal. Bagian penjualan kembali ke langkah 2.
	4.3	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data pelanggan baru gagal. Bagian penjualan kembali ke langkah 4.1.
	9.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi penjualan gagal. Bagian penjualan kembali ke langkah 8.

E. *Flow of Events Use Case Menerima Retur Penjualan*

Proses menerima retur penjualan merupakan proses yang dilakukan ketika pelanggan mengembalikan barang yang dibeli karena kondisi rusak. Data retur penjualan perlu dicatat agar toko memiliki dokumentasi retur barang yang lengkap dan dapat dijadikan informasi bagi pihak manajemen. *Flow of events* proses menerima retur penjualan disajikan pada tabel 3.6.

Tabel 3. 6 *Flow of Events* Menerima Retur Penjualan

Nama Use Case	Menerima retur penjualan	
Kebutuhan terkait	Retur penjualan dilakukan ketika pelanggan mengembalikan barang yang rusak. Retur harus berdasarkan data transaksi penjualan yang pernah dilakukan untuk memastikan kevalidan barang yang diretur.	
Tujuan	Menangani retur penjualan.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat retur penjualan.	
Kondisi akhir sukses	Retur penjualan sukses dilakukan.	
Kondisi akhir gagal	Jika tidak ditemukan transaksi penjualan dari barang yang akan diretur maka retur tidak dapat dilakukan.	
Aktor utama	Bagian Penjualan	
Aktor sekunder	Pelanggan	
Pemicu	Bagian penjualan melihat <i>history</i> penjualan dan memilih "Retur".	
Primary Flow	Langkah	Aksi

	1	Bagian penjualan membuka <i>history</i> penjualan.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data <i>history</i> penjualan yang pernah dilakukan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 2.1 .
	3	Bagian penjualan memilih transaksi yang akan diretur.
	4	Sistem melakukan koneksi <i>database</i> untuk mengambil data barang dari transaksi yang dipilih. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 4.1 .
	5	Bagian penjualan memilih barang yang akan diretur dan memasukkan jumlahnya. Jika jumlah yang diretur melebihi jumlah yang dibeli, masuk ke langkah 5.1 .
	6	Sistem melakukan koneksi ke <i>database</i> dan menyimpan transaksi retur. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 6.1 .
	7	Sistem mencetak nota retur dan bagian penjualan menyerahkannya ke pelanggan.
	8	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	5.1	Sistem menampilkan pesan jumlah yang diretur tidak dapat melebihi jumlah yang terjual dan meminta bagian penjualan mengulangi langkah ke-5.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>history</i> penjualan gagal. Bagian penjualan kembali ke langkah 1 .
	4.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data barang gagal. Bagian penjualan kembali ke langkah 3 .
	6.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi retur gagal. Bagian penjualan kembali ke langkah 5 .

F. *Flow of Events Use Case Menerima Pembayaran Kredit*

Proses pembayaran kredit atau cicilan dicatat oleh bagian penjualan berdasarkan data transaksi penjualan kredit yang telah disimpan. Nominal pembayaran kredit dan jangka waktu pembayaran bersifat fleksibel sesuai kemampuan pelanggan asalkan dapat diterima oleh bagian penjualan. *Flow of events* proses menerima pembayaran kredit disajikan pada tabel 3.7.

Tabel 3. 7 *Flow of Events* Menerima Pembayaran Kredit

Nama Use Case	Menerima pembayaran kredit	
Kebutuhan terkait	Pembayaran kredit dilakukan berdasarkan transaksi kredit yang telah dilakukan. Sistem harus dapat menghitung jumlah yang harus dibayar tiap periode secara otomatis beserta denda jika terjadi keterlambatan.	
Tujuan	Menangani pembayaran kredit.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat pembayaran kredit.	
Kondisi akhir sukses	Pembayaran kredit sukses dilakukan.	
Kondisi akhir gagal	Jika tidak ditemukan transaksi penjualan kredit yang telah dilakukan maka pembayaran kredit tidak dapat dilakukan.	
Aktor utama	Bagian Penjualan	
Aktor sekunder	Pelanggan	
Pemicu	Bagian penjualan melihat <i>history</i> penjualan kredit dan memilih "Pembayaran Kredit".	
Primary Flow	Langkah	Aksi
	1	Bagian Penjualan membuka <i>history</i> penjualan kredit.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data <i>history</i> penjualan kredit yang pernah dilakukan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 2.1 .
	3	Bagian penjualan memilih transaksi kredit yang akan dibayar dan memilih menu pembayaran kredit. Jika transaksi kredit yang dipilih sudah lunas, maka masuk ke langkah 3.1 .
	4	Sistem melakukan koneksi <i>database</i> untuk menghitung dan menampilkan jumlah yang harus dibayar oleh

		pelanggan.
	5	Bagian penjualan memasukkan jumlah uang yang dibayarkan pelanggan. Jika jumlah uang yang dibayarkan kurang, masuk langkah 5.1 .
	6	Sistem melakukan koneksi ke <i>database</i> dan menyimpan transaksi pembayaran kredit. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk langkah 6.1 .
	7	Sistem mencetak nota pembayaran kredit dan bagian penjualan menyerahkannya ke pelanggan.
	8	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan bahwa transaksi kredit sudah lunas. Bagian penjualan kembali ke langkah 3 .
	5.1	Sistem menampilkan pesan "uang yang dibayarkan kurang" dan meminta bagian penjualan mengulangi langkah ke-5.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>history</i> penjualan kredit gagal. Bagian penjualan kembali ke langkah 1 .
	6.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi pembayaran kredit gagal. Bagian penjualan kembali ke langkah 5 .

G. *Flow of Events Use Case Mencatat Transaksi Penerimaan*

Proses mencatat transaksi penerimaan dilakukan ketika bagian pengelola persediaan menerima barang dari *supplier*. Transaksi penerimaan perlu dicatat agar pihak manajemen dapat memantau transaksi penerimaan yang telah dilakukan beserta *supplier*-nya. Stok yang didapat dari penerimaan barang dapat disimpan langsung menjadi stok siap jual atau disimpan pada stok gudang yang belum boleh dijual. *Flow of events* proses mencatat transaksi penerimaan disajikan pada tabel 3.8.

Tabel 3. 8 *Flow of Events* Mencatat Transaksi Penerimaan

Nama Use Case		Mencatat transaksi penerimaan
Kebutuhan terkait	Transaksi penerimaan dari <i>supplier</i> dicatat dan <i>update</i> stok dilakukan sesuai yang ditentukan bagian pengelola stok persediaan. Stok dapat dimasukkan stok siap jual atau stok terkini yang disimpan di gudang dan tidak dapat dijual.	
Tujuan	Menangani penerimaan barang.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat penerimaan barang.	
Kondisi akhir sukses	Penerimaan barang sukses dilakukan.	
Kondisi akhir gagal	Jika masukan uang yang dibayarkan lebih kecil dari total yang harus dibayar, maka muncul pesan "Uang yang dibayarkan kurang".	
Aktor utama	Bagian Pengelola Stok Persediaan	
Aktor sekunder	<i>Supplier</i>	
Pemicu	Bagian pengelola stok persediaan memilih menu "Penerimaan".	
Primary Flow	Langkah	Aksi
	1	Bagian pengelola stok persediaan memilih menu "Penerimaan".
	2	Bagian pengelola stok persediaan memasukkan nama atau <i>id</i> barang dan jumlah yang diterima. Jika barang yang dimasukkan belum terdaftar, maka masuk ke langkah 2.1.
	3	Bagian pengelola stok persediaan memasukkan nama <i>supplier</i> . Jika nama <i>supplier</i> belum terdaftar, maka masuk ke langkah 3.1.
	4	Bagian pengelola persediaan menentukan apakah stok dimasukkan ke stok siap jual.
	5	Sistem menampilkan total harga.
	6	Bagian pengelola stok persediaan memasukkan jumlah uang yang toko bayarkan. Jika jumlah uang yang dimasukkan kurang dari total yang harus dibayarkan, masuk ke langkah 6.1.
	7	Sistem menampilkan jumlah kembalian.
	8	Sistem melakukan koneksi ke <i>database</i> , menyimpan data transaksi dan <i>update</i> stok. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 8.1.
	9	Sistem mencetak nota penerimaan dan bagian pengelola stok persediaan

		menyerahkannya ke <i>supplier</i> .
	10	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	2.1	Bagian pengelola stok persediaan memasukkan data barang baru pada <i>form</i> yang disediakan dan menyimpannya. Jika ada <i>format</i> data yang salah, masuk ke langkah 2.2 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.3 .
	2.2	Sistem menampilkan pesan <i>format</i> data yang dimasukkan salah dan meminta bagian pengelola stok persediaan kembali ke langkah 2.1 untuk mengisi ulang masukan data barang baru dengan <i>format</i> yang benar.
	3.1	Bagian pengelola stok persediaan memasukkan data <i>supplier</i> baru pada <i>form</i> yang disediakan dan menyimpannya. Jika ada <i>format</i> data yang salah, masuk ke langkah 3.2 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.3 .
	3.2	Sistem menampilkan pesan <i>format</i> data yang dimasukkan salah dan meminta bagian pengelola stok persediaan kembali ke langkah 3.1 untuk mengisi ulang masukan data <i>supplier</i> baru dengan <i>format</i> yang benar.
	6.1	Sistem menampilkan pesan "uang yang dibayarkan kurang" dan meminta bagian pengelola stok persediaan mengulang langkah ke- 6 .
Error Flow	Langkah	Aksi Percabangan
	2.3	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data barang baru gagal. Bagian pengelola stok persediaan kembali ke langkah 2.1 .
	3.3	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data <i>supplier</i> baru gagal. Bagian pengelola stok persediaan kembali ke langkah 3.1 .
	8.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi gagal. Bagian pengelola stok persediaan kembali ke langkah 8 .

H. *Flow of Events Use Case Mengelola Data Supplier*

Proses mengelola data *supplier* dilakukan untuk memastikan data *supplier* yang dimiliki ialah data terkini yang berhasil didapatkan dari *supplier*. Proses ini meliputi menambah, mengubah, menghapus dan mencari data *supplier*. Proses ini dilakukan ketika toko mendapatkan informasi terbaru tentang *supplier* mereka. *Flow of events* proses mengelola data *supplier* disajikan pada tabel 3.9.

Tabel 3. 9 *Flow of Events Mengelola Data Supplier*

Nama Use Case	Mengelola data supplier	
Kebutuhan terkait	Proses mengelola data <i>supplier</i> meliputi proses menambah, mengubah, menghapus dan mencari data <i>supplier</i> .	
Tujuan	Memastikan bahwa data <i>supplier</i> yang dimiliki merupakan data terkini berdasarkan yang didapatkan dari <i>supplier</i> . Data ini dapat digunakan toko untuk berinteraksi dengan <i>supplier</i> .	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola data <i>supplier</i> .	
Kondisi akhir sukses	Data <i>supplier</i> telah diperbaharui.	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Bagian Pengelola Stok Persediaan	
Aktor sekunder	Tidak ada	
Pemicu	Bagian pengelola stok persediaan memilih menu " <i>Supplier</i> ".	
Primary Flow	Langkah	Aksi
	1	Bagian pengelola stok persediaan memilih menu " <i>Supplier</i> "
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data <i>supplier</i> dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 2.1 .
	3	Bagian pengelola stok persediaan melakukan operasi mencari, menambah, mengubah atau menghapus data <i>supplier</i> .

	4	Sistem melakukan koneksi <i>database</i> untuk menyimpan semua perubahan yang dilakukan bagian pengelola stok persediaan terhadap data <i>supplier</i> . Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data <i>supplier</i> berhasil.
	6	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data <i>supplier</i> yang dimasukkan salah dan meminta bagian pengelola stok persediaan kembali ke langkah 3 untuk mengisi ulang masukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>supplier</i> gagal. Bagian pengelola stok persediaan kembali ke langkah 1 .
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mencari, menambah, mengubah atau menghapus data <i>supplier</i> gagal. Bagian pengelola stok persediaan kembali ke langkah 3 .

I. *Flow of Events Use Case Melihat Laporan*

Melihat laporan merupakan proses yang sangat penting bagi pihak manajemen. Laporan-laporan yang dihasilkan akan membantu manajemen dalam mengambil keputusan. Sistem akan memberikan laporan terkini. Laporan yang dihasilkan ialah tren penjualan, stok terkini, *history* transaksi pelanggan dan *history* transaksi *supplier*. *Flow of events* proses melihat laporan disajikan pada tabel 3.10.

Tabel 3. 10 *Flow of Events* Melihat Laporan

Nama <i>Use Case</i>		Melihat laporan
Kebutuhan terkait	Proses melihat laporan diperlukan pihak manajemen untuk membantu mereka dalam pengambilan keputusan. Informasi yang diperlukan adalah tren penjualan, stok terkini, <i>history</i> transaksi pelanggan dan <i>supplier</i> .	
Tujuan	Mendapatkan informasi terkini dari toko yang dapat membantu pihak manajemen dalam pengambilan keputusan.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk melihat laporan.	
Kondisi akhir sukses	Mendapatkan informasi yang diperlukan.	
Kondisi akhir gagal	Jika <i>query</i> untuk laporan gagal dieksekusi, maka sistem akan memberi pesan <i>error</i> .	
Aktor utama	Manajemen	
Aktor sekunder	Tidak ada	
Pemicu	Pihak manajemen memilih menu "Laporan"	
Primary Flow	Langkah	Aksi
	1	Manajemen memilih menu "Laporan"
	2	Manajemen memilih jenis laporan yang diinginkan.
	3	Manajemen memasukkan parameter-parameter laporan sesuai informasi yang dibutuhkan. Jika <i>format</i> parameter masukan salah, maka masuk ke langkah 3.1.
	4	Sistem melakukan koneksi <i>database</i> untuk mengambil data dan mengolahnya menjadi informasi sesuai dengan tampilan dan jenis laporan yang dipilih. Jika terjadi <i>error</i> saat melakukan koneksi <i>database</i> , maka masuk ke langkah 4.1.
	5	Manajemen mencetak laporan.
	6	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan <i>format</i> data yang dimasukkan salah dan meminta manajemen mengisi ulang parameter laporan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	4.1	Menampilkan pesan koneksi <i>database</i> untuk menampilkan laporan gagal. Manajemen kembali ke langkah 3.

J. *Flow of Events Use Case Mengatur Hak Akses Karyawan*

Pihak manajemen dapat menentukan hak akses karyawan terhadap sistem. Hal ini memudahkan pembagian tanggung jawab dan mencari penyebab terjadinya kesalahan atau kekurangan dalam sistem. *Flow of events* proses mengatur hak akses karyawan disajikan pada tabel 3.11.

Tabel 3. 11 *Flow of Events* Mengatur Hak Akses Karyawan

Nama <i>Use Case</i>		Mengatur hak akses karyawan
Kebutuhan terkait	Proses mengatur hak akses karyawan akan membantu manajemen untuk melakukan pembagian tugas dalam sistem. Hak akses dibagi menjadi proses <i>insert</i> , <i>update</i> dan <i>delete</i> terhadap transaksi penjualan, penerimaan dan pengelolaan data <i>master</i> karyawan, <i>supplier</i> dan pelanggan termasuk membaca laporan-laporan yang disediakan sistem.	
Tujuan	Membagi tugas dan tanggung jawab karyawan dengan pembagian hak akses terhadap sistem.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengatur hak akses karyawan.	
Kondisi akhir sukses	Data karyawan dan hak aksesnya telah diperbaharui.	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Manajemen	
Aktor sekunder	Tidak ada	
Pemicu	Pihak manajemen memilih menu "Karyawan"	
Primary Flow	Langkah	Aksi
	1	Manajemen memilih menu "Karyawan"
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data karyawan dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 2.1 .
	3	Manajemen melakukan operasi mencari, menambah, mengubah atau menghapus data karyawan beserta hak aksesnya.
	4	Sistem melakukan koneksi <i>database</i> untuk menyimpan semua perubahan yang dilakukan manajemen terhadap data karyawan beserta hak aksesnya. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> masuk

		ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data karyawan beserta hak aksesnya berhasil.
	6	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data karyawan beserta hak akses yang dimasukkan salah dan meminta manajemen kembali ke langkah 3 untuk mengisi ulang masukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data karyawan gagal. Manajemen kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mencari, menambah, mengubah atau menghapus data karyawan beserta hak akses gagal. Manajemen kembali ke langkah 3 .

K. Flow of Events Use Case Mengubah Paket Layanan

Sistem informasi penjualan yang dibangun merupakan solusi berbasis *cloud* sehingga pihak manajemen dapat menentukan paket layanan yang sesuai dengan kebutuhan, dihitung dari jumlah maksimal transaksi penjualan tiap bulan dan jumlah transaksi penjualan yang disimpan. *Flow of events* proses mengubah paket layanan disajikan pada tabel 3.12.

Tabel 3. 12 *Flow of Events* Mengubah Paket Layanan

Nama Use Case		Mengubah paket layanan	
Kebutuhan terkait	Kebutuhan dibagi menjadi jumlah maksimal transaksi penjualan dalam 30 hari dan jumlah maksimal transaksi penjualan yang disimpan. Jika paket layanan yang diajukan sama dengan paket layanan sebelumnya, maka tanggal batas akhir layanan diperpanjang. Jika paket layanan yang diajukan berbeda, maka paket layanan baru daiktifkan dan yang lama dihanguskan.		
Tujuan	Mengubah paket layanan sesuai kebutuhan bisnis berdasarkan jumlah maksimal transaksi penjualan dalam 30 hari dan jumlah maksimal transaksi penjualan yang disimpan.		
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengubah paket layanan.		
Kondisi akhir sukses	Paket layanan berhasil diubah.		
Kondisi akhir gagal	Jika ada <i>format</i> pengisian data yang salah, maka sistem akan memberi tahu bahwa pengubahan layanan gagal.		
Aktor utama	Manajemen		
Aktor sekunder	Tidak ada		
Pemicu	Pihak manajemen memilih menu "Karyawan"		
Primary Flow	Langkah	Aksi	
	1	Manajemen memilih menu "Ubah Layanan".	
	2	Manajemen memasukkan paket layanan yang diinginkan, jumlah maksimal penyimpanan transaksi penjualan yang diperlukan beserta jumlah bulan. Jika <i>format</i> data masukan salah, masuk ke langkah 2.1 .	
	3	Sistem melakukan koneksi <i>database</i> untuk menyimpan data permintaan pengubahan layanan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.2 .	
	4	Sistem menampilkan tagihan yang harus di- <i>transfer</i> oleh manajemen.	
	5	Manajemen memasukkan data konfirmasi <i>transfer</i> . Jika <i>format</i> data masukan salah, masuk ke langkah 5.1 .	
	6	Sistem melakukan koneksi <i>database</i> untuk menyimpan konfirmasi <i>transfer</i> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 6.1 .	

	7	Sistem mengaktifkan paket layanan. Jika terjadi <i>error</i> saat mengaktifkan layanan, masuk ke langkah 7.1 .
	8	<i>Use case</i> berakhir.
Alternate Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan <i>format</i> data perubahan paket layanan yang dimasukkan salah dan meminta manajemen mengisi ulang masukan dengan <i>format</i> yang benar.
	5.1	Sistem menampilkan pesan <i>format</i> data konfirmasi <i>transfer</i> yang dimasukkan salah dan meminta manajemen mengisi ulang masukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data permintaan perubahan layanan gagal. Manajemen kembali ke langkah 2 .
	6.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan konfirmasi <i>transfer</i> gagal. Manajemen kembali ke langkah 5 .
	7.1	Sistem menampilkan pesan aktivasi layanan baru gagal dan meminta manajemen mengulang langkah ke- 5 .

L. *Flow of Events Use Case Promosi di Facebook*

Bagian penjualan dapat melakukan promosi barang melalui Facebook dengan memanfaatkan Facebook API. Promosi ini akan menampilkan nama barang, stok terkini dan harga terkini pada status Facebook bagian penjualan sehingga dapat dilihat pengguna Facebook yang lainnya. *Flow of events* proses promosi di Facebook disajikan pada tabel 3.13.

Tabel 3. 13 *Flow of Events* Promosi di Facebook

Nama Use Case		Promosi barang di Facebook	
Kebutuhan terkait	Untuk melakukan promosi barang di Facebook, sistem harus mengambil data stok siap jual terkini beserta harga terkini dari barang yang akan dipromokan. Promosi dilakukan melalui status Facebook pengguna yang sedang aktif <i>login</i> saat proses promosi dilakukan.		
Tujuan	Menangani promosi barang di Facebook.		
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengakses data barang.		
Kondisi akhir sukses	Promosi barang di Facebook sukses dilakukan.		
Kondisi akhir gagal	Promosi barang di Facebook gagal dilakukan.		
Aktor utama	Bagian Penjualan		
Aktor sekunder	Tidak ada		
Pemicu	Bagian penjualan membuka data barang, memilih barang dan memilih menu "Promo di Facebook".		
Primary Flow	Langkah	Aksi	
	1	Bagian penjualan memilih menu "Barang"	
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data barang dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah 2.1.	
	3	Bagian penjualan memilih salah satu barang lalu memilih menu "Promo di Facebook". Bagian penjualan juga dapat memilih barang terlaris atau barang dengan diskon tertinggi.	
	4	Sistem memeriksa apakah bagian penjualan telah melakukan <i>login</i> di Facebook. Jika belum, masuk ke langkah 4.1.	
	5	Sistem menulis promo barang di status Facebook bagian penjualan dan menampilkan pesan sukses. Jika terjadi <i>error</i> saat menulis status Facebook, masuk ke langkah 5.1.	
	6	<i>Use case</i> berakhir.	
Alternate Flow	Langkah	Aksi Percabangan	
	4.1	Bagian penjualan harus melakukan <i>login</i> pada <i>form</i> yang disediakan dan akan diverifikasi oleh Facebook API secara langsung. Setelah <i>login</i> , sistem melanjutkan langkah ke-5. Jika terjadi <i>error</i> sistem saat <i>login</i> , masuk ke langkah	

		4.2.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data barang gagal. Manajemen kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan gagal <i>login</i> pada Facebook dan mengarahkan bagian penjualan kembali ke langkah 4.1 untuk <i>login</i> ulang.
	5.1	Sistem menampilkan pesan gagal menulis status pada Facebook dan mengarahkan bagian penjualan kembali ke langkah 3 .

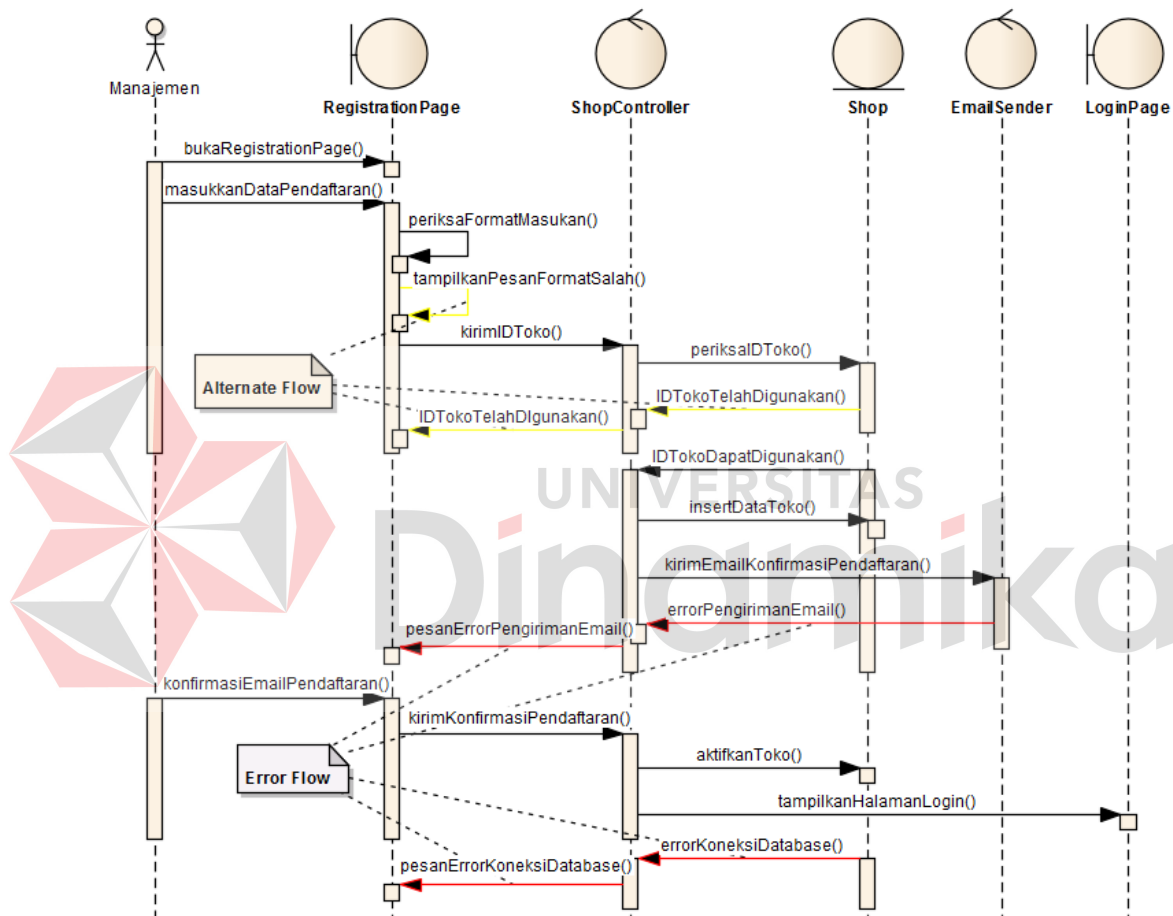
3.3.4 Diagram Sekuensial

Setelah menggambarkan *Flow of Events*, langkah selanjutnya adalah menggambar diagram sekuensial dari masing-masing *use case*. Diagram sekuensial adalah diagram yang digunakan untuk menunjukkan aliran fungsionalitas dalam masing-masing *use case* yang disusun dalam urutan waktu (Sholiq; 2010).

A. Diagram Sekuensial Mendaftarkan Toko

Diagram sekuensial pada gambar 3.12 menunjukkan proses mendaftar toko yang dimulai dari pihak manajemen mengakses *boundary RegistrationPage* dan mengisi *form* pendaftaran. Setelah itu, sistem memeriksa *format* data pendaftaran. Jika ada kesalahan *format*, sistem akan menyampaikan ke manajemen. Jika tidak ada, sistem memeriksa apakah *id* toko yang didaftarkan telah digunakan melalui *control ShopController*. Jika telah digunakan, sistem meminta penggunaan *id* toko lainnya. Jika belum, sistem akan menyimpan data

toko dan mengirim *email* konfirmasi pendaftaran pada alamat *email* yang didaftarkan. Setelah itu manajemen melakukan konfirmasi pendaftaran melalui *link* yang dikirimkan ke alamat *email* yang didaftarkan dan sistem akan mengaktifkan akun toko tersebut lalu mengarahkannya ke *boundary LoginPage*.

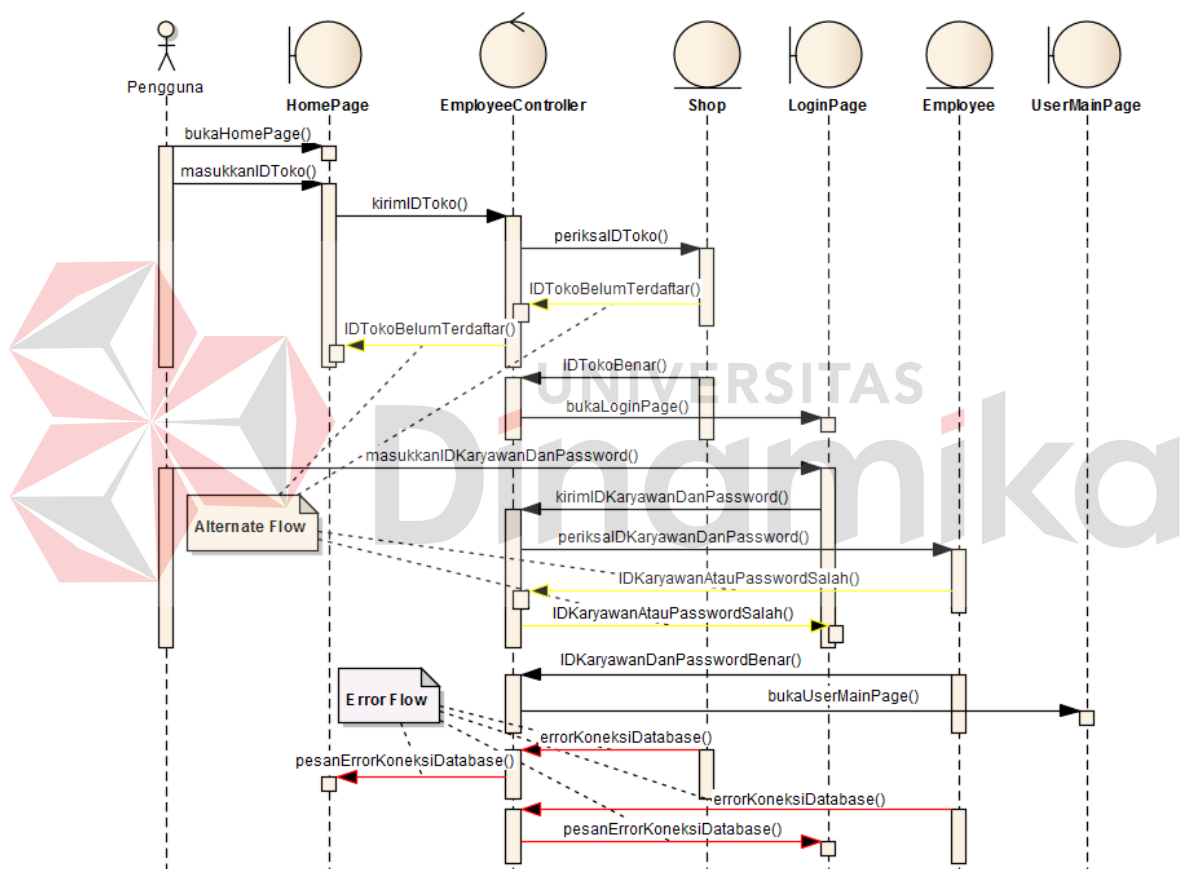


Gambar 3. 12 Diagram Sekuensial Mendaftarkan Toko

B. Diagram Sekuensial *Login*

Diagram sekuensial pada gambar 3.13 menunjukkan proses *login* yang dimulai dari pengguna mengakses *boundary HomePage* dan memasukkan *id* toko. Lalu sistem mengirim *id* toko ke *control EmployeeController* untuk memeriksa apakah *id* toko yang dimasukkan telah terdaftar. Jika belum, sistem meminta

pengguna mengisi *id* toko yang benar. Jika *id* toko benar, sistem membuka *boundary LoginPage* dan pengguna memasukkan *id* karyawan beserta *password*. Sistem akan mengirimkan *id* karyawan dan *password* ke *control EmployeeController* untuk diperiksa. Jika *id* karyawan atau *password* salah, sistem meminta pengguna memasukkan *id* karyawan dan *password* yang benar. Jika sudah benar, sistem mengarahkan pengguna ke *boundary UserMainPage*.

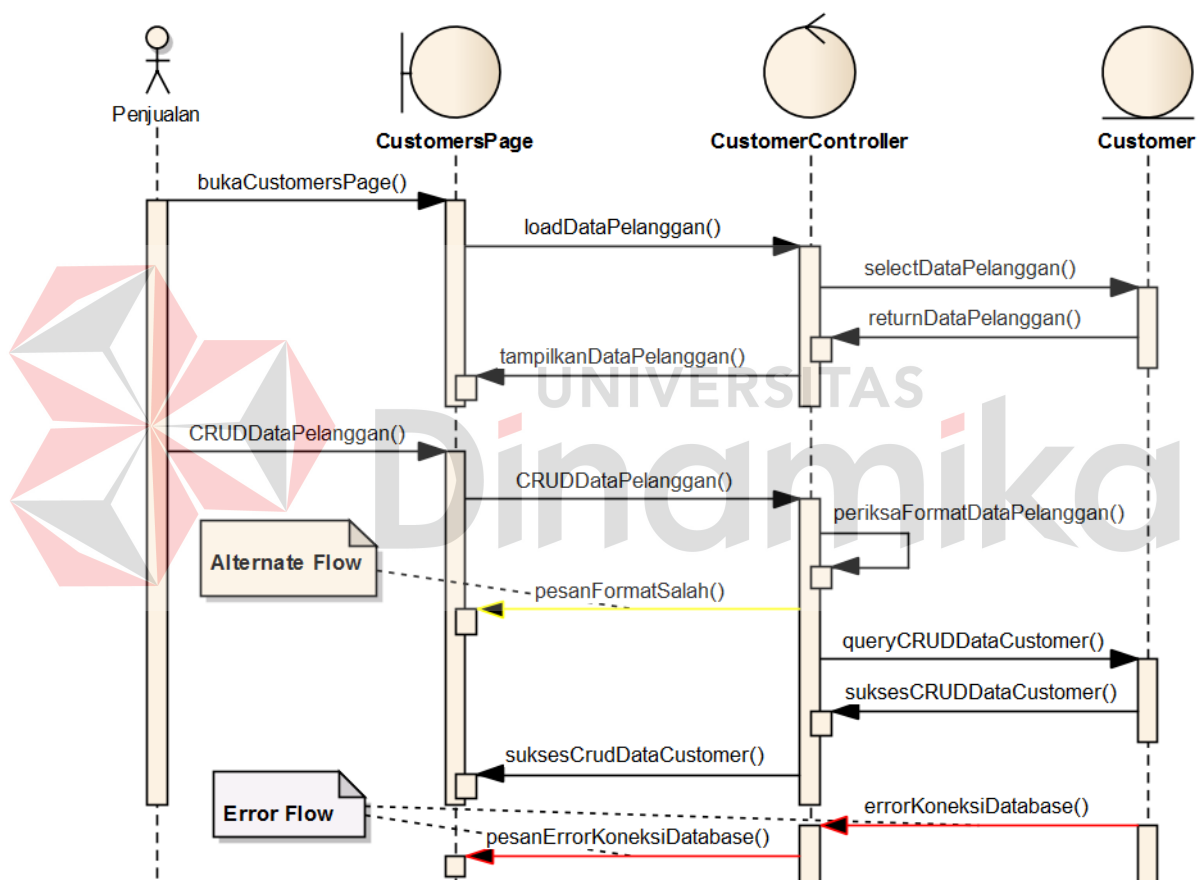


Gambar 3. 13 Diagram Sekuensial Login

C. Diagram Sekuensial Mengelola Data Pelanggan

Diagram sekuensial pada gambar 3.14 menunjukkan proses mengelola data pelanggan yang dimulai dari bagian penjualan mengakses *boundary CustomersPage*. Lalu *boundary CustomersPage* meminta *control*

CustomerController mengambil data pelanggan dari *database* untuk ditampilkan. Setelah itu, bagian penjualan melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary CustomersPage*. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan *format*, sistem meminta bagian penjualan mengubahnya. Jika *format* benar, data pelanggan disimpan di *database* melalui *control CustomerController* dan pesan sukses disampaikan.

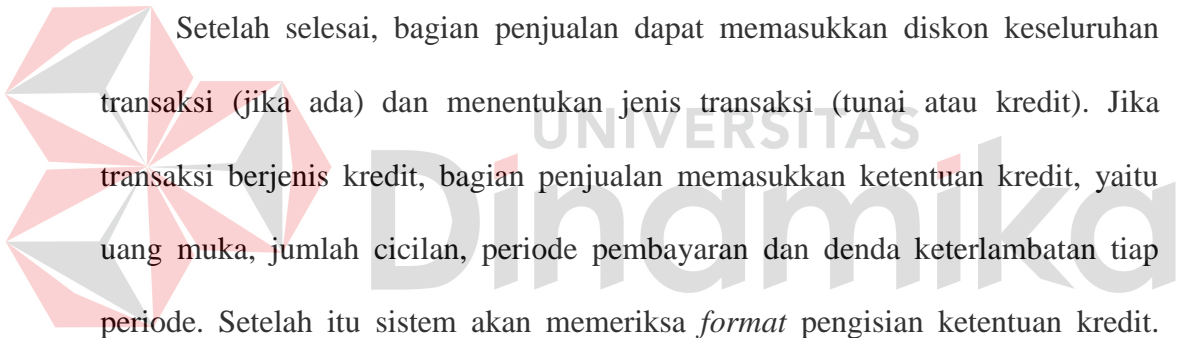


Gambar 3. 14 Diagram Sekuensial Mengelola Data Pelanggan

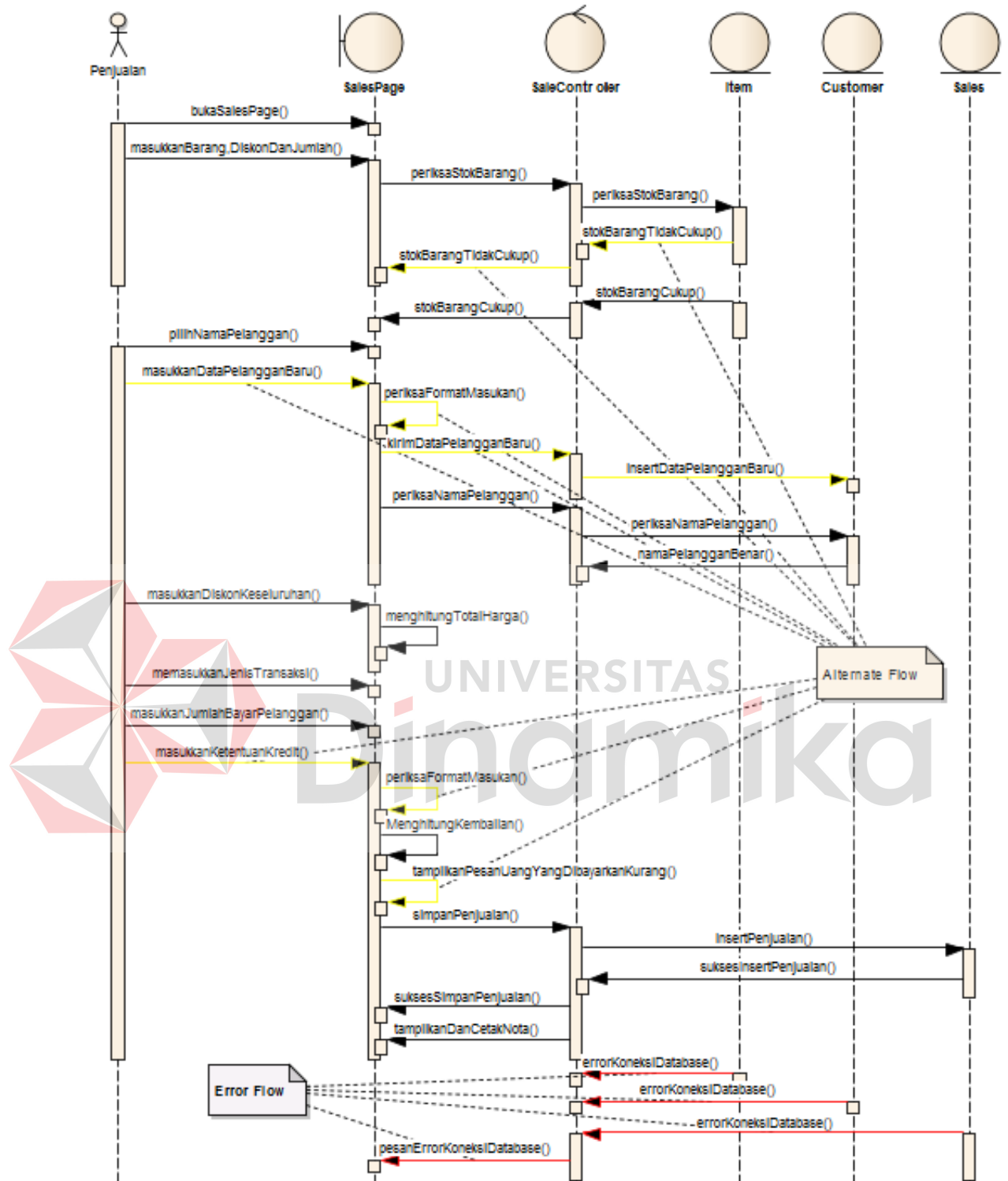
D. Diagram Sekuensial Mencatat Transaksi Penjualan

Diagram sekuensial pada gambar 3.15 menunjukkan proses mencatat transaksi penjualan yang dimulai dari bagian penjualan mengakses *boundary SalesPage* dan memasukkan barang, jumlah beserta diskon pada masing-masing

barang (jika ada). Setelah dimasukkan, sistem memeriksa melalui *control SaleController* apakah stok siap jual dari barang-barang tersebut mencukupi. Jika tidak, sistem menyampaikan ke bagian penjualan. Setelah itu, bagian penjualan memasukkan nama pelanggan. Sistem akan mengecek melalui *control SaleController* apakah nama pelanggan yang dimasukkan terdata di *database*. Jika nama pelanggan tidak ada, bagian penjualan mengisi data pelanggan baru pada *form* yang disediakan. Sistem akan memeriksa apakah *format* masukan data pelanggan baru sudah benar. Jika belum, sistem akan meminta bagian penjualan untuk mengisi data pelanggan baru dengan benar. Setelah itu data pelanggan baru akan disimpan.



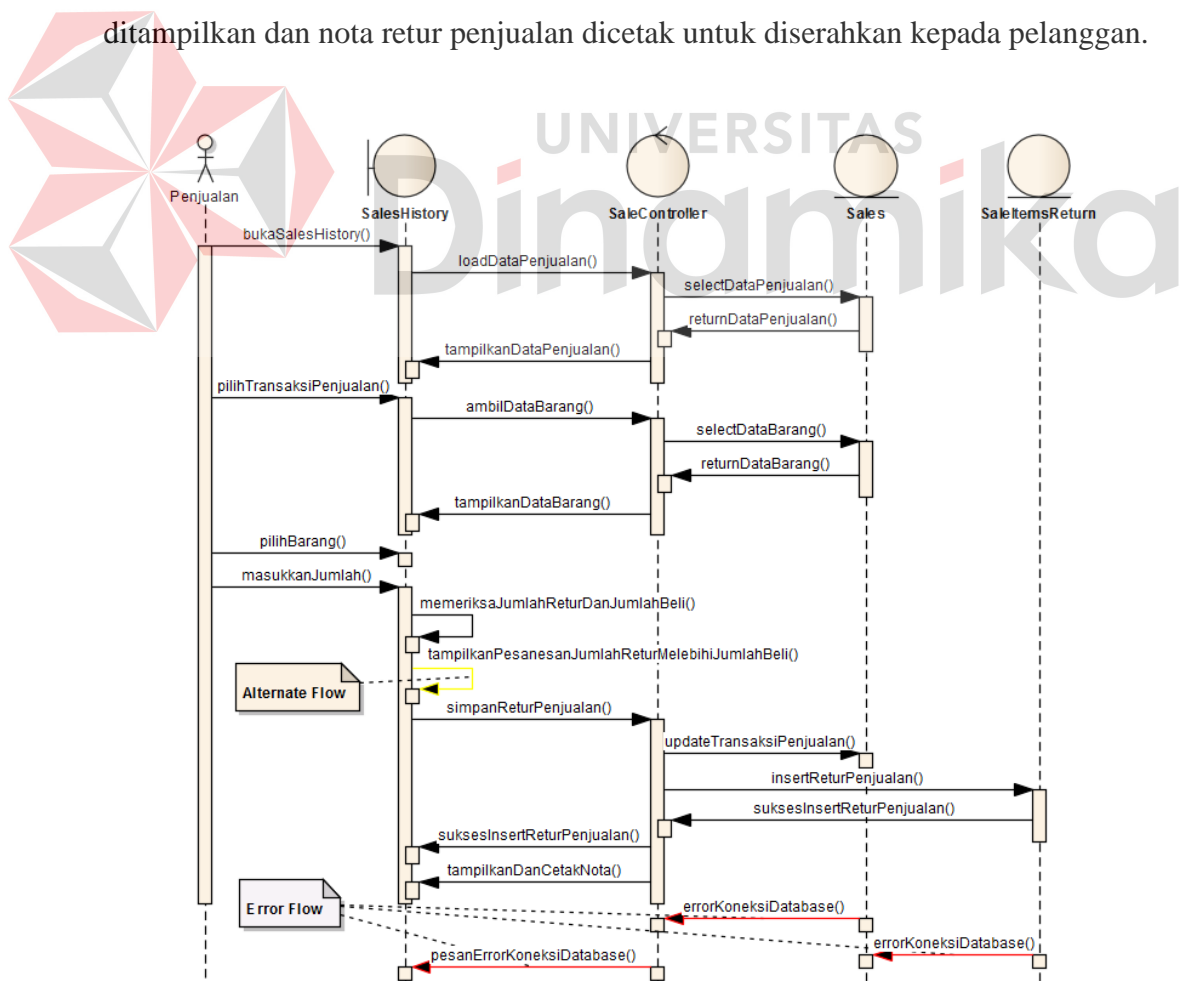
Setelah selesai, bagian penjualan dapat memasukkan diskon keseluruhan transaksi (jika ada) dan menentukan jenis transaksi (tunai atau kredit). Jika transaksi berjenis kredit, bagian penjualan memasukkan ketentuan kredit, yaitu uang muka, jumlah cicilan, periode pembayaran dan denda keterlambatan tiap periode. Setelah itu sistem akan memeriksa *format* pengisian ketentuan kredit. Jika ada kesalahan, sistem akan menyampaikannya kepada bagian penjualan. Jika transaksi berjenis tunai, bagian penjualan akan menyampaikan total harga dan menerima pembayaran dari pelanggan. Lalu bagian penjualan memasukkan jumlah yang dibayar oleh pelanggan dan sistem akan menampilkan kembaliannya. Jika jumlah uang yang dibayarkan kurang, sistem akan menyampaikannya ke bagian penjualan. Setelah itu transaksi akan disimpan dan nota transaksi penjualan akan dicetak dan diserahkan oleh bagian penjualan ke pelanggan.



Gambar 3. 15 Diagram Sekuensial Mencatat Transaksi Penjualan

E. Diagram Sekuensial Menerima Retur Penjualan

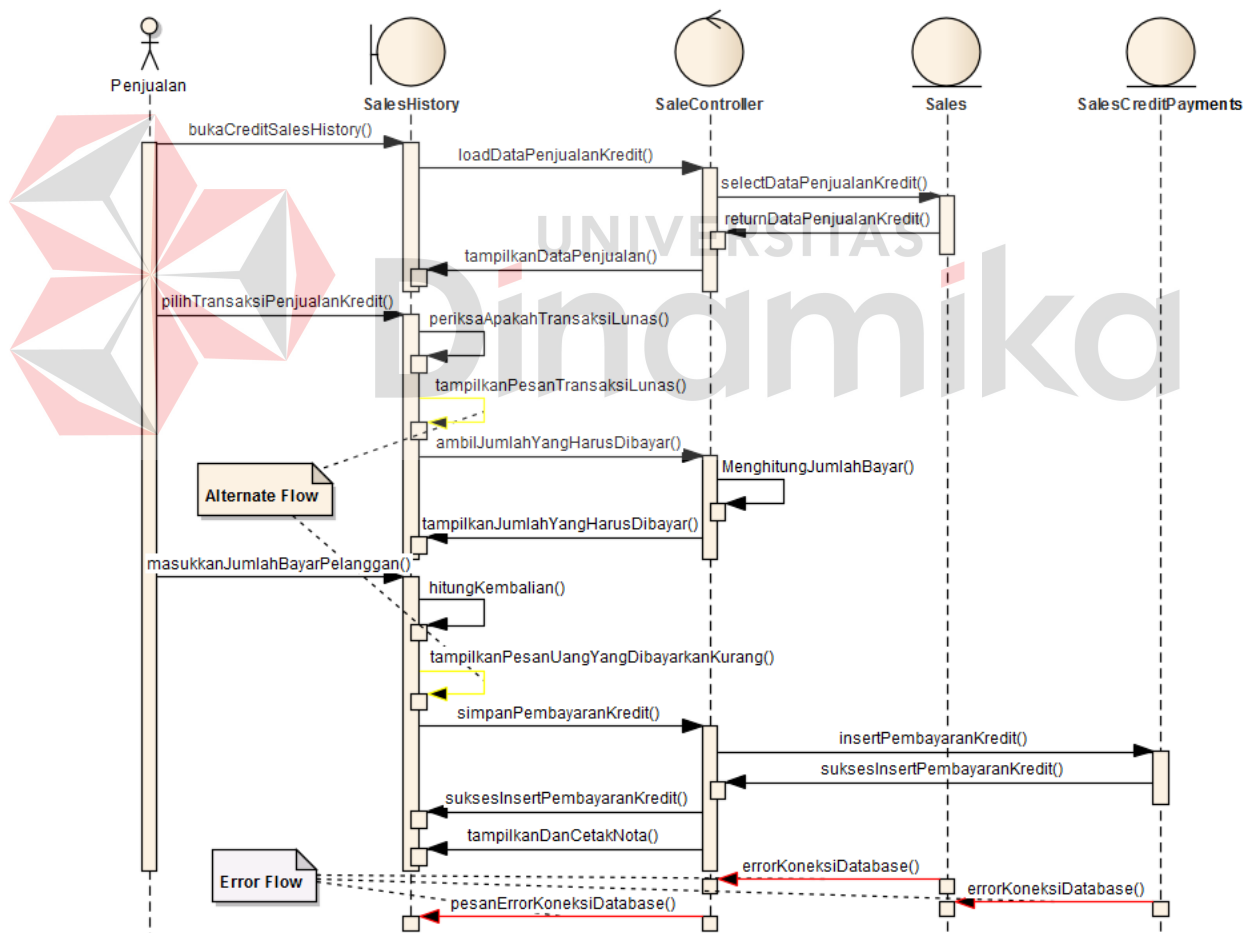
Diagram sekuensial pada gambar 3.16 menunjukkan proses menerima retur penjualan yang dimulai dari bagian penjualan mengakses *boundary SalesHistory* dan sistem menampilkan *history* penjualan melalui *control SaleController*. Bagian penjualan memilih transaksi penjualan dan sistem menampilkan data barang pada transaksi tersebut melalui. Lalu bagian penjualan memilih barang dan memasukkan jumlah yang diretur. Lalu sistem memeriksa apakah jumlah yang diretur melebihi jumlah yang dibeli. Jika melebihi, sistem menyampaikannya ke bagian penjualan. Jika tidak, sistem menyimpan data retur penjualan sekaligus memperbaharui data transaksi penjualan yang diretur. Setelah itu pesan sukses ditampilkan dan nota retur penjualan dicetak untuk diserahkan kepada pelanggan.



Gambar 3. 16 Diagram Sekuensial Menerima Retur Penjualan

F. Diagram Sekuensial Menerima Pembayaran Kredit

Diagram sekuensial pada gambar 3.17 menunjukkan proses menerima pembayaran kredit yang dimulai dari bagian penjualan mengakses *boundary* *SalesHistory* dan sistem menampilkan *history* penjualan kredit melalui *control* *SaleController*. Bagian penjualan memilih transaksi kredit yang akan dibayar dan memasukkan nominal pembayarannya. Setelah itu sistem akan menyimpan data pembayaran kredit dan pesan sukses ditampilkan. Lalu nota pembayaran kredit dicetak dan diserahkan ke pelanggan oleh bagian penjualan.



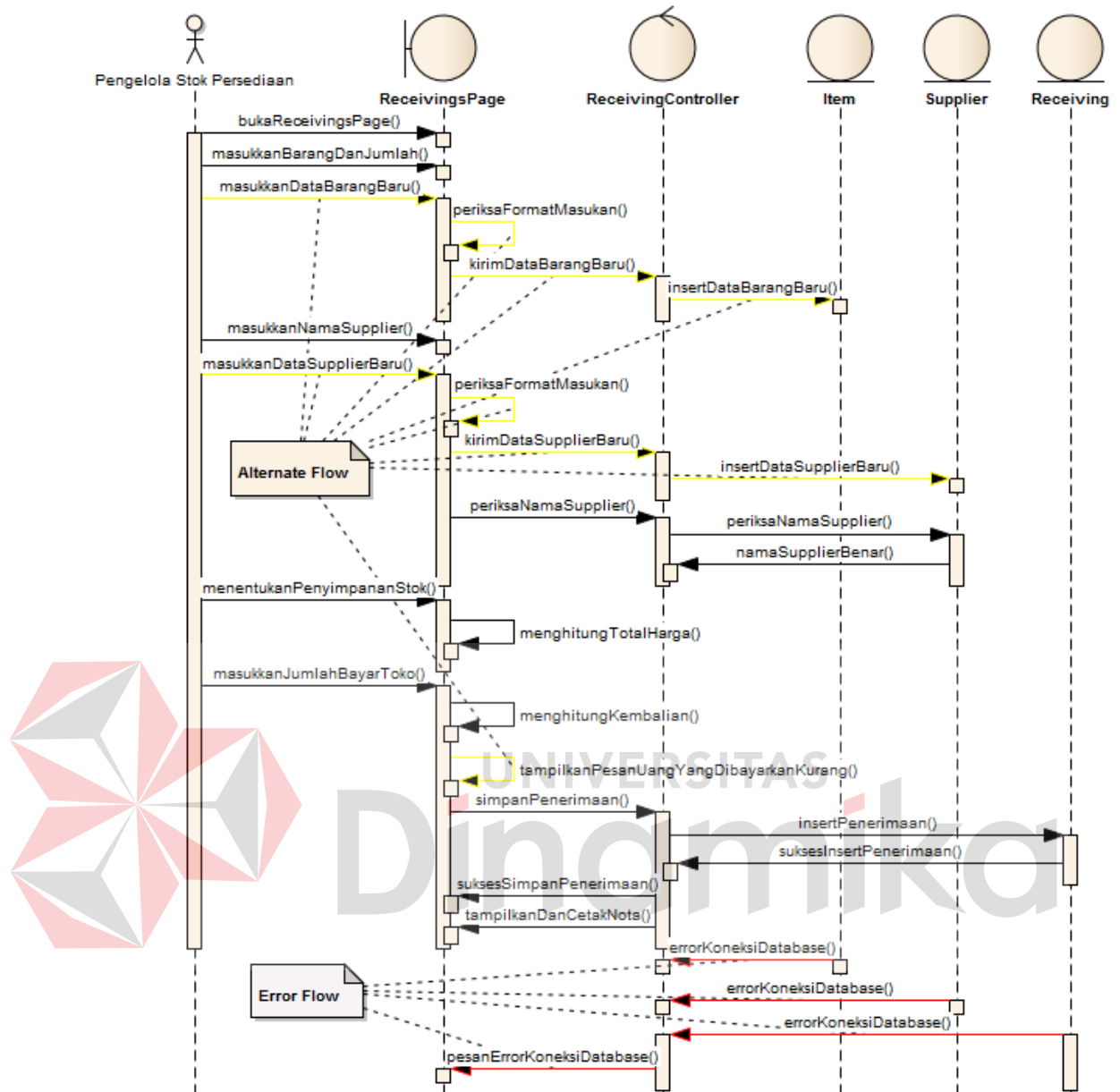
Gambar 3. 17 Diagram Sekuensial Menerima Pembayaran Kredit

G. Diagram Sekuensial Mencatat Transaksi Penerimaan

Diagram sekuensial pada gambar 3.18 menunjukkan proses mencatat transaksi penerimaan yang dimulai dari bagian pengelola stok persediaan mengakses *boundary ReceivingPage* dan memasukkan barang beserta jumlah yang diterimanya. Jika barang belum terdata, bagian pengelola stok persediaan dapat memasukkan data barang baru melalui *form* yang disediakan. Sistem akan memeriksa apakah *format* data barang baru yang dimasukkan sudah benar. Jika belum, sistem meminta bagian pengelola stok persediaan untuk memasukkannya dengan benar. Jika sudah, sistem akan menyimpan data barang baru tersebut.

Langkah selanjutnya adalah memasukkan nama *supplier*. Sistem akan memeriksa melalui *control ReceivingController* apakah nama *supplier* yang dimasukkan terdata di *database*. Jika nama *supplier* tidak ada, maka bagian pengelola stok persediaan dapat mengisi data *supplier* baru pada *form* yang disediakan. Sistem akan memeriksa apakah *format* data *supplier* baru yang dimasukkan sudah benar. Jika belum, sistem meminta bagian pengelola stok persediaan untuk memasukkannya dengan benar. Jika sudah, sistem akan menyimpan data *supplier* baru tersebut.

Setelah itu, bagian pengelola stok persediaan menentukan apakah stok yang dimasukkan akan langsung menjadi stok siap jual. Setelah selesai, sistem akan menghitung total yang harus dibayar. Lalu bagian pengelola stok persediaan memasukkan jumlah uang yang mereka bayarkan dan sistem akan membantu menghitung kembaliannya. Setelah itu transaksi disimpan dan nota transaksi penerimaan akan dicetak dan diserahkan oleh bagian pengelola stok persediaan ke *supplier*.

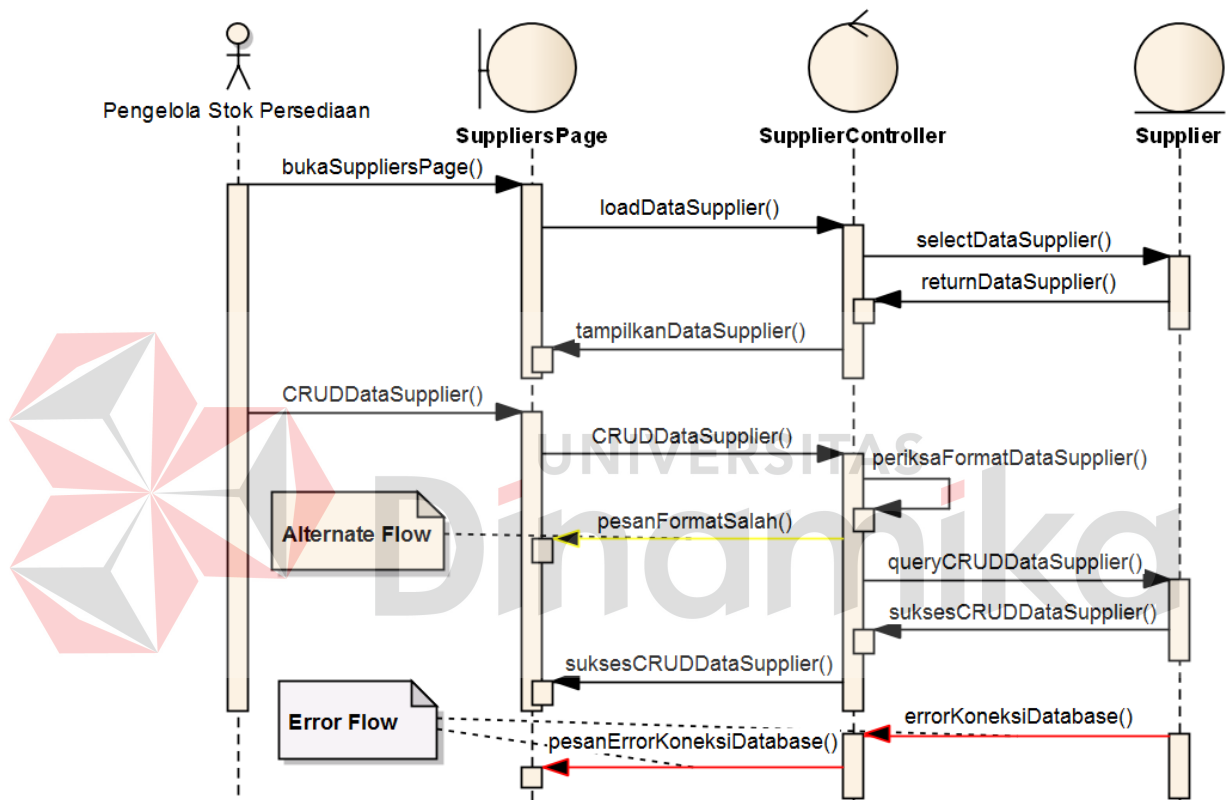


Gambar 3. 18 Diagram Sekuensial Mencatat Transaksi Penerimaan

H. Diagram Sekuensial Mengelola Data *Supplier*

Diagram sekuensial pada gambar 3.19 menunjukkan proses mengelola data *supplier* yang dimulai dari bagian pengelola stok persediaan mengakses *boundary SuppliersPage*. Lalu *boundary SuppliersPage* meminta *control SupplierController* untuk mengambil data *Supplier* dari *database* untuk ditampilkan. Setelah itu, bagian pengelola stok persediaan melakukan operasi

CRUD (*Create, Read, Update, Delete*) pada *boundary SuppliersPage*. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan *format*, sistem meminta bagian pengelola stok persediaan mengubahnya. Jika *format* benar, data *supplier* disimpan di *database* melalui *control SupplierController* dan pesan sukses disampaikan.

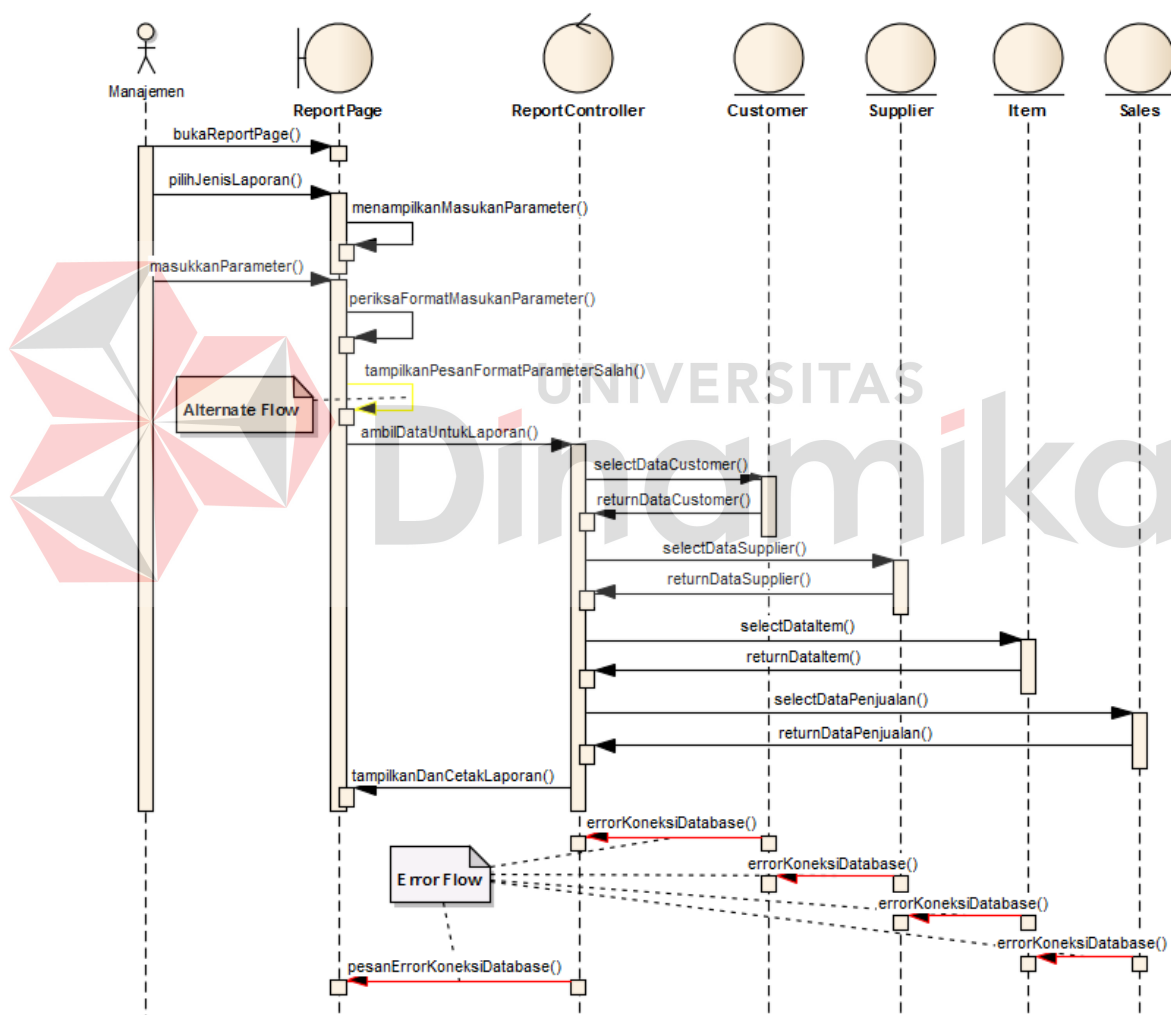


Gambar 3. 19 Diagram Sekuensial Mengelola Data *Supplier*

I. Diagram Sekuensial Melihat Laporan

Diagram sekuensial pada gambar 3.20 menunjukkan proses melihat laporan yang dimulai dari pihak manajemen yang mengakses *boundary ReportPage* dan memilih jenis laporan yang diinginkan. *Boundary ReportPage* akan menampilkan masukan parameter yang perlu diisi bagian manajemen untuk mendapatkan laporan yang diinginkan. Setelah pihak manajemen memasukkan parameter yang

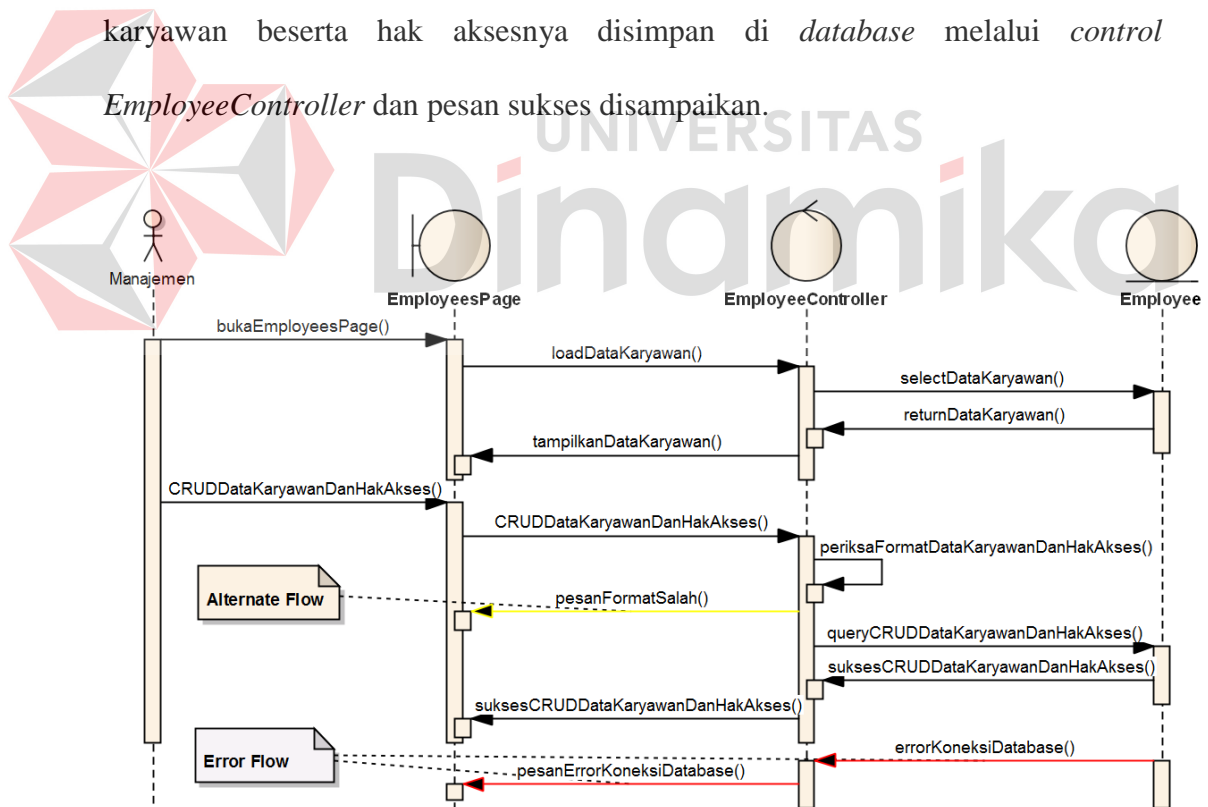
diperlukan, sistem akan memeriksa apakah ada kesalahan *format* pada masukan parameter. Jika ada, sistem akan menyampaikan ke manajemen. Jika sudah benar, *boundary ReportPage* meminta *control ReportController* untuk mengambil data dari *database* sesuai jenis laporan dan parameter-parameter yang dimasukkan manajemen. Setelah itu, data akan diolah menjadi informasi dan ditampilkan kepada pihak manajemen untuk dicetak.



Gambar 3. 20 Diagram Sekuensial Melihat Laporan

J. Diagram Sekuensial Mengatur Hak Akses Karyawan

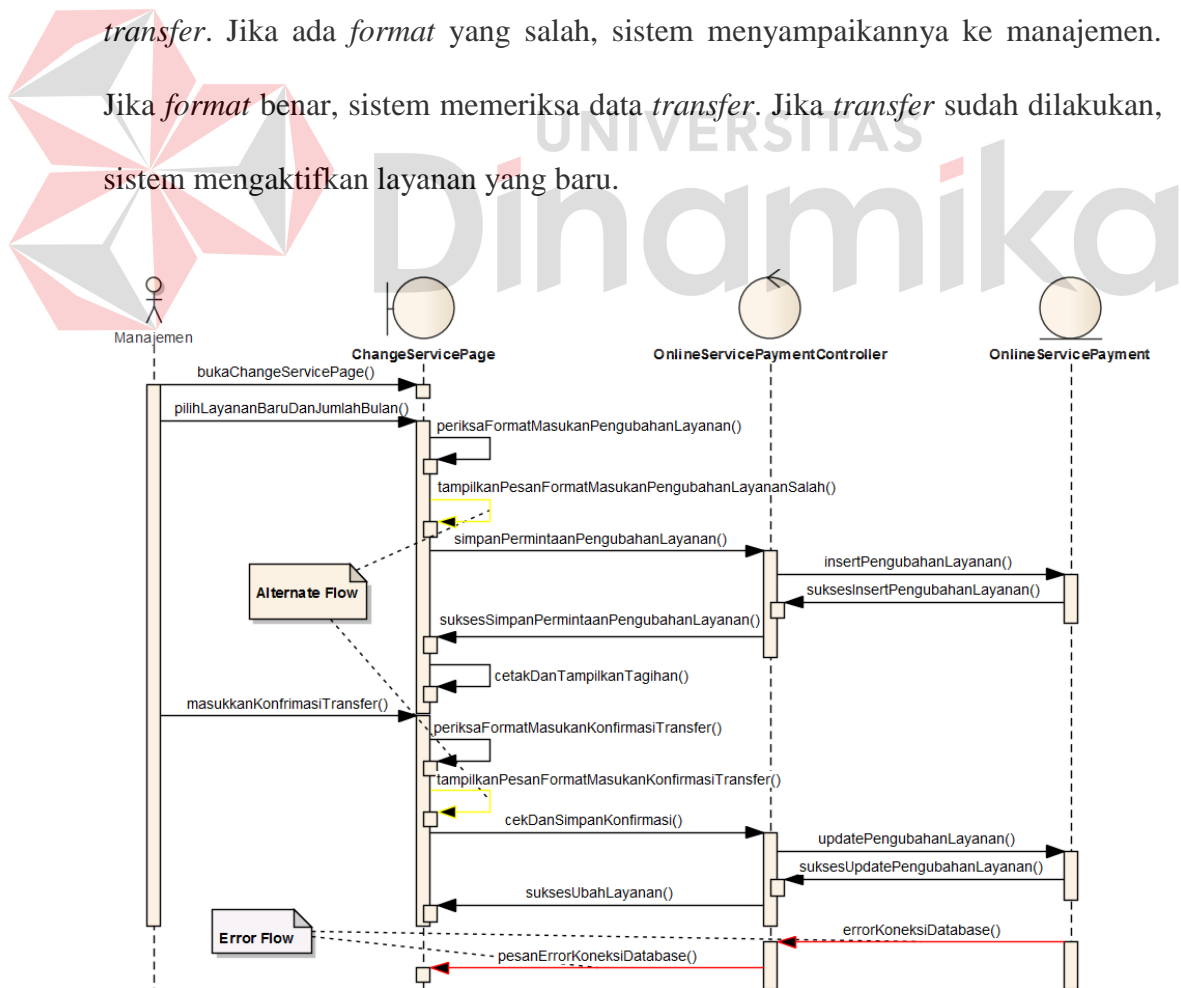
Diagram sekuensial pada gambar 3.21 menunjukkan proses mengatur hak akses karyawan yang dimulai dari pihak manajemen mengakses *boundary* *EmployeesPage*. Lalu *boundary* *EmployeesPage* meminta *control* *EmployeeController* untuk mengambil data *Employee* dari *database* untuk ditampilkan. Setelah itu, bagian pengelola stok persediaan melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *EmployeesPage*. Operasi CRUD dilakukan pada data karyawan beserta hak aksesnya terhadap sistem. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan *format*, sistem meminta manajemen mengubahnya. Jika *format* benar, data karyawan beserta hak aksesnya disimpan di *database* melalui *control* *EmployeeController* dan pesan sukses disampaikan.



Gambar 3. 21 Diagram Sekuensial Megatur Hak Akses Karyawan

K. Diagram Sekuensial Mengubah Paket Layanan

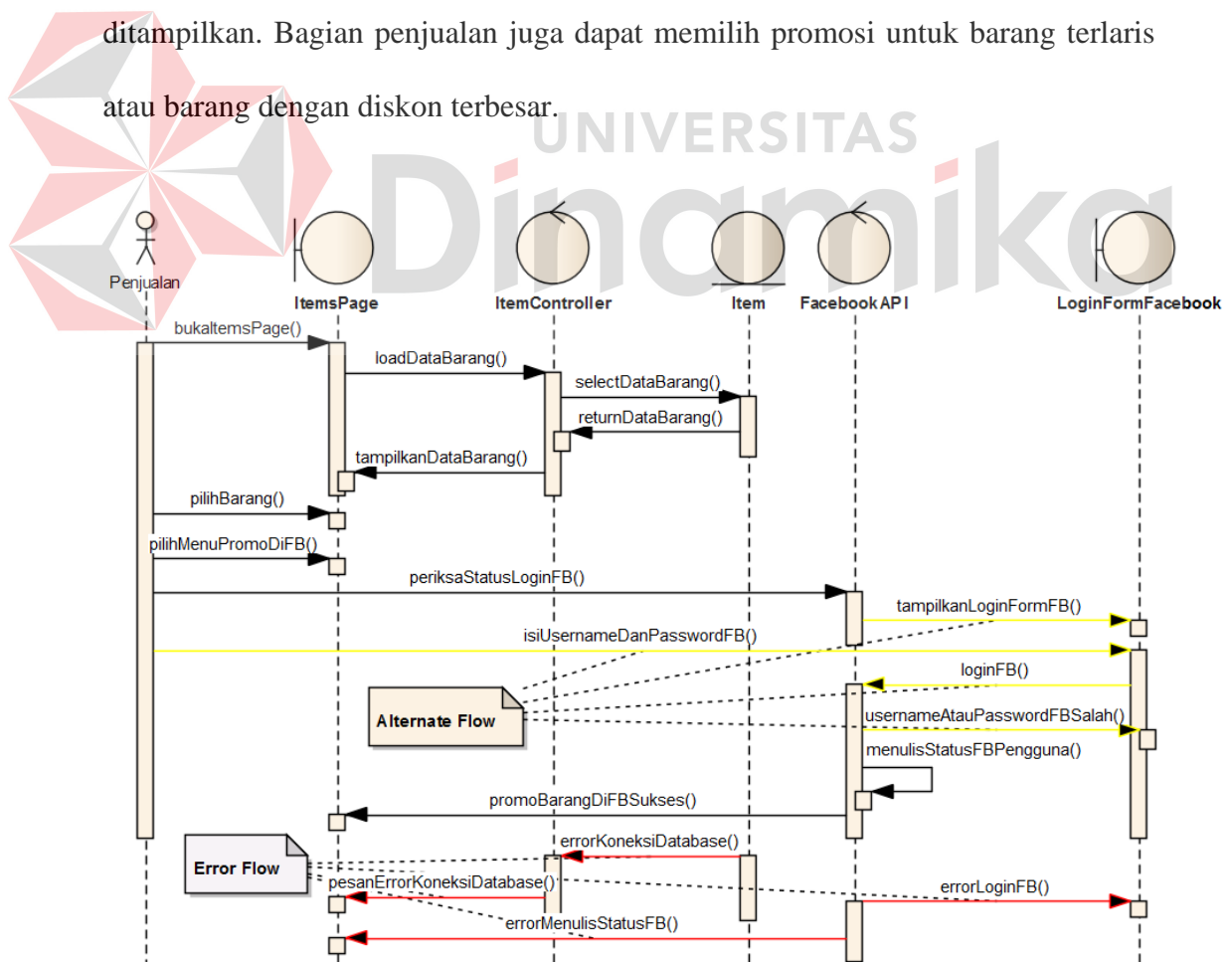
Diagram sekuensial pada gambar 3.22 menunjukkan proses mengubah paket layanan yang dimulai dari manajemen mengakses *boundary ChangeServicePage* dan memilih layanan yang diinginkan beserta jumlah bulan berlangganan. Lalu sistem memeriksa *format* masukan permintaan perubahan layanan. Jika *format* salah, sistem menyampaikannya ke manajemen. Jika *format* benar, sistem menyimpan permintaan perubahan layanan melalui *Control OnlineServicePaymentController* dan menampilkan nota tagihan. Lalu manajemen melakukan *transfer* sesuai tagihan dan melakukan konfirmasi pada *form* yang disediakan. Sistem akan memeriksa *format* masukan konfirmasi *transfer*. Jika ada *format* yang salah, sistem menyampaikannya ke manajemen. Jika *format* benar, sistem memeriksa data *transfer*. Jika *transfer* sudah dilakukan, sistem mengaktifkan layanan yang baru.



Gambar 3. 22 Diagram Sekuensial Mengubah Layanan

L. Diagram Sekuensial Promosi di Facebook

Diagram sekuensial pada gambar 3.23 menunjukkan proses promosi di Facebook yang dimulai dari bagian penjualan mengakses *boundary ItemsPage*. Lalu *boundary ItemsPage* meminta *control ItemController* untuk mengambil data *Item* dari *database* untuk ditampilkan. Setelah itu bagian penjualan memilih data barang yang akan dipromosikan dan memilih menu promosi di Facebook. Sistem akan terhubung dengan Facebook API untuk memeriksa apakah pengguna sudah *login*. Jika belum, sistem meminta pengguna untuk *login*. Jika sudah, Facebook API akan meminta pengguna mengkonfirmasi status promosi yang akan di-*post* melalui Facebook. Setelah itu pesan sukses promosi di Facebook akan ditampilkan. Bagian penjualan juga dapat memilih promosi untuk barang terlaris atau barang dengan diskon terbesar.



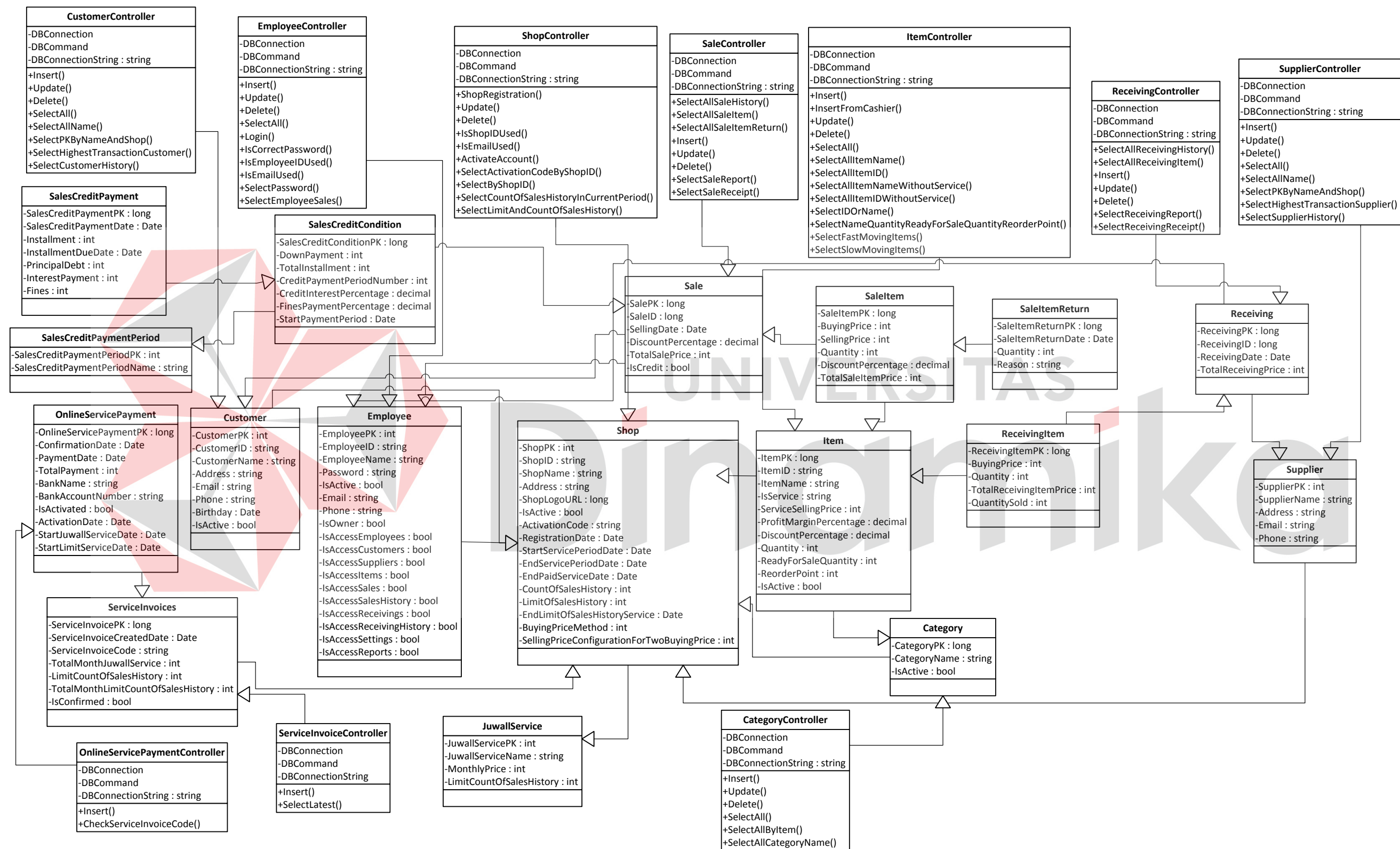
Gambar 3. 23 Diagram Sekuensial Promosi di Facebook

3.3.5 Class Diagram Sistem Informasi Penjualan

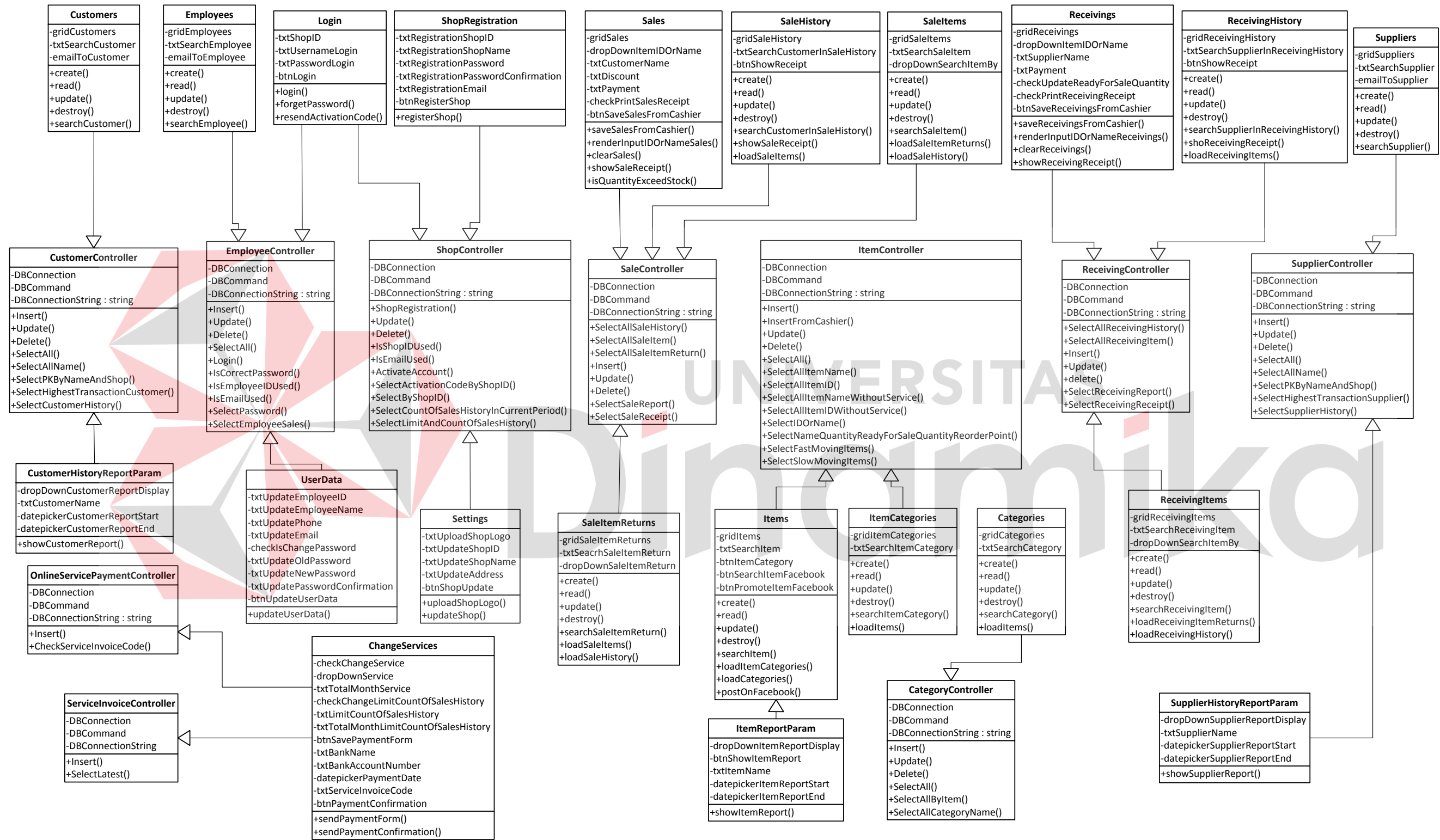
Class diagram adalah diagram yang menunjukkan interaksi antar kelas dalam sistem. Diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram kelas membantu tim pengembang mendapatkan pola kelas-kelas dalam sistem, struktur sistem sebelum menuliskan kode program dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif rancangan (Sholih; 2010).

Class yang dibuat dalam sistem ini menggunakan *pattern* MVC (*Model-View-Controller*). *MVC pattern* adalah sebuah *pattern* dalam teknik pemrograman yang memisahkan ketiga bagian program menjadi tiga buah objek yang terpisah (Azis; 2005). *Controller* merupakan penerima tamu dari permintaan yang datang (*HTTP Request*). Fungsinya adalah untuk mengevaluasi permintaan yang datang, kemudian mengarahkannya ke objek yang tepat untuk memproses permintaan tersebut. *View* adalah hasil akhir yang dikirimkan ke pengguna aplikasi. *Format HTML* merupakan contoh umum hasil akhir yang dikirimkan dalam aplikasi web. *Model* merupakan representasi dari proses bisnis di dalam aplikasi perangkat lunak. Lebih mudahnya, *model* adalah bagian yang bertugas untuk mengolah data mentah menjadi data yang mengandung arti yang diinginkan oleh pengguna.

Class diagram akan dibagi menjadi dua gambar, yaitu gambar 3.24 untuk *class diagram* yang menggambarkan interaksi *model* dan *controller* serta gambar 3.25 untuk *class diagram* yang menggambarkan interaksi *view* dan *controller*.



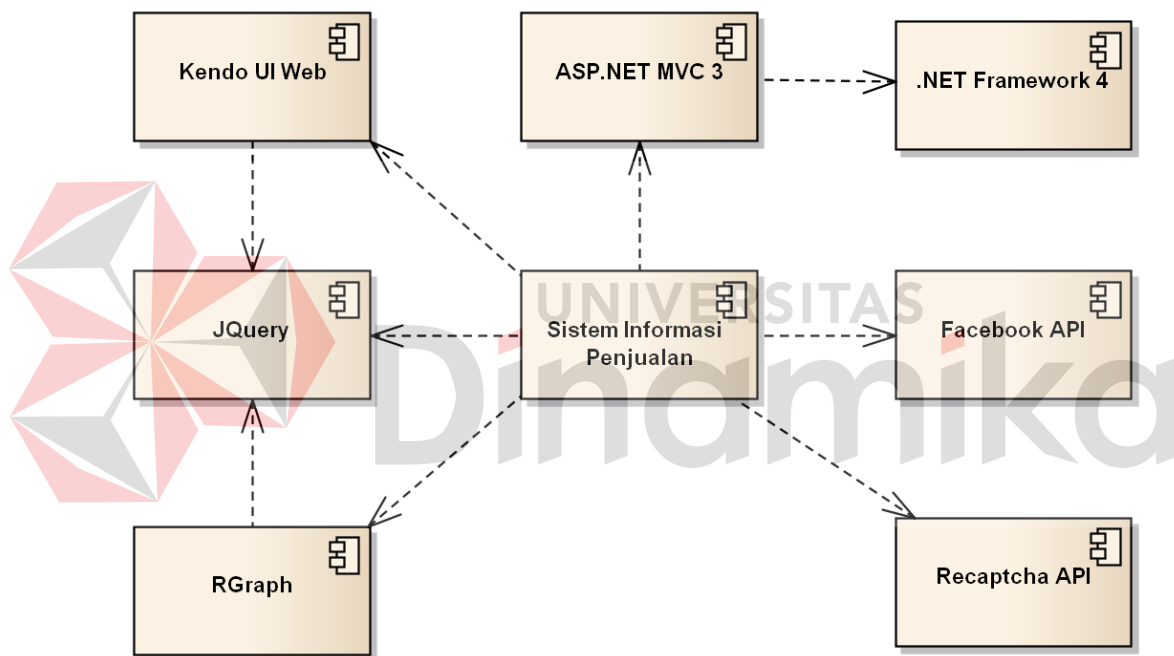
Gambar 3. 24 Class Diagram Model dan Controller



Gambar 3. 25 Class Diagram Controller dan View

3.3.6 Component Diagram Sistem Informasi Penjualan

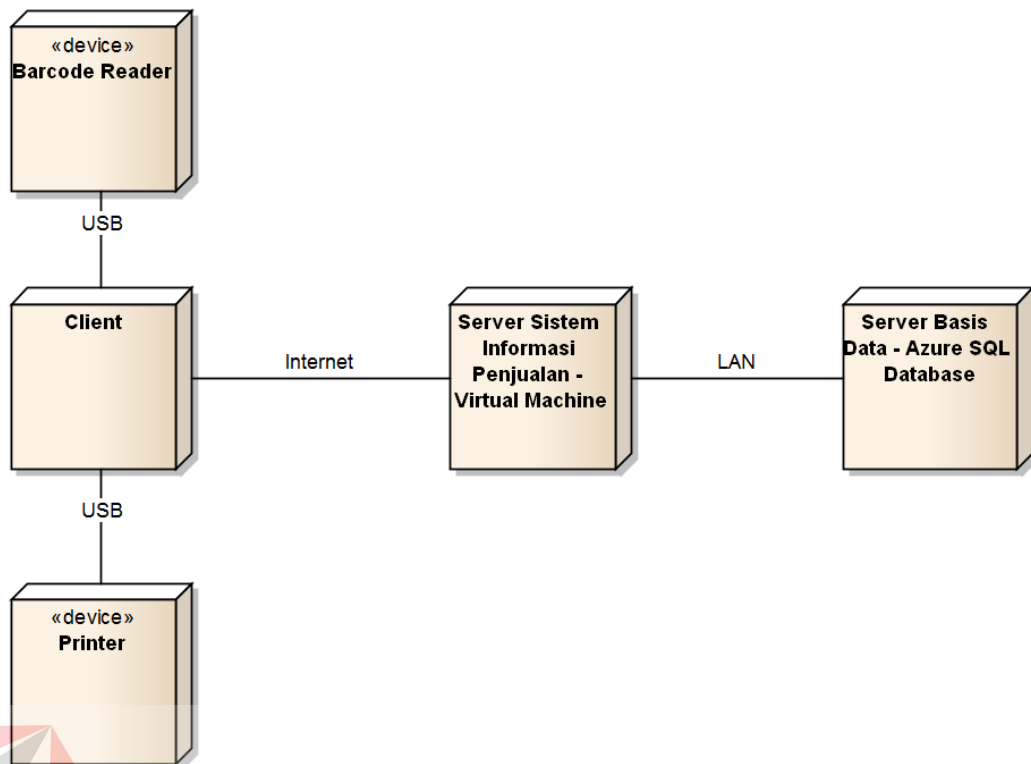
Component diagram adalah diagram yang menunjukkan model secara fisik atau komponen perangkat lunak pada sistem dan hubungannya antar mereka (Sholiq; 2010). Komponen sendiri merupakan modul fisik dari kode. Komponen bisa mencantumkan pustaka kode program dan berkas-berkas runtime sekaligus. Gambar 3.26 di bawah menggambarkan *component diagram* dari sistem informasi penjualan yang akan dibangun.



Gambar 3. 26 *Component Diagram*

3.3.7 Deployment Diagram Sistem Informasi Penjualan

Deployment diagram adalah diagram yang menampilkan rancangan fisik jaringan tempat berbagai komponen akan diletakkan disana. Diagram ini menghubungkan titik-titik alat pemroses dan peralatan lainnya dalam sebuah *layout* fisik (Sholiq; 2010). Gambar 3.27 menggambarkan *deployment diagram* dari sistem informasi penjualan yang akan dibangun.



Gambar 3.27 Deployment Diagram

3.3.8 Rencana Penggunaan Infrastruktur Cloud

Sistem informasi penjualan yang dibangun merupakan sistem berbasis *cloud* karena sistem digunakan oleh UKM sebagai *Software As A Service (SAAS)* dan sistem menggunakan infrastruktur *cloud* sebagai *application server (IAAS)* dan *database server (PAAS)*. *Provider* cloud yang digunakan untuk *IAAS* dan *PAAS* adalah Windows Azure dari Microsoft. Berikut akan dijelaskan penggunaan *cloud* pada sistem informasi penjualan yang dibangun.

A. *Virtual Private Server* sebagai *IAAS* bagi *Application Server*

Application server digunakan untuk menyimpan aplikasi dan menangani permintaan atau *request* yang diterima oleh pengguna. Teknologi *cloud* yang digunakan adalah *Infrastructure As A Service (IAAS)* dalam bentuk *Virtual*

Private Server. *IAAS* dipilih karena pengguna memiliki kontrol penuh untuk menginstall program-program atau komponen-komponen di dalam *server*. Hal ini sangat cocok dengan sistem informasi penjualan yang dibangun karena proses implementasinya memerlukan komponen-komponen khusus yang perlu diatur dan dikonfigurasi sendiri. Walaupun demikian, pengguna sudah tidak perlu memikirkan *availability* dan *reliability* dari *server* karena hal tersebut telah menjadi tanggung jawab *cloud provider*.

B. *SQL Azure sebagai PAAS Bagi Database Server*

Database server digunakan untuk menyimpan data-data milik pengguna sistem informasi penjualan. Teknologi *cloud* yang digunakan adalah *Platform As A Service (PAAS)* dalam bentuk *SQL Azure*. *PAAS* dipilih karena teknologi *SQL Azure* sesuai dengan *DBMS* yang digunakan yaitu *SQL Server*. Dengan menggunakan *SQL Azure*, pengguna tidak perlu memikirkan konfigurasi, optimasi, *availability* dan *reliability* karena merupakan tanggung jawab *cloud provider*. Pengguna hanya perlu meng-*upgrade database* sesuai kebutuhan sistem dan pengguna dan meng-*upload file database* tersebut di *SQL Azure*. Pemisahan *database server* dan *application server* ditujukan agar pembagian kerja antara pemrosesan aplikasi dan data dapat menjadi lebih ringan.

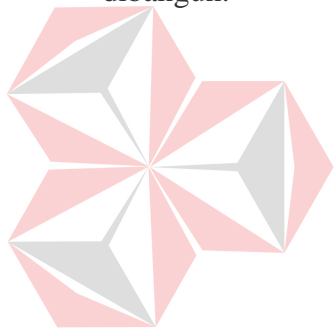
C. *Sistem Informasi Penjualan sebagai SAAS*

Sistem informasi penjualan yang dibangun termasuk dalam kategori *software as a service*. Hal ini dikarenakan pengguna (UKM) dapat langsung menggunakan layanan sistem informasi penjualan dengan mendaftar pada *website* yang digunakan dan mengkonfigurasi layanan sesuai dengan yang dibutuhkan.

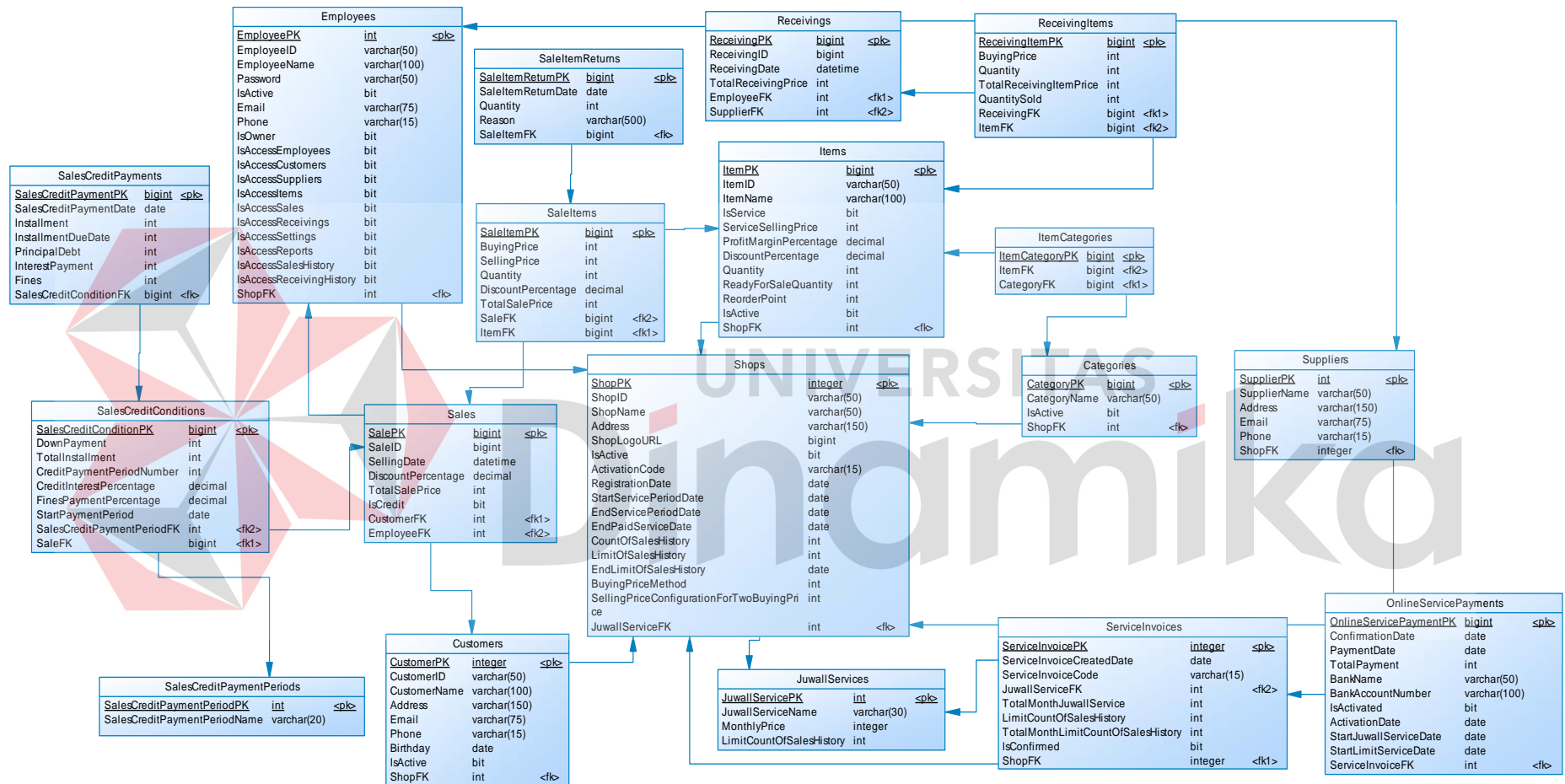
UKM hanya perlu membayar biaya berlangganan sesuai yang dibutuhkan. Dengan demikian, UKM tidak perlu menyediakan *server* dan tidak perlu melakukan *maintenance* sistem sendiri karena hal tersebut telah menjadi tanggung jawab *service provider*. UKM hanya perlu menyediakan perangkat akses untuk mengakses sistem informasi penjualan.

3.3.9 *Physical Data Model*

Physical data model menggambarkan tabel-tabel, tipe data dan relasi pada *database* dari sistem yang akan dibangun. Gambar 3.28 menggambarkan *physical data model* dari sistem informasi penjualan berbasis *cloud* dari sistem yang akan dibangun.



UNIVERSITAS
Dinamika



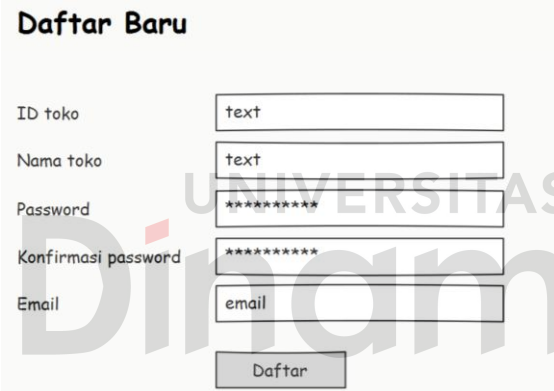
Gambar 3. 28 Physical Data Model

3.3.10 Desain *User Interface*

Desain *user interface* diperlukan untuk menghasilkan tampilan yang digunakan pengguna untuk berinteraksi dengan sistem. Desain *user interface* harus dibuat semudah mungkin untuk digunakan oleh pengguna.

A. Desain *User Interface* Halaman Pendaftaran Toko

Desain *user interface* halaman pendaftaran toko digambarkan pada gambar 3.29. Halaman ini digunakan oleh pengguna yang akan mendaftarkan toko mereka untuk menggunakan sistem informasi penjualan berbasis *cloud*.



Daftar Baru

ID toko

Nama toko

Password

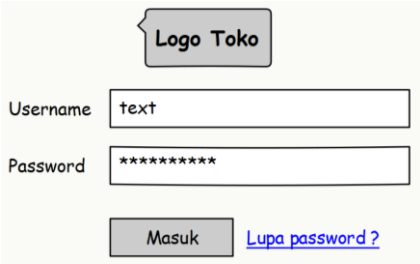
Konfirmasi password

Email

Gambar 3. 29 Desain *User Interface* Halaman Pendaftaran Toko

B. Desain *User Interface* Halaman Login

Desain *user interface* halaman *login* digambarkan pada gambar 3.30. Halaman ini digunakan oleh pengguna untuk *login* ke dalam sistem.



Username

Password

[Lupa password ?](#)

Gambar 3. 30 Desain *User Interface* Form Login

C. Desain *User Interface* Halaman Penjualan

Desain *user interface* halaman penjualan digambarkan pada gambar 3.31.

Halaman ini digunakan oleh bagian penjualan untuk mencatat data penjualan.

Gambar 3. 31 Desain *User Interface* Halaman Penjualan (Tunai)

Gambar 3.31 menunjukkan *form* pembayaran jika transaksi dilakukan secara tunai. Namun jika transaksi dilakukan secara kredit atau cicilan, maka akan muncul *form* pembayaran seperti gambar 3.32.

Gambar 3. 32 Desain *User Interface Form* Penjualan Kredit

D. Desain *User Interface* Halaman *History Penjualan*

Desain *user interface* halaman *history* penjualan digambarkan pada gambar 3.33. Halaman ini digunakan oleh bagian penjualan untuk melihat transaksi-transaksi penjualan yang pernah dilakukan.

No. Transaksi	Tanggal	Jenis	Pelanggan	Diskon	Total Harga	Karyawan	Detail Barang
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Gambar 3. 33 Desain *User Interface* Halaman *History Penjualan*

Pada tabel *history* penjualan di gambar 3.33, terdapat kolom *detail* barang yang berisi *button* untuk melihat barang-barang yang dibeli pada transaksi tersebut. Tampilan *detail* barang seperti di gambar 3.34.

ID barang	Nama barang	Harga	Jumlah	Diskon	Total harga	Retur
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Gambar 3. 34 Desain *User Interface* *History Penjualan - Detail Barang*

E. Desain *User Interface* Halaman Retur Penjualan

Untuk mencatat data retur penjualan, bagian penjualan harus masuk ke *history* penjualan dan memilih detail barang dari transaksi yang akan diretur. Setelah memilih barang yang akan diretur, *form* retur yang ditampilkan seperti pada gambar 3.35.

Gambar 3. 35 Desain *User Interface* Form Input Retur Penjualan

Apabila bagian penjualan ingin melihat semua data retur pada transaksi yang dipilih, maka bagian penjualan memilih menu “Barang Retur” di halaman *history* penjualan bagian detail barang. Halaman yang ditampilkan seperti pada gambar 3.36.

Tanggal retur	ID barang	Nama barang	Harga	Jumlah	Alasan
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-

Gambar 3. 36 Desain *User Interface* Halaman Retur Penjualan

F. Desain *User Interface* Halaman Pembayaran Kredit

Untuk mencatat pembayaran kredit yang dilakukan pelanggan, bagian penjualan harus masuk ke *history* penjualan dan memilih transaksi kredit yang akan dibayar. Setelah itu, bagian penjualan memilih menu pembayaran kredit dan akan muncul *form* pembayaran kredit seperti pada gambar 3.37.

Pembayaran kredit			
Tanggal	dd-mm-yyyy	Hutang pokok	0
No. Transaksi	1	Bunga	0
Nama pelanggan	ABCDE	Denda	0
Angsuran	0	Total bayar	0
Jatuh tempo angsuran	dd-mm-yyyy	Uang	<input type="text" value="number"/>
		Kembalian	0
		<input checked="" type="checkbox"/> cetak nota	<input type="button" value="Simpan"/>

Gambar 3. 37 Desain *User Interface* Form Pembayaran Kredit

Apabila bagian penjualan ingin melihat *history* pembayaran kredit, bagian penjualan memilih transaksi kredit yang diinginkan dan memilih menu *history* pembayaran kredit. Halaman yang akan ditampilkan adalah seperti gambar 3.38.

Tanggal bayar	Angsuran	Jatuh tempo	Hutang pokok	Bunga	Denda	Total
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

No. Transaksi
0

Nama pelanggan
ABCDE

Jumlah angsuran
0

Angsuran terbayar
0

Total hutang
0

Hutang terbayar
0

Gambar 3. 38 Desain *User Interface* History Pembayaran Kredit

G. Desain *User Interface* Halaman Penerimaan

Desain *user interface* halaman penerimaan digambarkan pada gambar 3.39. Halaman ini digunakan oleh bagian pengelola stok persediaan untuk mencatat data penerimaan barang dari *supplier*.

Gambar 3. 39 Desain *User Interface* Halaman Penerimaan

H. Desain *User Interface* Halaman *History* Penerimaan

Desain *user interface* halaman *history* penerimaan digambarkan pada gambar 3.40. Halaman ini digunakan oleh bagian pengelola stok persediaan untuk melihat transaksi-transaksi penerimaan yang pernah dilakukan.

Gambar 3. 40 Desain *User Interface* Halaman *History* Penerimaan

Pada tabel *history* penerimaan di gambar 3.40, terdapat kolom *detail* barang yang berisi *button* untuk melihat barang-barang yang diterima pada transaksi tersebut. Tampilan *detail* barang seperti di gambar 3.41.

ID barang	Nama barang	Harga	Jumlah	Total harga	Retur
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-

Gambar 3. 41 Desain *User Interface* History Penerimaan - *Detail* Barang

I. Desain *User Interface* Halaman Pelanggan

Desain *user interface* halaman pelanggan digambarkan pada gambar 3.42.

Halaman ini digunakan oleh bagian penjualan untuk mengelola data pelanggan.

ID pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	Hapus
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Gambar 3. 42 Desain *User Interface* Halaman Pelanggan

J. Desain *User Interface* Halaman Karyawan

Desain *user interface* halaman karyawan digambarkan pada gambar 3.43. Halaman ini digunakan oleh pihak manajemen untuk mengelola data karyawan beserta hak aksesnya terhadap sistem.

ID karyawan	Nama	Email	Telepon	Ubah	Hapus
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-

Gambar 3. 43 Desain *User Interface* Halaman Karyawan

Untuk menambah atau mengubah data karyawan beserta hak aksesnya, sistem akan memberikan tampilan seperti pada gambar 3.44.

Nama	text	ID Karyawan	text
Email	email	Password	*****
Telepon	text	Konfirmasi password	*****

Hak akses

- Kasir penjualan (Tambah)
- Kasir penerimaan barang (Tambah)
- History Penjualan (Retur, Ubah, Hapus)
- History Penerimaan Barang (Retur, Ubah, Hapus)
- Karyawan (Tambah, Ubah, Hapus)
- Barang (Tambah, Ubah, Hapus)
- Pelanggan (Tambah, Ubah, Hapus)
- Supplier (Tambah, Ubah, Hapus)
- Laporan (Laporan semua aktivitas kasir)
- Setting kasir (Pengaturan kasir)

Gambar 3. 44 Desain *User Interface* Form Karyawan

K. Desain *User Interface* Halaman Barang

Desain *user interface* halaman karyawan digambarkan pada gambar 3.45. Halaman ini digunakan oleh bagian pengelola stok persediaan untuk mengelola data barang.

Gambar 3. 45 Desain *User Interface* Halaman Barang

Untuk mempromosikan data barang tertentu di Facebook, pengguna memilih barang pada halaman barang lalu memilih menu “Promo di Facebook” dan sistem akan menampilkan *form* seperti pada gambar 3.46. Promosi dapat dilakukan pada masing-masing data barang, barang terlaris atau barang dengan diskon terbesar.

Gambar 3. 46 Desain *User Interface* Form Promo Barang di Facebook

L. Desain *User Interface* Halaman Kategori

Desain *user interface* halaman kategori digambarkan pada gambar 3.47. Halaman ini digunakan oleh bagian pengelola stok persediaan untuk mengelola data kategori barang.

Gambar 3. 47 Desain *User Interface* Halaman Kategori

Untuk memasukkan barang ke dalam kategori barang tertentu, bagian pengelolaan persediaan harus memilih barang yang akan dimasukkan pada halaman barang dan memilih menu “Atur Kategori”. Sistem akan menampilkan seperti gambar 3.48.

Gambar 3. 48 Desain *User Interface* Halaman Pengaturan Kategori

M. Desain User Interface Halaman Supplier

Desain *user interface* halaman *supplier* digambarkan pada gambar 3.49. Halaman ini digunakan oleh bagian pengelola stok persediaan untuk mengelola data karyawan *supplier*.

The screenshot shows a web application interface for managing suppliers. At the top, there is a navigation bar with several menu items: Logo, Penjualan, Penerimaan, Pelanggan, Karyawan, Barang, Supplier, Laporan, and Setting. A dropdown menu is currently open over the 'Supplier' item, showing the word 'toko'. Below the navigation bar, the main content area is titled 'Supplier'. It contains a table with the following structure:

Nama	Email	Telepon	Hapus
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

To the right of the table, there is a search section with the label 'Cari supplier', a text input field containing 'text', and an 'Email ke supplier' button.

Gambar 3. 49 Desain User Interface Halaman Supplier

N. Desain User Interface Input Parameter Laporan Barang

Desain *user interface* halaman *input parameter* laporan barang digambarkan pada gambar 3.50 dan 3.51. Halaman ini digunakan oleh pihak manajemen untuk memilih jenis laporan dan memasukkan *parameter* laporan barang sesuai informasi yang dibutuhkan.

The screenshot shows a form for configuring report parameters. It is divided into two main sections: 'Jenis laporan barang' and 'Parameter laporan'. In the 'Jenis laporan barang' section, there is a dropdown menu with the text 'Pilih jenis laporan barang yang anda inginkan' and the selected option 'Laporan stok terkini'. Below this is a 'Deskripsi laporan' section. In the 'Parameter laporan' section, there is a dropdown menu with the text 'Tampilan laporan' and the selected option 'Grafis'. A 'Tampilkan' button is located to the right of the 'Parameter laporan' section.

Gambar 3. 50 Desain User Interface Parameter Laporan Stok Terkini

Laporan yang dapat dipilih adalah laporan stok terkini dan tren penjualan. Tampilan laporan yang dapat dipilih adalah tabular atau grafis. Gambar 3.50 menampilkan *parameter* dari laporan stok terkini. Gambar 3.51 menampilkan *parameter* dari laporan tren penjualan.

Gambar 3. 51 Desain *User Interface Parameter* Laporan Tren Penjualan

O. Desain *User Interface Input Parameter* Laporan *History* Pelanggan

Desain *user interface* halaman *input parameter* laporan *history* pelanggan digambarkan pada gambar 3.52. Halaman ini digunakan oleh pihak manajemen untuk memasukkan *parameter* laporan *history* pelanggan sesuai informasi yang dibutuhkan.

Gambar 3. 52 Desain *User Interface Parameter History* Pelanggan

P. Desain User Interface Input Parameter Laporan History Supplier

Desain *user interface* halaman *input parameter* laporan *history supplier* digambarkan pada gambar 3.53. Halaman ini digunakan oleh pihak manajemen untuk memasukkan *parameter* laporan *history* pelanggan sesuai informasi yang dibutuhkan.

Gambar 3. 53 Desain *User Interface* Parameter *History Supplier*

Q. Desain User Interface Halaman Ubah Layanan

Desain *user interface* halaman *ubah layanan* digambarkan pada gambar 3.54. Halaman ini digunakan oleh pihak manajemen untuk memasukkan data paket layanan yang diinginkan sesuai dengan kebutuhan toko.

Gambar 3. 54 Desain *User Interface* Halaman *Ubah Layanan*

R. Desain *User Interface* Halaman Konfirmasi Pembayaran

Desain *user interface* halaman ubah layanan digambarkan pada gambar 3.55. Halaman ini digunakan oleh pihak manajemen untuk memasukkan konfirmasi pembayaran melalui *transfer* ATM yang dilakukan sesuai nota tagihan pengubahan layanan yang diterima.

Konfirmasi pembayaran	
Nama bank	text
Nomer rekening	text
Jumlah pembayaran	number
Tanggal transfer	date
Kode tagihan	text
Kirim	

Gambar 3. 55 Desain *User Interface* Halaman Konfirmasi Pembayaran

3.3.11 Desain *Input/Output*

Desain *input/output* digunakan untuk menggambarkan *input* dan *output* dari lingkungan eksternal sistem yang akan dimasukkan ke dalam sistem. Dalam sistem informasi penjualan yang akan dibangun, desain *input* tidak digambarkan karena semua *input* dimasukkan secara langsung ke dalam sistem. Desain *output* yang dihasilkan sistem adalah nota transaksi beserta laporan yang dibutuhkan manajemen.

A. Desain *Output* Nota Transaksi Penjualan

Nota transaksi penjualan adalah *output* untuk setiap transaksi penjualan yang dimasukkan di menu penjualan. Nota penjualan dibagi menjadi dua jenis, yaitu nota penjualan tunai dan kredit. Nota penjualan tunai berisi tanggal

transaksi, nomer transaksi, nama pelanggan, nama barang yang dibeli beserta jumlah, harga satuan, diskon satuan dan total harga. Hasil *output* nota penjualan tunai dapat dilihat di gambar 3.56.

Nota penjualan - [Nama Toko]				
<div style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> logo </div>				
Tanggal			No. Transaksi	
			Nama pelanggan	
Nama barang	Jumlah	Harga satuan	Diskon	Total harga
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Tanda terima, [Nama toko] Diskon penjualan
Total harga

Gambar 3. 56 Desain *Output* Nota Transaksi Penjualan Tunai

Nota penjualan kredit berisi tanggal transaksi, nomer transaksi, nama pelanggan, diskon penjualan, total harga, uang muka, sisa hutang, angsuran, periode pembayaran, bunga, denda tiap periode dan awal periode pembayaran.

Hasil *output* nota penjualan tunai dapat dilihat di gambar 3.57.

Nota penjualan kredit - [Nama Toko]				
<div style="border: 1px solid black; border-radius: 50%; width: 30px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> logo </div>				
Tanggal			No. Transaksi	
			Nama pelanggan	
Nama barang	Jumlah	Harga satuan	Diskon	Total harga
-	-	-	-	-
-	-	-	-	-


Diskon penjualan	0	Sisa hutang	0	Bunga (%)	0
Total harga	0	Angsuran	0	Denda tiap periode (%)	0
Uang muka	0	Periode pembayaran	0	Awal periode pembayaran	0

Tanda terima,
[Nama toko]

Gambar 3. 57 Desain *Output* Nota Transaksi Penjualan Kredit

B. Desain *Output* Nota Retur Transaksi Penjualan

Nota transaksi retur penjualan adalah *output* untuk setiap transaksi retur penjualan yang dimasukkan di menu penjualan. Nota retur penjualan berisi tanggal retur, nomer transaksi, nama pelanggan, nama barang beserta jumlah, harga satuan, diskon satuan, total harga dan alasan retur. Hasil *output* nota retur penjualan dapat dilihat di gambar 3.58.

Nota retur penjualan - [Nama Toko]					
					
Tanggal			No. Transaksi		
			Nama pelanggan		
Nama barang	Alasan	Jumlah	Harga satuan	Diskon	Total harga
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
Tanda terima,				Total harga	
[Nama toko]					

Gambar 3. 58 Desain *Output* Nota Retur Transaksi Penjualan

C. Desain *Output* Nota Transaksi Pembayaran Kredit

Nota transaksi pembayaran kredit adalah *output* untuk setiap transaksi pembayaran kredit yang dimasukkan di menu penjualan. Nota pembayaran kredit berisi tanggal transaksi, nomer transaksi, nama pelanggan, angsuran, sisa angsuran, tanggal jatuh tempo, hutang pokok, bunga, denda, total bayar dan tanggal jatuh tempo selanjutnya. Hasil *output* nota pembayaran kredit dapat dilihat di gambar 3.59.

Nota pembayaran kredit - [Nama Toko]			
<div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> logo </div>			
Tanggal	dd-mm-yyyy	Hutang pokok	0
No. Transaksi	0	Bunga	0
Nama pelanggan	ABCDE	Denda	0
Angsuran	1	Total bayar	0
Tanggal jatuh tempo	dd-mm-yyyy	Tanggal jatuh tempo selanjutnya	dd-mm-yyyy
Sisa angsuran	1		
			Tanda terima,
			[Nama toko]

Gambar 3. 59 Desain *Output* Nota Transaksi Pembayaran Kredit

D. Desain *Output* Nota Transaksi Penerimaan Barang

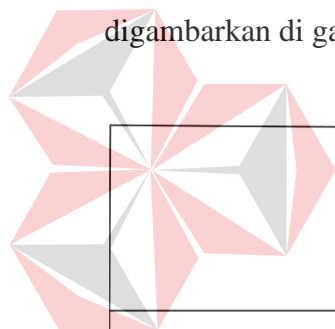
Nota transaksi penerimaan barang adalah *output* untuk setiap transaksi penerimaan barang yang dimasukkan di menu penerimaan. Nota penerimaan barang berisi tanggal transaksi, nomer transaksi, nama *supplier*, *id* barang yang dibeli beserta nama barang, jumlah, harga satuan dan total harga. Hasil *output* nota penerimaan barang dapat dilihat di gambar 3.60.

Nota penerimaan barang - [Nama Toko]				
<div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> logo </div>				
Tanggal		No. Transaksi		
		Nama supplier		
ID barang	Nama barang	Jumlah	Harga satuan	Total harga
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
Tanda terima,		Total harga		
[Nama toko]				

Gambar 3. 60 Desain *Output* Nota Transaksi Penerimaan Barang

E. Desain *Output* Laporan Stok Terkini

Laporan stok terkini adalah *output* dari sistem untuk pihak manajemen agar dapat memantau dan mengetahui stok terkini yang tersedia. Hal ini membantu pihak manajemen untuk mengambil keputusan *order* ke *supplier* sehingga toko tidak kehabisan stok. Laporan stok terkini dapat dilihat dalam dua tampilan, yaitu grafis dan tabular. Laporan stok terkini dalam bentuk tabular berisi tanggal stok terkini diakses, *id* barang, nama barang, harga beli, harga jual, stok tersedia, stok siap jual dan stok minimum. Bentuk tabular digambarkan di gambar 3.61. Laporan stok terkini dalam bentuk grafis berisi tanggal stok terkini diakses, nama barang, stok tersedia, stok siap jual dan stok minimum. Bentuk grafis digambarkan di gambar 3.62.



UNIVERSITAS
Dinamika

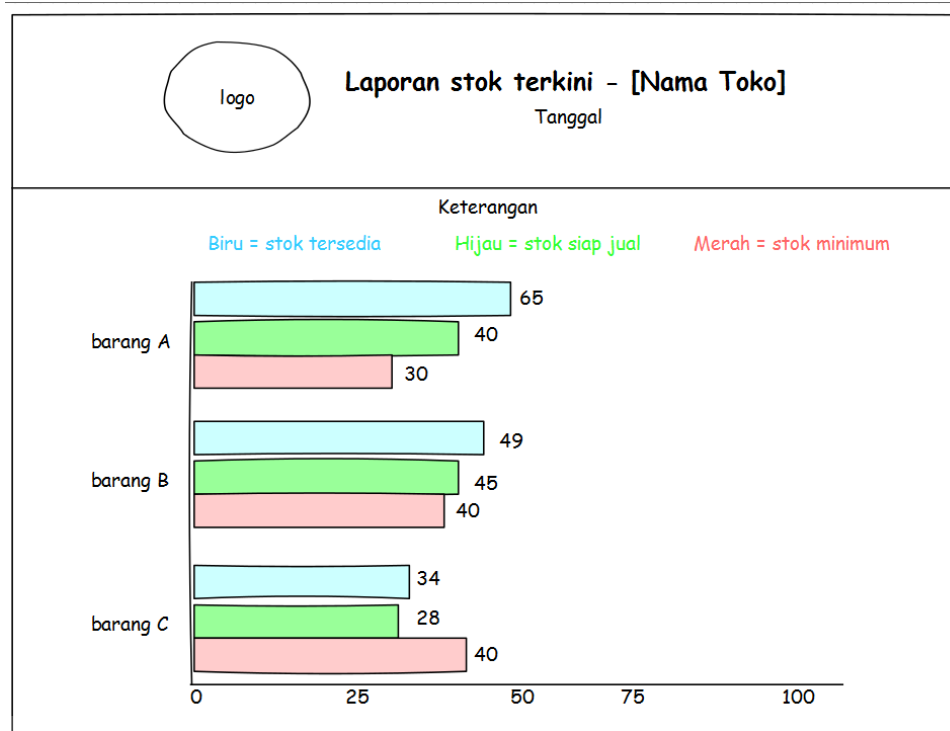
Laporan stok terkini - [Nama Toko]

logo

Tanggal

ID barang	Nama barang	Stok tersedia	Stok siap jual	Stok minimum
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-


Gambar 3. 61 Desain *Output* Laporan Tabular Stok Terkini



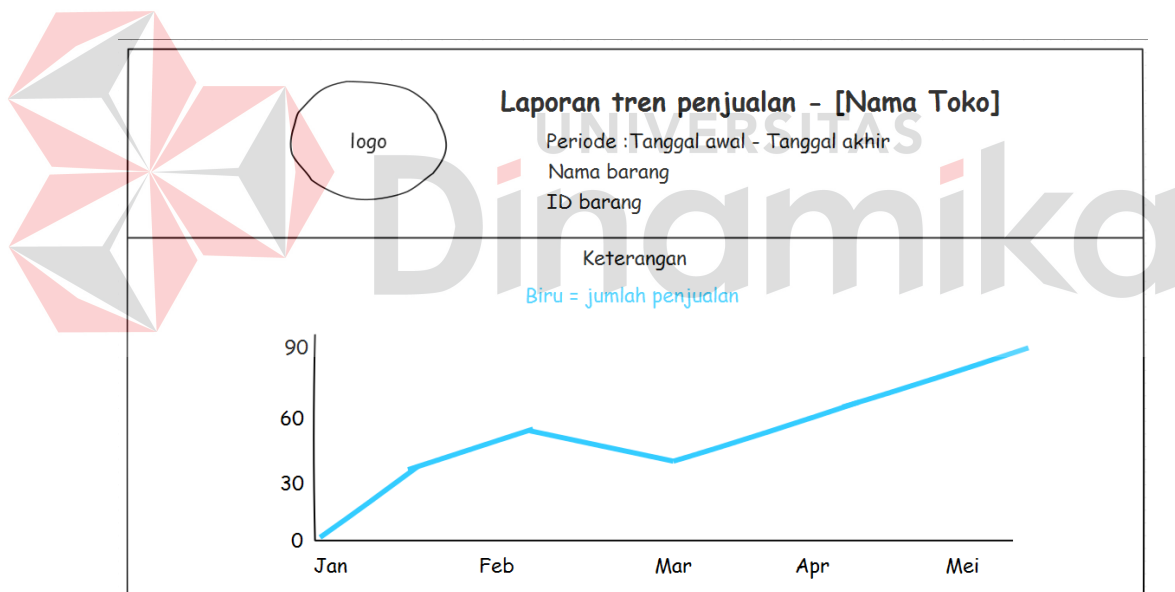
Gambar 3. 62 Desain *Output* Laporan Grafis Stok Terkini

F. Desain *Output* Laporan Tren Penjualan

Laporan tren penjualan adalah *output* dari sistem untuk membantu pihak manajemen dalam mengetahui tren penjualan produk yang dijualnya untuk mengambil keputusan seperti jumlah *reorder* ke *supplier*, pemberian diskon dan strategi promosi. Laporan tren penjualan dapat dilihat dalam 2 bentuk, yaitu tabular dan grafis. Penjualan juga dapat dilihat dalam periode harian, bulanan atau tahunan. Laporan tren penjualan berisi tanggal akses laporan, *id* barang, nama barang dan jumlah penjualan tiap periode. Laporan dalam bentuk tabular digambarkan di gambar 3.63. Laporan dalam bentuk grafis digambarkan di gambar 3.64.

Laporan tren penjualan - [Nama Toko]	
	
Tanggal	ID barang Nama barang
Periode	Jumlah penjualan
-	-
-	-
-	-

Gambar 3. 63 Desain *Output* Laporan Tabular Tren Penjualan




Gambar 3. 64 Desain *Output* Laporan Grafis Tren Penjualan

G. Desain *Output* Laporan *History* Pelanggan

Laporan *history* pelanggan adalah *output* dari sistem yang membantu pihak manajemen untuk melihat transaksi-transaksi yang pernah dilakukan pelanggan-pelanggannya. Informasi ini membantu mengetahui siapa saja pelanggan utama mereka. Laporan *history* pelanggan dapat dilihat dalam 2 bentuk, yaitu grafis dan

tabular. Laporan dalam bentuk tabular berisi data pelanggan, periode transaksi beserta *detail* transaksi yang pernah dilakukannya, seperti *id* barang, nama barang, jumlah, harga beli, harga jual, diskon, total harga beserta keuntungan. Selain itu, informasi total penjualan dan total keuntungan juga ditampilkan. Laporan dalam bentuk grafis berisi nama pelanggan, periode transaksi, jumlah penjualan, nilai penjualan, jumlah keuntungan dan barang-barang yang dibeli beserta jumlahnya. Laporan dalam bentuk tabular digambarkan di gambar 3.65. Laporan dalam bentuk grafis digambarkan di gambar 3.66.



Laporan history pelanggan - [Nama Toko]

(logo)

Nama pelanggan
Alamat
Email
Telepon
Periode

01 Januari 2013

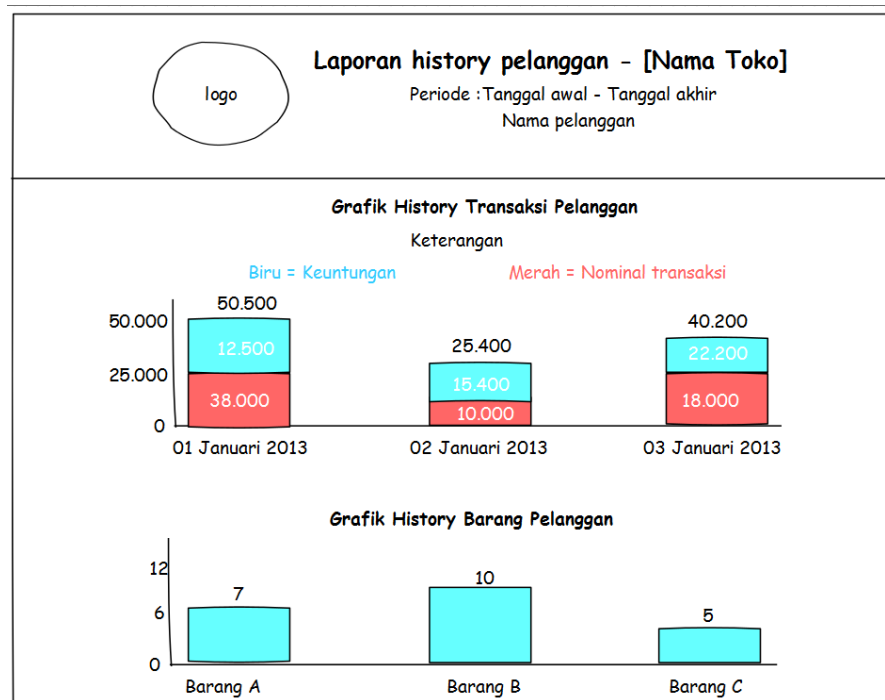
ID barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

02 Januari 2013

ID barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Total penjualan Total keuntungan


Gambar 3. 65 Desain *Output* Laporan Tabular *History* Pelanggan



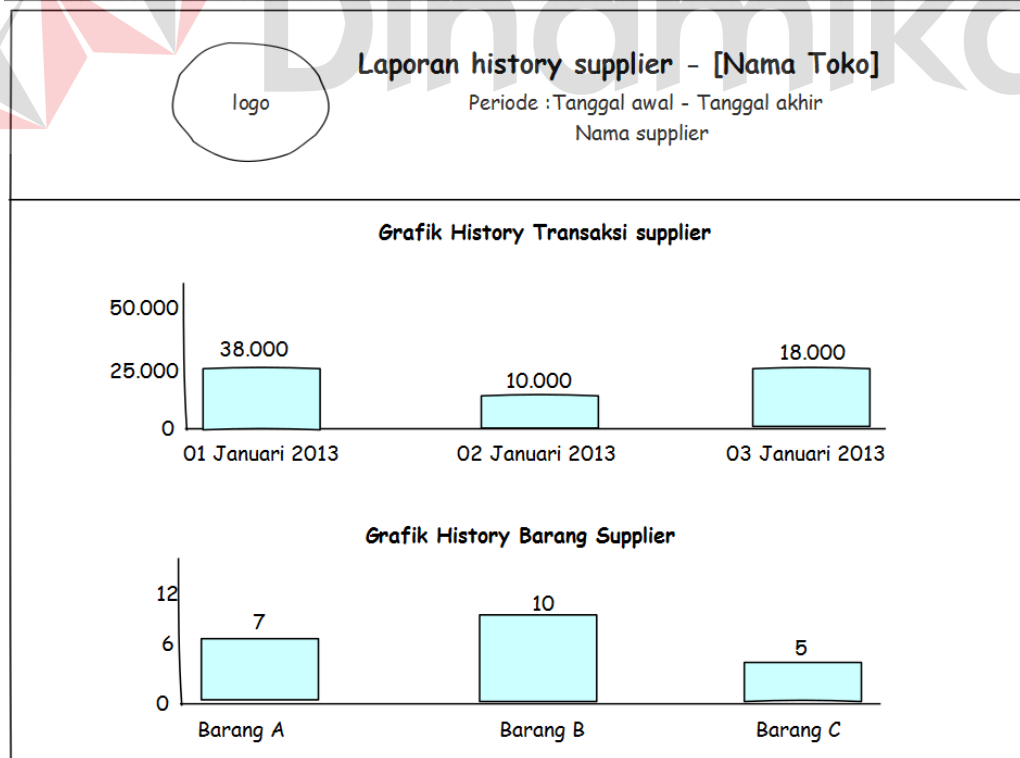
Gambar 3. 66 Desain *Output* Laporan Grafis *History* Pelanggan

H. Desain *Output* Laporan *History Supplier*

Laporan *history supplier* adalah *output* dari sistem yang membantu pihak manajemen untuk melihat transaksi-transaksi yang pernah dilakukan *supplier* mereka. Informasi ini membantu mengetahui siapa saja *supplier* utama mereka. Laporan *history supplier* dapat dilihat dalam 2 bentuk, yaitu grafis dan tabular. Laporan dalam bentuk tabular berisi data *supplier*, periode transaksi beserta *detail* transaksi yang pernah dilakukannya, seperti *id* barang, nama barang, jumlah, harga satuan dan total harga. Selain itu, informasi total pengeluaran juga ditampilkan. Laporan dalam bentuk grafis berisi nama *supplier*, periode transaksi, jumlah penjualan dan barang-barang yang diterima toko beserta jumlahnya. Laporan dalam bentuk tabular digambarkan di gambar 3.67. Laporan dalam bentuk grafis digambarkan di gambar 3.68.

Laporan history supplier - [Nama Toko]				
				
Nama supplier Alamat Email Telepon Periode				
01 Januari 2013				
ID barang	Nama barang	Jumlah	Harga satuan	Total harga
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
02 Januari 2013				
ID barang	Nama barang	Jumlah	Harga satuan	Total harga
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
Total pengeluaran				

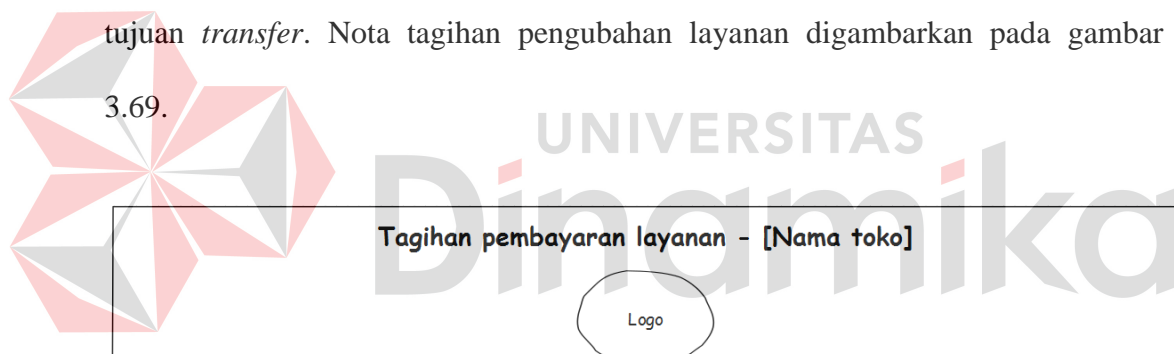
Gambar 3. 67 Desain *Output* Laporan Tabular *History Supplier*



Gambar 3. 68 Desain *Output* Laporan Grafis *History Supplier*

I. Desain Output Nota Tagihan Pengubahan Layanan

Nota tagihan pengubahan layanan merupakan *output* dari sistem untuk pihak manajemen ketika melakukan pengubahan layanan. Nota ini memberikan informasi mengenai layanan yang diinginkan beserta jumlah tagihan untuk layanan tersebut. Pihak manajemen harus membayar tagihan ini melalui *transfer* ATM sebelum dapat menggunakan layanan yang diinginkan. Isi dari nota tagihan pengubahan layanan adalah data toko, tanggal tagihan, kode tagihan untuk konfirmasi *transfer*, layanan yang diinginkan, jumlah bulan, harga bulanan, total harga dan total harga keseluruhan. Nota tagihan juga berisi informasi nama bank, nomer rekening dan nama pemilik rekening penyedia layanan yang menjadi tujuan *transfer*. Nota tagihan pengubahan layanan digambarkan pada gambar 3.69.



Tagihan pembayaran layanan - [Nama toko]

Logo

Ditagihkan kepada, Nama toko	Dibayarkan kepada, Nama penyedia layanan	Tanggal tagihan Kode tagihan
---------------------------------	---	---------------------------------

Layanan	Jumlah bulan	Harga bulanan	Total harga
-	-	-	-
-	-	-	-

Silahkan ditransfer ke rekening sebelum tanggal jatuh tempo yaitu, XX-XX-XXXX. Berikut adalah detail rekening :

Nama bank

No. rekening

Atas nama

Terima Kasih

Gambar 3. 69 Desain *Output* Nota Tagihan Pengubahan Layanan

3.3.12 Rancangan Pengujian dan Evaluasi Sistem

Sistem informasi penjualan yang telah dirancang dan dibangun harus diuji untuk mengetahui apakah sistem berjalan sesuai dengan yang diharapkan. Secara garis besar, uji coba yang dilakukan adalah pendaftaran toko, proses *login*, pengelolaan data *master*, pencatatan transaksi, pemberian laporan kepada pihak manajemen serta proses perubahan layanan. Pengujian akan dilakukan dengan metode *Black Box Testing*. *Black Box Testing* fokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. (Al Fatta; 2007). Cara pengujian dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan. Secara *detail*, hal-hal yang diujikan adalah sebagai berikut:

Tabel 3. 14 Rencana Pengujian SI Penjualan Berbasis *Cloud*

Requirement yang diuji	Fungsi yang diuji
Pendaftaran Toko	Mendaftarkan toko
	Mengirim ulang <i>email</i> aktivasi
	Mengaktivasi toko
Login	Melakukan login
	Menjalankan fitur "Lupa Password"
Data Barang	Melakukan operasi CRU (<i>Create, Read, Update</i>) pada data barang
	Menjalankan fitur "Promosi di Facebook"
	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data kategori barang
	Mengkategorikan barang
Data Karyawan	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data karyawan beserta hak aksesnya
Data Pelanggan	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data pelanggan
Data Supplier	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data <i>supplier</i>

Data Penjualan	Mencatat transaksi penjualan
	Mencatat retur penjualan
	Mencatat pembayaran kredit
Data Penerimaan	Mencatat transaksi penerimaan
Laporan	Menampilkan laporan stok terkini
	Menampilkan laporan tren penjualan
	Menampilkan laporan <i>history</i> pelanggan
	Menampilkan laporan <i>history supplier</i>
Ubah Layanan	Memasukkan perubahan layanan
	Mengkonfirmasi <i>transfer</i>

A. Desain Uji Coba Pendaftaran Toko

Desain uji coba pendaftaran toko bertujuan untuk menguji apakah fungsi mendaftarkan toko, mengirim ulang *email* aktivasi dan mengaktifkan toko dapat berjalan sesuai yang diharapkan. Desain uji coba pendaftaran toko dapat dilihat pada tabel 3.15.

Tabel 3. 15 Desain Uji Coba Pendaftaran Toko

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
1	Mendaftarkan toko	<i>ID</i> toko, nama toko, Nama toko, Password, Konfirmasi password, Email	Pendaftaran berhasil	
2	Mengirim ulang <i>email</i> aktivasi ke <i>email</i> pendaftar	<i>ID</i> toko	Email aktivasi telah dikirim ulang ke <i>email</i> pendaftar.	
3	Mengaktifkan toko	<i>ID</i> toko, kode aktivasi	Akun toko telah diaktifkan dan dapat mengakses halaman <i>login</i> .	

B Desain Uji Coba *Login*

Desain uji coba *login* bertujuan untuk menguji apakah fungsi melakukan *login* dan menjalankan fitur "Lupa Password" dapat berjalan sesuai yang diharapkan. Desain uji coba *login* dapat dilihat pada tabel 3.16.

Tabel 3. 16 Desain Uji Coba *Login*

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
4	Melakukan <i>login</i>	<i>ID toko, username, password</i>	<i>Login berhasil dan masuk ke halaman utama sistem</i>	
5	Melakukan <i>login</i> dengan <i>format</i> salah	<i>ID toko, username, password</i>	Muncul pesan " <i>Username atau password salah</i> "	
6	Menjalankan fitur " <i>Lupa Password</i> "	<i>ID toko, username</i>	Muncul pesan bahwa <i>password</i> telah dikirim melalui <i>email</i>	

C. Desain Uji Coba Data Barang

Desain uji coba data barang bertujuan untuk menguji apakah fungsi melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data barang, menjalankan fitur "Promosi di Facebook", melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data kategori barang dan mengkategorikan barang dapat berjalan sesuai yang diharapkan. Desain uji coba data barang dapat dilihat pada tabel 3.17.

Tabel 3. 17 Desain Uji Coba Data Barang

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
7	Menampilkan data barang	Memilih menu " <i>Barang</i> "	Data barang	

8	Memasukkan data barang	ID barang, nama barang, jenis, stok terkini, stok siap jual, stok minimum, harga beli, harga jual	Data barang berhasil disimpan dan ditampilkan pada tabel barang	
9	Mengubah data barang	ID barang, nama barang, jenis, stok terkini, stok siap jual, stok minimum, harga beli, harga jual	Data barang berhasil diubah dan ditampilkan pada tabel barang	
10	Menjalankan fitur "Promosi di Facebook"	ID atau nama barang	Nama barang, harga jual dan stok terkini berhasil dipromokan melalui status Facebook	
11	Menampilkan data kategori barang	Memilih menu "Kategori Barang"	Data kategori barang	
12	Memasukkan data kategori barang	Nama kategori	Data kategori barang berhasil disimpan dan ditampilkan pada tabel barang	
13	Mengubah data kategori barang	Nama kategori	Data kategori barang berhasil diubah dan ditampilkan pada tabel barang	
14	Menghapus data kategori barang	Nama kategori	Data kategori barang berhasil dihapus	
15	Mengkategorikan barang	ID atau nama barang, nama kategori	Data barang telah dimasukkan ke dalam kategori	

D. Desain Uji Coba Data Karyawan

Desain uji coba data karyawan bertujuan untuk menguji apakah fungsi melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data karyawan

dan hak aksesnya dapat berjalan sesuai yang diharapkan. Desain uji coba data karyawan dapat dilihat pada tabel 3.18.

Tabel 3. 18 Desain Uji Coba Data Karyawan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
16	Menampilkan data karyawan	Memilih menu "Karyawan"	Data karyawan	
17	Memasukkan data karyawan	<i>ID</i> karyawan, nama, <i>email</i> , telepon, hak akses	Data karyawan berhasil disimpan dan ditampilkan pada tabel karyawan	
18	Mengubah data karyawan	<i>ID</i> karyawan, nama, <i>email</i> , telepon, hak akses	Data karyawan berhasil diubah dan ditampilkan pada tabel karyawan	
19	Menghapus data karyawan	<i>ID</i> atau nama karyawan	Data karyawan berhasil dihapus	

E. Desain Uji Coba Data Pelanggan

Desain uji coba data pelanggan bertujuan untuk menguji apakah fungsi melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data pelanggan dapat berjalan sesuai yang diharapkan. Desain uji coba data pelanggan dapat dilihat pada tabel 3.19.

Tabel 3. 19 Desain Uji Coba Data Pelanggan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
20	Menampilkan data pelanggan	Memilih menu "Pelanggan"	Data pelanggan	
21	Memasukkan data pelanggan	<i>ID</i> pelanggan, nama, alamat, <i>email</i> , telepon, tanggal lahir	Data pelanggan berhasil disimpan dan ditampilkan pada tabel pelanggan	

22	Mengubah data pelanggan	ID pelanggan, nama, alamat, <i>email</i> , telepon, tanggal lahir	Data pelanggan berhasil diubah dan ditampilkan pada tabel pelanggan	
23	Menghapus data pelanggan	ID atau nama pelanggan	Data pelanggan berhasil dihapus	

F. Desain Uji Coba Data *Supplier*

Desain uji coba data *supplier* bertujuan untuk menguji apakah fungsi melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data *supplier* dapat berjalan sesuai yang diharapkan. Desain uji coba data *supplier* dapat dilihat pada tabel 3.20.

Tabel 3. 20 Desain Uji Coba Data *Supplier*

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
24	Menampilkan data <i>supplier</i>	Memilih menu " <i>Supplier</i> "	Data <i>supplier</i>	
25	Memasukkan data <i>supplier</i>	Nama <i>supplier</i> , alamat, <i>email</i> , telepon	Data <i>supplier</i> berhasil disimpan dan ditampilkan pada tabel <i>supplier</i>	
26	Mengubah data <i>supplier</i>	Nama <i>supplier</i> , alamat, <i>email</i> , telepon	Data <i>supplier</i> berhasil diubah dan ditampilkan pada tabel <i>supplier</i>	
27	Menghapus data <i>supplier</i>	Nama <i>supplier</i>	Data <i>supplier</i> berhasil dihapus	

G. Desain Uji Coba Data Penjualan

Desain uji coba data penjualan bertujuan untuk menguji apakah fungsi mencatat transaksi penjualan, mencatat retur penjualan dan mencatat pembayaran

kredit dapat berjalan sesuai yang diharapkan. Desain uji coba data penjualan dapat dilihat pada tabel 3.21.

Tabel 3. 21 Desain Uji Coba Data Penjualan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
28	Mencatat transaksi penjualan	<i>ID</i> atau nama barang, jumlah, diskon satuan, nama pelanggan, diskon keseluruhan, uang yang dibayarkan	Total harga, pesan transaksi penjualan berhasil disimpan dan nota penjualan	
29	Mencatat retur penjualan	<i>ID</i> atau nama barang, jumlah retur dan alasan.	Pesan retur penjualan berhasil disimpan dan nota retur penjualan	
30	Mencatat pembayaran kredit	Jumlah uang yang dibayarkan pelanggan	Pesan pembayaran kredit telah disimpan dan nota pembayaran kredit	

H. Desain Uji Coba Data Penerimaan

Desain uji coba data penerimaan bertujuan untuk menguji apakah fungsi mencatat transaksi penerimaan dan mencatat retur pembelian dapat berjalan sesuai yang diharapkan. Desain uji coba data penerimaan dapat dilihat pada tabel 3.22.

Tabel 3. 22 Desain Uji Coba Data Penerimaan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
31	Mencatat transaksi penerimaan	<i>ID</i> atau nama barang, jumlah, nama <i>supplier</i> , uang yang dibayarkan	Total harga dan pesan transaksi penerimaan berhasil disimpan	

I. Desain Uji Coba Laporan

Desain uji coba laporan bertujuan untuk menguji apakah fungsi menampilkan laporan stok terkini, menampilkan laporan tren penjualan, menampilkan laporan *history* pelanggan, menampilkan laporan *history supplier* dapat berjalan sesuai yang diharapkan. Desain uji coba data penerimaan dapat dilihat pada tabel 3.23.

Tabel 3. 23 Desain Uji Coba Laporan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
32	Menampilkan laporan stok terkini	Memilih menu "Laporan Stok Terkini"	Tampilan laporan stok terkini dalam bentuk grafis atau tabular	
33	Menampilkan laporan tren penjualan	Tampilan laporan, nama barang, periode laporan	Tampilan laporan tren penjualan dalam bentuk grafis atau tabular	
34	Menampilkan laporan <i>history</i> pelanggan	Tampilan laporan, nama pelanggan, periode laporan	Tampilan laporan <i>history</i> pelanggan dalam bentuk grafis atau tabular	
35	Menampilkan laporan <i>history supplier</i>	Tampilan laporan, nama <i>supplier</i> , periode laporan	Tampilan laporan <i>history supplier</i> dalam bentuk grafis atau tabular	

J. Desain Uji Coba Ubah Layanan

Desain uji coba ubah layanan bertujuan untuk menguji apakah fungsi memasukkan perubahan layanan dan mengkonfirmasi *transfer* dapat berjalan sesuai yang diharapkan. Desain uji coba data penerimaan dapat dilihat pada tabel 3.24.

Tabel 3. 24 Desain Uji Coba Ubah Layanan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
36	Memasukkan perubahan layanan	Paket layanan, jumlah bulan paket layanan, batas data penjualan, jumlah bulan batas data penjualan	Pesan permintaan perubahan layanan berhasil disimpan dan nota tagihan	
37	Mengkonfirmasi <i>transfer</i>	Nama bank, nomer rekening, jumlah pembayaran, tanggal <i>transfer</i> , kode tagihan	Pesan konfirmasi <i>transfer</i> berhasil disimpan	

K. Desain Uji Coba *Cloud Performance*

Uji coba *cloud performance* digunakan untuk mengetahui *performance* layanan *cloud* yang digunakan, yaitu *CPU Percentage & Network Out, Database Monitor, CPU, Memory* dan *Disk*. Pengujian ini dilakukan untuk membuktikan bahwa sistem yang dibangun memiliki *performance* yang baik dalam penggunaan normal. Pada saat pengujian, sistem akan digunakan oleh 2 orang bagian penjualan untuk mencatat masing-masing 10 transaksi penjualan, 1 orang bagian pengelola stok persediaan untuk mencatat 10 transaksi penerimaan barang dan 1 orang pihak manajemen yang melihat laporan penjualan pada saat transaksi berlangsung. Semua transaksi dilakukan secara bersamaan. Desain uji coba *cloud performance* dapat dilihat pada tabel 3.25.

Tabel 3. 25 Desain Uji Coba *Cloud Performance*

No	Pengujian	Status	Hasil
1	<i>CPU Percentage & Network Out</i> melalui <i>dashboard Windows Azure</i>		

2	<i>DB Monitor</i> melalui <i>dashboard Windows Azure</i>		
3	<i>CPU</i> melalui <i>website Site24x7</i> dari Soho		
4	<i>Memory</i> melalui <i>website Site24x7</i> dari Soho		
5	<i>Disk</i> melalui <i>website Site24x7</i> dari Soho		

L. Desain Uji Coba Proses Bisnis pada Sistem

Pengujian yang penting dalam tahap implementasi adalah uji coba penggunaan sistem dalam sebuah *business cycle* pada tempat studi kasus. Pengujian dilakukan untuk memastikan sistem yang dibangun tidak menghasilkan *error* ataupun kesalahan perhitungan saat digunakan dalam proses bisnis.

Pengujian dilakukan dengan membuat sebuah kasus transaksi. Kasus transaksi dimulai dari penentuan HPP, penerimaan barang dan penjualan barang. Semua perhitungan, mulai dari stok yang berkurang, HPP yang diberikan dan harga jual yang diberikan harus sesuai. Desain uji coba kesesuaian hasil akhir sistem dapat dilihat pada tabel 3.26.

Tabel 3. 26 Desain Uji Coba Proses Bisnis

No.	Proses Bisnis	Input	Output yang diharapkan	Status
1	Menentukan HPP	Metode HPP yang digunakan.	Pengaturan HPP telah disimpan.	
2	Mencatat transaksi penerimaan barang	<i>ID</i> atau nama barang, jumlah, harga dan nama <i>supplier</i> .	Nota penerimaan barang. Stok barang bertambah, harga beli tersimpan dengan benar sesuai yang diinputkan.	

3	Mencatat transaksi penjualan	ID atau nama barang jumlah, harga, diskon dan nama pelanggan.	Nota penjualan. Stok barang berkurang, harga beli dan jual tersimpan sesuai metode yang dipilih.	
4	Menampilkan informasi barang terlaris (<i>fast moving</i>).	Kategori barang, periode transaksi, jumlah data, pengurutan (<i>sort</i>)	Informasi barang terlaris	
5	Menampilkan informasi barang tidak laris (<i>slow moving</i>).	Kategori barang, periode transaksi, jumlah data, pengurutan (<i>sort</i>)	Informasi barang tidak laris	
6	Menampilkan informasi <i>supplier</i> bagi barang tertentu	Nama barang, periode transaksi	Informasi <i>supplier</i> , barang yang di- <i>supply</i> beserta rata-rata harga	

M. Desain Evaluasi Kesesuaian Hasil Akhir Sistem

Evaluasi kesesuaian hasil akhir sistem adalah sebuah pengujian untuk mengetahui apakah sistem yang dibangun telah memenuhi kebutuhan-kebutuhan yang ada di latar belakang permasalahan. Sistem yang dibangun juga harus mengatasi permasalahan-permasalahan sistem penjualan yang telah digunakan sebelumnya. Desain uji coba kesesuaian hasil akhir sistem dapat dilihat pada tabel 3.27.

Tabel 3. 27 Desain Evaluasi Hasil Akhir Sistem

	Kebutuhan	Hasil	Bukti
1	Informasi <i>up to date</i> mengenai tren penjualan, stok terkini serta <i>history</i> transaksi dan <i>supplier</i> yang diperoleh tanpa melakukan rekap		
2	Sistem yang <i>portable</i> , dapat diakses kapanpun dan dimanapun dengan koneksi internet melalui <i>web browser</i> , termasuk <i>mobile browser</i> .		
3	Biaya <i>maintenance</i> , perangkat teknologi serta SDM pengelola sistem yang murah.		
4	Jaminan keamanan data dan ketersediaan sistem		

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Kebutuhan Sistem

Sistem informasi penjualan berbasis *cloud* membutuhkan *server* dan perangkat lunak dengan kondisi tertentu agar dapat berjalan dengan baik. Kebutuhan *server* dan perangkat lunak dibagi menjadi dua, yaitu kebutuhan di *server* yang merupakan tanggung jawab penyedia layanan sistem informasi penjualan berbasis *cloud* dan kebutuhan oleh pengguna sistem.

4.1.1 Kebutuhan Server

Sistem informasi penjualan ini diletakkan di *server* milik penyedia layanan infrastruktur *cloud*, yaitu Microsoft (Windows Azure). Karena menggunakan infrastruktur *cloud*, penggunaan perangkat keras *server* di *data center* tidak dapat diketahui dan diidentifikasi dengan jelas karena merupakan tanggung jawab dari penyedia layanan infrastruktur *cloud*. Hal yang dapat diketahui dengan pasti adalah spesifikasi *server* yang dapat dikonfigurasi sesuai dengan kebutuhan. Spesifikasi *server* yang dibutuhkan adalah:

- a. *Virtual Machine* Windows Azure (*Shared core*)
- b. 768 MB *memory*

Untuk menggunakan sistem informasi penjualan berbasis *cloud*, toko tidak perlu menyediakan *server* melainkan hanya membutuhkan perangkat akses sebagai berikut:

- a. Komputer, *laptop*, *tablet* atau *smartphone*.

- b. *Barcode scanner* dan *printer* jika diperlukan.
- c. Koneksi internet.

4.1.2 Kebutuhan Perangkat Lunak

Untuk mengakses sistem informasi penjualan berbasis *cloud* ini, pengguna tidak perlu memenuhi kebutuhan perangkat lunak *server*, melainkan hanya kebutuhan perangkat lunak *client*. Kebutuhan perangkat lunak *server* adalah:

- a. Microsoft Windows Server 2012
- b. Azure SQL Database Business Edition

Kebutuhan perangkat lunak *client* untuk mengakses sistem informasi penjualan berbasis *cloud* yang adalah:

- a. *A-Grade Browser*

4.2 Pembuatan Program

Sistem informasi penjualan berbasis *cloud* ini dibangun dengan menggunakan bahasa pemrograman ASP.NET MVC 3 dan *database* SQL Server 2012. Pembangunan program juga dibantu oleh *library-library* javascript seperti Kendo UI Web, JQuery dan RGraph. *Library-library* ini memiliki fungsi yang khusus. Kendo UI Web digunakan untuk tampilan kontrol beserta operasinya ke pengguna, seperti *grid*, *auto-complete*, *datetime picker* dan lainnya. JQuery digunakan untuk mempermudah operasi-operasi javascript serta komunikasi ke *server* melalui teknologi *asynchronous javascript*. RGraph digunakan untuk menampilkan laporan grafis dalam bentuk *chart*. *Pattern* yang digunakan dalam penulisan kode program adalah MVC.

IDE (*Integrated Development Environment*) yang digunakan untuk menulis kode program dan mengatur susunan *file* adalah Visual Studio 2012. Semua *file* program akan di-*build* menjadi *file* .bin oleh Visual Studio. *File* .bin ini beserta *file* Javascript, CSS dan *View* (.cshtml) akan disimpan di *server*. *File database* akan dirancang dan dibuat di *editor* SQL Server 2012 Management Studio. *Editor* ini juga akan membantu tahap *deploy database* ke Azure SQL Database secara *online*.

4.3 Implementasi Sistem

Setelah sistem informasi penjualan selesai dibuat, sistem akan di-*deploy* pada *server* yang terletak di *cloud*. *Provider cloud* yang digunakan adalah Windows Azure. Informasi mengenai Windows Azure dapat dilihat di windowsazure.com. Dalam pembahasan implementasi sistem di *cloud* ini, pengguna yang dimaksud adalah penyedia layanan sistem informasi penjualan berbasis *cloud* yang akan men-*deploy* sistem ke Windows Azure. Untuk menggunakan Windows Azure, pengguna harus memiliki Microsoft *account* yang pendaftarannya dapat dilakukan di windowsazure.com.

4.3.1 Registrasi Akun Windows Azure

Untuk memiliki Microsoft *account*, pengguna perlu mendaftar di windowsazure.com dan mengisi data seperti pada gambar 4.1. Semua data harus diisi dengan benar karena akun ini digunakan dalam proses pembayaran biaya sewa layanan infrastruktur *cloud* serta untuk *login* dan konfigurasi *server*.

Microsoft account

Already have a Microsoft account? If you use **Hotmail**, **SkyDrive**, **Xbox LIVE** and want to claim a new email address, [sign in](#), and then [rename your account](#) or [create an alias](#).

Who are you?

Name

Birth date

Gender

How would you like to sign in?

Microsoft account name
 @

[Or use an email address you already have](#)

Create a password

8-character minimum; case sensitive

Reenter password

Phone number

Alternate email address

[Or choose a security question](#)

Where are you from?

Country/region

Postal Code

Help us make sure you're not a robot
 Enter the characters you see
[New](#) | [Audio](#)

Send me email with promotional offers from Microsoft. (You can unsubscribe at any time.)

Click **I accept** to agree to the [Microsoft services agreement](#) and [privacy & cookies statement](#).

I accept

Gambar 4. 1 Registrasi Microsoft Account

Setelah memiliki Microsoft *account*, langkah selanjutnya adalah mendaftarkan *account* tersebut pada Windows Azure. Data pertama yang diminta adalah nomer telepon. Pengguna dapat memilih untuk menerima SMS atau telepon dari Microsoft untuk melakukan verifikasi. Setelah itu, pengguna diminta memasukkan kode yang diterima. Tampilan registrasi nomer telepon adalah seperti pada gambar 4.2.

Gambar 4. 2 Registrasi Windows Azure - Nomer Telepon

Setelah verifikasi nomer telepon selesai, langkah selanjutnya adalah memberikan informasi *credit card* yang digunakan sebagai media pembayaran tagihan. Data yang perlu diisi dengan lengkap adalah seperti pada gambar 4.3.

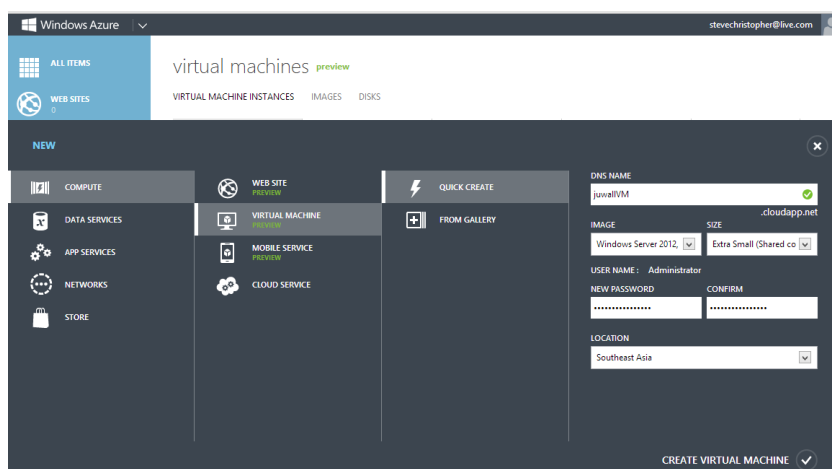
Gambar 4. 3 Registrasi Windows Azure - *Credit Card*

Setelah verifikasi *credit card* berhasil, pengguna sudah dapat melakukan konfigurasi layanan *cloud* yang diperlukan untuk sistem informasi penjualan ini.

Sistem informasi penjualan berbasis *cloud* akan menggunakan dua jenis layanan *cloud*, yaitu *Virtual Machine* dan *Azure SQL Database*. *Virtual Machine* digunakan sebagai *application server* dan *Azure SQL Database* digunakan sebagai *database server*. Pemisahan *server* aplikasi dan *database* ditujukan agar pemrosesan yang terjadi di aplikasi dan *database* tidak saling mengganggu sehingga meringankan beban kerja masing-masing *server*. Infrastruktur sistem yang diimplementasikan di *cloud* adalah seperti pada gambar 3.9 di bab 3.

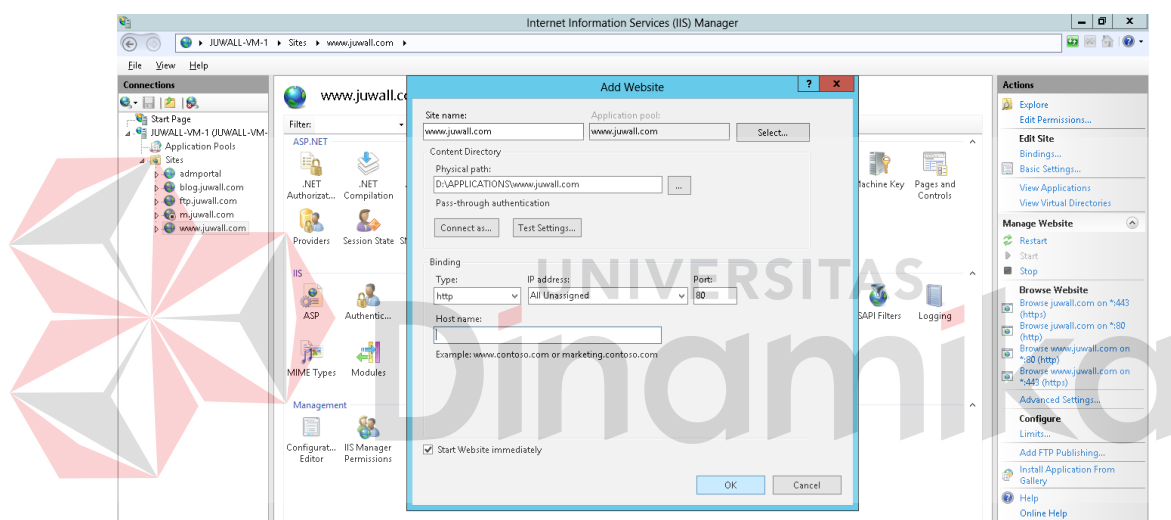
4.3.2 Konfigurasi *Virtual Machine*

Untuk membuat *Virtual Machine* pada Windows Azure, pengguna perlu mengakses manage.azure.com dan *login* menggunakan Microsoft *account* yang telah didaftarkan. Pada *dashboard* yang disediakan, pengguna membuat *Virtual Machine* baru dengan spesifikasi *operating system* Windows Server 2012 dan *size* *Extra Small (Shared core, 768 MB Memory)*. Lalu pengguna memasukkan DNS dan *password* yang digunakan untuk mengakses *Virtual Machine*. *Location* dari *server* yang dipilih adalah *Southeast Asia*. Tampilan pembuatan *Virtual Machine* baru adalah seperti pada gambar 4.4.



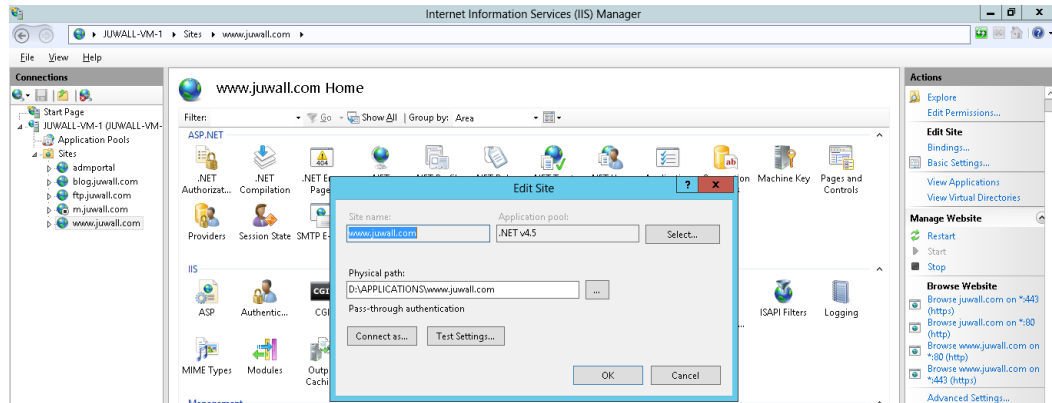
Gambar 4. 4 Pembuatan *Virtual Machine*

Setelah *Virtual Machine* dibuat, langkah selanjutnya adalah masuk ke *Virtual Machine* tersebut menggunakan *software* Remote Desktop Connection dan mengkonfigurasi *application server*. Setelah masuk ke *Virtual Machine*, *copy* data *web* application yang diperlukan, yaitu *config file*, HTML, CSS, Javascript, *View* dan *bin folder*. Setelah itu buka Internet Information Service dan buat *folder website* baru dengan konfigurasi seperti pada gambar 4.5. Lokasi *Physical Path* diarahkan pada lokasi *file web application* yang telah di-*copy* ke *server*.



Gambar 4. 5 Pembuatan *folder website* pada Internet Information Service

Setelah *folder website* berhasil dibuat di Internet Information Service, langkah selanjutnya adalah mengubah *Application Pool* untuk disesuaikan dengan *framework* yang digunakan, yaitu .NET Framework 4.5. Konfigurasi ini dapat dilakukan dengan memilih *Basic Settings* dan akan muncul *window* konfigurasi seperti pada gambar 4.6.

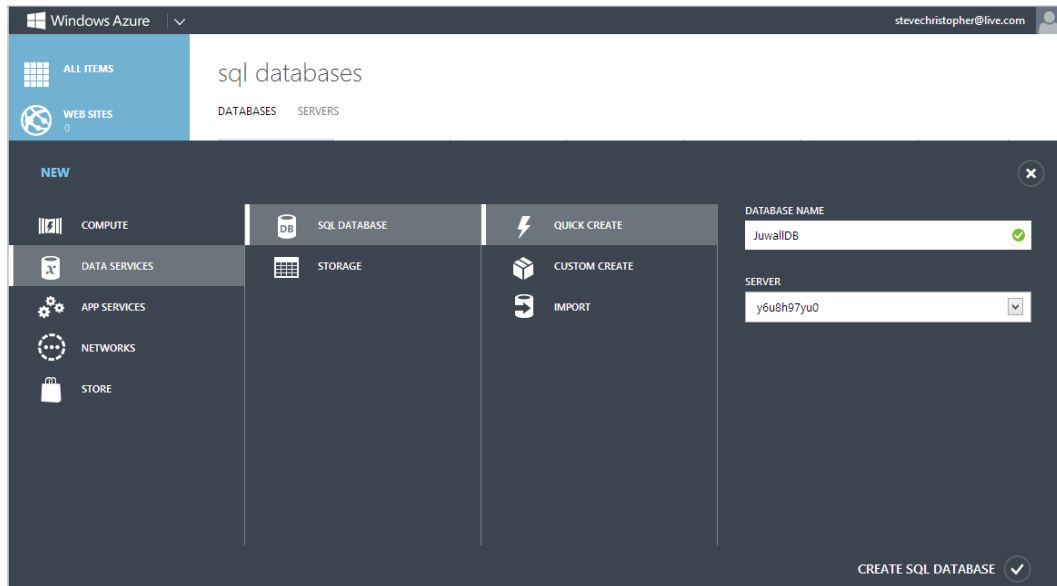


Gambar 4. 6 Konfigurasi *Application Pool*

Setelah konfigurasi selesai, langkah selanjutnya adalah melakukan konfigurasi *database server* dengan menggunakan *Azure SQL Database*.

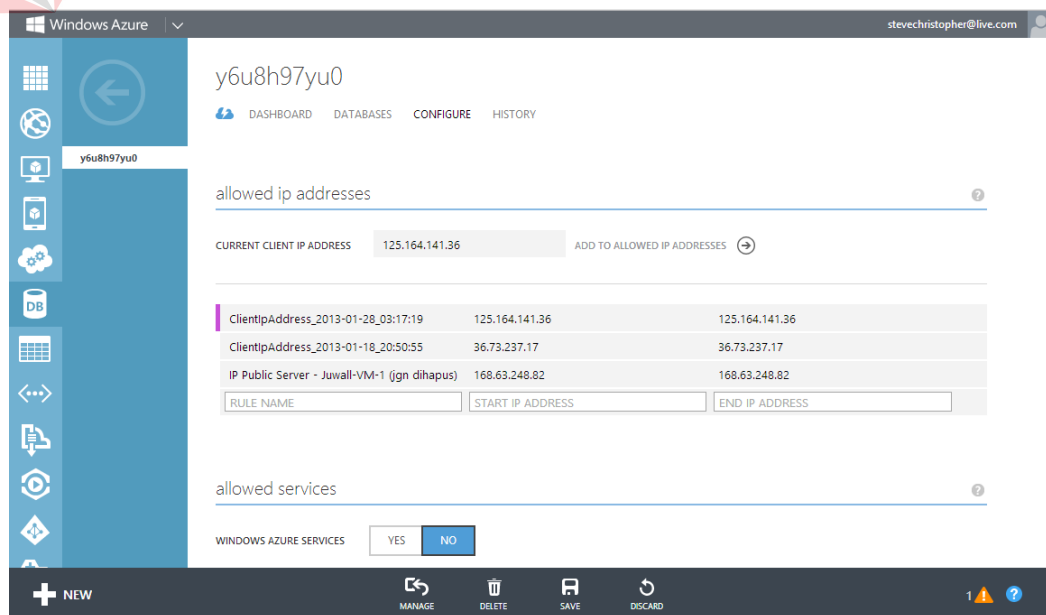
4.3.3 Konfigurasi *Azure SQL Database*

Untuk membuat *Azure SQL Database*, pengguna perlu mengakses manage.azure.com dan *login* menggunakan *Microsoft account* yang telah didaftarkan. Pada *dashboard* yang disediakan, pengguna membuat *Azure SQL Database* baru dengan mengisi nama *database* dan memilih *server* yang digunakan seperti pada gambar 4.7. Nama *database* dan *server* digunakan saat *men-deploy* dan melakukan *remote connection* terhadap *database* dari *SQL Server Management Studio*. Konfigurasi ukuran *database* beserta edisi yang digunakan akan dilakukan pada saat *deploy* melalui *SQL Server Management Studio*.



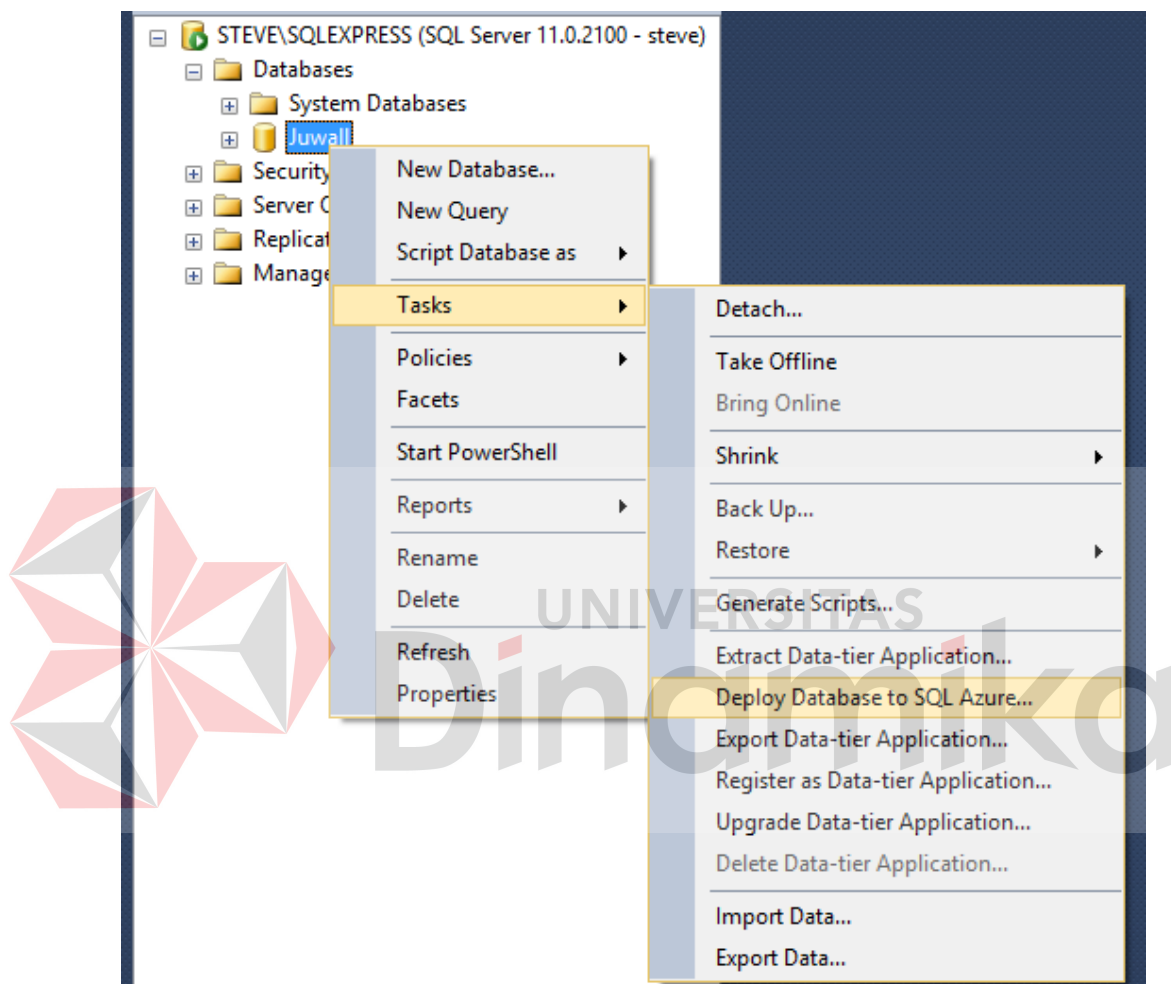
Gambar 4. 7 Pembuatan Azure SQL Database

Setelah selesai dibuat, akses komputer pengguna ke *database* harus diijinkan agar proses *deploy* dapat berjalan. Untuk melakukannya, pilih Azure SQL Database yang telah dibuat, pilih menu *Manage allowed IP Addresses* dan pilih *Add to followed IP Addresses* seperti pada gambar 4.8.



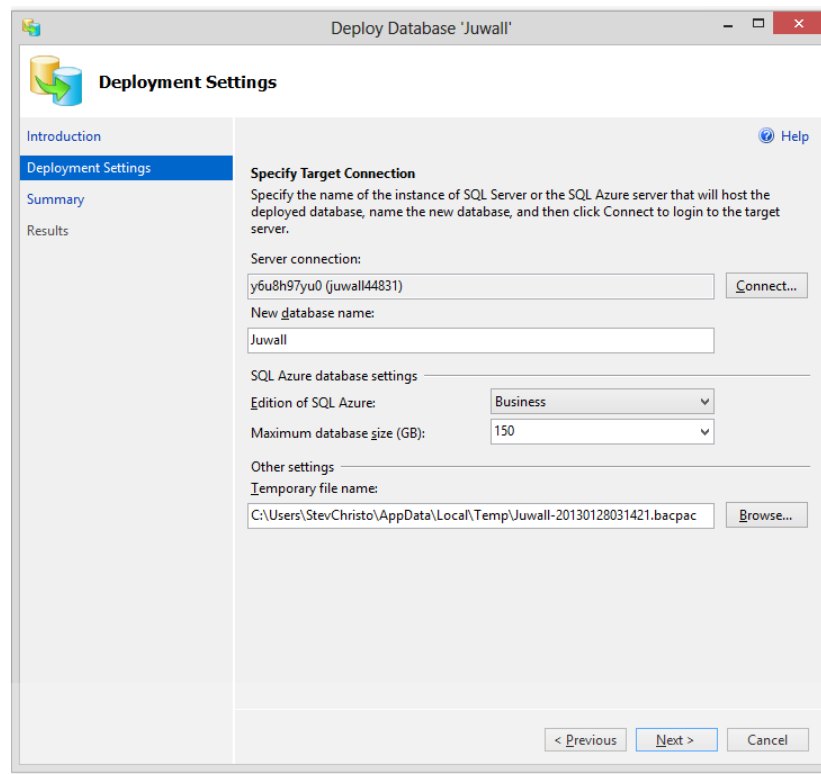
Gambar 4. 8 Konfigurasi IP Address

Tahap selanjutnya adalah *deploy database*. Buka SQL Server Management Studio, pilih *database* dari sistem informasi penjualan lalu pilih menu *Task* dan *Deploy Database to SQL Azure* seperti pada gambar 4.9.



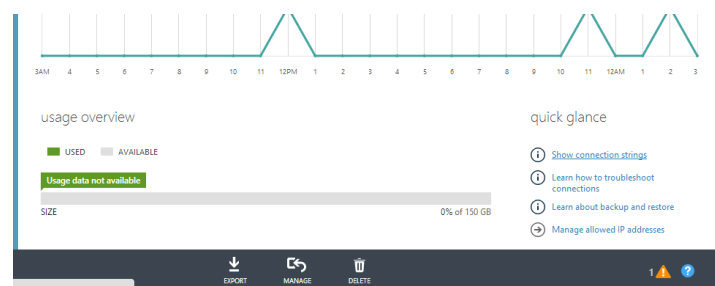
Gambar 4. 9 Menu *Deploy Database* ke Azure SQL Database

Setelah itu, isi *form* konfigurasi *database* seperti pada gambar 4.10. *Server Connection* diisi dengan melakukan koneksi ke Azure SQL Database dengan menggunakan nama *server* yang didaftarkan pada saat pembuatan. Nama *database* harus diisi dan edisi yang digunakan adalah *Business* dengan kapasitas *maximum* 150 GB. Setelah itu pilih *Next* dan sistem akan men-*deploy database* ke Azure SQL Database yang telah dibuat.



Gambar 4. 10 Deploy Database ke Azure SQL Database

Setelah proses *deploy database* berhasil, pengguna perlu mengubah *connection string* pada *config file* yang tersimpan di *application server* sesuai dengan *database* yang telah ter-*deploy* di Azure SQL Database. *Connection string* dapat dilihat pada *dashboard* Windows Azure dengan memilih Azure SQL Database lalu *Show connection strings* seperti pada gambar 4.11. Setelah itu, sistem informasi penjualan berbasis *cloud* sudah dapat diakses.



Gambar 4. 11 Konfigurasi *Connection String*

4.4 Proses Bisnis

4.4.1 Registrasi Pengguna

Untuk menggunakan sistem informasi penjualan berbasis *cloud*, pengguna perlu melakukan proses registrasi toko dengan mengakses halaman utama seperti pada gambar 4.12 dan memilih menu daftar.



Gambar 4. 12 Halaman Awal

Setelah memilih menu daftar sistem akan menampilkan *form* registrasi toko seperti pada gambar 4.13. Pengguna harus mengisi *id* toko, nama toko, *password*, konfirmasi *password* dan *email*. Pengguna juga harus menyetujui *terms of service*. Setelah pendaftaran selesai, pesan sukses seperti pada gambar 4.14 akan muncul dan sistem akan mengirimkan *email* aktivasi akun ke alamat *email* yang didaftarkan.

Terms of service

- Juwall dapat memblokir akun pengguna apabila ditemukan indikasi kecurangan atau penyalahgunaan layanan Juwall, misalkan percobaan berinteraksi dengan sistem tanpa melalui user interface yang disediakan oleh Juwall.
- Pengguna bertanggung jawab secara pribadi terhadap konten ataupun data yang diunggah atau disimpan di Juwall. Juwall tidak dapat dituntut untuk segala pelanggaran yang ditemukan dalam data ataupun konten yang diunggah atau disimpan di Juwall oleh pengguna.
- Juwall tidak dapat dituntut untuk kelalaian pengguna dalam pengolahan data, sehingga terjadi kesalahan pada data pengguna.
- Juwall akan menjaga ketersediaan sistem sebaik mungkin. Namun Juwall tidak dapat dituntut untuk kejadian-kejadian *force majeure* yang mengakibatkan matinya sistem ataupun rusaknya data, misalkan bencana alam, peperangan, kerusakan dan sejenisnya.

Dengan ini saya menyetujui semua kondisi diatas sebelum melakukan pendaftaran

Daftar baru

ID toko: tokobesar

Nama toko: Toko Besar

Password:

Konfirmasi password:

Email: tokobesar@live.com

Daftar

* ID Toko merupakan identitas toko yang dapat digunakan sebagai alamat layanan Juwall milik toko anda. (eg : juwall.com/pos/IDToko)

* Setelah terdaftar, toko anda secara otomatis akan menggunakan layanan Gratis. Anda dapat mengubah menjadi layanan Super atau Premium setelah masuk ke sistem.

Copyright ©2013, Juwall

Gambar 4. 13 Halaman Registrasi Akun

Pendaftaran berhasil !

Terima kasih toko **tokobesar** telah bergabung dengan Juwall.

Kami telah mengirimkan link untuk mengaktifkan akun anda ke email **stevchristo@gmail.com**.

Silahkan cek inbox email anda dan aktifkan akun anda agar dapat masuk ke halaman anda.

Apabila anda belum menerima email aktivasi dari kami, periksa folder Spam anda. Jika tidak ada email dari kami, silahkan klik link dibawah ini untuk mengulang pengiriman email.

[Kirim ulang email!](#)

Setelah melakukan aktivasi melalui email, anda dapat membuka toko anda melalui link www.juwall.com/pos/tokobesar atau dengan mengetik ID toko pada halaman utama www.juwall.com.

Gambar 4. 14 Halaman Registrasi Akun Telah Berhasil Dilakukan

Email aktivasi akun yang dikirimkan sistem setelah proses registrasi toko berisi *link* yang perlu dibuka oleh pengguna untuk mengaktifkan akun seperti pada gambar 4.15. Setelah *link* aktivasi akun dibuka, sistem akan mengaktifkan akun dan mengarahkan pengguna ke halaman *login*.



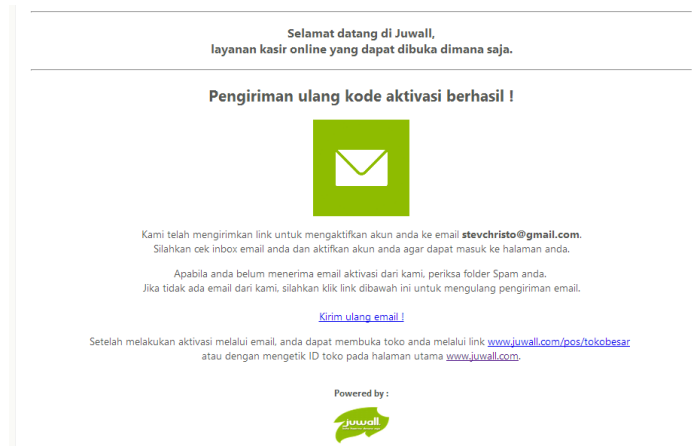
Gambar 4. 15 *Email* Aktivasi Akun

Jika pengguna mengakses sistem dengan membuka halaman utama dan memasukkan *id* toko yang belum diaktivasi, maka akan muncul halaman seperti pada gambar 4.16 yang memberi informasi bahwa aktivasi akun belum dilakukan.



Gambar 4. 16 Halaman Akun Belum Diaktivasi

Pengguna dapat melakukan pengiriman ulang *email* aktivasi ke alamat *email* yang didaftarkan melalui *link* pengiriman ulang. Setelah *email* berhasil dikirim ulang, maka muncul halaman seperti gambar 4.17 dan *email* telah berhasil dikirimkan ke alamat *email* yang didaftarkan.

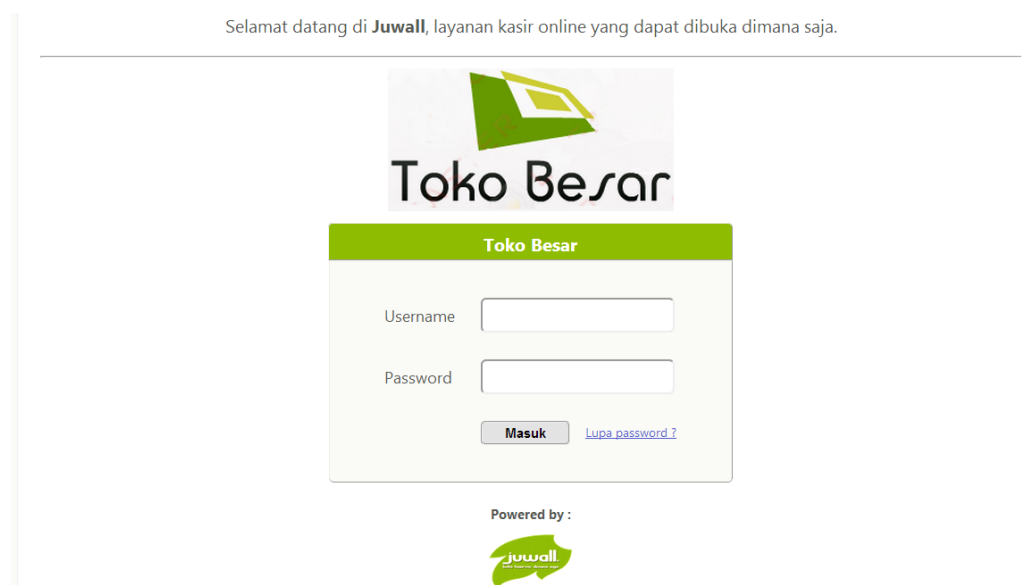


Gambar 4. 17 Halaman Pengiriman Ulang *Email* Aktivasi Berhasil

4.4.2 Login

Proses *login* dilakukan pengguna sebelum masuk ke dalam sistem.

Pengguna perlu memasukkan *id* toko pada halaman utama dan sistem akan menampilkan halaman *login* toko seperti pada gambar 4.18. Pada halaman ini, pengguna dapat melakukan *login* dengan mengisi *username* dan *password* untuk diverifikasi oleh sistem. Jika pengguna baru pertama kali mengakses sistem, *username* yang digunakan sama dengan *id* toko.



Gambar 4. 18 Halaman *Login*

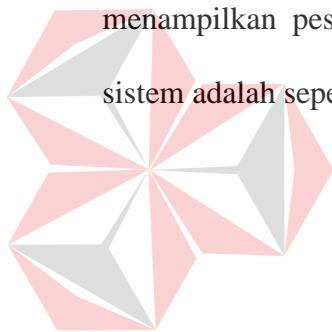
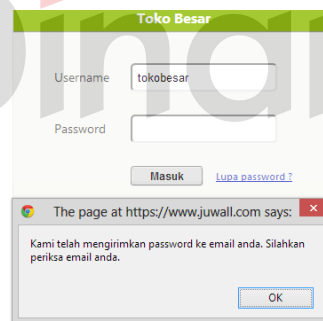
Jika pengguna mengisi *username* atau *password* yang salah, maka akan muncul pesan *error* seperti pada gambar 4.19.



The screenshot shows a login form for 'Toko Besar'. The 'Username' field contains '12345' and the 'Password' field contains '.....'. Below the fields are buttons for 'Masuk' and 'Lupa password?'. A red error message banner at the bottom reads 'Username atau password salah!'.

Gambar 4. 19 Pesan Kesalahan *Username* atau *Password*

Jika lupa *password*, pengguna perlu mengisi *id* toko dan memilih *link* lupa password. Setelah selesai, sistem akan mengirimkan *password* melalui *email* dan menampilkan pesan sukses seperti pada gambar 4.20. *Email* yang dikirimkan sistem adalah seperti pada gambar 4.21.

The screenshot shows the same login form as in Gambar 4.19, but with 'tokobesar' entered in the 'Username' field. A dialog box is overlaid on the form, displaying the message: 'The page at https://www.juwall.com says: Kami telah mengirimkan password ke email anda. Silahkan periksa email anda.' with an 'OK' button.

Gambar 4. 20 Pengiriman *Password* ke *Email* Berhasil



Gambar 4. 21 *Email* Informasi *Password*

4.4.3 Maintenance Data Karyawan Beserta Hak Akses

Pihak manajemen dapat membuka halaman karyawan untuk menambah, mengubah, menghapus dan melihat data karyawan yang memiliki hak akses ke dalam sistem. Halaman karyawan adalah seperti pada gambar 4.22.

The screenshot displays the 'Karyawan' management interface. At the top, there is a navigation menu with icons for 'Penjualan', 'Penerimaan', 'Pelanggan', 'Karyawan', 'Barang', 'Supplier', 'Laporan', and 'Setting'. Below this, the 'Karyawan' section contains a table with the following data:

ID Karyawan	Nama	Email	Telepon	Ubah	Delete
dwicahyo	Dwi Cahyo	dwicahyo91@gmail.com	08273622232	Ubah	Delete
riyadi	Riyadi	riyadi1982@yahoo.com	081263728381	Ubah	Delete
tokobesar	tokobesar	tokobesar@live.com		Ubah	Delete

Additional elements include a 'Save changes' button, a 'Cancel changes' button, and a sidebar with a search box labeled 'Cari karyawan' and buttons for 'Karyawan baru' and 'Email ke karyawan'. The page footer shows 'UNIVERSITAS Dinamika' and '1 - 3 of 3 items'.

Gambar 4. 22 Halaman Karyawan

Untuk menambah data karyawan baru, pilih menu karyawan baru. Setelah itu, muncul *form* karyawan baru yang perlu diisi. Data yang diisi adalah *id* karyawan, nama, *email*, telepon, *password* dan hak akses. Hak akses yang dapat diberikan adalah kasir penjualan, kasir penerimaan barang, history penjualan, *history* penerimaan barang, karyawan, barang, pelanggan, *supplier*, laporan dan *setting* kasir. Tampilan *form* karyawan baru adalah seperti pada gambar 4.23.

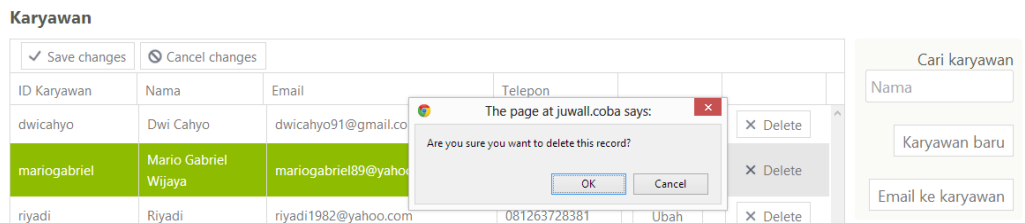
Gambar 4. 23 *Form* Karyawan Baru

Untuk mengubah data karyawan, pilih data karyawan yang diinginkan lalu pilih ubah. Setelah itu akan muncul *form* ubah data karyawan seperti pada gambar 4.24. Data yang diisi sama seperti *form* karyawan baru.

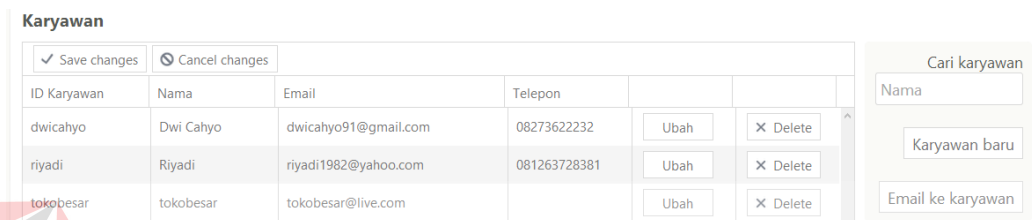
Gambar 4. 24 *Form* Ubah Karyawan

Untuk menghapus data karyawan, pihak manajemen memilih data karyawan yang ingin dihapus dan memilih menu *Delete*. Setelah itu akan muncul *dialog* konfirmasi penghapusan dan pilih *OK*. Setelah selesai, pilih *Save changes* dan data akan dihapus. Tampilan penghapusan data karyawan adalah seperti pada

gambar 4.25. Setelah dihapus, data akan hilang dari tabel seperti pada gambar 4.26.



Gambar 4. 25 Penghapusan Data Karyawan



Gambar 4. 26 Penghapusan Data Karyawan Berhasil

4.4.4 Maintenance Data Barang

Bagian pengelola stok persediaan dapat membuka halaman barang untuk menambah, mengubah, menghapus dan melihat data barang. Halaman barang adalah seperti pada gambar 4.27.

Barang

ID Barang	Nama	Stok terkini	Siap jual	Stok Minim	Harga Beli	Harga Jual	Profit (%)	Diskon (%)
coolpad001	Cooling Pad Targus	34	34	1	150000	225000	50	0
fd004	Flash Disk Kingston 16GB	14	14	20	100000	150000	50	0
99-81201-002	ARGOX Barcode Scanner	4	4	10	600000	900000	50	5
763649023341	Ext HD Seagate GoFlex 500 GB	3	3	5	485000	727500	50	0
4894299006...	Capdase Lumia 710	0	0	2	0	0	50	0
coolpad002	Cooling Pad	0	0	2	0	0	50	0

1 - 10 of 32 items

Gambar 4. 27 Halaman Barang

Untuk memasukkan data barang, pengelola stok persediaan memilih *Add new record* pada tabel barang dan mengisikan data barang baru. Setelah selesai, pilih *Save changes* dan data barang baru akan tersimpan. Tampilan pengisian data barang baru adalah seperti pada gambar 4.28 dan ketika sukses disimpan seperti pada gambar 4.29.

Barang

ID Barang	Nama	Jenis	Stok terkini	Stok siap jual	Stok Minim	Harga Beli	Harga Jual	Diskon	
fd003	Flash Disk Kingston 16GB	Barang	20	20	20	100000	135000	5000	X Delete
coolpad...	Cooling Pad	Barang	1	1	1	150000	185000	0	X Delete

Gambar 4. 28 Pengisian Data Barang

Barang

ID Barang	Nama	Jenis	Stok terkini	Stok siap jual	Stok Minim	Harga Beli	Harga Jual	Diskon	
fd003	Flash Disk Kingston 16GB	Barang	20	20	20	100000	135000	5000	X Delete
coolpad...	Cooling Pad	Barang	1	1	1	150000	185000	0	X Delete

Gambar 4. 29 Pengisian Data Barang Berhasil

Untuk mengubah data barang, bagian pengelola stok persediaan memilih data barang yang ingin diubah dan mengisikan data baru. Setelah selesai, pilih *Save changes* dan data akan disimpan. Tampilan perubahan data barang adalah seperti pada gambar 4.30 dan ketika sukses disimpan seperti pada gambar 4.31.

Barang

+ Add new record ✓ Save changes ⌛ Cancel changes

ID Barang	Nama	Jenis	Stok terkini	Stok siap jual	Stok Minim	Harga Beli	Harga Jual	Diskon	
fd003	Flash Disk Kingston 16GB	Barang	20	20	20	100000	135000	10000	✕ Delete

Cari barang
Nama
Atur Kategori

Gambar 4. 30 Perubahan Data Barang

Barang

+ Add new record ✓ Save changes ⌛ Cancel changes

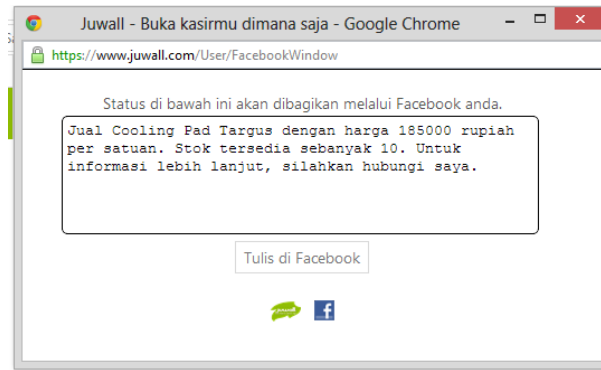
ID Barang	Nama	Jenis	Stok terkini	Stok siap jual	Stok Minim	Harga Beli	Harga Jual	Diskon	
fd003	Flash Disk Kingston 16GB	Barang	20	20	20	100000	135000	10000	✕ Delete

Cari barang
Nama
Atur Kategori

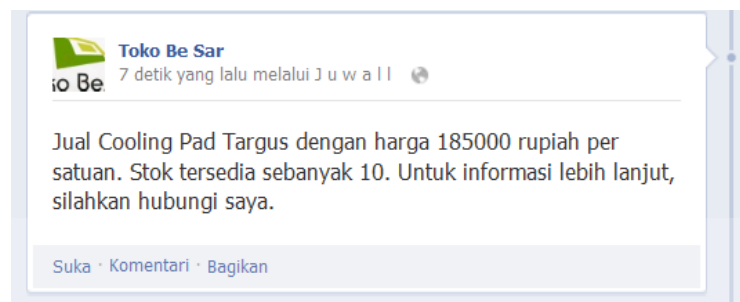
Gambar 4. 31 Perubahan Data Barang Berhasil

4.4.5 Menjalankan Fitur Promosi Di Facebook

Untuk mempromosikan barang di Facebook, pengguna memilih data barang yang diinginkan lalu memilih menu promo di Facebook. Setelah itu akan muncul *form* penulisan status Facebook seperti pada gambar 4.32 dan pengguna dapat memberi tambahan kata sebelum menulisnya di Facebook. Setelah ditulis di Facebook, promosi akan tertulis sebagai status pengguna seperti pada gambar 4.33.



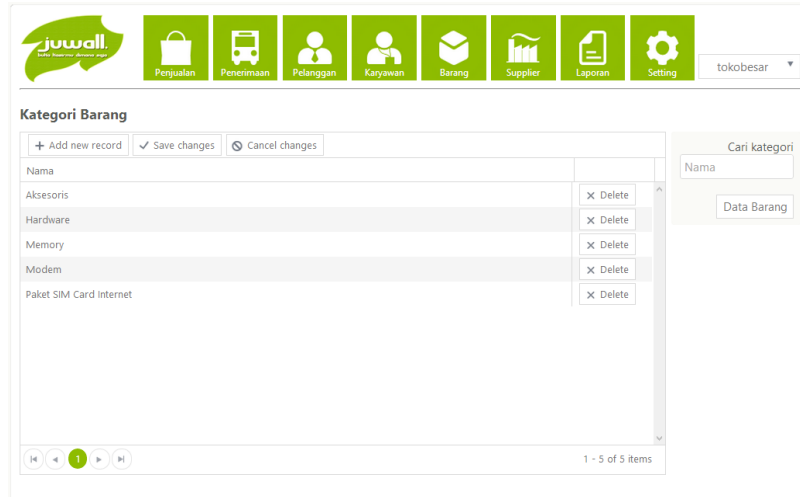
Gambar 4. 32 Form Promosi Barang Di Facebook



Gambar 4. 33 Tampilan Promosi Di Facebook

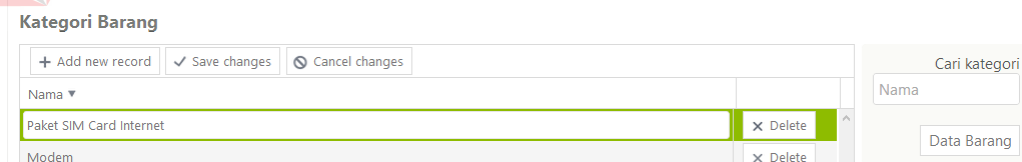
4.4.6 Maintenance Data Kategori Barang

Bagian pengelola stok persediaan dapat membuka halaman kategori barang untuk menambah, mengubah, menghapus dan melihat data kategori barang. Untuk menampilkan halaman barang, bagian pengelola stok persediaan memilih menu data kategori pada halaman barang. Halaman kategori barang adalah seperti pada gambar 4.34.



Gambar 4. 34 Halaman Kategori Barang

Untuk memasukkan data kategori barang, bagian pengelola stok persediaan memilih *Add new record* pada tabel barang dan mengisi nama kategori. Setelah selesai, pilih *Save changes* dan data kategori barang baru akan tersimpan. Tampilan pengisian data kategori barang baru adalah seperti pada gambar 4.35 dan ketika sukses disimpan seperti pada gambar 4.36.



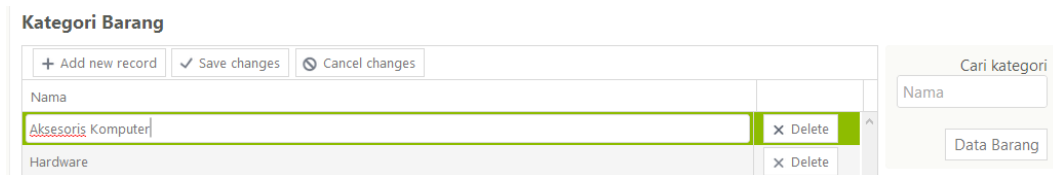
Gambar 4. 35 Pengisian Data Kategori Barang



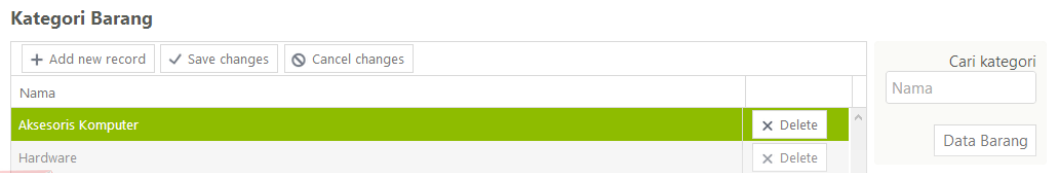
Gambar 4. 36 Pengisian Data Kategori Barang Berhasil

Untuk mengubah data kategori barang, bagian pengelola stok persediaan memilih data kategori yang ingin diubah dan mengisi data baru. Setelah

selesai, pilih *Save changes* dan data akan disimpan. Tampilan perubahan data kategori barang adalah seperti pada gambar 4.37 dan ketika sukses disimpan seperti pada gambar 4.38.

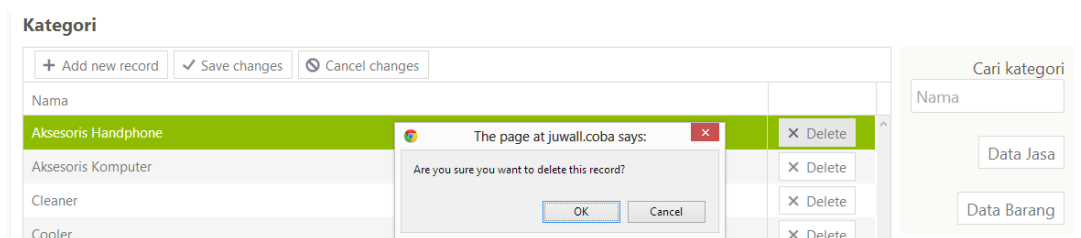


Gambar 4. 37 Perubahan Data Kategori Barang

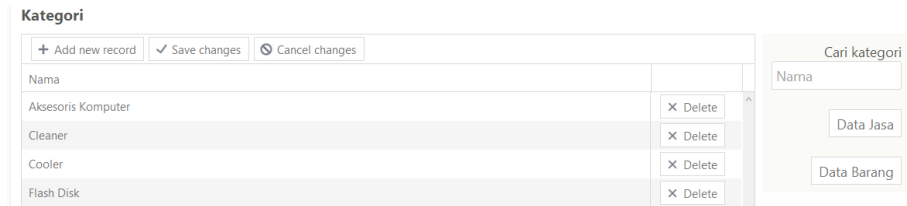


Gambar 4. 38 Perubahan Data Kategori Barang Berhasil

Untuk menghapus data kategori barang, bagian pengelola stok persediaan memilih data kategori barang yang ingin dihapus dan memilih menu *Delete*. Setelah itu akan muncul *dialog* konfirmasi penghapusan dan pilih *OK*. Setelah selesai, pilih *Save changes* dan data akan dihapus. Tampilan penghapusan data kategori barang adalah seperti pada gambar 4.39. Setelah dihapus, data akan hilang dari tabel seperti pada gambar 4.40.



Gambar 4. 39 Penghapusan Data Kategori Barang



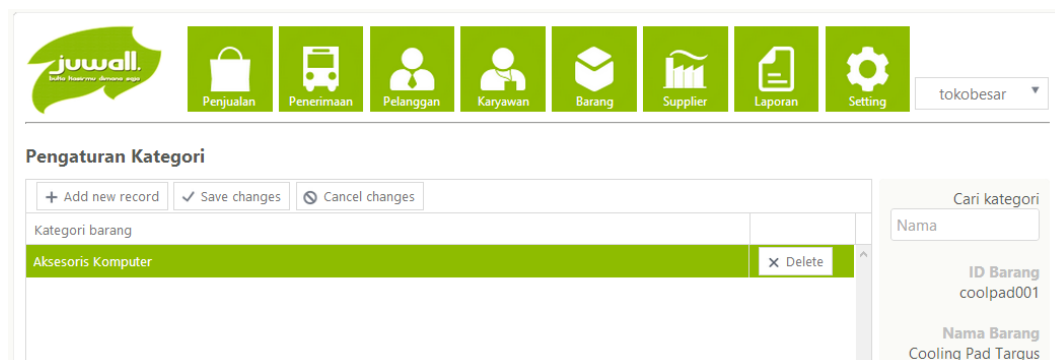
Gambar 4. 40 Penghapusan Data Kategori Barang Berhasil

4.4.7 Mengkategorikan Barang

Untuk mengkategorikan barang, bagian pengelola stok persediaan memilih data barang yang diinginkan lalu memilih menu atur kategori. Setelah itu akan muncul tabel pengaturan kategori. Pengguna memilih *Add new record* lalu memasukkan nama kategori dan pilih *Save changes* untuk menyimpan pengaturan kategori. Tampilan pengaturan kategori barang adalah seperti gambar 4.41 dan ketika sukses disimpan seperti pada gambar 4.42.



Gambar 4. 41 Pengkategorian Barang



Gambar 4. 42 Pengkategorian Barang Berhasil

4.4.8 Maintenance Data Pelanggan

Bagian penjualan dapat membuka halaman pelanggan untuk menambah, mengubah, menghapus dan melihat data pelanggan. Halaman pelanggan adalah seperti pada gambar 4.43.

ID Pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	
cus0003	Budi	Pondok Tjandra Indah - CA-491		085739210293	24-July-1979	X Delete
cus0004	CV Megah Jaya			031-5039281	07-March-1984	X Delete
cus0005	Diana Ramadhan	Puri Indah U-8 Sidoarjo		08121728274	31-May-1990	X Delete
cus0002	Mutiara Anjani				08-August-1992	X Delete
cus0001	Ravee Agastya	Jl. Sulawesi 17 Surabaya	ravagastya84@...	031-5031066	06-June-1984	X Delete

Gambar 4. 43 Halaman Data Pelanggan

Untuk memasukkan data pelanggan, bagian penjualan memilih *Add new record* pada tabel pelanggan dan mengisikan data pelanggan baru. Setelah selesai, pilih *Save changes* dan data pelanggan baru akan tersimpan. Tampilan pengisian data pelanggan baru adalah seperti pada gambar 4.44 dan ketika sukses disimpan seperti pada gambar 4.45.

ID Pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	
cus0006	Dian Herlinda S	Puri Indah Permai J3-21	dianherlindas@yah	08193827381	20-March-1992	X Delete

Gambar 4. 44 Pengisian Data Pelanggan

Pelanggan

+ Add new record ✓ Save changes ⌂ Cancel changes

ID Pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	
cus0006	Dian Herlinda S	Puri Indah Permai J3-21	dianherlindas@...	08193827381	20-March-1992	X Delete

Cari pelanggan

Nama

Email ke pelanggan

Gambar 4. 45 Pengisian Data Pelanggan Berhasil

Untuk mengubah data pelanggan, bagian penjualan memilih data pelanggan yang ingin diubah dan mengisikan data baru. Setelah selesai, pilih *Save changes* dan data akan disimpan. Tampilan pengubahan data pelanggan adalah seperti pada gambar 4.46 dan ketika sukses disimpan seperti pada gambar 4.47.

Pelanggan

+ Add new record ✓ Save changes ⌂ Cancel changes

ID Pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	
cus0006	Dian Herlinda Sari	Puri Indah Permai J3-28	dianherlindas@yah	08193827381	20-March-1992	X Delete

Cari pelanggan

Nama

Email ke pelanggan

Gambar 4. 46 Pengubahan Data Pelanggan

Pelanggan

+ Add new record ✓ Save changes ⌂ Cancel changes

ID Pelanggan	Nama	Alamat	Email	Telepon	Tanggal lahir	
cus0006	Dian Herlinda Sari	Puri Indah Permai J3-28	dianherlindas@...	08193827381	20-March-1992	X Delete

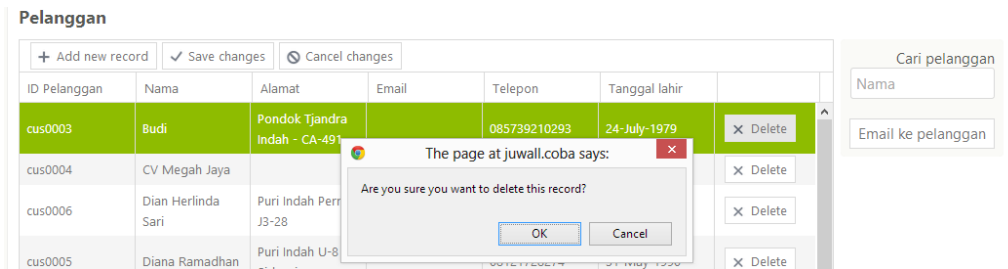
Cari pelanggan

Nama

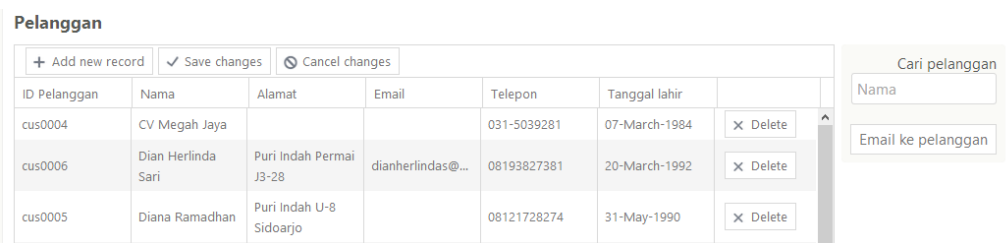
Email ke pelanggan

Gambar 4. 47 Pengubahan Data Pelanggan Berhasil

Untuk menghapus data pelanggan, bagian penjualan memilih data pelanggan yang ingin dihapus dan memilih menu *Delete*. Setelah itu akan muncul *dialog* konfirmasi penghapusan dan pilih *OK*. Setelah selesai, pilih *Save changes* dan data akan dihapus. Tampilan penghapusan data pelanggan adalah seperti pada gambar 4.48. Setelah dihapus, data akan hilang dari tabel seperti pada gambar 4.49.



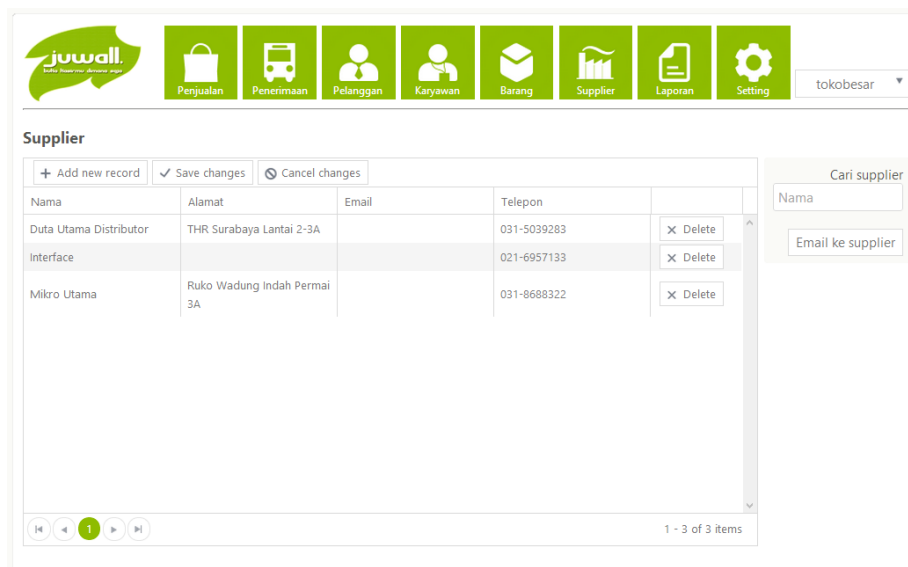
Gambar 4. 48 Penghapusan Data Pelanggan



Gambar 4. 49 Penghapusan Data Pelanggan Berhasil

4.4.9 Maintenance Data Supplier

Bagian pengelola stok persediaan dapat membuka halaman *supplier* untuk menambah, mengubah, menghapus dan melihat data *supplier*. Halaman *supplier* adalah seperti pada gambar 4.50.



Gambar 4. 50 Halaman *Supplier*

Untuk memasukkan data *supplier*, bagian pengelola stok persediaan memilih *Add new record* pada tabel *supplier* dan mengisikan data *supplier* baru. Setelah selesai, pilih *Save changes* dan data *supplier* baru akan tersimpan. Tampilan pengisian data *supplier* baru adalah seperti pada gambar 4.51 dan ketika sukses disimpan seperti pada gambar 4.52.

The screenshot shows a web interface for managing suppliers. At the top, there are three buttons: '+ Add new record', '✓ Save changes', and '⊗ Cancel changes'. Below these is a table with columns: Nama, Alamat, Email, and Telepon. The first row is highlighted in green and contains 'Mitra Jaya', 'Ruko Klampis 8-A Surabaya', and '031-509287'. The second row contains 'Duta Utama Distributor', 'THR Surabaya Lantai 2-3A', and '031-5039283'. To the right of the table is a search section titled 'Cari supplier' with a 'Nama' input field and an 'Email ke supplier' button.

Gambar 4. 51 Pengisian Data *Supplier*

This screenshot is similar to the previous one, but the 'Email' field for the 'Mitra Jaya' entry has been updated to 'info@mitrajaya.com'. The 'Save changes' button is now active, indicating the data has been successfully updated.

Gambar 4. 52 Pengisian Data *Supplier* Berhasil

Untuk mengubah data *supplier*, bagian pengelola stok persediaan memilih data *supplier* yang ingin diubah dan mengisikan data baru. Setelah selesai, pilih *Save changes* dan data akan disimpan. Tampilan pengubahan data *supplier* adalah seperti pada gambar 4.53 dan ketika sukses disimpan seperti pada gambar 4.54.

This screenshot is identical to the previous one, showing the successful update of the 'Mitra Jaya' supplier's email address to 'info@mitrajaya.com'.

Gambar 4. 53 Pengubahan Data *Supplier*

Supplier

+ Add new record ✓ Save changes ⌛ Cancel changes				
Nama	Alamat	Email	Telepon	
Mitra Jaya	Ruko Klampis 8-A Surabaya	info@mitrajaya.com	031-509288	X Delete
Duta Utama Distributor	THR Surabaya Lantai 2-3A		031-5039283	X Delete

Cari supplier

Nama

Email ke supplier

Gambar 4. 54 Pengubahan Data *Supplier* Berhasil

Untuk menghapus data *supplier*, bagian pengelola stok persediaan memilih data *supplier* yang ingin dihapus dan memilih menu *Delete*. Setelah itu akan muncul *dialog* konfirmasi penghapusan dan pilih *OK*. Setelah selesai, pilih *Save changes* dan data akan dihapus. Tampilan penghapusan data *supplier* adalah seperti pada gambar 4.55. Setelah dihapus, data akan hilang dari tabel seperti pada gambar 4.56.

Supplier

+ Add new record ✓ Save changes ⌛ Cancel changes				
Nama	Alamat	Email	Telepon	
Alamraya Electronics	HiTech Mall Lantai 1A no. 38 Jl. Kusuma Bangsa 116-118 Surabaya		(031) 5476582	X Delete
Asiaraya Computronics	Mangga 2 Mall Lantai 4 Blok A-62 Jakarta			X Delete
Digital One				X Delete
Duta Utama Distributor	THR Surabaya Lantai 2-3A			X Delete

Cari supplier

Nama

Email ke supplier

The page at juwall.coba says:

Are you sure you want to delete this record?

OK Cancel

Gambar 4. 55 Penghapusan Data *Supplier*

Supplier

+ Add new record ✓ Save changes ⌛ Cancel changes				
Nama	Alamat	Email	Telepon	
Asiaraya Computronics	Mangga 2 Mall Lantai 4 Blok A-62 Jakarta		(021) 6019408	X Delete
Digital One			+62 21 62302406	X Delete
Duta Utama Distributor	THR Surabaya Lantai 2-3A		031-5039283	X Delete

Cari supplier

Nama

Email ke supplier

Gambar 4. 56 Penghapusan Data *Supplier* Berhasil

4.4.10 Mencatat Transaksi Penjualan

Untuk melakukan pencatatan transaksi penjualan, bagian penjualan memilih menu penjualan pada menu utama. Penjualan dapat dilakukan dalam 2 cara, yaitu

tunai dan kredit. Tampilan pencatatan penjualan tunai adalah seperti pada gambar 4.57. Bagian penjualan mengisi barang yang dibeli pelanggan, jumlah, diskon dan jumlah uang yang dibayarkan oleh pelanggan. Setelah itu sistem akan menghitung jumlah kembalian dan menyimpan transaksi penjualan. Setelah transaksi tunai disimpan, maka akan muncul nota penjualan tunai seperti pada gambar 4.58.

The screenshot shows the 'Penjualan' (Sales) module. At the top, there are navigation icons for Penjualan, Penerimaan, Pelanggan, Karyawan, Barang, Supplier, Laporan, and Setting. The main area is titled 'Penjualan' and includes a search bar for 'Nama Barang' and a dropdown for 'Nama pelanggan (optional)'. Below this is a table of items:

ID Barang	Nama	Jumlah	Harga	Diskon	Total	
fd003	Flash Disk Kingston 16GB	1	135000	10000	125000	X Delete
modem001	Modem Vodafone Rev A	1	200000	0	200000	X Delete
coolpad001	Cooling Pad Targus	1	185000	0	185000	X Delete

On the right, a summary panel shows:

- Total harga: 510,000
- Diskon: 10,000
- Total bayar: 500,000
- Uang: 500,000
- Kembalian: 0
- Buttons: cetak nota, Simpan

Gambar 4. 57 Pencatatan Transaksi Penjualan Tunai

The receipt is titled 'Nota penjualan - Toko Besar' and includes the following information:

- Tanggal: 29 January 2013
- No. Transaksi: 10
- Nama pelanggan: Mutiara Anjani

Nama barang	Jumlah	Harga satuan	Diskon	Total harga
Flash Disk Kingston 16GB	1	135,000	10,000	125,000
Modem Vodafone Rev A	1	200,000		200,000
Cooling Pad Targus	1	185,000		185,000

Summary of the receipt:

- Diskon penjualan: 10,000
- Total harga: 500,000

Buttons at the bottom: Cetak, Keluar

Gambar 4. 58 Nota Penjualan Tunai

Jika penjualan dilakukan secara kredit, maka muncul form ketentuan kredit yang harus diisi terlebih dahulu. Tampilan *form* penjualan kredit adalah seperti pada gambar 4.59. Bagian penjualan mengisi diskon, uang muka, jumlah angsuran, periode pembayaran, persentase bunga, persentase denda dan tanggal awal periode. Setelah transaksi kredit disimpan, maka akan muncul nota penjualan kredit seperti pada gambar 4.60.



The image shows a screenshot of a web application form titled "Transaksi kredit". The form contains the following fields and values:

Field	Value
Total harga	875,000
Diskon	0
Total bayar	875,000
Uang muka	375,000
Total hutang	500,000
Jumlah angsuran	5
Periode pembayaran	1 bulanan
Bunga (%)	10
Denda Tiap Periode (%)	10
Awal periode	29/01/2013

At the bottom of the form, there is a checkbox labeled "cetak nota" which is checked, and a "Simpan" button.

Gambar 4. 59 *Form* Penjualan Kredit

Nota penjualan kredit

Nota penjualan kredit - Toko Besar



Tanggal : 29 January 2013

No. Transaksi : 11
Nama pelanggan : Mutiara Anjani

Nama barang	Jumlah	Harga satuan	Diskon	Total harga
Flash Disk Kingston 8GB	5	80,000	5,000	375,000
Flash Disk Kingston 16GB	4	135,000	10,000	500,000

Diskon penjualan : Sisa hutang : 500,000 Bunga (%) : 10 %
 Total harga : 875,000 Angsuran : 5 Denda tiap periode (%) : 10 %
 Uang muka : 375,000 Periode pembayaran : 1 bulanan Awal periode pembayaran : 29 January 2013

Tanda terima,

Toko Besar

juwall.com - layanan kasir online

Gambar 4. 60 Nota Penjualan Kredit

4.4.11 Mencatat Retur Penjualan

Untuk mencatat retur penjualan, bagian penjualan masuk ke bagian *detail* barang di halaman *history* penjualan. Setelah itu, bagian penjualan memilih barang yang akan diretur dan memasukkan jumlah beserta alasannya. Tampilan *form* mencatat retur penjualan adalah seperti pada gambar 4.61. Setelah retur penjualan dicatat, sistem akan menampilkan nota retur penjualan seperti pada gambar 4.62.

penjualan - barang terjual

Nama barang	Harga
Cooling Pad Targus	185000
Flash Disk Kingston 16GB	135000
Modem Vodafone Rev A	200000

Retur penjualan

Nama barang Cooling Pad Targus

Jumlah retur 1

Alasan Kipas tidak dapat berputar

Simpan

Cari barang

Nama barang

Nama Barang

No. Transaksi 10

Barang retur

Histori penjualan

Gambar 4. 61 Form Pencatatan Retur Penjualan

Nota retur penjualan

Nota retur penjualan - Toko Besar


Toko Besar

Tanggal : 29 January 2013 No. Transaksi : 10
Nama pelanggan : Mutiara Anjani

Nama barang	Alasan	Jumlah	Harga satuan	Diskon	Total harga
Cooling Pad Targus	Kipas tidak dapat berputar	1	185,000	0	185,000

Tanda terima, Total harga : **185,000**

Toko Besar juwall.com - layanan kasir online

Gambar 4. 62 Nota Retur Penjualan

4.4.12 Mencatat Pembayaran Kredit

Untuk mencatat pembayaran kredit, bagian penjualan masuk ke halaman *history* penjualan dan memilih transaksi kredit yang akan dibayar. Setelah itu bagian penjualan memilih menu pembayaran kredit dan sistem akan menampilkan *form* pembayaran kredit seperti pada gambar 4.63. Setelah pembayaran disimpan, maka akan muncul nota pembayaran kredit seperti pada gambar 4.64.

Histori penjualan

No. Transaksi	Tanggal
11	29-Janua
5	26-Janua

Pembayaran kredit

Tanggal	29 January 2013	Hutang pokok	100,000
No. Transaksi	11	Bunga	10,000
Nama pelanggan	Mutiara Anjani	Denda	0
Angsuran	1	Total bayar	110,000
Jatuh tempo angsuran	28 February 2013	Uang	110,000
		Kembalian	0

cetak nota

Gambar 4. 63 Form Pencatatan Pembayaran Kredit

Nota pembayaran kredit

Nota penjualan kredit - Toko Besar




Toko Besar

Tanggal	29 January 2013	Hutang pokok	100,000
No. Transaksi	11	Bunga	10,000
Nama pelanggan	Mutiara Anjani	Denda	0
Angsuran	1	Total bayar	110,000
Tanggal jatuh tempo	28 February 2013	Tanggal jatuh tempo selanjutnya	28 February 2013
Sisa angsuran	4		

Gambar 4. 64 Nota Pembayaran Kredit

4.4.13 Mencatat Transaksi Penerimaan

Untuk mencatat transaksi penerimaan barang, bagian pengelola stok persediaan memilih menu penerimaan pada menu utama. Tampilan pencatatan penerimaan barang adalah seperti pada gambar 4.65. Setelah transaksi disimpan, sistem akan memunculkan nota penerimaan barang seperti pada gambar 4.66.



Penjualan
 Penerimaan
 Pelanggan
 Karyawan
 Barang
 Supplier
 Laporan
 Setting

tokobesar ▾

Penerimaan

Nama Barang ▾ : atau

Nama supplier (optional) : atau

ID Barang	Nama	Jumlah	Harga satuan	Total	
fd002	Flash Disk Kingston 8GB	5	65000	325000	<input type="button" value="X Delete"/>
fd003	Flash Disk Kingston 16GB	5	100000	500000	<input type="button" value="X Delete"/>

Total harga **825,000**

Uang

Kembalian **25,000**

stok barang siap jual

cetak nota

Gambar 4. 65 Pencatatan Transaksi Penerimaan

Nota penerimaan barang

Penerimaan Barang - Toko Besar


Toko Besar

Tanggal : 29 January 2013 No. Transaksi : 5
Nama supplier : Interface

ID. Barang	Nama barang	Jumlah	Harga satuan	Total harga
fd002	Flash Disk Kingston 8GB	5	65,000	325,000
fd003	Flash Disk Kingston 16GB	5	100,000	500,000
Tanda terima,				Total harga : 825,000

Toko Besar juwal.com - layanan kasir online

Gambar 4. 66 Nota Penerimaan Barang

4.4.14 Menampilkan Laporan Stok Terkini

Untuk menampilkan laporan stok terkini, pihak manajemen memilih menu laporan pada halaman utama dan memilih laporan stok terkini seperti pada gambar 4.67. Setelah itu pihak manajemen memilih tampilan laporan dan sistem akan menampilkan laporan stok terkini seperti pada gambar 4.68 untuk tabular dan gambar 4.69 untuk grafis.

Jenis laporan barang

Pilih jenis laporan barang yang anda inginkan

Deskripsi laporan

Laporan stok terkini akan memberi informasi stok barang yang tersedia di toko, stok siap jual serta stok minimum yang telah ditetapkan sehingga anda dapat mengetahui apakah saat ini toko anda sedang kekurangan stok barang tertentu.

Tampilan laporan

Parameter laporan

Pilih kategori barang.

NB : pastikan kategori barang terisi secara otomatis dari pilihan yang ada. Anda cukup mengetik huruf depan dari kategori barang dan kami akan mencari data kategori barang yang sesuai. Jika tidak diisi maka akan ditampilkan barang dari semua kategori.

Gambar 4. 67 Form Laporan Stok Terkini

Laporan stok terkini - Toko Besar

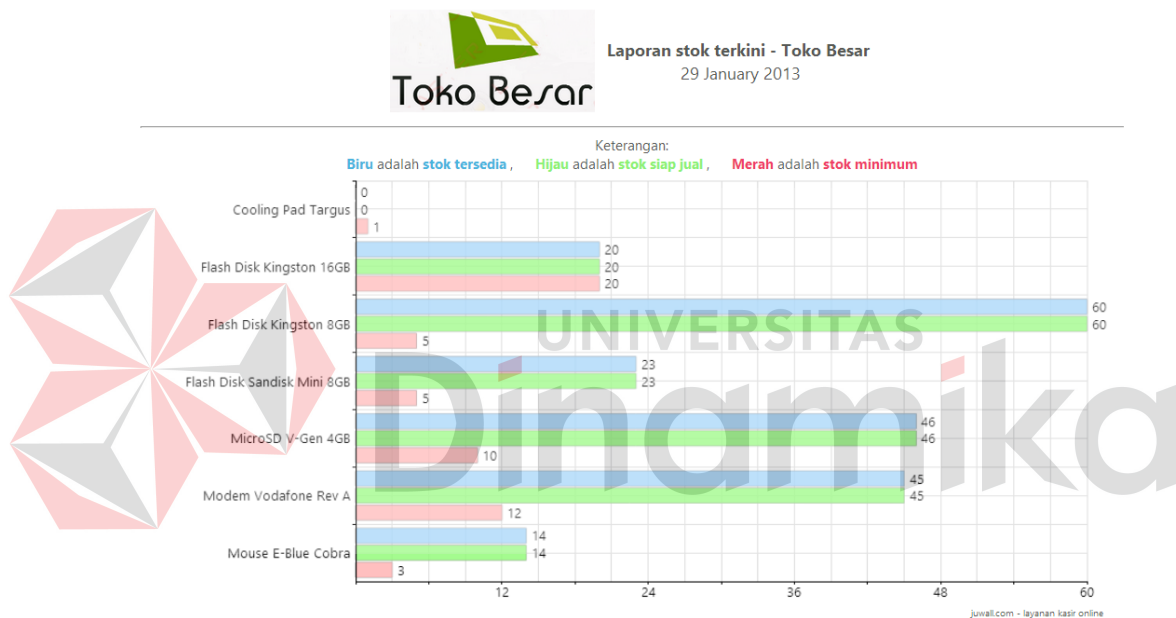


Tanggal : 11 March 2013

ID. barang	Nama	Stok tersedia	Stok siap jual	Stok minimum
89620951712000450203	Modem Smartfren EVDO Rev A	6	6	5
modem001	Modem Vodafone Rev A	3	3	12

juwall.com - layanan kasir online

Gambar 4. 68 Laporan Tabular Stok Terkini



Gambar 4. 69 Laporan Grafis Stok Terkini

4.4.15 Menampilkan Laporan Tren Penjualan

Untuk menampilkan laporan tren penjualan, pihak manajemen memilih menu laporan pada halaman utama dan memilih laporan tren penjualan seperti pada gambar 4.70. Setelah itu pilih periode transaksi, tampilan laporan, nama barang beserta tanggal awal dan akhir periode. Lalu sistem menampilkan laporan tren penjualan seperti pada gambar 4.71 untuk tabular dan gambar 4.72 untuk grafis.

Jenis laporan barang

Pilih jenis laporan barang yang anda inginkan

Tren penjualan barang

Deskripsi laporan

Laporan tren penjualan barang akan memberi informasi transaksi penjualan dan penerimaan yang dilakukan terhadap barang tertentu.

Periode transaksi

Harian

Tampilan laporan

Tabular

Parameter laporan

Pilih nama barang.

Flash Disk Sandisk Mini 8GE

Pilih periode barang.

29/01/2013

hingga

29/01/2013

Tampilkan

NB : pastikan nama barang terisi secara otomatis dari pilihan yang ada. Anda cukup mengetik huruf depan dari nama barang dan kami akan mencari data nama barang yang sesuai.

Gambar 4. 70 Form Laporan Tren Penjualan

Laporan tren penjualan - Toko Besar



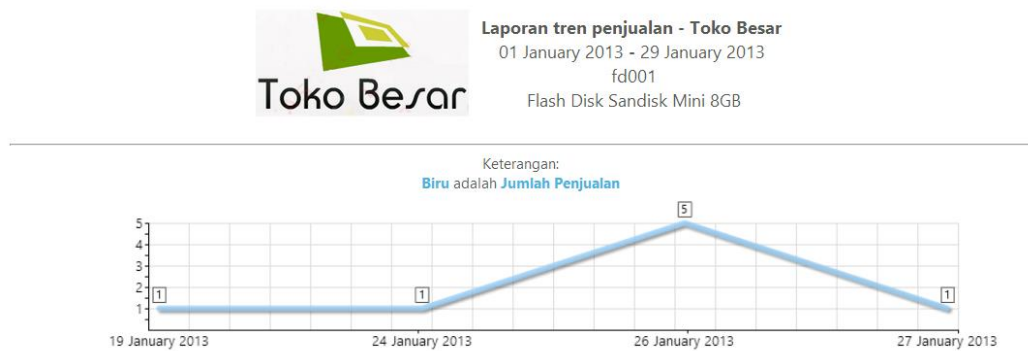
Toko Besar

Tanggal : 01 January 2013 - 29 January 2013 ID barang : fd001
Nama barang : Flash Disk Sandisk Mini 8GB

Periode	Jumlah penjualan
19 January 2013	1
24 January 2013	1
26 January 2013	5
27 January 2013	1
Jumlah	8

juwall.com - layanan kasir online

Gambar 4. 71 Laporan Tabular Tren Penjualan



Gambar 4. 72 Laporan Grafis Tren Penjualan

4.4.16 Menampilkan Laporan *History* Pelanggan

Untuk menampilkan laporan *history* pelanggan, pihak manajemen memilih menu laporan pada halaman utama dan memilih laporan *history* pelanggan seperti pada gambar 4.73. Setelah itu pilih tampilan laporan, nama pelanggan beserta tanggal awal dan akhir periode. Lalu sistem menampilkan laporan *history* pelanggan seperti pada gambar 4.74 untuk tabular dan gambar 4.75 untuk grafis.

Jenis laporan pelanggan

Pilih jenis laporan pelanggan yang anda inginkan

History pelanggan

Parameter laporan pelanggan

Pilih nama pelanggan.

Mutiara Anjani

Pilih periode.

01/01/2013

hingga

29/01/2013

Tampilkan

Deskripsi laporan

Laporan history pelanggan akan memberi informasi transaksi-transaksi yang pernah dilakukan oleh pelanggan tertentu.

Tampilan laporan

Grafis

NB : pastikan nama pelanggan terisi secara otomatis dari pilihan yang ada. Anda cukup mengetik huruf depan dari nama pelanggan dan kami akan mencari data nama pelanggan yang sesuai.

Gambar 4. 73 Form Laporan *History* Pelanggan

Laporan history pelanggan - Toko Besar



Nama pelanggan : **Mutiara Anjani**
 Alamat :
 Email :
 Telepon :
 Periode : **01 January 2013** hingga **29 January 2013**

25 January 2013

ID. barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
mouse001	Mouse E-Blue Cobra	1	100,000	150,000	10,000	140,000	40,000
						140,000	40,000

27 January 2013

ID. barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
fd001	Flash Disk Sandisk Mini 8GB	1	60,000	80,000		80,000	20,000
mouse001	Mouse E-Blue Cobra	1	100,000	150,000	10,000	140,000	40,000
						210,000	60,000

29 January 2013

ID. barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
fd003	Flash Disk Kingston 16GB	1	100,000	135,000	10,000	125,000	25,000
modem001	Modem Vodafone Rev A	1	150,000	200,000		200,000	50,000
coolpad001	Cooling Pad Targus		150,000	185,000			
						325,000	75,000

29 January 2013

ID. barang	Nama barang	Jumlah	Harga beli	Harga jual	Diskon	Total harga	Keuntungan
fd002	Flash Disk Kingston 8GB	5	65,000	80,000	5,000	375,000	50,000
fd003	Flash Disk Kingston 16GB	4	100,000	135,000	10,000	500,000	100,000
						875,000	150,000

Total penjualan : **1,550,000** Total keuntungan : **325,000**

Gambar 4. 74 Laporan Tabular *History* Pelanggan

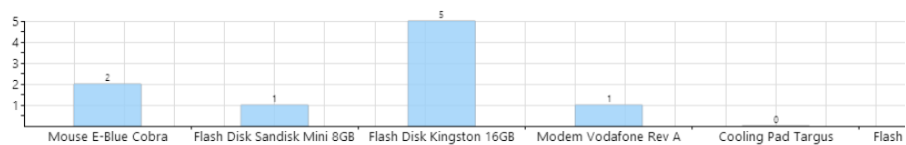


Laporan history pelanggan - Toko Besar
 01 January 2013 - 29 January 2013

Grafik History Transaksi Pelanggan



Grafik History Barang Pelanggan



Gambar 4. 75 Laporan Grafis *History* Pelanggan

4.4.17 Menampilkan Laporan *History Supplier*

Untuk menampilkan laporan *history supplier*, pihak manajemen memilih menu laporan pada halaman utama dan memilih laporan *history supplier* seperti pada gambar 4.76. Setelah itu pilih tampilan laporan, nama *supplier* beserta tanggal awal dan akhir periode. Lalu sistem menampilkan laporan *history supplier* seperti pada gambar 4.77 untuk tabular dan gambar 4.78 untuk grafis.

Gambar 4. 76 Form Laporan *History Supplier*

Laporan history supplier - Toko Besar



Toko Besar

Nama supplier : **Interface**
 Alamat :
 Email :
 Telepon : **021-6957133**
 Periode : **01 January 2013** hingga **29 January 2013**

27 January 2013

ID. barang	Nama barang	Jumlah	Harga satuan	Total harga
microSD001	MicroSD V-Gen 4GB	34	25,000	850,000
				850,000

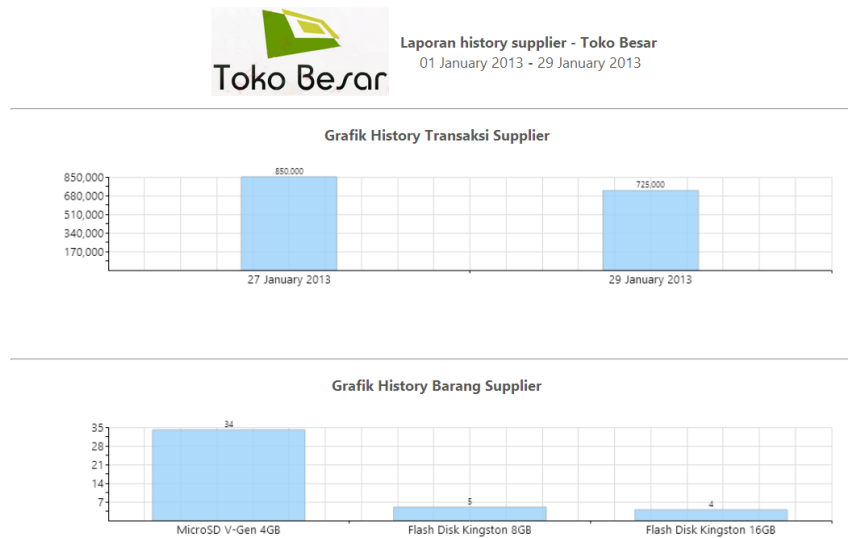
29 January 2013

ID. barang	Nama barang	Jumlah	Harga satuan	Total harga
fd002	Flash Disk Kingston 8GB	5	65,000	325,000
fd003	Flash Disk Kingston 16GB	4	100,000	400,000
				725,000

Total pengeluaran : **1,575,000**

juwall.com - layanan kasir online

Gambar 4. 77 Laporan Tabular *History Supplier*



Gambar 4. 78 Laporan Grafis *History Supplier*

4.4.18 Mengubah Layanan

Proses mengubah layanan dibagi menjadi dua yaitu mengisi *form* perubahan layanan dan mengkonfirmasi *transfer* pembayaran tagihan perubahan layanan yang telah dilakukan. Untuk mengisi *form* perubahan layanan, pihak manajemen memilih menu *setting* pada halaman utama lalu memilih ubah layanan. Setelah itu isi paket layanan, batas data penjualan dan jumlah bulan seperti pada gambar 4.79. Setelah selesai, sistem akan menampilkan nota tagihan perubahan layanan seperti pada gambar 4.80.

Form pembayaran

Ubah paket layanan

Paket layanan: Super

Jumlah bulan layanan: 6 *1 bulan = 30 hari

Ubah batas data penjualan

Batas data penjualan: 500,000 *kelipatan 500,000

Jumlah bulan batas data penjualan: 6 *1 bulan = 30 hari

Kirim

Gambar 4. 79 Form Perubahan Layanan

Tagihan pembayaran layanan - Toko Besar



Toko Besar

Ditagihkan kepada, Toko Besar di tempat	Dibayarkan kepada, Juwall Surabaya, Indonesia 0857 3008 1004	Tanggal tagihan 29 January 2013 Kode tagihan caahbh0zp2h <small>(Kode ini akan digunakan saat konfirmasi transaksi)</small>
--	--	---

Layanan	Jumlah bulan	Harga bulanan	Total
Paket Super	6	25,000	150,000
Penyimpanan data penjualan (kapasitas 500,000)	6	50,000	300,000
Total			450,000

Silahkan ditransfer ke rekening Juwall, sebelum tanggal jatuh tempo yaitu **05 February 2013**. Berikut adalah detail rekening Juwall :

Nama bank Bank Mandiri
No. rekening 141-00-1266-5998
Atas nama MUHAMMAD AFIFUL HASHIF

Terima kasih,

Juwall
juwall.com - layanan lebih online

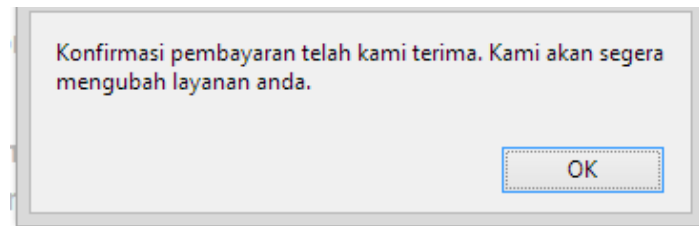
Gambar 4. 80 Nota Tagihan Pengubahan Layanan

Setelah mendapatkan nota tagihan, pihak manajemen melakukan *transfer* uang ke rekening yang dicantumkan di nota tagihan. Setelah itu, pihak manajemen melakukan konfirmasi *transfer* agar sistem dapat mengubah layanan sesuai yang dibayarkan. Untuk mengkonfirmasi *transfer*, pihak manajemen memilih menu *setting* pada halaman utama lalu memilih ubah layanan. Setelah itu buka *form* konfirmasi pembayaran dan isi nama *bank*, nomer rekening, jumlah pembayaran, tanggal *transfer* dan kode tagihan seperti pada gambar 4.81. Setelah selesai, sistem akan menampilkan pesan sukses seperti pada gambar 4.82.

Konfirmasi pembayaran ×

Nama bank	<input type="text" value="BNI"/>
Nomer rekening	<input type="text" value="1029372921"/>
Jumlah pembayaran	<input type="text" value="450,000"/>
Tanggal transfer	<input type="text" value="29/01/2013"/>
Kode tagihan	<input type="text" value="caahbh0zp2h"/>
<input type="button" value="Kirim"/>	

Gambar 4. 81 Form Konfirmasi Transfer



Gambar 4. 82 Konfirmasi *Transfer* berhasil

4.5 Uji Coba dan Evaluasi Sistem

Setelah sistem diimplementasikan, langkah selanjutnya adalah melakukan uji coba dan evaluasi sistem. Tujuan dari uji coba dan evaluasi sistem adalah untuk mengetahui apakah sistem yang telah dibangun ini sudah berjalan dengan baik sesuai dengan tujuan atau *output* yang diharapkan. Uji coba dan evaluasi sistem dilakukan sesuai dengan rancangan yang dibuat pada bab 3 dan menggunakan metode *black box testing*. Uji coba dan evaluasi dilakukan terhadap fungsi-fungsi sistem yang telah ditulis pada rancangan di bab 3.

4.5.1 Uji Coba Pendaftaran Toko

Proses uji coba pendaftaran toko dilakukan terhadap tiga *test case*. *Test case* pertama adalah mendaftarkan toko dengan mengisi *id* toko, nama toko, *password*, konfirmasi *password* dan *email* pada halaman pendaftaran toko. Setelah itu sistem akan mengirimkan *email* aktivasi ke *email* yang didaftarkan oleh pengguna. *Test case* kedua adalah pengiriman ulang *email* aktivasi jika pengguna tidak menerima *email* yang dikirimkan sistem. *Test case* ketiga adalah mengaktivasi toko dengan membuka *link* aktivasi yang tercantum di *email* aktivasi. Hasil uji coba pendaftaran toko dapat dilihat pada tabel 4.1.

Tabel 4. 1 Uji Coba Pendaftaran Toko

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Mendaftarkan toko	<i>ID toko, nama toko, Nama toko, Password, Konfirmasi password, Email</i>	Pendaftaran berhasil	Sukses (gambar 4.15)
2	Mengirim ulang <i>email</i> aktivasi ke <i>email</i> pendaftar	<i>ID toko</i>	Email aktivasi telah dikirim ulang ke <i>email</i> pendaftar	Sukses (gambar 4.18)
3	Mengaktivasi toko	<i>ID toko, kode aktivasi</i>	Akun toko telah diaktifkan dan dapat mengakses halaman <i>login</i>	Sukses (gambar 4.19)

4.5.2 Uji Coba Login

Proses uji coba *login* dilakukan terhadap tiga *test case*. *Test case* pertama adalah melakukan *login* ke dalam sistem dengan mengisi *id toko, username* dan *password*. *Test case* kedua adalah melakukan *login* ke dalam sistem dengan *format* yang salah sehingga sistem harus menampilkan pesan kesalahan. *Test case* ketiga adalah menguji fitur lupa *password*. Pengguna mengisi *id toko, username* lalu memilih menu lupa *password*. Setelah itu sistem akan mengirim *password* ke *email* pengguna yang telah terdaftar. Hasil uji coba *login* dapat dilihat pada tabel 4.2.

Tabel 4. 2 Uji Coba Login

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
4	Melakukan <i>login</i>	<i>ID toko, username, password</i>	<i>Login</i> berhasil dan masuk ke halaman utama sistem	Sukses (gambar 4.54)
5	Melakukan <i>login</i> dengan <i>format</i> salah	<i>ID toko, username,</i>	Muncul pesan " <i>Username</i> atau	Sukses (gambar

		<i>password</i>	<i>password</i> salah"	4.20)
6	Menjalankan fitur "Lupa <i>Password</i> "	<i>ID</i> toko, <i>username</i>	Muncul pesan bahwa <i>password</i> telah dikirim melalui <i>email</i>	Sukses (gambar 4.21)

4.5.3 Uji Coba Data Barang

Proses uji coba data barang dilakukan untuk menguji CRUD (*Create, Read, Update, Delete*) pada data barang, promosi barang di Facebook berdasarkan data yang disimpan, (*Create, Read, Update, Delete*) pada data kategori barang serta mengkategorikan barang pada kategori tertentu. Hasil uji coba data barang dapat dilihat pada tabel 4.3.

Tabel 4. 3 Uji Coba Data Barang

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
7	Menampilkan data barang	Memilih menu "Barang"	Data barang	Sukses (gambar 4.27)
8	Memasukkan data barang	<i>ID</i> barang, nama barang, jenis, stok terkini, stok siap jual, stok minimum, harga beli, harga jual	Data barang berhasil disimpan dan ditampilkan pada tabel barang	Sukses (gambar 4.29)
9	Mengubah data barang	<i>ID</i> barang, nama barang, jenis, stok terkini, stok siap jual, stok minimum, harga beli, harga jual	Data barang berhasil diubah dan ditampilkan pada tabel barang	Sukses (gambar 4.31)
10	Menjalankan fitur "Promosi di Facebook"	<i>ID</i> atau nama barang	Nama barang, harga jual dan stok terkini berhasil dipromokan melalui status Facebook	Sukses (gambar 4.33)

11	Menampilkan data kategori barang	Memilih menu "Kategori Barang"	Data kategori barang	Sukses (gambar 4.34)
12	Memasukkan data kategori barang	Nama kategori	Data kategori barang berhasil disimpan dan ditampilkan pada tabel barang	Sukses (gambar 4.36)
13	Mengubah data kategori barang	Nama kategori	Data kategori barang berhasil diubah dan ditampilkan pada tabel barang	Sukses (gambar 4.38)
14	Menghapus data kategori barang	Nama kategori	Data kategori barang berhasil dihapus	Sukses (gambar 4.39)
15	Mengkategorikan barang	ID atau nama barang, nama kategori	Data barang telah dimasukkan ke dalam kategori	Sukses (gambar 4.41)

4.5.4 Uji Coba Data Karyawan

Proses uji coba data karyawan dilakukan untuk menguji CRUD (*Create, Read, Update, Delete*) pada data karyawan beserta hak aksesnya. Hasil uji coba data karyawan dapat dilihat pada tabel 4.4.

Tabel 4. 4 Uji Coba Data Karyawan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
16	Menampilkan data karyawan	Memilih menu "Karyawan"	Data karyawan	Sukses (gambar 4.23)
17	Memasukkan data karyawan	ID karyawan, nama, <i>email</i> , telepon, hak akses	Data karyawan berhasil disimpan dan ditampilkan pada tabel karyawan	Sukses (gambar 4.24)
18	Mengubah data karyawan	ID karyawan, nama, <i>email</i> , telepon, hak akses	Data karyawan berhasil diubah dan ditampilkan pada tabel	Sukses (gambar 4.25)

			karyawan	
19	Menghapus data karyawan	ID atau nama karyawan	Data karyawan berhasil dihapus	Sukses (gambar 4.26)

4.5.5 Uji Coba Data Pelanggan

Proses uji coba data pelanggan dilakukan untuk menguji CRUD (*Create, Read, Update, Delete*) pada data pelanggan. Hasil uji coba data pelanggan dapat dilihat pada tabel 4.5.

Tabel 4.5 Uji Coba Data Pelanggan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
20	Menampilkan data pelanggan	Memilih menu "Pelanggan"	Data pelanggan	Sukses (gambar 4.42)
21	Memasukkan data pelanggan	ID pelanggan, nama, alamat, <i>email</i> , telepon, tanggal lahir	Data pelanggan berhasil disimpan dan ditampilkan pada tabel pelanggan	Sukses (gambar 4.44)
22	Mengubah data pelanggan	ID pelanggan, nama, alamat, <i>email</i> , telepon, tanggal lahir	Data pelanggan berhasil diubah dan ditampilkan pada tabel pelanggan	Sukses (gambar 4.46)
23	Menghapus data pelanggan	ID atau nama pelanggan	Data pelanggan berhasil dihapus	Sukses (gambar 4.47)

4.5.6 Uji Coba Data *Supplier*

Proses uji coba data *supplier* dilakukan untuk menguji CRUD (*Create, Read, Update, Delete*) pada data *supplier*. Hasil uji coba data *supplier* dapat dilihat pada tabel 4.6.

Tabel 4. 6 Uji Coba Data *Supplier*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
24	Menampilkan data <i>supplier</i>	Memilih menu " <i>Supplier</i> "	Data <i>supplier</i>	Sukses (gambar 4.48)
25	Memasukkan data <i>supplier</i>	Nama <i>supplier</i> , alamat, <i>email</i> , telepon	Data <i>supplier</i> berhasil disimpan dan ditampilkan pada tabel <i>supplier</i>	Sukses (gambar 4.50)
26	Mengubah data <i>supplier</i>	Nama <i>supplier</i> , alamat, <i>email</i> , telepon	Data <i>supplier</i> berhasil diubah dan ditampilkan pada tabel <i>supplier</i>	Sukses (gambar 4.52)
27	Menghapus data <i>supplier</i>	Nama <i>supplier</i>	Data <i>supplier</i> berhasil dihapus	Sukses (gambar 4.53)

4.5.7 Uji Coba Data Penjualan

Proses uji coba data penjualan dilakukan terhadap tiga *test case*. *Test case* pertama adalah mencatat transaksi penjualan, baik tunai ataupun kredit. Proses pencatatan transaksi penjualan dinyatakan berhasil ketika sistem telah menyimpan data penjualan dengan benar dan menampilkan nota penjualan. *Test case* kedua adalah mencatat retur penjualan. Proses pencatatan retur penjualan dinyatakan berhasil jika sistem telah menyimpan data retur dengan benar dan menampilkan nota retur penjualan. *Test case* ketiga adalah mencatat transaksi pembayaran kredit. Proses pencatatan transaksi pembayaran kredit dinyatakan berhasil jika sistem telah menyimpan data pembayaran kredit dengan benar dan menampilkan nota pembayaran kredit. Hasil uji coba data penjualan dapat dilihat pada tabel 4.7.

Tabel 4. 7 Uji Coba Data Penjualan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
28	Mencatat transaksi penjualan	<i>ID</i> atau nama barang, jumlah, diskon satuan, nama pelanggan, diskon keseluruhan, uang yang dibayarkan	Total harga, pesan transaksi penjualan berhasil disimpan dan nota penjualan	Sukses (gambar 4.55 dan 4.57)
29	Mencatat retur penjualan	<i>ID</i> atau nama barang, jumlah retur dan alasan.	Pesan retur penjualan berhasil disimpan dan nota retur penjualan	Sukses (gambar 4.59)
30	Mencatat pembayaran kredit	Jumlah uang yang dibayarkan pelanggan	Pesan pembayaran kredit telah disimpan dan nota pembayaran kredit	Sukses (gambar 4.61)

4.5.8 Uji Coba Data Penerimaan

Proses uji coba data penjualan dilakukan terhadap dua *test case*. *Test case* pertama adalah mencatat transaksi penerimaan. Proses pencatatan transaksi penerimaan dinyatakan berhasil ketika sistem telah menyimpan data penerimaan dengan benar dan menampilkan nota penerimaan. *Test case* kedua adalah pencatatan retur pembelian. Proses pencatatan retur pembelian dinyatakan berhasil jika sistem telah menyimpan data retur pembelian dengan benar. Hasil uji coba data penerimaan dapat dilihat pada tabel 4.8.

Tabel 4. 8 Uji Coba Data Penerimaan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
31	Mencatat transaksi penerimaan	<i>ID</i> atau nama barang, jumlah, nama <i>supplier</i> , uang yang dibayarkan	Total harga dan pesan transaksi penerimaan berhasil disimpan	Sukses (gambar 4.63)

4.5.9 Uji Coba Laporan

Proses uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem, yaitu laporan stok terkini, laporan tren penjualan, laporan *history* pelanggan dan laporan *history supplier*. Uji coba laporan dinyatakan berhasil jika laporan yang dihasilkan sesuai dengan *parameter* yang dimasukkan ke dalam sistem. Hasil uji coba data penerimaan dapat dilihat pada tabel 4.9.

Tabel 4. 9 Uji Coba Laporan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
32	Menampilkan laporan stok terkini	Memilih menu "Laporan Stok Terkini"	Tampilan laporan stok terkini dalam bentuk grafis atau tabular	Sukses (gambar 4.65 dan 4.66)
33	Menampilkan laporan tren penjualan	Tampilan laporan, nama barang, periode laporan	Tampilan laporan tren penjualan dalam bentuk grafis atau tabular	Sukses (gambar 4.68 dan 4.69)
34	Menampilkan laporan <i>history</i> pelanggan	Tampilan laporan, nama pelanggan, periode laporan	Tampilan laporan <i>history</i> pelanggan dalam bentuk grafis atau tabular	Sukses (gambar 4.71 dan 4.72)
35	Menampilkan laporan <i>history supplier</i>	Tampilan laporan, nama <i>supplier</i> , periode laporan	Tampilan laporan <i>history supplier</i> dalam bentuk grafis atau tabular	Sukses (gambar 4.74 dan 4.75)

4.5.10 Uji Coba Ubah Layanan

Proses uji coba ubah layanan dilakukan terhadap dua *test case*. *Test case* pertama adalah memasukkan data perubahan layanan dengan mengisi paket layanan, batas penyimpanan data penjualan beserta jumlah bulan yang diinginkan pada halaman perubahan layanan. Setelah itu sistem akan mengirimkan nota tagihan pembayaran dan meminta pengguna melakukan *transfer* sesuai jumlah yang tercantum pada nota tagihan tersebut. *Test case* kedua adalah konfirmasi *transfer* yang telah dilakukan dengan mengisi nama bank, nomer rekening, jumlah pembayaran, tanggal *transfer* dan kode tagihan. Hasil uji coba ubah layanan dapat dilihat pada tabel 4.10.

Tabel 4. 10 Uji Coba Ubah Layanan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
36	Memasukkan perubahan layanan	Paket layanan, jumlah bulan paket layanan, batas data penjualan, jumlah bulan batas data penjualan	Pesan permintaan perubahan layanan berhasil disimpan dan nota tagihan	Sukses (gambar 4.77)
37	Mengkonfirmasi <i>transfer</i>	Nama bank, nomer rekening, jumlah pembayaran, tanggal <i>transfer</i> , kode tagihan	Pesan konfirmasi <i>transfer</i> berhasil disimpan	Sukses (gambar 4.79)

4.5.11 Uji Coba *Cloud Performance*

Untuk mengetahui *performance* dari layanan *cloud* yang digunakan, perlu beberapa pengujian sesuai dengan desain uji coba seperti yang tertulis pada bab 3, yaitu *CPU Percentage & Network Out, DB Monitor, CPU, Memory* dan *Disk*.

Untuk *CPU Percentage & Network Out* dan *DB Monitor* dapat dilihat dari *dashboard Windows Azure* saat penggunaan sistem. Sedangkan untuk *CPU*, *Memory* dan *Disk* dilihat menggunakan *third-party application* yaitu *Site24x7* dari Soho. Sistem diuji coba dengan cara penggunaan oleh tempat studi kasus dalam proses bisnis mereka. Hasil uji coba ubah *cloud performance* dapat dilihat pada tabel 4.11 dengan *detail* pengujian pada bagian lampiran.

Tabel 4. 11 Uji Coba *Cloud Performance*

No	Pengujian	Status	Hasil
1	<i>CPU Percentage & Network Out</i> melalui <i>dashboard Windows Azure</i>	Berjalan dengan normal.	Lampiran 6 bagian <i>dashboard Windows Azure</i>
2	<i>DB Monitor</i> melalui <i>dashboard Windows Azure</i>	Berjalan dengan normal.	Lampiran 6 bagian <i>dashboard Windows Azure</i>
3	<i>CPU</i> melalui <i>website Site24x7</i> dari Soho	Berjalan dengan normal.	Lampiran 6 bagian <i>site24x7 - Soho</i>
4	<i>Memory</i> melalui <i>website Site24x7</i> dari Soho	Berjalan dengan normal.	Lampiran 6 bagian <i>site24x7 - Soho</i>
5	<i>Disk</i> melalui <i>website Site24x7</i> dari Soho	Berjalan dengan normal.	Lampiran 6 bagian <i>site24x7 - Soho</i>

4.5.12 Uji Coba Proses Bisnis

Untuk menguji proses bisnis pada tempat studi kasus dengan menggunakan sistem, perlu dibuat sebuah skenario atau kasus penentuan HPP, pencatatan transaksi penerimaan barang dan transaksi penjualan. Skenario proses bisnis yang digunakan adalah sebagai berikut.

Metode penentuan HPP yang digunakan adalah *First In First Out*. Pengaturan metode dilakukan seperti pada gambar 4.83.

Gambar 4. 83 Pengaturan Metode Penentuan HPP

Barang yang akan digunakan adalah *Capdase Lumia 710*. Stok awal persediaan adalah 0, seperti yang ditunjukkan pada gambar 4.84.

Barang

ID Barang	Nama	Stok terkini	Siap jual	Stok Minim	Harga Beli	Harga Jual	Profit (%)	Diskon (%)
4894299006...	Capdase Lumia 710	0	0	2	0	0	50	0

Gambar 4. 84 Stok Awal Persediaan Barang

Skenario transaksi penerimaan barang pertama dilakukan dengan jumlah barang sebanyak 6 dan harga beli dari *supplier* sebesar 85.000 rupiah seperti pada gambar 4.85. Nota transaksi penerimaan barang adalah seperti pada gambar 4.86.

Penerimaan

Nama Barang : atau

Nama supplier (optional) : Asiaraya Computronics atau

ID Barang	Nama	Jumlah	Harga satuan	Total	
4894299006395	Capdase Lumia 710	6	85000	510000	<input type="button" value="X Delete"/>

Total harga **510,000**

Uang

Kembalian **90,000**

stok barang siap jual

cetak nota

Gambar 4. 85 Transaksi Penerimaan Barang Pertama

Nota penerimaan barang

Penerimaan Barang - Toko Besar

Tanggal : 11 March 2013 No. Transaksi : 14
 Nama supplier : Asiaraya Computronics

ID. Barang	Nama barang	Jumlah	Harga satuan	Total harga
4894299006395	Capdase Lumia 710	6	85.000	510.000

Tanda terima, Total harga : 510,000

Toko Besar

Gambar 4. 86 Nota Transaksi Penerimaan Barang Pertama

Setelah transaksi selesai dilakukan, data stok barang beserta harga jual dan harga beli akan disesuaikan oleh sistem seperti pada gambar 4.87.

Barang

ID Barang	Nama	Stok terkini	Siap jual	Stok Minim	Harga Beli	Harga Jual	Profit (%)	Diskon (%)
4894299006...	Capdase Lumia 710	6	6	2	85000	127500	50	0

Gambar 4. 87 Stok Barang Setelah Transaksi Penerimaan Barang Pertama

Skenario transaksi penerimaan barang kedua dilakukan dengan jumlah barang sebanyak 2 dan harga beli dari *supplier* sebesar 98.000 rupiah seperti pada gambar 4.88. Nota transaksi penerimaan barang adalah seperti pada gambar 4.89.

Penerimaan

Nama Barang : atau

Nama supplier (optional) : atau

ID Barang	Nama	Jumlah	Harga satuan	Total	
4894299006395	Capdase Lumia 710	2	98000	196000	<input type="button" value="X Delete"/>

Total harga: **196,000**

Uang:

Kembalian: **4,000**

stok barang siap jual

cetak nota

Gambar 4. 88 Transaksi Penerimaan Barang Kedua

Nota penerimaan barang

Penerimaan Barang - Toko Besar

Toko Besar

Tanggal : 11 March 2013

No. Transaksi : 15

Nama supplier : **Asiaraya Computronics**

ID. Barang	Nama barang	Jumlah	Harga satuan	Total harga
4894299006395	Capdase Lumia 710	2	98,000	196,000

Tanda terima, Total harga : **196,000**

Toko Besar

juwall.com - layanan kasir online

Gambar 4. 89 Nota Transaksi Penerimaan Barang Kedua

Setelah transaksi selesai dilakukan, data stok barang beserta harga jual dan harga beli akan disesuaikan oleh sistem seperti pada gambar 4.90. Karena metode penentuan HPP yang digunakan adalah *First In First Out*, maka harga beli dan

harga jual tidak berubah sampai stok barang pada transaksi penerimaan barang pertama habis.

Barang

+ Add new record ✓ Save changes ✕ Cancel changes

ID Barang	Nama	Stok terkini	Siap jual	Stok Minim	Harga Beli	Harga Jual	Profit (%)	Diskon (%)
4894299006...	Capdase Lumia 710	8	8	2	85000	127500	50	0

Gambar 4. 90 Stok Barang Setelah Transaksi Penerimaan Barang Kedua

Setelah transaksi penerimaan selesai dilakukan, skenario bisnis selanjutnya adalah melakukan transaksi penjualan. Transaksi penjualan akan dilakukan terhadap barang *Capdase Lumia 710* sebanyak 6 buah seperti pada gambar 4.91.

Nota transaksi penerimaan barang adalah seperti pada gambar 4.92.

Penjualan

Nama Barang :

Nama pelanggan (optional) : atau

ID Barang	Nama	Jumlah	Harga	Diskon (%)	Total
489429900...	Capdase Lumia 710	6	127500	0	765000

Tunai Kredit

Total harga 765,000

Diskon (%) 0.00

Total bayar **765,000**

Uang 800,000

Kembalian **35,000**

cetak nota

Gambar 4. 91 Transaksi Penjualan Barang

Nota penjualan

Nota penjualan - Toko Besar


Toko Besar

Tanggal : **11 March 2013** No. Transaksi : **2**
Nama pelanggan : **Mutiara Anjani**

Nama barang	Jumlah	Harga satuan	Diskon	Total harga
Capdase Lumia 710	6	127,500	0.00 %	765,000

Tanda terima, Diskon penjualan : **0.00 %**
Total harga : **765,000**

Toko Besar juwall.com - layanan kasir online

Gambar 4. 92 Nota Transaksi Penjualan

Setelah transaksi selesai dilakukan, data stok barang beserta harga jual dan harga beli akan disesuaikan oleh sistem seperti pada gambar 4.93. Karena semua barang pada transaksi penerimaan pertama telah habis terjual, maka harga yang ditampilkan adalah harga dari transaksi penerimaan kedua.

Barang

ID Barang	Nama	Stok terkini	Siap jual	Stok Minim	Harga Beli	Harga Jual	Profit (%)	Diskon (%)
4894299006...	Capdase Lumia 710	2	2	2	98000	147000	50	0

Gambar 4. 93 Stok Barang Setelah Transaksi Penjualan

Pihak manajemen dapat melihat informasi barang terlaris berdasarkan transaksi-transaksi yang telah dilakukan pada periode tertentu. Pihak manajemen dapat memilih menu laporan barang dan memasukkan *parameter* laporan sesuai dengan yang dibutuhkan seperti pada gambar 4.94. Laporan barang terlaris dapat dilihat dalam 2 bentuk, yaitu grafis seperti pada gambar 4.95 dan tabular seperti pada gambar 4.96.

Parameter laporan

Pilih kategori barang.

Aksesoris Handphone

Pilih periode barang.

01/10/2012

hingga

20/03/2013

Jumlah data yang diinginkan

10

Urutkan berdasarkan

Jumlah terjual

Tampilkan

NB : pastikan kategori barang terisi secara otomatis dari pilihan yang ada. Anda cukup mengetik huruf depan dari kategori barang dan kami akan mencari data kategori barang yang sesuai. Jika tidak diisi maka akan ditampilkan barang dari semua kategori.

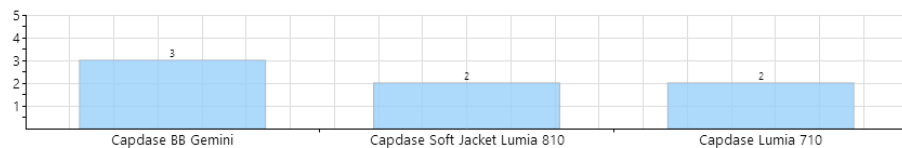


Gambar 4.94 *Input Parameter Laporan Barang Terlaris*

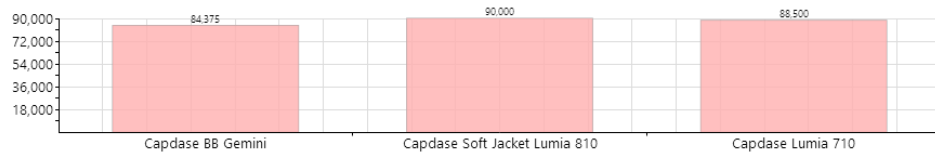


Laporan barang atau jasa terlaris (Fast Moving) - Toko Besar
01 October 2012 - 20 March 2013

Grafik Jumlah Barang Terjual



Grafik Jumlah Keuntungan



Gambar 4.95 *Laporan Grafis Barang Terlaris*

Laporan barang atau jasa terlaris (Fast Moving) - Toko Besar

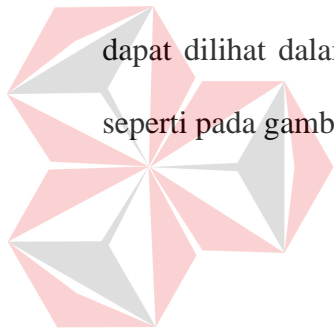


Periode : 01 October 2012 hingga 20 March 2013

No	Nama	Jumlah terjual	Jumlah keuntungan
1	Capdase BB Gemini	3	84,375
2	Capdase Soft Jacket Lumia 810	2	90,000
3	Capdase Lumia 710	2	88,500

Gambar 4. 96 Laporan Tabular Barang Terlaris

Pihak manajemen dapat melihat informasi barang tidak laris berdasarkan transaksi-transaksi yang telah dilakukan pada periode tertentu. Pihak manajemen dapat memilih menu laporan barang dan memasukkan *parameter* laporan sesuai dengan yang dibutuhkan seperti pada gambar 4.97. Laporan barang tidak laris dapat dilihat dalam 2 bentuk, yaitu grafis seperti pada gambar 4.98 dan tabular seperti pada gambar 4.99.



UNIVERSITAS
Dinamika

Parameter laporan

Pilih kategori barang.

Flash Disk

Pilih periode barang.

01/10/2012

hingga

20/03/2013

Jumlah data yang diinginkan

10

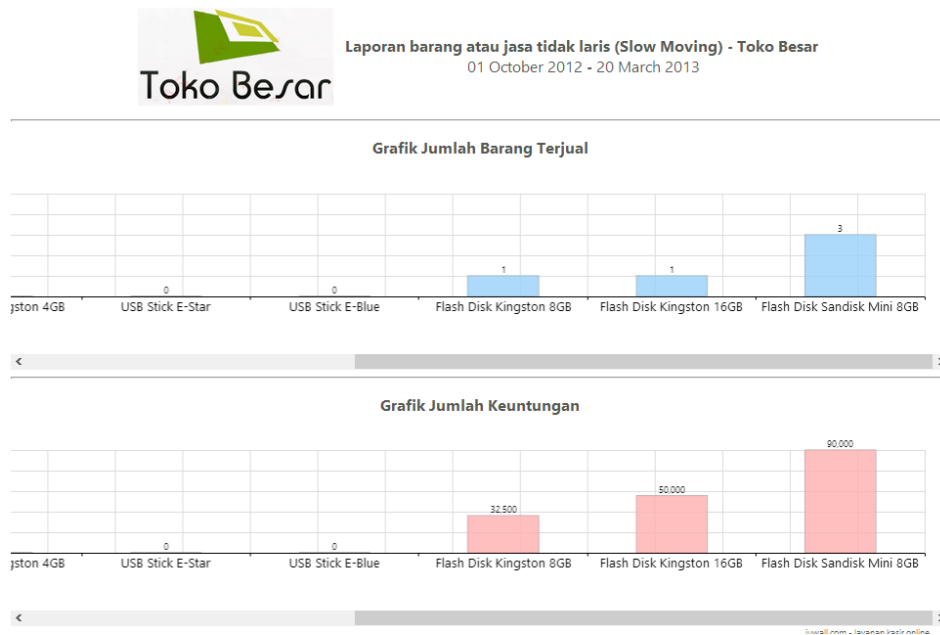
Urutkan berdasarkan

Jumlah terjual

Tampilkan

NB : pastikan kategori barang terisi secara otomatis dari pilihan yang ada. Anda cukup mengetik huruf depan dari kategori barang dan kami akan mencari data kategori barang yang sesuai. Jika tidak diisi maka akan ditampilkan barang dari semua kategori.

Gambar 4. 97 Input Parameter Laporan Barang Tidak Laris



Gambar 4. 98 Laporan Grafis Barang Tidak Laris

Laporan barang atau jasa tidak laris (Slow Moving) - Toko Besar
Periode : 01 October 2012 hingga 20 March 2013

No	Nama	Jumlah terjual	Jumlah keuntungan
1	Flash Disk San Disk Mini 4GB	0	0
2	Flash Disk Kingston 2GB	0	0
3	Flash Disk Kingston 4GB	0	0
4	USB Stick E-Star	0	0
5	USB Stick E-Blue	0	0
6	Flash Disk Kingston 8GB	1	32.500
7	Flash Disk Kingston 16GB	1	50.000
8	Flash Disk Sandisk Mini 8GB	3	90.000

Gambar 4. 99 Laporan Tabular Barang Tidak Laris

Ketika stok barang akan habis dan mendekati batas minimum, bagian pengelola stok persediaan dapat meminta penambahan stok barang. Sistem dapat memberikan informasi *supplier-supplier* yang pernah men-*supply* barang yang habis tersebut beserta rata-rata harga yang diberikan pada rentang periode tertentu. Informasi ini dapat dijadikan pertimbangan untuk mengetahui *supplier* mana yang memberikan harga termurah. Untuk mendapatkan informasi tersebut, bagian

pengelola stok persediaan memilih menu laporan *supplier* dan memasukkan *parameter-parameter* laporan sesuai dengan yang dibutuhkan seperti pada gambar 4.100. Laporan barang tidak laris dapat dilihat dalam 2 bentuk, yaitu grafis seperti pada gambar 4.101 dan tabular seperti pada gambar 4.102.

Parameter laporan supplier

Nama barang

Pilih periode.

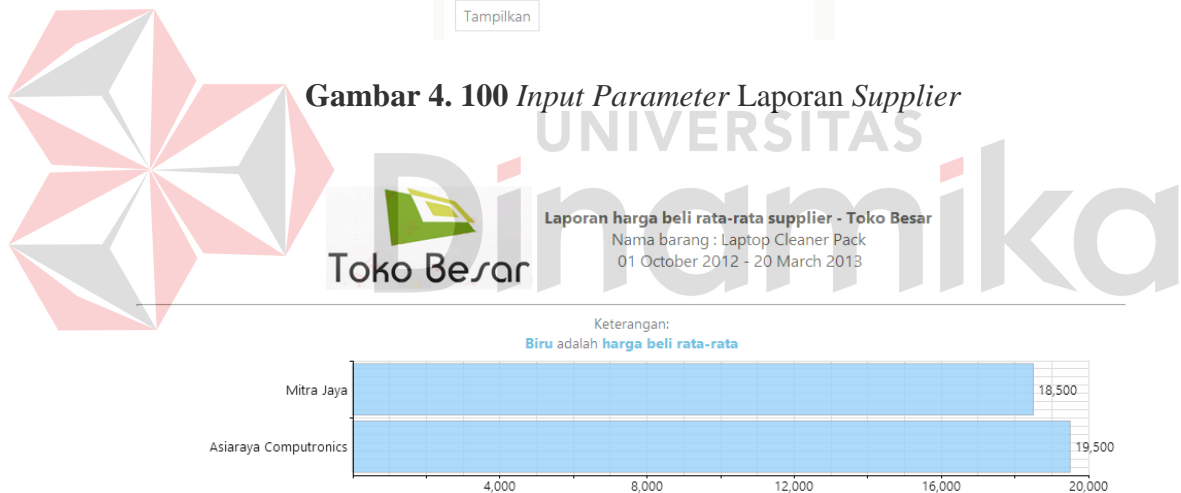
01/10/2012

hingga

20/03/2013

Tampilkan

Gambar 4. 100 *Input Parameter Laporan Supplier*



Gambar 4. 101 *Laporan Grafis Supplier*

Laporan harga beli rata-rata supplier - Toko Besar



Tanggal : 20 March 2013

Nama barang : Laptop Cleaner Pack

Nama supplier	Harga beli rata-rata
Mitra Jaya	18,500
Asiaraya Computronics	19,500

Gambar 4. 102 *Laporan Tabular Supplier*

Berdasarkan skenario pengujian di atas dapat disimpulkan bahwa sistem telah berjalan dengan benar dan menghasilkan perhitungan yang sesuai dengan inputan dan metode yang ditentukan oleh pengguna. Hasil uji coba dapat dilihat pada tabel 4.12.

Tabel 4. 12 Uji Coba Proses Bisnis

No.	Proses Bisnis	Input	Output yang diharapkan	Status
1	Menentukan HPP	Metode HPP yang digunakan.	Pengaturan HPP telah disimpan.	Sukses (gambar 4.80)
2	Mencatat transaksi penerimaan barang	ID atau nama barang, jumlah, harga dan nama <i>supplier</i> .	Nota penerimaan barang. Stok barang bertambah, harga beli tersimpan dengan benar sesuai yang diinputkan.	Sukses (gambar 4.81 – 4.87)
3	Mencatat transaksi penjualan	ID atau nama barang jumlah, harga, diskon dan nama pelanggan.	Nota penjualan. Stok barang berkurang, harga beli dan jual tersimpan sesuai metode yang dipilih.	Sukses (gambar 4.88 – 4.90)
4	Menampilkan informasi barang terlaris (<i>fast moving</i>).	Kategori barang, periode transaksi, jumlah data, pengurutan (<i>sort</i>)	Informasi barang terlaris	Sukses (gambar 4.91 – 4.93)
5	Menampilkan informasi barang tidak laris (<i>slow moving</i>).	Kategori barang, periode transaksi, jumlah data, pengurutan (<i>sort</i>)	Informasi barang tidak laris	Sukses (gambar 4.94 – 4.96)

6	Menampilkan informasi <i>supplier</i> bagi barang tertentu	Nama barang, periode transaksi	Informasi <i>supplier</i> , barang yang di- <i>supply</i> beserta rata-rata harga	Sukses (gambar 4.97-4.99)
---	--	--------------------------------	---	---------------------------

4.5.13 Evaluasi Kesesuaian Hasil Akhir Sistem

Sistem informasi penjualan berbasis *cloud* yang telah dibangun harus mampu memenuhi kebutuhan-kebutuhan yang ada di latar belakang permasalahan. Sistem informasi penjualan berbasis *cloud* ini juga harus mengatasi permasalahan-permasalahan sistem penjualan yang telah digunakan sebelumnya.

Oleh karena itu, evaluasi kesesuaian hasil akhir sistem dilakukan untuk memastikan semua kebutuhan yang terdapat pada latar belakang dan yang termasuk dalam ruang lingkup sistem telah dipenuhi oleh sistem yang dibangun. Hasil evaluasi dapat dilihat di tabel 4.13.

Tabel 4. 13 Evaluasi Kesesuaian Hasil Akhir Sistem

	Kebutuhan	Hasil	Bukti
1	Informasi <i>up to date</i> mengenai tren penjualan, stok terkini serta <i>history</i> transaksi dan <i>supplier</i> yang diperoleh tanpa melakukan rekap	Sukses. Semua laporan yang dihasilkan sistem didapatkan tanpa melalui rekap, melainkan hanya perlu memasukkan <i>parameter</i> laporan sesuai yang dibutuhkan.	Gambar 4.65, hingga 4.75 dan Lampiran 1. Hasil Laporan Perbandingan Stok Terkini dan Tren Penjualan
2	Sistem yang <i>portable</i> , dapat diakses kapanpun dan dimanapun dengan koneksi internet melalui <i>web browser</i> , termasuk <i>mobile browser</i> .	Sukses. Sistem dapat diakses kapanpun dan dimanapun dengan koneksi internet dan melalui <i>web browser</i> termasuk <i>mobile browser</i> . <i>Domain</i> internet yang digunakan adalah www.juwall.com	Lampiran 2. Uji Coba sistem pada <i>web browser</i> termasuk <i>mobile browser</i> .

3	Biaya <i>maintenance</i> , perangkat teknologi serta SDM pengelola sistem yang murah.	Sukses. Pengguna tidak perlu membeli infrastruktur sistem serta merekrut SDM untuk melakukan <i>maintenance</i> sistem. Pengguna cukup membayar biaya bulanan sesuai dengan kebutuhan.	Lampiran 3. Daftar harga layanan sistem informasi penjualan berbasis <i>cloud</i> .
4	Jaminan keamanan data dan ketersediaan sistem	Sukses. Sistem akan menggunakan infrastruktur yang tersimpan di <i>data center</i> Microsoft dengan layanan <i>cloud</i> Windows Azure yang memiliki SLA sebagai jaminan keamanan data dan ketersediaan sistem.	Lampiran 4. <i>Service Level Agreement</i> Windows Azure

4.6 *Service Level Agreement (SLA)*

Setelah tahap pengujian selesai dilakukan, langkah selanjutnya adalah penyusunan *Service Level Agreement (SLA)* agar layanan sistem informasi penjualan berbasis *cloud* yang dibangun dapat memberikan jaminan bagi pengguna serta memiliki tanggung jawab yang jelas apabila pihak penyedia layanan tidak dapat memenuhi jaminan tersebut. *SLA* yang dibuat disesuaikan dengan *cloud provider* yang digunakan, yaitu *Windows Azure*. *SLA* dari *Windows Azure* dapat dilihat pada lampiran 5. *Service Level Agreement* yang dibuat dapat dilihat pada tabel 4.14.

Tabel 4. 14 *Service Level Agreement*

Jaminan		Konsekuensi
1	Ketersediaan sistem kurang dari 99.9%	Potongan biaya layanan sebesar 10%
2	Ketersediaan sistem kurang dari 99%	Potongan biaya layanan sebesar 25%

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan uji coba dan evaluasi terhadap sistem informasi penjualan berbasis *cloud*, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil uji coba didapatkan bahwa sistem dapat menghasilkan *output* berupa laporan tren penjualan, stok terkini, *history* transaksi *supplier* dan pelanggan sesuai dengan kebutuhan manajemen serta dapat diakses kapanpun dan dimanapun melalui jaringan internet. Laporan tren penjualan digunakan untuk mengetahui *lifetime* sebuah produk dan menentukan jumlah pemesanan kepada *supplier* pada periode berikutnya. Laporan stok terkini digunakan untuk memantau kondisi stok terkini yang dimiliki sehingga tidak terjadi kehabisan stok yang dapat merugikan toko. Laporan *history* transaksi dari pelanggan dan *supplier* digunakan untuk mengetahui siapa yang telah melakukan banyak transaksi. Informasi tersebut diperlukan sebagai *input* bagi strategi marketing dalam menjaga relasi dengan pelanggan atau *supplier* utama. Akses yang dapat dilakukan kapanpun dan dimanapun membantu pihak manajemen untuk mendapatkan informasi penjualan tanpa harus datang ke toko.
2. Berdasarkan hasil uji coba didapatkan bahwa sistem ini dapat digunakan sebagai media promosi barang di jejaring sosial Facebook berdasarkan data terkini. Promosi dapat dilakukan untuk masing-masing barang, barang terlaris atau diskon terbesar. DenganW promosi melalui jejaring sosial

berdasarkan data terkini, toko diharapkan bisa menjangkau pasar yang lebih luas.

3. Berdasarkan hasil uji coba didapatkan bahwa sistem ini menyediakan 3 metode HPP yang dapat dipilih oleh pengguna, yaitu *Average Costing*, *First In First Out* dan *Last In First Out*. Berdasarkan hasil uji coba pada bab 4, perhitungan harga dan stok barang setelah transaksi penjualan dan penerimaan sesuai dengan metode yang dipilih.

5.2 Saran

Adapun beberapa saran yang dapat diberikan kepada peneliti berikutnya apabila ingin mengembangkan sistem yang telah dibuat ini agar menjadi lebih baik adalah sebagai berikut:

1. Aplikasi mendatang sebaiknya dapat menghasilkan laporan-laporan akuntansi seperti laporan laba rugi dan neraca.
2. Aplikasi mendatang sebaiknya memiliki *Web API (Application Programming Interface)* yang dapat diakses dari sistem eksternal dengan mekanisme keamanan tertentu sehingga dapat diintegrasikan dengan sistem informasi yang sudah ada.
3. Aplikasi mendatang sebaiknya dapat diintegrasikan dengan *e-commerce* sehingga dapat mencatat transaksi-transaksi yang terjadi di *e-commerce*.
4. Aplikasi mendatang sebaiknya dapat menangani kebutuhan UKM yang memiliki *multi* gudang atau *multi* stok.

DAFTAR PUSTAKA

- Alamsyah, Fahrizal. 2009. *e-Business : Membangun Bisnis Hosting dan Domain*. Jakarta: PT Elex Media Komputindo
- Al Fatta, Hanif. 2007. *Analisis & Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: Penerbit Andi
- Arifin, Johar. 2007. *Aplikasi Excel Untuk Perencanaan Bisnis (Business Plan)*. Jakarta: PT Elex Media Komputindo
- Ayodya, Wulan. 2010. *Cara Awal Buka Usaha Roti dan Kue*. Jakarta: PT Elex Media Komputindo
- Azis, M. Farid. 2005. *Object Oriented Programming dengan PHP 5*. Jakarta: PT Elex Media Komputindo
- Kurniawan, Erick, 2010. *Cepat Mahir ASP.NET 3.5 Untuk Aplikasi Web Interaktif*. Yogyakarta: Penerbit Andi
- Hall, James. A. 2009. *Accounting Information System*. Jakarta: Salemba Empat
- Hendri. 2007. *Object Oriented Analysis oleh Hendri*. 20 Juni 2007. URL: <http://rpl07.wordpress.com/2007/06/20/object-oriented-analysis-oleh-hendri-5105-100-162/>
- Iskandar, Ridwan. 2009. *Pengertian Penjualan*. 19 Mei 2009. URL: <http://ridwaniskandar.files.wordpress.com/2009/05/91-pengertian-penjualan.pdf>
- Karmatoli, Puby. 2012. *Sistem Informasi Penjualan* 20 Januari 2012. URL: <http://12puby.blogspot.com/2012/01/sistem-informasi-penjualan.html>
- Kusmayadi, Dani. 2011. *Lemahnya Informasi Pasar Menjadi Permasalahan UKM di Karawang*. 17 April 2011. URL: <http://karawang-in-bisnis.blogspot.com/2011/04/lemahnya-informasi-pasar-menjadi.html>
- Kusrini, Andri Koniyo. 2007. *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft SQL Server*. Yogyakarta: Penerbit Andi
- Marconi. 2011. *Teori Akuntansi - Sistem Informasi Penjualan* 16 Juli 2011. URL: <http://marconiekonomi.blogspot.com/2011/07/sistem-informasi-penjualan.html>

- Rohmat, 2011. *UKM Bertambah Bukti Kegagalan Pemerintah*. 25 Juli 2011. URL: <http://economy.okezone.com/read/2011/07/25/320/483763/ukm-bertambah-bukti-kegagalan-pemerintah>
- Sugiono, Arief, Yanuar Nanok Soenarno, Synthia Madya Kusumawati. 2009. *Akuntansi dan Pelaporan Keuangan untuk Bisnis Skala Kecil dan Menengah*. Jakarta: Grasindo
- Sulistiyastuti, Dyah Ratih. 2004, *Dinamika Usaha Kecil dan Menengah (UKM) : Analisis Konsentrasi Regional UKM di Indonesia 1999-2001*. Jurnal Ekonomi Pembangunan, 9 (2), 149
- Syaikhu, Akhmad. 2010. *Komputasi Awan (Cloud Computing) Perpustakaan Pertanian*. Jurnal Pustakawan Indonesia, 10 (1), 1-4
- Tambunan, Tulus. 2002. *Usaha Kecil dan Menengah di Indonesia: Beberapa Isu Penting*. Jakarta: Salemba Empat
- Yoga, Paulus. 2011. *Kontribusi Koperasi dan UKM Terhadap PDB Capai 56.6%*. 1 April 2011. URL: <http://www.infobanknews.com/2011/04/kontribusi-koperasi-dan-ukm-terhadap-pdb-capai-566/>
- Yuhefizard. 2008. *Database Management Menggunakan Microsoft Access 2003*. PT Elex Media Komputindo



UNIVERSITAS
Dinamika