BAB I

PENDAHULUAN

1. 1. Latar Belakang

Perguruan tinggi adalah salah satu lembaga pendidikan yang mempersiapkan mahasiswa untuk dapat terjun kedalam masyarakat, khususnya pada disiplin ilmu yang telah dipelajari selama mengikuti perkuliahan. Perguruan tinggi tidak terlepas dari masyarakat atau institusi-institusi yang ada, hal ini sesuai dengan Tri Dharma Perguruan Tinggi yaitu "Pengabdian Masyarakat". Untuk mewujudkan cita-cita sebuah Perguruan Tinggi, yang berorientasi pada penciptaan sarjana yang berkualitas, maka perlu adanya Praktik Kerja Lapangan (PKL).

Praktik Kerja Lapangan (PKL) adalah sebuah kegiatan wajib bagi mahasiswa - mahasiswi guna menerapkan teori-teori yang diterima saat proses pembelajaran di bangku kuliah kedalam dunia kerja. Hubungan antara teori dan praktik merupakan hal penting untuk membandingkan teori dengan keadaan sebenarnya di lapangan. Selain itu, dibutuhkan juga cara berpikir praktis dan logis untuk menghadapi dunia usaha. Untuk itu, STIKI Malang mewajibkan setiap mahasiswanya melaksanakan Praktik Kerja Lapangan di instansi pemerintah atau perusahaan swasta sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan sarjana.

Sebagai syarat kelulusan sarjana dari STIKI Malang, penulis memilih Elecomp Software House sebagai tempat untuk melaksanakan PKL dengan tujuan untuk mengkaji secara terperinci serta mendalam tentang teori dan praktik mengenai perkuliahan dengan harapan pihak Elecomp Software House dapat memberikan orientasi mengenai praktik serta motivasi kerja di lapangan.

Salah satu proyek yang diberikan oleh Elecomp Software House yaitu implementasi sistem informasi penjualan berbasis *desktop* dengan menggunakan bahasa Java. Dalam melakukan implementasi, dibagi menjadi beberapa bagian agar mempermudah fokus *developer* dalam proses pengembangan, untuk itu dibagi menjadi dua tahap yaitu *interface* dan pengkodingan sistem.

1. 2. Rumusan Masalah

Adapun rumusan masalah yang dapat disimpulkan adalah:

1. Bagaimana cara mengimplementasikan ilmu yang didapat selama perkuliahan sesuai dengan masalah dan tugas yang dihadapi?
2. Bagaimana cara mengembangkan sistem informasi penjualan pada proyek yang diberikan oleh Elecomp Software House?

1. 3. Ruang Lingkup

Dalam penyusunan Tugas Praktik Kerja Lapangan, pembahasan masalah mengenai pengembangan sistem informasi penjualan akan dibatasi pada hal-hal berikut ini:

1. Pengerjaan modul transaksi meliputi Toko Transaksi, Return Transaksi, dan Laporan Toko
2. Modul master meliputi Pegawai, Supplier, Salesman, dan Customer.

1. 4. Tujuan dan Manfaat

1. 4. 1. Tujuan

Tujuan yang dicapai dalam penelitian adalah:

1. Mempelajari serta berpartisipasi langsung dalam dunia kerja
2. Mengembangkan wawasan dan keterampilan dalam dunia kerja
3. Membantu subjek penelitian dalam menyelesaikan proyek pihak ketiga.

1. 4. 2. Manfaat

Manfaat yang kami per oleh dari Praktik Kerja Lapangan adalah:

1. Melatih tanggung jawab serta disiplin dalam dunia kerja
2. Memperoleh pengalaman kerja
3. Meningkatkan kemampuan dalam melakukan *problem solving*
4. Meningkatkan kerja sama antara subjek penelitian dengan universitas.

1. 5. Metodologi Penelitian

1. 5. 1. Tempat dan Waktu Penelitian

Praktik Kerja Lapangan dilakukan di Elecomp Software House yang berlokasi di Jalan Danau Paniai Utara 1 C7G5 Malang terhitung mulai tanggal 12 Maret 2018 hingga 12 Mei 2018. Adapun jadwal kerja setiap hari adalah sebagai berikut:

Nomor	Hari	Jam Masuk	Jam Pulang
1	Senin	13.00	17.00
2	Rabu	13.00	17.00
3	Kamis	11.00	17.00
4	Jumat	08.00	17.00
5	Sabtu	08.00	13.00

Tabel 1.1 Tabel Jadwal Kerja PKL

1. 6. Sistematika Penulisan

BAB I PENDAHULUAN

Berisi tentang Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian, Metodologi Penelitian.

BAB II TINJAUAN UMUM

Memuat gambaran umum perusahaan mengenai sejarah, visi dan misi, serta struktur organisasi dan deskripsi layanan perusahaan.

BAB III URAIAN KEGIATAN

Memuat tabel dan uraian kegiatan selama kegiatan praktik kerja lapangan pada Elecomp Software House.

BAB IV PEMBAHASAN MASALAH

Memuat tentang permasalahan yang terjadi pada perusahaan.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari Praktik kerja Lapangan serta saran.

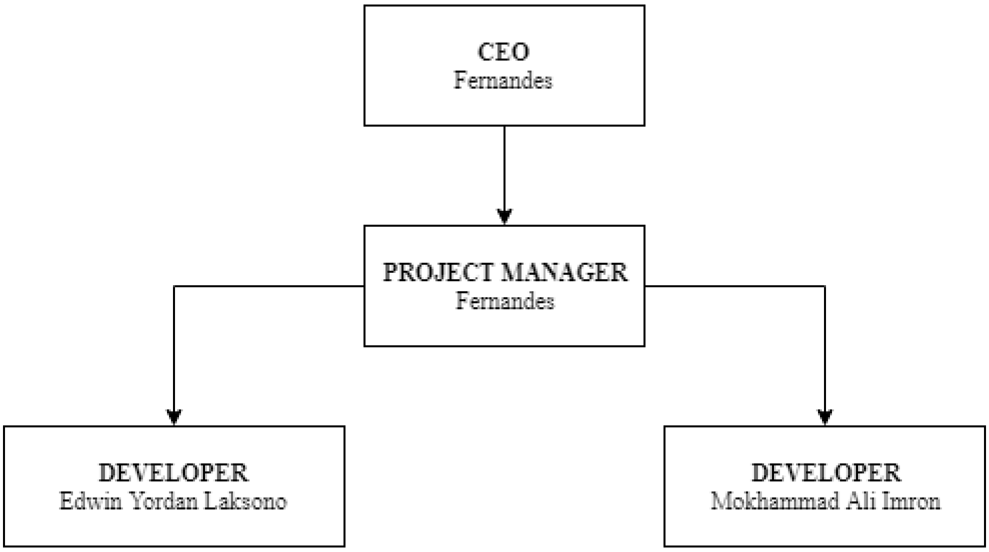
BAB II

TINJAUAN UMUM

2. 1. Deskripsi Perusahaan

Elecomp Software House berlokasi di Jalan Danau Paniai Utara 1 C7G5, Madyopuro, Kedungkandang, Kota Malang, Jawa Timur . Elecomp Software House merupakan perusahaan jasa yang menawarkan pembuatan *software* berbasis *desktop*, *mobile* maupun *web* serta animasi, *game* android, dan *game* komputer.

2. 2. Struktur Organisasi



Gambar 2.1 Struktur Organisasi Elecomp Software House

2. 3. Profil Perusahaan

Visi :

“Menjadi salah satu *software house* yang dapat diperhitungkan di komunitas internasional dengan tidak mengabaikan nilai sosial”.

Misi :

1. Mengembangkan industri produk IT secara kompetitif dan dapat diterima oleh semua orang

2. Memberikan layanan terbaik kepada klien
3. Mengembangkan kerja sama dengan sesama perusahaan yang saling menguntungkan
4. Mengembangkan produk IT yang inovatif
5. Dapat secara aktif di kegiatan sosial dan berkontribusi di bidang IT.

2. 4. *Job Description Staff*

Berikut penjelasan serta tugas yang dilakukan berdasarkan struktur organisasi:

1. *Chief Executive Officer*

   Merupakan pemilik perusahaan yang bertanggung jawab atas izin perusahaan sekaligus pimpinan tertinggi pada perusahaan yang memiliki wewenang penuh terhadap operasional perusahaan.

   Adapun tugas dan wewenang CEO adalah:

   a) Penanggung jawab seluruh kegiatan yang dilakukan perusahaan
   b) Pemberi modal bagi perusahaan
   c) Melaksanakan manajemen transformasi dalam rangka terwujudnya tata nilai berkelanjutan pada perusahaan
   d) Menetapkan langkah-langkah strategi dalam mengembangkan perusahaan
   e) Menyukseskan pembangunan sarana dan prasarana teknologi informasi secara efektif.

2. Project Manager

   Merupakan orang yang ditunjuk untuk menggerakkan organisasi proyek dan memimpinnya dalam objektif proyek.

Adapun tugas dan wewenang Project Manager:

a) Mengidentifikasi dan menyelesaikan potensi masalah yang akan timbul agar dapat diantisipasi secara dini
b) Melakukan koordinasi terhadap staf
c) Menyiapkan rencana kerja operasi proyek meliputi aspek teknis, waktu, administrasi, dan keuangan proyek
d) Melaksanakan dan mengontrol operasional proyek sehingga operasi proyek dapat berjalan sesuai dengan rencana (*on-track*)
e) Mengkomunikasikan laporan kemajuan pekerjaan dalam bentuk lisan dan tertulis
f) Mengontrol proyek yang ditangani (harus sesuai dengan anggaran, spesifikasi, dan waktu).

3. Developer

Merupakan orang yang bertugas melakukan implementasi berdasarkan rancangan yang telah dibuat.

Adapun tugas Developer adalah:

a) Membangun atau mengembangkan *software* terutama pada tahap *construction* dengan melakukan koding menggunakan Bahasa Pemrograman yang ditentukan
b) Mengimplementasikan *requirement* dan desain proses bisnis ke dalam bentuk program dengan menggunakan algoritma atau logika dan Bahasa Pemrograman tertentu
c) Melakukan testing terhadap *software* apabila diperlukan.

BAB III

URAIAN KEGIATAN

3. 1. Tabel Kegiatan Praktik Kerja Lapangan

Dalam menunjang pembuatan laporan ini, maka setiap kegiatan yang dilakukan pada saat Praktik Kerja Lapangan di Elecomp Software House dimuat pada poin 3.2. Berikut rincian kegiatan selama dua bulan Praktik Kerja Lapangan:

Objek Penelitian	: Elecomp Software House
Alamat	: Jalan Danau Paniai Utara 1 C7G5 Malang
Pelaksana	:151111068 / I Made Bisma Gargita 151111098 / Windy 151111104 / Mitsaq Enigma al Afghan

3. 2. Uraian Pelaksanaan Praktik Kerja Lapangan

Tabel 3.1 Tabel Uraian Kegiatan PKL Minggu ke-1

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 12/3/ 2018	I Made Bisma Gargita	Percobaan CRUD Code Igniter	Proses
	Windy	Percobaan CRUD Code Igniter	Proses
	Mitsaq Enigma al Afghan	Percobaan CRUD Code Igniter	Proses
Rabu 14/3/ 2018	I Made Bisma Gargita	Proyek CRUD	Tercapai
	Windy	Proyek CRUD	Tercapai
	Mitsaq Enigma al Afghan	Proyek CRUD	Tercapai
Kamis 15/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses
Jumat 16/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses

Sabtu 17/3/ 2018	I Made Bisma Gargita	Libur	-
	Windy	Libur	-
	Mitsaq Enigma al Afghan	Libur	-

Tabel 3.2 Tabel Uraian Kegiatan PKL Minggu ke-2

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 19/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses
Rabu 21/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses
Kamis 22/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses
Jumat 23/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses
Sabtu 24/3/ 2018	I Made Bisma Gargita	Proyek Web Resort	Proses
	Windy	Proyek Web Resort	Proses
	Mitsaq Enigma al Afghan	Proyek Web Resort	Proses

Tabel 3.3 Tabel Uraian Kegiatan PKL Minggu ke-3

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 26/3/ 2018	I Made Bisma Gargita	Finishing Web Resort	Tercapai
	Windy	Finishing Web Resort	Tercapai
	Mitsaq Enigma al Afghan	Finishing Web Resort	Tercapai

Rabu 28/3/ 2018	I Made Bisma Gargita	Proyek Web Elecomp Point Of Sales	Proses
	Windy	Proyek Web Elecomp Point Of Sales	Proses
	Mitsaq Enigma al Afghan	Proyek Web Elecomp Point Of Sales	Proses
Kamis 29/3/ 2018	I Made Bisma Gargita	Proyek Web Elecomp Point Of Sales	Proses
	Windy	Proyek Web Elecomp Point Of Sales	Proses
	Mitsaq Enigma al Afghan	Proyek Web Elecomp Point Of Sales	Proses
Jumat 30/3/ 2018	I Made Bisma Gargita	Libur	-
	Windy	Libur	-
	Mitsaq Enigma al Afghan	Libur	-
Sabtu 31/3/ 2018	I Made Bisma Gargita	Proyek Web Elecomp Point Of Sales	Proses
	Windy	Proyek Web Elecomp Point Of Sales	Proses
	Mitsaq Enigma al Afghan	Proyek Web Elecomp Point Of Sales	Proses

Tabel 3.4 Tabel Uraian Kegiatan PKL Minggu ke-4

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 2/4/ 2018	I Made Bisma Gargita	Proyek Web Elecomp Point Of Sales	Tidak Tercapai
	Windy	Proyek Web Elecomp Point Of Sales	Tidak Tercapai
	Mitsaq Enigma al Afghan	Proyek Web Elecomp Point Of Sales	Tidak Tercapai
Rabu 4/4/ 2018	I Made Bisma Gargita	Proyek Web WGFree	Proses
	Windy	Proyek Web WGFree	Proses
	Mitsaq Enigma al Afghan	Proyek Web WGFree	Proses
Kamis 5/4/ 2018	I Made Bisma Gargita	Proyek Web WGFree	Proses
	Windy	Proyek Web WGFree	Proses
	Mitsaq Enigma al Afghan	Proyek Web WGFree	Proses
Jumat 6/4/ 2018	I Made Bisma Gargita	Proyek Web WGFree	Proses
	Windy	Proyek Web Tea Shop	Proses
	Mitsaq Enigma al Afghan	Proyek Web WGFree	Proses

Sabtu 7/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Proyek Web WGFree Proyek Web Tea Shop Proyek Web WGFree	Proses Proses Proses

Tabel 3.5 Tabel Uraian Kegiatan PKL Minggu ke-5

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 9/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Finishing Proyek Web WGFree Proyek Web Tea Shop Finishing Proyek Web WGFree	Tercapai Proses Tercapai
Rabu 11/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Proyek Web Tea Shop Proyek Web Tea Shop Proyek Web Tea Shop	Proses Proses Proses
Kamis 12/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Proyek Web Tea Shop Proyek Web Tea Shop Proyek Web Tea Shop	Proses Proses Proses
Jumat 13/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Libur Libur Libur	- - -
Sabtu 14/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Proyek Web Tea Shop Proyek Web Tea Shop Proyek Desktop Toko Cat	Tercapai Tercapai Proses

Tabel 3.6 Tabel Uraian Kegiatan PKL Minggu ke-6

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 16/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Master Supplier Toko Cat – Master Customer Toko Cat – Master Pegawai	Proses Proses Proses

Rabu 18/4/2018	I Made Bisma Gargita	Toko Cat – Master Supplier	Proses
	Windy	Toko Cat – Master Customer	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Pegawai	Proses
Kamis 19/4/2018	I Made Bisma Gargita	Toko Cat – Master Supplier	Tidak Tercapai
	Windy	Toko Cat – Master Customer	Tidak Tercapai
	Mitsaq Enigma al Afghan	Toko Cat – Master Pegawai	Tercapai
Jumat 20/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Customer	Proses
Sabtu 21/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Customer	Proses

Tabel 3.7 Tabel Uraian Kegiatan PKL Minggu ke-7

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 23/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Customer	Tercapai
Rabu 25/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Supplier	Proses
Kamis 26/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Supplier	Tercapai
Jumat 27/4/2018	I Made Bisma Gargita	Toko Cat – Toko Return	Proses
	Windy	Toko Cat – Toko Transaksi	Proses
	Mitsaq Enigma al Afghan	Toko Cat – Master Salesman	Proses

Sabtu 28/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Master Salesman	Proses Proses Proses

Tabel 3.8 Tabel Uraian Kegiatan PKL Minggu ke-8

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 30/4/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Master Salesman	Proses Proses Tercapai
Rabu 2/5/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat - Toko Transaksi	Proses Proses Proses
Kamis 3/5/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Toko Transaksi	Proses Proses Proses
Jumat 4/5/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Toko Transaksi	Proses Proses Proses
Sabtu 5/5/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Toko Transaksi	Proses Proses Proses

Tabel 3.9 Tabel Uraian Kegiatan PKL Minggu ke-9

Hari	Mahasiswa	Pekerjaan	Keterangan
Senin 7/5/ 2018	I Made Bisma Gargita Windy Mitsaq Enigma al Afghan	Toko Cat – Toko Return Toko Cat – Toko Transaksi Toko Cat – Toko Transaksi	Proses Proses Proses

Rabu	I Made Bisma Gargita	Toko Cat – Toko Return	Tercapai
9/5/	Windy	Toko Cat – Toko Transaksi	Tercapai
2018	Mitsaq Enigma al Afghan	Toko Cat – Master Salesman	Tercapai
Kamis	I Made Bisma Gargita	Libur	-
10/5/	Windy	Libur	-
2018	Mitsaq Enigma al Afghan	Libur	-
Jumat	I Made Bisma Gargita	Toko Cat – Laporan Toko	Proses
11/5/	Windy	Toko Cat – Laporan Toko	Proses
2018	Mitsaq Enigma al Afghan	Toko Cat – Laporan Toko	Proses
Sabtu	I Made Bisma Gargita	Toko Cat – Laporan Toko	Tercapai
12/5/	Windy	Toko Cat – Laporan Toko	Tercapai
2018	Mitsaq Enigma al Afghan	Toko Cat – Laporan Toko	Tercapai

BAB IV

PEMBAHASAN MASALAH

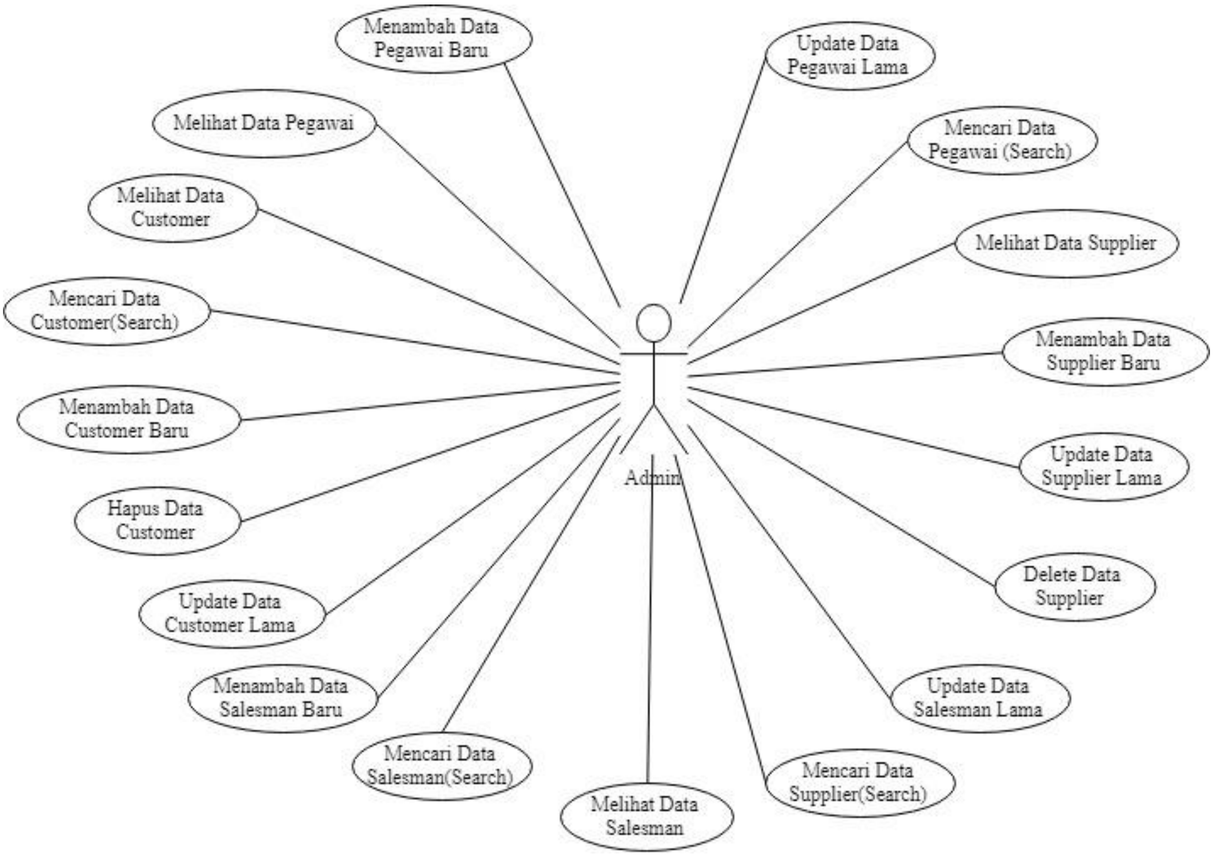
4.1. Permasalahan

Proyek yang diberikan oleh Elecomp Software House mengenai sistem informasi penjualan toko cat berbasis desktop dengan menggunakan bahasa Java. Seputar permasalahan yang dikerjakan yaitu pengembangan dalam fitur penjualan di modul toko dan master. Adapun modul toko yang dikerjakan meliputi toko transaksi, toko return penjualan, dan toko laporan serta modul master meliputi master pegawai, master supplier, master salesman, dan master customer.

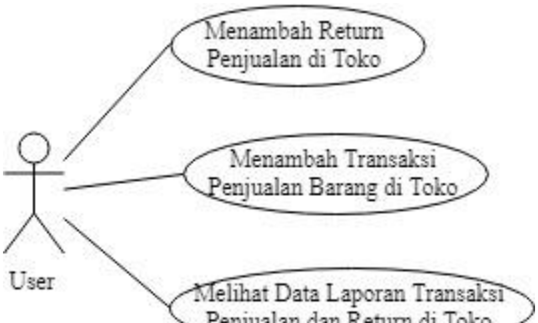
4.2. Usulan Pemecahan Masalah

4.3. Rancangan Sistem

4.3.1. Usecase Diagram

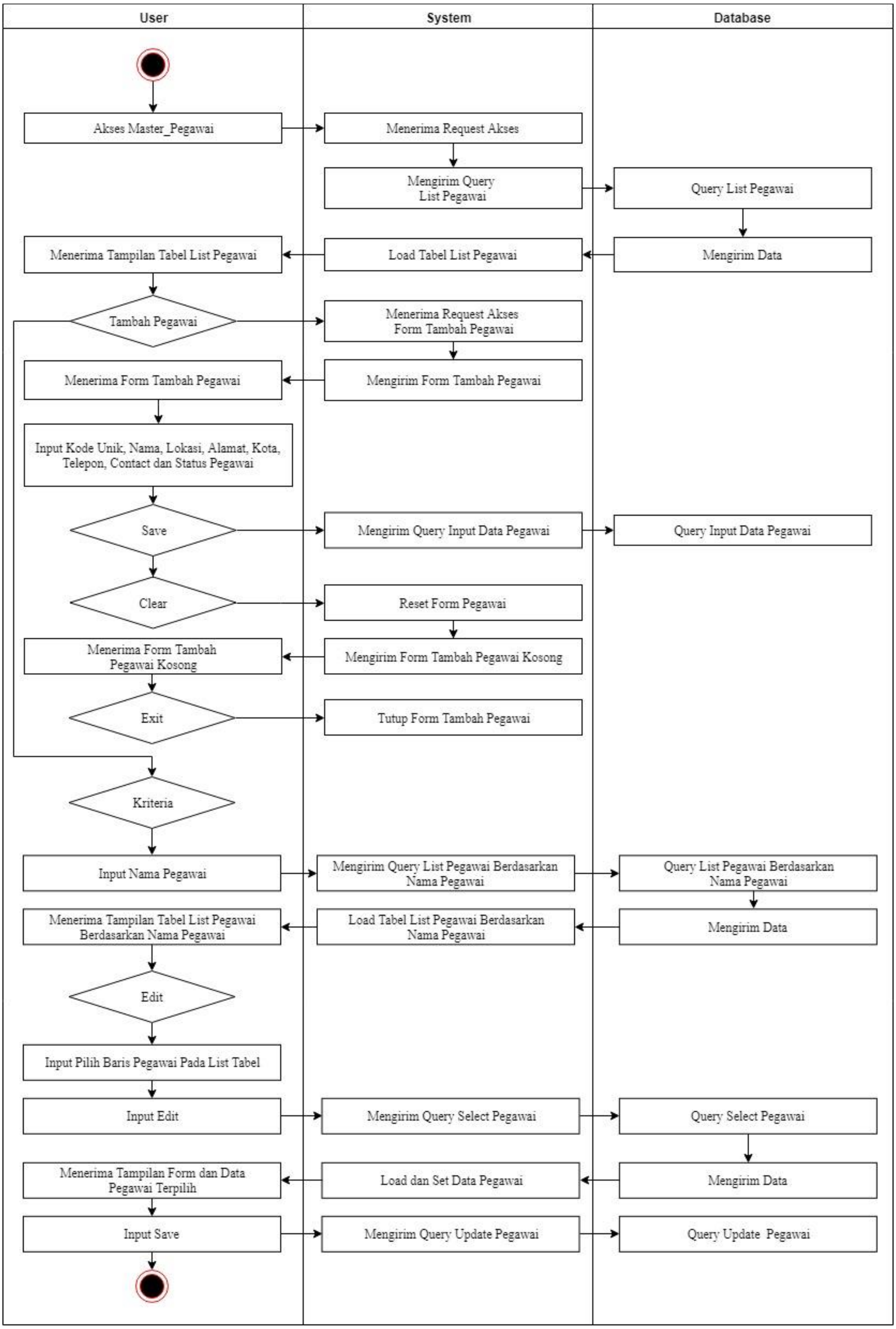

Menambah Data Pegawai Baru
Update Data Pegawai Lama
Melihat Data Pegawai
Mencari Data Pegawai (Search)
Melihat Data Customer
Melihat Data Supplier
Mencari Data Customer(Search)
Menambah Data Supplier Baru
Menambah Data Customer Baru
Update Data Supplier Lama
Admin
Hapus Data Customer
Delete Data Supplier
Update Data Customer Lama
Menambah Data Salesman Baru
Update Data Salesman Lama
Mencari Data Salesman(Search)
Mencari Data Supplier(Search)
Melihat Data Salesman

Gambar 4.1 Use Case Diagram Administrator

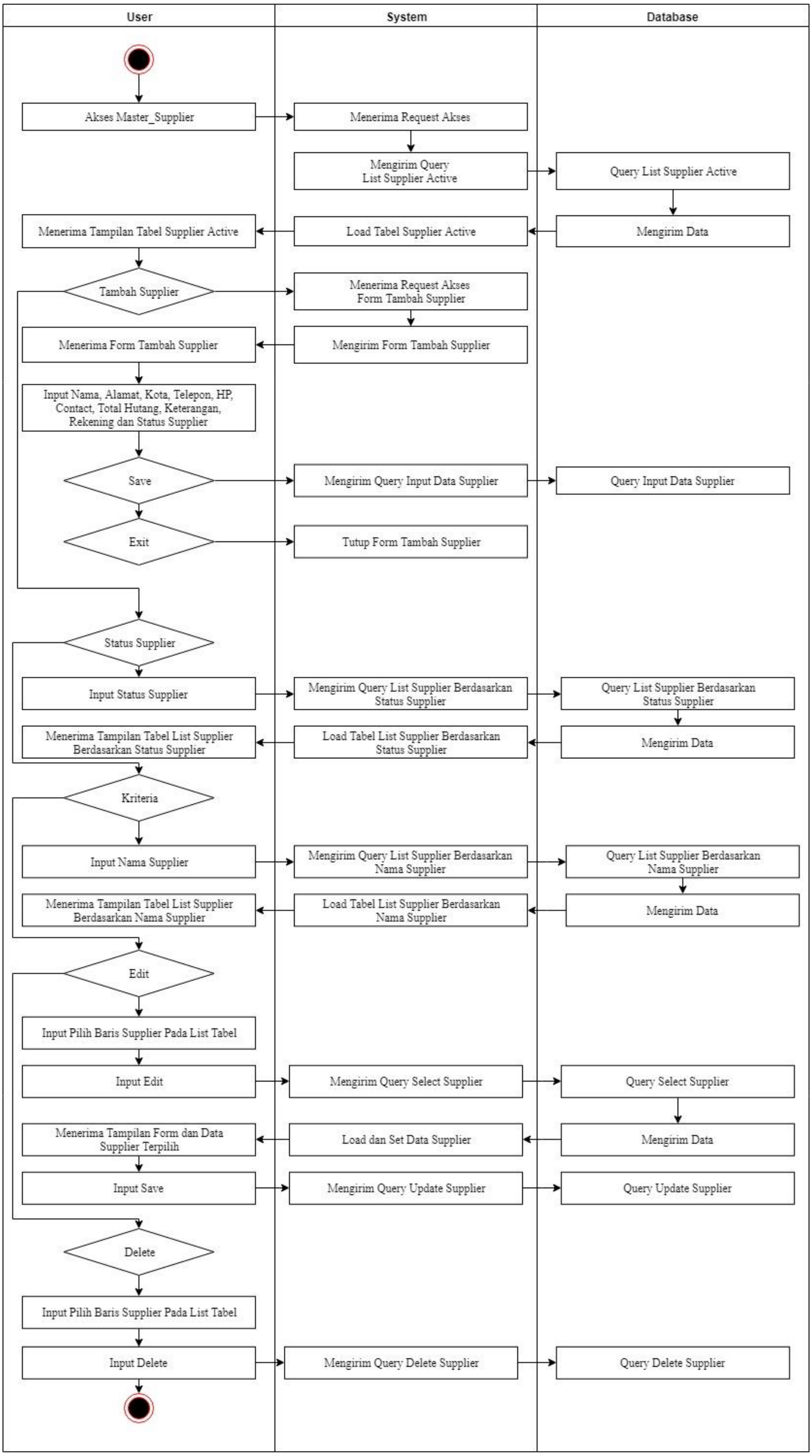

Menambah Return Penjualan di Toko
Menambah Transaksi Penjualan Barang di Toko
User
Melihat Data Laporan Transaksi Penjualan dan Return di Toko

Gambar 4.2 Use Case Diagram User

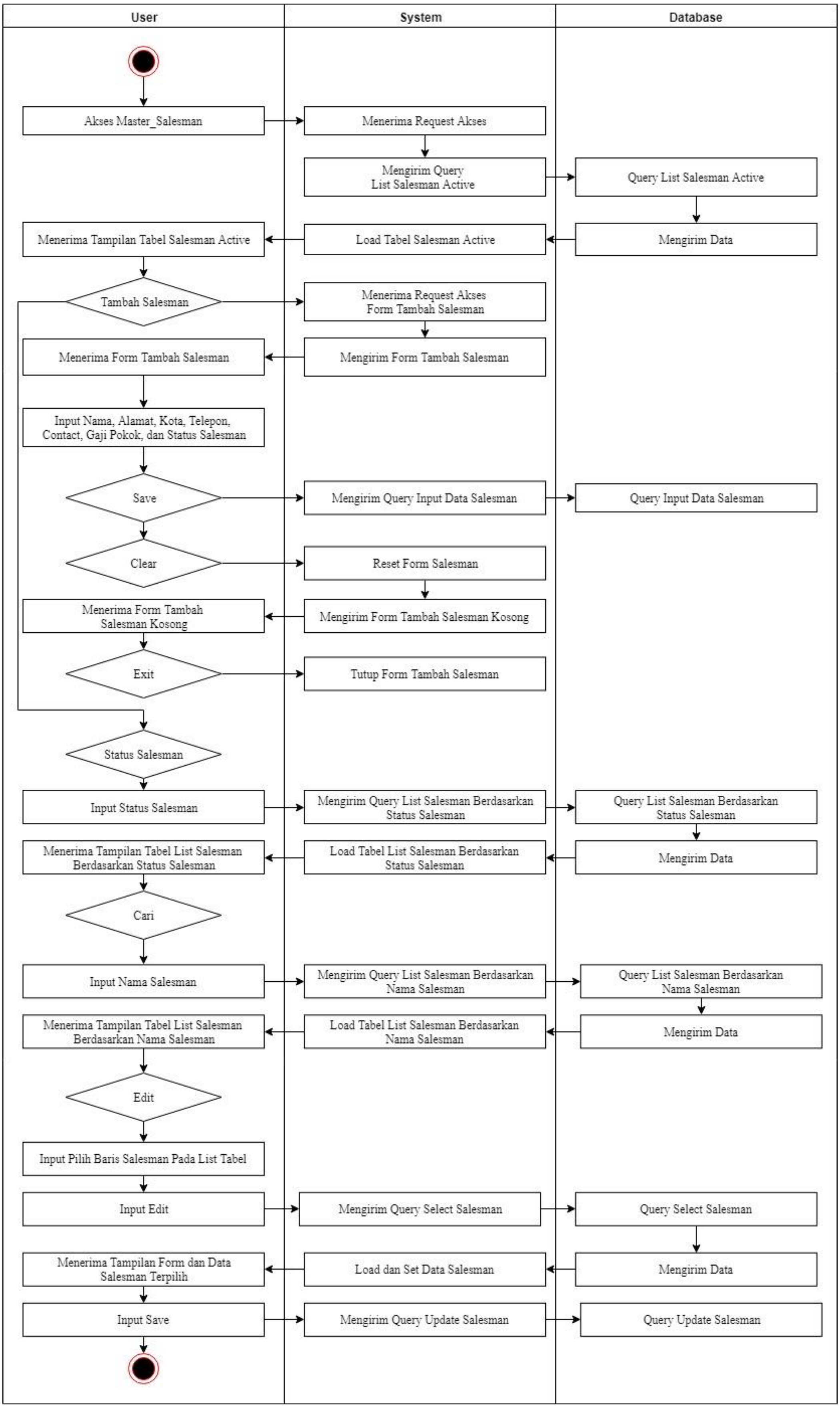
4.3.2. Activity Diagram


User
System
Database
Akses Master_Pegawai
Menerima Request Akses
Mengirim Query List Pegawai
Query List Pegawai
Menerima Tampilan Tabel List Pegawai
Load Tabel List Pegawai
Mengirim Data
Tambah Pegawai
Menerima Request Akses Form Tambah Pegawai
Menerima Form Tambah Pegawai
Mengirim Form Tambah Pegawai
Input Kode Unik, Nama, Lokasi, Alamat, Kota, Telepon, Contact dan Status Pegawai
Save
Mengirim Query Input Data Pegawai
Query Input Data Pegawai
Clear
Reset Form Pegawai
Menerima Form Tambah Pegawai Kosong
Mengirim Form Tambah Pegawai Kosong
Exit
Tutup Form Tambah Pegawai
Kriteria
Input Nama Pegawai
Mengirim Query List Pegawai Berdasarkan Nama Pegawai
Query List Pegawai Berdasarkan Nama Pegawai
Menerima Tampilan Tabel List Pegawai Berdasarkan Nama Pegawai
Load Tabel List Pegawai Berdasarkan Nama Pegawai
Mengirim Data
Edit
Input Pilih Baris Pegawai Pada List Tabel
Input Edit
Mengirim Query Select Pegawai
Query Select Pegawai
Menerima Tampilan Form dan Data Pegawai Terpilih
Load dan Set Data Pegawai
Mengirim Data
Input Save
Mengirim Query Update Pegawai
Query Update Pegawai

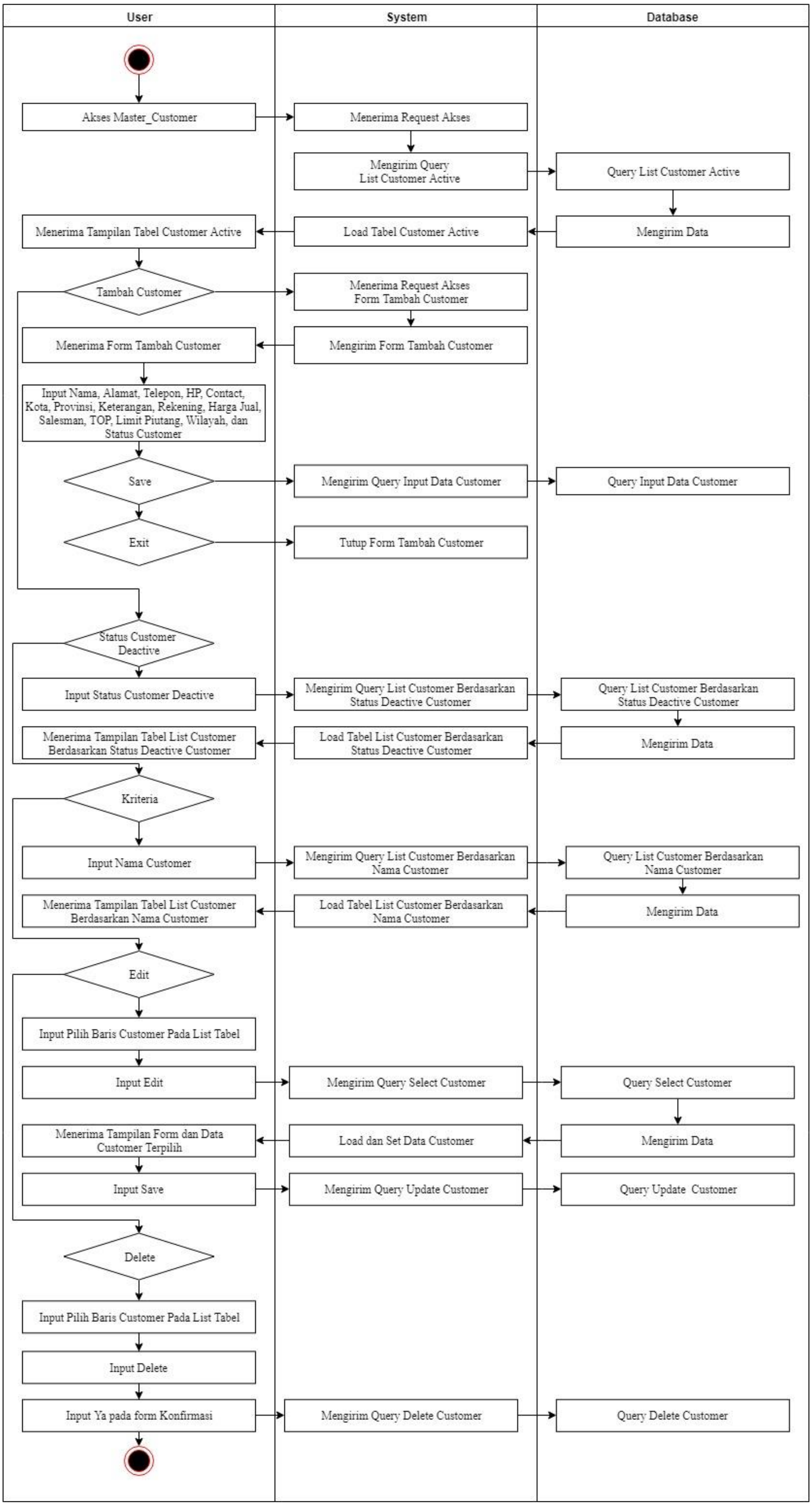
Gambar 4.3 Activity Diagram Master Pegawai


User
System
Database
Akses Master_Supplier
Menerima Request Akses
Mengirim Query List Supplier Active
Query List Supplier Active
Menerima Tampilan Tabel Supplier Active
Load Tabel Supplier Active
Mengirim Data
Tambah Supplier
Menerima Request Akses Form Tambah Supplier
Menerima Form Tambah Supplier
Mengirim Form Tambah Supplier
Input Nama, Alamat, Kota, Telepon, HP, Contact, Total Hutang, Keterangan, Rekening dan Status Supplier
Save
Mengirim Query Input Data Supplier
Query Input Data Supplier
Exit
Tutup Form Tambah Supplier
Status Supplier
Input Status Supplier
Mengirim Query List Supplier Berdasarkan Status Supplier
Query List Supplier Berdasarkan Status Supplier
Menerima Tampilan Tabel List Supplier Berdasarkan Status Supplier
Load Tabel List Supplier Berdasarkan Status Supplier
Mengirim Data
Kriteria
Input Nama Supplier
Mengirim Query List Supplier Berdasarkan Nama Supplier
Query List Supplier Berdasarkan Nama Supplier
Menerima Tampilan Tabel List Supplier Berdasarkan Nama Supplier
Load Tabel List Supplier Berdasarkan Nama Supplier
Mengirim Data
Edit
Input Pilih Baris Supplier Pada List Tabel
Input Edit
Mengirim Query Select Supplier
Query Select Supplier
Menerima Tampilan Form dan Data Supplier Terpilih
Load dan Set Data Supplier
Mengirim Data
Input Save
Mengirim Query Update Supplier
Query Update Supplier
Delete
Input Pilih Baris Supplier Pada List Tabel
Input Delete
Mengirim Query Delete Supplier
Query Delete Supplier

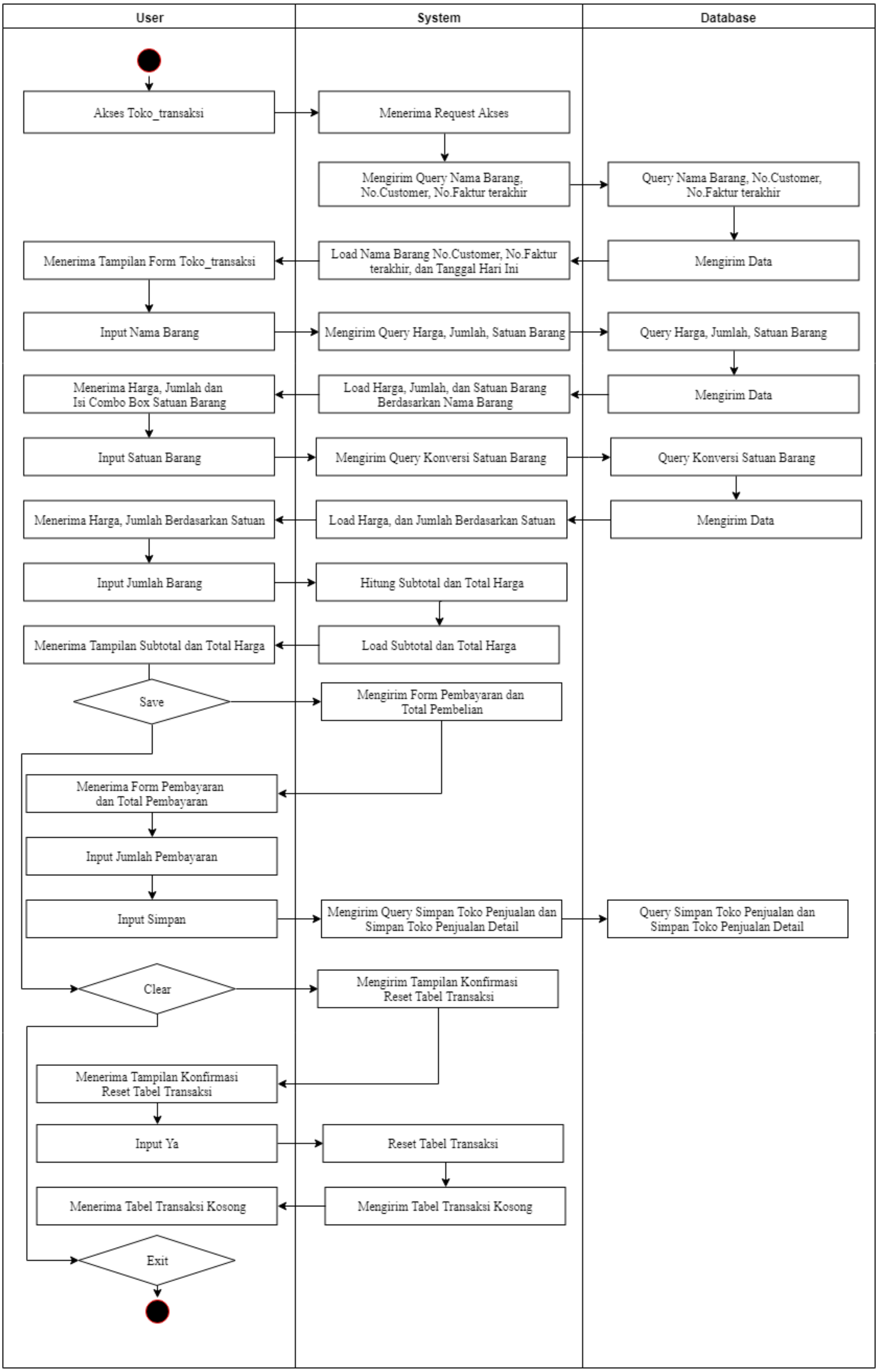
Gambar 4.4 Activity Diagram Master Supplier


User
System
Database
Akses Master_Salesman
Menerima Request Akses
Mengirim Query List Salesman Active
Query List Salesman Active
Menerima Tampilan Tabel Salesman Active
Load Tabel Salesman Active
Mengirim Data
Tambah Salesman
Menerima Request Akses Form Tambah Salesman
Menerima Form Tambah Salesman
Mengirim Form Tambah Salesman
Input Nama, Alamat, Kota, Telepon, Contact, Gaji Pokok, dan Status Salesman
Save
Mengirim Query Input Data Salesman
Query Input Data Salesman
Clear
Reset Form Salesman
Menerima Form Tambah Salesman Kosong
Mengirim Form Tambah Salesman Kosong
Exit
Tutup Form Tambah Salesman
Status Salesman
Input Status Salesman
Mengirim Query List Salesman Berdasarkan Status Salesman
Query List Salesman Berdasarkan Status Salesman
Menerima Tampilan Tabel List Salesman Berdasarkan Status Salesman
Load Tabel List Salesman Berdasarkan Status Salesman
Mengirim Data
Cari
Input Nama Salesman
Mengirim Query List Salesman Berdasarkan Nama Salesman
Query List Salesman Berdasarkan Nama Salesman
Menerima Tampilan Tabel List Salesman Berdasarkan Nama Salesman
Load Tabel List Salesman Berdasarkan Nama Salesman
Mengirim Data
Edit
Input Pilih Baris Salesman Pada List Tabel
Input Edit
Mengirim Query Select Salesman
Query Select Salesman
Menerima Tampilan Form dan Data Salesman Terpilih
Load dan Set Data Salesman
Mengirim Data
Input Save
Mengirim Query Update Salesman
Query Update Salesman

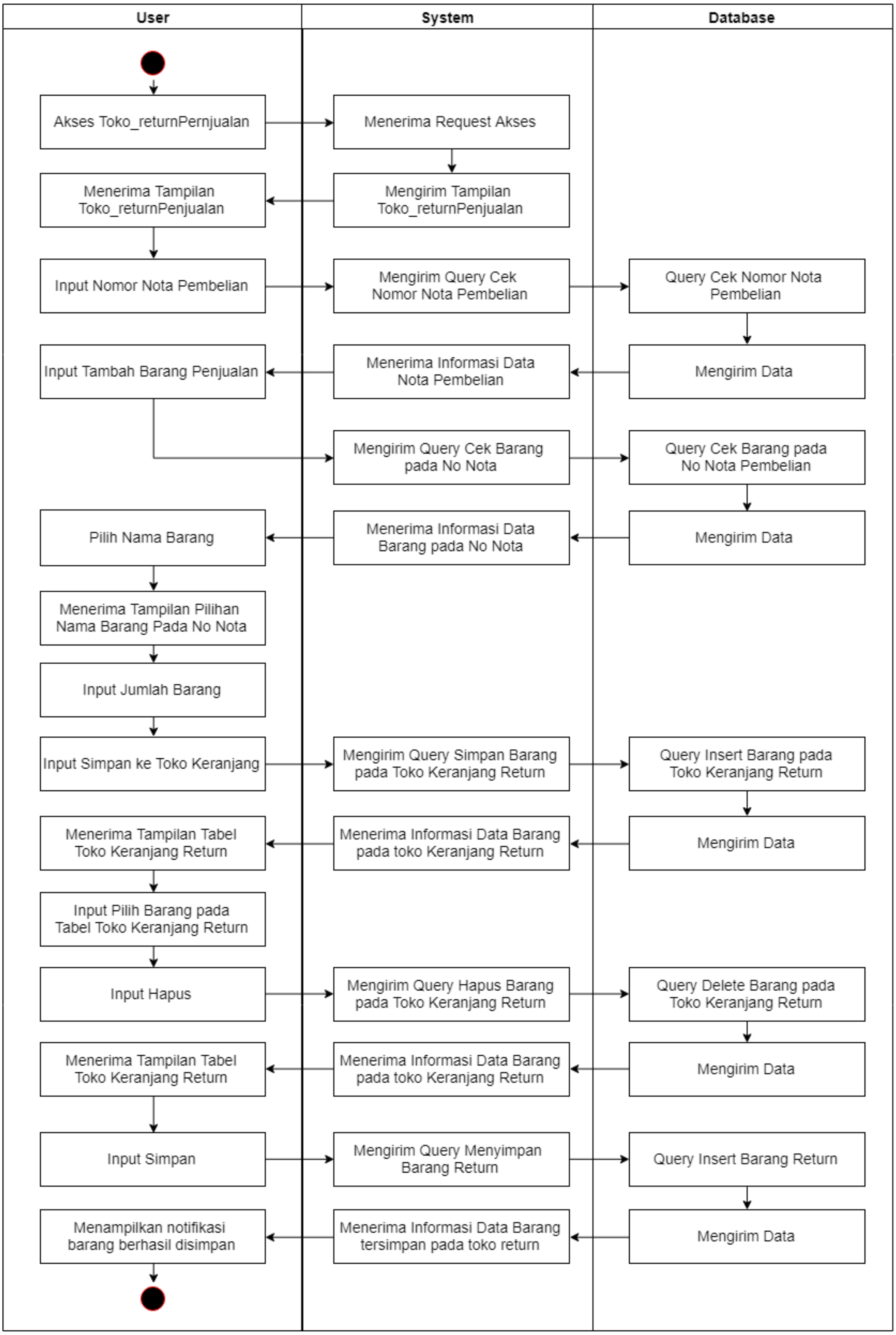
Gambar 4.5 Activity Diagram Master Salesman


User
System
Database
Akses Master_Customer
Menerima Request Akses
Mengirim Query List Customer Active
Query List Customer Active
Menerima Tampilan Tabel Customer Active
Load Tabel Customer Active
Mengirim Data
Tambah Customer
Menerima Request Akses Form Tambah Customer
Menerima Form Tambah Customer
Mengirim Form Tambah Customer
Input Nama, Alamat, Telepon, HP, Contact, Kota, Provinsi, Keterangan, Rekening, Harga Jual, Salesman, TOP, Limit Piutang, Wilayah, dan Status Customer
Save
Mengirim Query Input Data Customer
Query Input Data Customer
Exit
Tutup Form Tambah Customer
Status Customer Deactive
Input Status Customer Deactive
Mengirim Query List Customer Berdasarkan Status Deactive Customer
Query List Customer Berdasarkan Status Deactive Customer
Menerima Tampilan Tabel List Customer Berdasarkan Status Deactive Customer
Load Tabel List Customer Berdasarkan Status Deactive Customer
Mengirim Data
Kriteria
Input Nama Customer
Mengirim Query List Customer Berdasarkan Nama Customer
Query List Customer Berdasarkan Nama Customer
Menerima Tampilan Tabel List Customer Berdasarkan Nama Customer
Load Tabel List Customer Berdasarkan Nama Customer
Mengirim Data
Edit
Input Pilih Baris Customer Pada List Tabel
Input Edit
Mengirim Query Select Customer
Query Select Customer
Menerima Tampilan Form dan Data Customer Terpilih
Load dan Set Data Customer
Mengirim Data
Input Save
Mengirim Query Update Customer
Query Update Customer
Delete
Input Pilih Baris Customer Pada List Tabel
Input Delete
Input Ya pada form Konfirmasi
Mengirim Query Delete Customer
Query Delete Customer

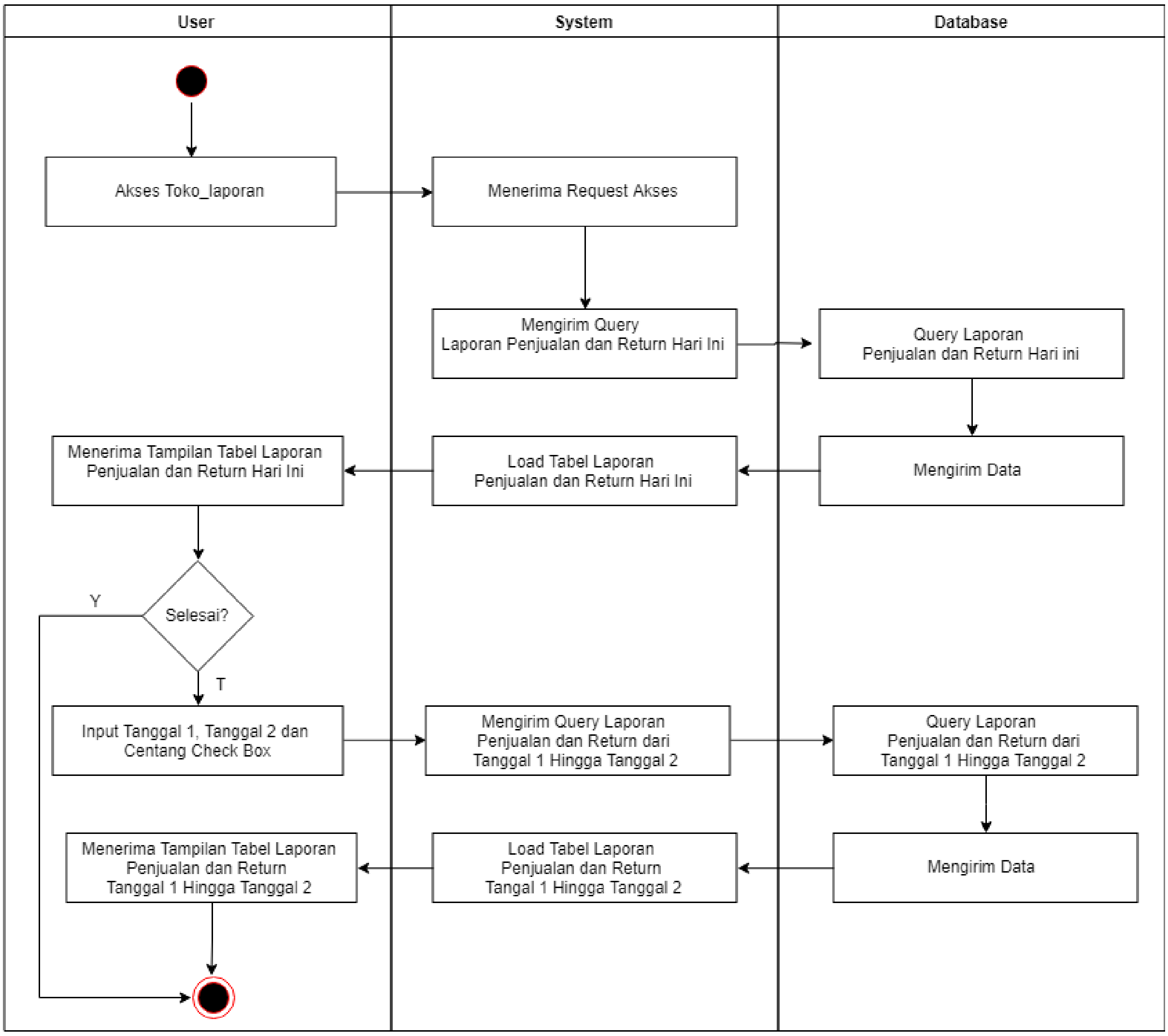
Gambar 4.6 Activity Diagram Master Customer


User
System
Database
Akses Toko_transaksi
Menerima Request Akses
Mengirim Query Nama Barang, No.Customer, No.Faktur terakhir
Query Nama Barang, No.Customer, No.Faktur terakhir
Menerima Tampilan Form Toko_transaksi
Load Nama Barang No.Customer, No.Faktur terakhir, dan Tanggal Hari Ini
Mengirim Data
Input Nama Barang
Mengirim Query Harga, Jumlah, Satuan Barang
Query Harga, Jumlah, Satuan Barang
Menerima Harga, Jumlah dan Isi Combo Box Satuan Barang
Load Harga, Jumlah, dan Satuan Barang Berdasarkan Nama Barang
Mengirim Data
Input Satuan Barang
Mengirim Query Konversi Satuan Barang
Query Konversi Satuan Barang
Menerima Harga, Jumlah Berdasarkan Satuan
Load Harga, dan Jumlah Berdasarkan Satuan
Mengirim Data
Input Jumlah Barang
Hitung Subtotal dan Total Harga
Menerima Tampilan Subtotal dan Total Harga
Load Subtotal dan Total Harga
Save
Mengirim Form Pembayaran dan Total Pembelian
Menerima Form Pembayaran dan Total Pembayaran
Input Jumlah Pembayaran
Input Simpan
Mengirim Query Simpan Toko Penjualan dan Simpan Toko Penjualan Detail
Query Simpan Toko Penjualan dan Simpan Toko Penjualan Detail
Clear
Mengirim Tampilan Konfirmasi Reset Tabel Transaksi
Menerima Tampilan Konfirmasi Reset Tabel Transaksi
Input Ya
Reset Tabel Transaksi
Menerima Tabel Transaksi Kosong
Mengirim Tabel Transaksi Kosong
Exit

Gambar 4.7. Activity Diagram Toko Transaksi

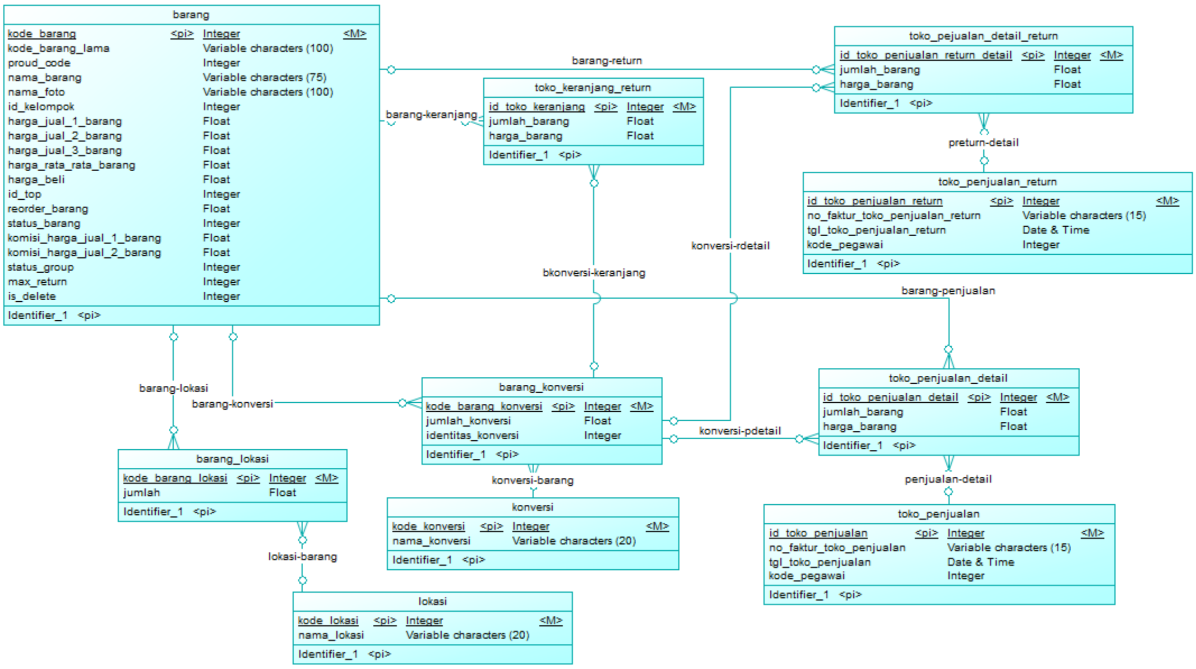

User
System
Database
Akses Toko_returnPernjualan
Menerima Request Akses
Menerima Tampilan Toko_returnPenjualan
Mengirim Tampilan Toko_returnPenjualan
Input Nomor Nota Pembelian
Mengirim Query Cek Nomor Nota Pembelian
Query Cek Nomor Nota Pembelian
Input Tambah Barang Penjualan
Menerima Informasi Data Nota Pembelian
Mengirim Data
Mengirim Query Cek Barang pada No Nota
Query Cek Barang pada No Nota Pembelian
Pilih Nama Barang
Menerima Informasi Data Barang pada No Nota
Mengirim Data
Menerima Tampilan Pilihan Nama Barang Pada No Nota
Input Jumlah Barang
Input Simpan ke Toko Keranjang
Mengirim Query Simpan Barang pada Toko Keranjang Return
Query Insert Barang pada Toko Keranjang Return
Menerima Tampilan Tabel Toko Keranjang Return
Menerima Informasi Data Barang pada toko Keranjang Return
Mengirim Data
Input Pilih Barang pada Tabel Toko Keranjang Return
Input Hapus
Mengirim Query Hapus Barang pada Toko Keranjang Return
Query Delete Barang pada Toko Keranjang Return
Menerima Tampilan Tabel Toko Keranjang Return
Menerima Informasi Data Barang pada toko Keranjang Return
Mengirim Data
Input Simpan
Mengirim Query Menyimpan Barang Return
Query Insert Barang Return
Menampilkan notifikasi barang berhasil disimpan
Menerima Informasi Data Barang tersimpan pada toko return
Mengirim Data

Gambar 4.8. Activity Diagram Toko Return Penjualan


User
System
Database
Akses Toko_laporan
Menerima Request Akses
Mengirim Query
Laporan Penjualan dan Return Hari Ini
Query Laporan
Penjualan dan Return Hari ini
Menerima Tampilan Tabel Laporan
Penjualan dan Return Hari Ini
Load Tabel Laporan
Penjualan dan Return Hari Ini
Mengirim Data
Y
Selesai?
T
Input Tanggal 1, Tanggal 2 dan
Centang Check Box
Mengirim Query Laporan
Penjualan dan Return dari
Tanggal 1 Hingga Tanggal 2
Query Laporan
Penjualan dan Return dari
Tanggal 1 Hingga Tanggal 2
Menerima Tampilan Tabel Laporan
Penjualan dan Return
Tanggal 1 Hingga Tanggal 2
Load Tabel Laporan
Penjualan dan Return
Tangal 1 Hingga Tanggal 2
Mengirim Data

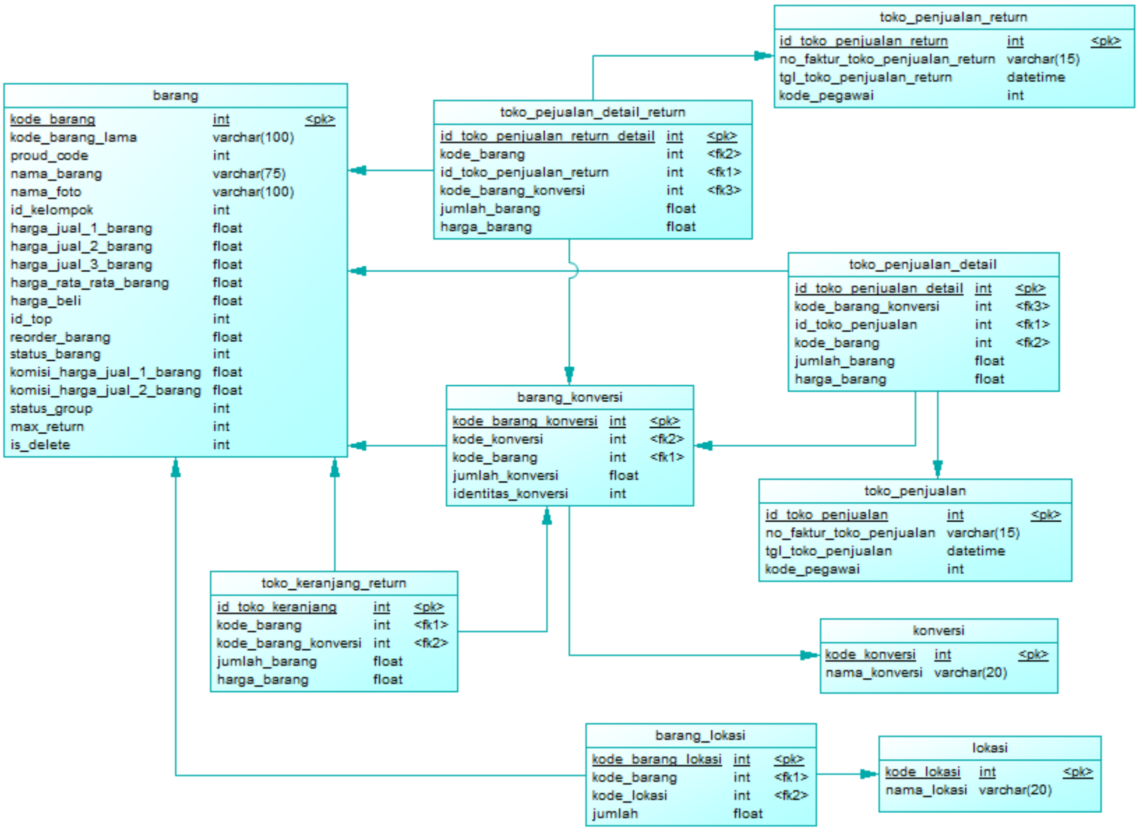
Gambar 4.9. Activity Diagram Toko Laporan

4.3.3. ERD Conceptual Diagram


barang
kode_barang <pi> Integer <M>
kode_barang_lama Variable characters (100)
proud_code Integer
nama_barang Variable characters (75)
nama_foto Variable characters (100)
id_kelompok Integer
harga_jual_1_barang Float
harga_jual_2_barang Float
harga_jual_3_barang Float
harga_rata_rata_barang Float
harga_beli Float
id_top Integer
reorder_barang Float
status_barang Integer
komisi_harga_jual_1_barang Float
komisi_harga_jual_2_barang Float
status_group Integer
max_return Integer
is_delete Integer
Identifier_1 <pi>
barang-return
barang-keranjang
toko_keranjang_return
id_toko_keranjang <pi> Integer <M>
jumlah_barang Float
harga_barang Float
Identifier_1 <pi>
toko_pejualan_detail_return
id_toko_penjualan_return_detail <pi> Integer <M>
jumlah_barang Float
harga_barang Float
Identifier_1 <pi>
preturn-detail
toko_penjualan_return
id_toko_penjualan_return <pi> Integer <M>
no_faktur_toko_penjualan_return Variable characters (15)
tgl_toko_penjualan_return Date & Time
kode_pegawai Integer
Identifier_1 <pi>
konversi-rdetail
bkonversi-keranjang
barang-penjualan
barang-lokasi
barang-konversi
barang_konversi
kode_barang_konversi <pi> Integer <M>
jumlah_konversi Float
identitas_konversi Integer
Identifier_1 <pi>
konversi-pdetail
toko_penjualan_detail
id_toko_penjualan_detail <pi> Integer <M>
jumlah_barang Float
harga_barang Float
Identifier_1 <pi>
barang_lokasi
kode_barang_lokasi <pi> Integer <M>
jumlah Float
Identifier_1 <pi>
konversi-barang
konversi
kode_konversi <pi> Integer <M>
nama_konversi Variable characters (20)
Identifier_1 <pi>
penjualan-detail
toko_penjualan
id_toko_penjualan <pi> Integer <M>
no_faktur_toko_penjualan Variable characters (15)
tgl_toko_penjualan Date & Time
kode_pegawai Integer
Identifier_1 <pi>
lokasi-barang
lokasi
kode_lokasi <pi> Integer <M>
nama_lokasi Variable characters (20)
Identifier_1 <pi>

Gambar 4.10. ERD Conceptual Diagram

4.3.4. ERD Physical Diagram


toko_penjualan_return
id_toko_penjualan_return int <pk>
no_faktur_toko_penjualan_return varchar(15)
tgl_toko_penjualan_return datetime
kode_pegawai int
barang
kode_barang int <pk>
kode_barang_lama varchar(100)
proud_code int
nama_barang varchar(75)
nama_foto varchar(100)
id_kelompok int
harga_jual_1_barang float
harga_jual_2_barang float
harga_jual_3_barang float
harga_rata_rata_barang float
harga_beli float
id_top int
reorder_barang float
status_barang int
komisi_harga_jual_1_barang float
komisi_harga_jual_2_barang float
status_group int
max_return int
is_delete int
toko_pejualan_detail_return
id_toko_penjualan_return_detail int <pk>
kode_barang int <fk2>
id_toko_penjualan_return int <fk1>
kode_barang_konversi int <fk3>
jumlah_barang float
harga_barang float
toko_penjualan_detail
id_toko_penjualan_detail int <pk>
kode_barang_konversi int <fk3>
id_toko_penjualan int <fk1>
kode_barang int <fk2>
jumlah_barang float
harga_barang float
barang_konversi
kode_barang_konversi int <pk>
kode_konversi int <fk2>
kode_barang int <fk1>
jumlah_konversi float
identitas_konversi int
toko_penjualan
id_toko_penjualan int <pk>
no_faktur_toko_penjualan varchar(15)
tgl_toko_penjualan datetime
kode_pegawai int
toko_keranjang_return
id_toko_keranjang int <pk>
kode_barang int <fk1>
kode_barang_konversi int <fk2>
jumlah_barang float
harga_barang float
konversi
kode_konversi int <pk>
nama_konversi varchar(20)
barang_lokasi
kode_barang_lokasi int <pk>
kode_barang int <fk1>
kode_lokasi int <fk2>
jumlah float
lokasi
kode_lokasi int <pk>
nama_lokasi varchar(20)

Gambar 4.11. ERD Physical Diagram

4.3.5. Tabel Database

Tabel 4.1. Tabel barang

Nama Field	Type	Length	Keterangan
kode_barang	Int	-	PK
kode_barang_lama	Varchar	100	
proud_code	Int	-	
nama_barang	Varchar	75	
nama_foto	Varchar	100	
id_kelompok	Int	-	
harga_jual_1_barang	Double	-	
harga_jual_2_barang	Double	-	
harga_jual_3_barang	Double	-	
harga_rata_rata_barang	Double	-	
harga_beli	Double	-	
id_top	Int	-	
reorder_barang	Double	-	
status_barang	Int	-	
komisi_harga_jual_1_barang	Double	-	
komisi_harga_jual_2_barang	Double	-	
status_group	Int	-	
max_return	Int	-	
is_delete	Int	-	

Tabel 4.2. Tabel konversi

Nama Field	Type	Length	Keterangan
kode_konversi	Int	-	PK
nama_konversi	Varchar	20	

Tabel 4.3. Tabel barang_konversi

Nama Field	Type	Length	Keterangan
kode_barang_konversi	Int	-	PK
kode_konversi	Int	-	FK
kode_barang	Int	-	FK
Jumlah_konversi	Double	-	
identitas_konversi	Int	-	

Tabel 4.4. Tabel lokasi

Nama Field	Type	Length	Keterangan
kode_lokasi	Int	-	PK
nama_lokasi	Varchar	20	

Tabel 4.5. Tabel barang_lokasi

Nama Field	Type	Length	Keterangan
kode_barang_lokasi	Int	-	PK
kode_barang	Int	-	FK
kode_lokasi	Int	-	FK
jumlah	Double	-	

Tabel 4.6. Tabel toko_penjualan

Nama Field	Type	Length	Keterangan
id_toko_penjualan	Int	-	PK
no_faktur_toko_penjualan	Varchar	15	
tgl_toko_penjualan	DateTime	-	
kode_pegawai	Int	-	

Tabel 4.7. Tabel toko_penjualan_detail

Nama Field	Type	Length	Keterangan
id_toko_penjualan_detail	Int	-	PK
kode_barang_konversi	Int	-	FK
id_toko_penjualan	Int	-	FK
kode_barang	Int	-	FK
jumlah_barang	Double	-	
harga_barang	Double	-	

Tabel 4.7. Tabel toko_penjualan_return

Nama Field	Type	Length	Keterangan
id_toko_penjualan_return	Int	-	PK
no_faktur_toko_penjualan_return	Varchar	15	
tgl_toko_penjualan_return	DateTime	-	
kode_pegawai	Int	-	

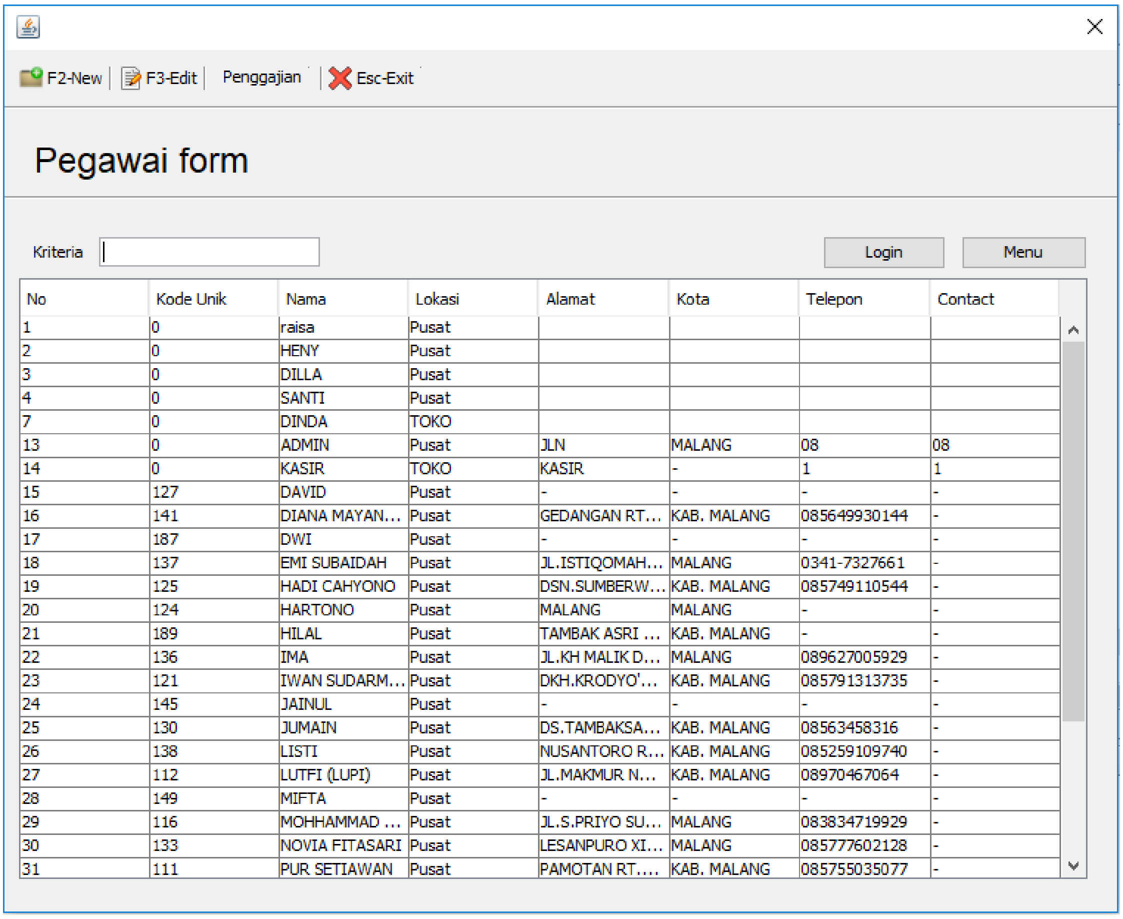
Tabel 4.7. Tabel toko_penjualan_detail_return

Nama Field	Type	Length	Keterangan
id_toko_penjualan_return_detail	Int	-	PK
kode_barang	Int	-	FK
id_toko_penjualan_return	Int	-	FK
kode_barang_konversi	Int	-	FK
jumlah_barang	Double	-	
harga_barang	Double	-	

4.4. Implementasi Program

Implementasi program disesuaikan dengan modul – modul yang sudah dikerjakan oleh penulis.

4.4.1. Form Master Pegawai



Pegawai form

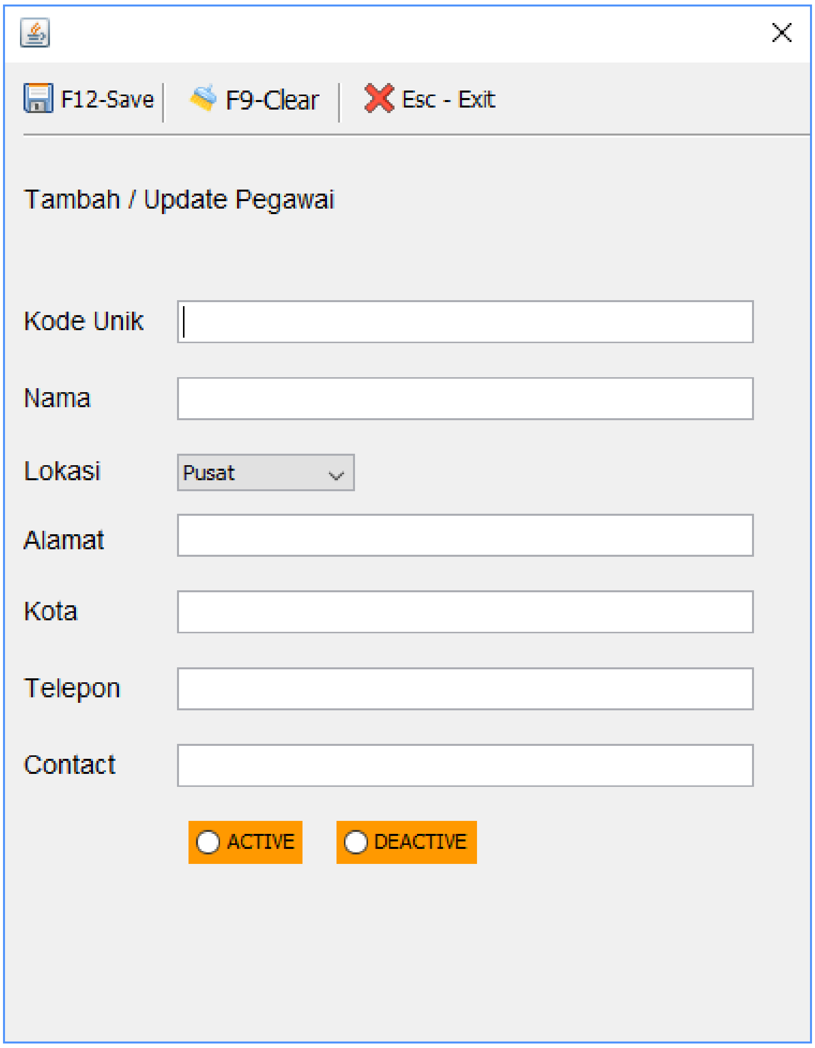
Kriteria

No	Kode Unik	Nama	Lokasi	Alamat	Kota	Telepon	Contact
1	0	raisa	Pusat				
2	0	HENY	Pusat				
3	0	DILLA	Pusat				
4	0	SANTI	Pusat				
7	0	DINDA	TOKO				
13	0	ADMIN	Pusat	JLN	MALANG	08	08
14	0	KASIR	TOKO	KASIR	-	1	1
15	127	DAVID	Pusat	-	-	-	-
16	141	DIANA MAYAN...	Pusat	GEDANGAN RT...	KAB. MALANG	085649930144	-
17	187	DWI	Pusat	-	-	-	-
18	137	EMI SUBAIDAH	Pusat	JL.ISTIQOMAH...	MALANG	0341-7327661	-
19	125	HADI CAHYONO	Pusat	DSN.SUMBERW...	KAB. MALANG	085749110544	-
20	124	HARTONO	Pusat	MALANG	MALANG	-	-
21	189	HILAL	Pusat	TAMBAK ASRI ...	KAB. MALANG	-	-
22	136	IMA	Pusat	JL.KH MALIK D...	MALANG	089627005929	-
23	121	IWAN SUDARM...	Pusat	DKH.KRODYO'...	KAB. MALANG	085791313735	-
24	145	JAINUL	Pusat	-	-	-	-
25	130	JUMAIN	Pusat	DS.TAMBAKSA...	KAB. MALANG	08563458316	-
26	138	LISTI	Pusat	NUSANTORO R...	KAB. MALANG	085259109740	-
27	112	LUTFI (LUPI)	Pusat	JL.MAKMUR N...	KAB. MALANG	08970467064	-
28	149	MIFTA	Pusat	-	-	-	-
29	116	MOHHAMMAD ...	Pusat	JL.S.PRIYO SU...	MALANG	083834719929	-
30	133	NOVIA FITASARI	Pusat	LESANPURO XI...	MALANG	085777602128	-
31	111	PUR SETIAWAN	Pusat	PAMOTAN RT....	KAB. MALANG	085755035077	-

Gambar 4.12. Form Master Pegawai

```
public class Master_Pegawai extends javax.swing.JDialog {
    private Connect connection;
    private ResultSet hasil;
    private ArrayList<ListPegawai> list;
    private ListPegawai listPegawai;
    private TableModel model;
    private PreparedStatement PS;

    public Master_Pegawai() {
        initComponents();
    }

    public Master_Pegawai(java.awt.Frame parent, boolean modal, Connect
    connection) {
        super(parent, modal);
        initComponents();
        this.connection = connection;
        tampilTabel("*");
    }
    private void tampilTabel(String param) {
        String data = "";
        try {
            data = "SELECT kode_pegawai, kode_unik, nama_lokasi,
            nama_pegawai, alamat_pegawai, kota_pegawai,
            telepon_pegawai,contact_pegawai, status_pegawai "
            + "FROM pegawai, lokasi "
            + "WHERE " + (param.equalsIgnoreCase("*") ? "" : "nama_pegawai
            like '%" + param + "%' and ") +
            "pegawai.kode_lokasi=lokasi.kode_lokasi "
            + "order by kode_pegawai";
            hasil = connection.ambilData(data);
            setModel(hasil);
        } catch (Exception e) {
            System.out.println("ERROR -> " + e.getMessage());
        } finally {}
    }


    private void setModel(ResultSet hasil) {
        try {
            list = new ArrayList<>();
            while (hasil.next()) {
                this.listPegawai = new ListPegawai();
                this.listPegawai.setKode_pegawai(hasil.getInt
                ("kode_pegawai"));
                this.listPegawai.setKode_unik(hasil.getInt("kode_unik"));
                this.listPegawai.setNama_pegawai(hasil.getString
                ("nama_pegawai"));
                this.listPegawai.setKode_lokasi(hasil.getString
                ("nama_lokasi"));
                this.listPegawai.setAlamat_pegawai(hasil.getString
                ("alamat_pegawai"));
                this.listPegawai.setKota_pegawai(hasil.getString
                ("kota_pegawai"));
                this.listPegawai.setTelepon_pegawai(hasil.getString
                ("telepon_pegawai"));
                this.listPegawai.setContact_pegawai(hasil.getString
                ("contact_pegawai"));
                this.listPegawai.setStatus_pegawai(hasil.getInt
                ("status_pegawai"));
                list.add(listPegawai); listPegawai = null;
            }
            model = new modelTabelPegawai(list);
            jTable6.setModel(model);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private int getNewId() {
        String sql = "SELECT kode_pegawai FROM pegawai ORDER BY
        kode_pegawai DESC LIMIT 1";
        int id = 0;
        try {
            hasil = connection.ambilData(sql);
            while (hasil.next()) {
                id = hasil.getInt("kode_pegawai") + 1;
            }
        } catch (SQLException e) {} return id;
    }
    private void insertData(ListPegawai listPegawai) {
        PS = null;
        if (listPegawai.getKode_pegawai() != 0) {
            try {
                String sql = "insert into pegawai values
                (?,?,?,?,?,?,?,?,?,?,?,?)";
                PS = connection.Connect().prepareStatement(sql);
                PS.setInt(1, listPegawai.getKode_pegawai());
                PS.setInt(2, listPegawai.getKode_unik());
                PS.setString(3, listPegawai.getKode_lokasi());
                PS.setInt(4, 0);
                PS.setString(5, listPegawai.getNama_pegawai());
                PS.setString(6, listPegawai.getAlamat_pegawai());
                PS.setString(7, listPegawai.getKota_pegawai());
                PS.setString(8, listPegawai.getTelepon_pegawai());
                PS.setString(9, listPegawai.getContact_pegawai());
                PS.setInt(10, listPegawai.getStatus_pegawai());
                PS.setString(11, "");
                PS.setString(12, "");
                PS.executeUpdate();
                tampilTabel("*");
            } catch (SQLException e) {
                System.out.println("Master_Pegawai_Line_86_" +
                e.toString());
            }
        }
    }

    private void updateData(ListPegawai listPegawai) {
        PS = null;
        try {
            String sql = "Update pegawai set
            kode_unik=?,nama_pegawai=?,kode_lokasi=?,alamat_pegawai=?,
            kota_pegawai=?, "
            + "telepon_pegawai=?,contact_pegawai=?,status_pegawai=? "
            + "where kode_pegawai=? ";
            PS = connection.Connect().prepareStatement(sql);
            PS.setInt(1, listPegawai.getKode_unik());
            PS.setString(2, listPegawai.getNama_pegawai());
            PS.setString(3, listPegawai.getKode_lokasi());
            PS.setString(4, listPegawai.getAlamat_pegawai());
            PS.setString(5, listPegawai.getKota_pegawai());
            PS.setString(6, listPegawai.getTelepon_pegawai());
            PS.setString(7, listPegawai.getContact_pegawai());
            PS.setInt(8, listPegawai.getStatus_pegawai());
            PS.setInt(9, listPegawai.getKode_pegawai());
            PS.executeUpdate();

            tampilTabel("*");
        } catch (SQLException e) {
            System.out.println("Master_Pegawai_Line_111_" + e.toString());
        }
    }

    private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
        listPegawai = new ListPegawai();
        Master_Pegawai_TambahPegawai tp = new
        Master_Pegawai_TambahPegawai(new Awal(rootPaneCheckingEnabled),
        rootPaneCheckingEnabled, listPegawai, getNewId());
        tp.setLocationRelativeTo(this);
        tp.setVisible(true);
        listPegawai.getKode_pegawai());
        insertData(listPegawai);
    }
    private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {
        if (jTable6.getSelectedRow() < 0) {
            JOptionPane.showMessageDialog(null, "Pilih Pegawai yang akan
            diedit");
        } else {
            try {
                Master_Pegawai_TambahPegawai tp = new
                Master_Pegawai_TambahPegawai(new Awal
                (rootPaneCheckingEnabled), rootPaneCheckingEnabled,
                listPegawai, true);
                tp.setLocationRelativeTo(this);
                tp.setVisible(true);
                updateData(listPegawai);
            } catch (Exception e) {
                System.out.println("Master_Pegawai_Line_360_" +
                e.getMessage());
            }
        }
    }

    private void jTable6MouseClicked(java.awt.event.MouseEvent evt) {
        this.listPegawai = list.get(jTable6.getSelectedRow());
    }

    private void jLabel7MouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
        tampilTabel(jTextField1.getText());
    }

    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane7;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator10;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    private javax.swing.JSeparator jSeparator6;
    private javax.swing.JSeparator jSeparator7;
    private javax.swing.JTable jTable6;
    private javax.swing.JTextField jTextField1;
```

4.4.2. Form Master Pegawai TambahPegawai



Tambah / Update Pegawai

Kode Unik

Nama

Lokasi: Pusat

Alamat

Kota

Telepon

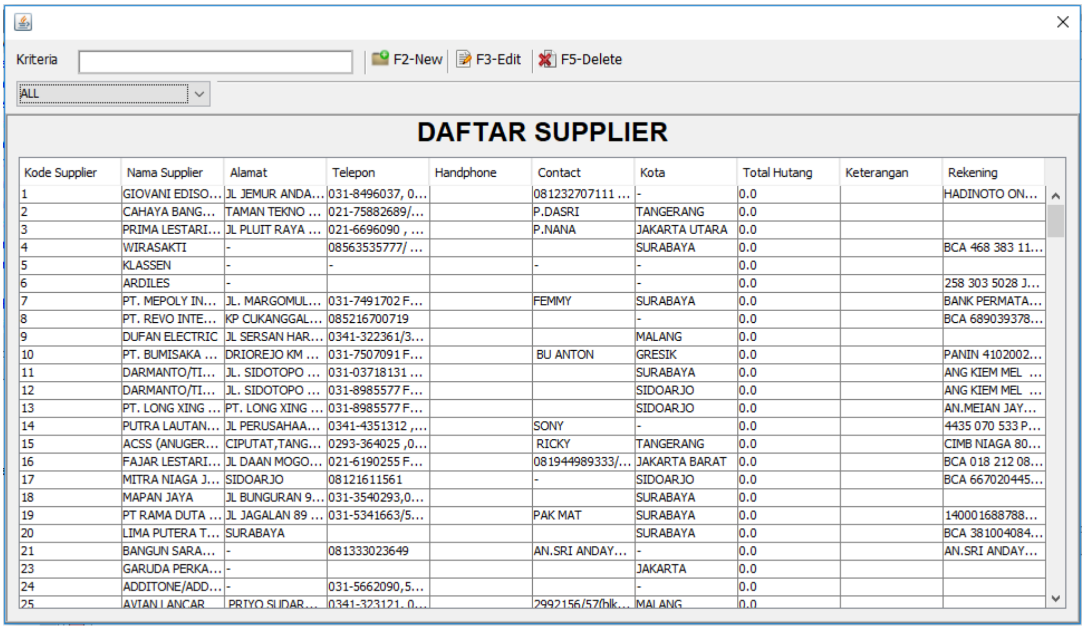
Contact

○ ACTIVE ○ DEACTIVE

Gambar 4.13 Form Master Pegawai TambahPegawai

```
public class Master_Pegawai_TambahPegawai extends javax.swing.JDialog {
    private ListPegawai listpegawai;
    private boolean isEdit = false;
    private int id;

    public Master_Pegawai_TambahPegawai() {//dont use this
        initComponents();
    }

    public Master_Pegawai_TambahPegawai(java.awt.Frame parent, boolean
    modal, ListPegawai listpegawai, boolean isEdit) {
        super(parent, modal);
        initComponents();
        this.isEdit = isEdit;
        this.listpegawai = listpegawai;
        setContent();
    }
    public Master_Pegawai_TambahPegawai(java.awt.Frame parent, boolean
    modal, ListPegawai listpegawai, int id) {
        super(parent, modal);
        initComponents();
        this.id = id;
        this.listpegawai = listpegawai;
    }

    private void setContent() {
        jTextField6.setText(String.valueOf(listpegawai.getKode_unik()));
        jTextField7.setText(listpegawai.getNama_pegawai());
        jComboBox1.setSelectedItem(listpegawai.getKode_lokasi());
        jTextField9.setText(listpegawai.getAlamat_pegawai());
        jTextField10.setText(listpegawai.getKota_pegawai());
        jTextField11.setText(listpegawai.getTelepon_pegawai());
        jTextField12.setText(listpegawai.getContact_pegawai());
        if (listpegawai.getStatus_pegawai() == 0) {
            active.setSelected(true);
        } else {
            deactive.setSelected(true);
        }
    }

    private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
        String lokasi = jComboBox1.getSelectedItem().toString();
        if (!isEdit) {
            listpegawai.setKode_pegawai(this.id);
        }
        listpegawai.setKode_unik(Integer.parseInt(jTextField6.getText()));
        listpegawai.setNama_pegawai(jTextField7.getText());
        listpegawai.setKode_lokasi(lokasi.equalsIgnoreCase("utara") ? "6" :
        lokasi.equalsIgnoreCase("gud63") ? "2" :
        lokasi.equalsIgnoreCase("toko") ? "4" :
        lokasi.equalsIgnoreCase("tengah") ? "5" : "1");
        listpegawai.setAlamat_pegawai(jTextField9.getText());
        listpegawai.setKota_pegawai(jTextField10.getText());
        listpegawai.setTelepon_pegawai(jTextField11.getText());
        listpegawai.setContact_pegawai(jTextField12.getText());
        listpegawai.setStatus_pegawai(active.isSelected() ? 0 : 1);
        this.dispose();
    }

    private javax.swing.JRadioButton active;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JRadioButton deactive;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel26;
    private javax.swing.JLabel jLabel27;
    private javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel29;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JLabel jLabel31;
    private javax.swing.JLabel jLabel32;
    private javax.swing.JLabel jLabel33;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField11;
    private javax.swing.JTextField jTextField12;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField9;
```

4.4.3. Form Master Supplier



DAFTAR SUPPLIER

Kode Supplier	Nama Supplier	Alamat	Telepon	Handphone	Contact	Kota	Total Hutang	Keterangan	Rekening
1	GIOVANI EDISO…	JL JEMUR ANDA…	031-8496037, 0…		081232707111 …	-	0.0		HADINOTO ON…
2	CAHAYA BANG…	TAMAN TEKNO …	021-75882689/…		P.DASRI	TANGERANG	0.0		
3	PRIMA LESTARI…	JL PLUIT RAYA …	021-6696090 , …		P.NANA	JAKARTA UTARA	0.0		
4	WIRASAKTI	-	08563535777/ …			SURABAYA	0.0		BCA 468 383 11…
5	KLASSEN	-	-		-	-	0.0		
6	ARDILES	-				-	0.0		258 303 5028 J…
7	PT. MEPOLY IN…	JL. MARGOMUL…	031-7491702 F…		FEMMY	SURABAYA	0.0		BANK PERMATA…
8	PT. REVO INTE…	KP CUKANGGAL…	085216700719			-	0.0		BCA 689039378…
9	DUFAN ELECTRIC	JL SERSAN HAR…	0341-322361/3…			MALANG	0.0		
10	PT. BUMISAKA …	DRIOREJO KM …	031-7507091 F…		BU ANTON	GRESIK	0.0		PANIN 4102002…
11	DARMANTO/TI…	JL. SIDOTOPO …	031-03718131 …			SURABAYA	0.0		ANG KIEM MEL …
12	DARMANTO/TI…	JL. SIDOTOPO …	031-8985577 F…			SIDOARJO	0.0		ANG KIEM MEL …
13	PT. LONG XING …	PT. LONG XING …	031-8985577 F…			SIDOARJO	0.0		AN.MEIAN JAY…
14	PUTRA LAUTAN…	JL PERUSAHAA…	0341-4351312 ,…		SONY	-	0.0		4435 070 533 P…
15	ACSS (ANUGER…	CIPUTAT,TANG…	0293-364025 ,0…		RICKY	TANGERANG	0.0		CIMB NIAGA 80…
16	FAJAR LESTARI…	JL DAAN MOGO…	021-6190255 F…		081944989333/…	JAKARTA BARAT	0.0		BCA 018 212 08…
17	MITRA NIAGA J…	SIDOARJO	08121611561		-	SIDOARJO	0.0		BCA 667020445…
18	MAPAN JAYA	JL BUNGURAN 9…	031-3540293,0…			SURABAYA	0.0		
19	PT RAMA DUTA …	JL JAGALAN 89 …	031-5341663/5…		PAK MAT	SURABAYA	0.0		140001688788…
20	LIMA PUTERA T…	SURABAYA				SURABAYA	0.0		BCA 381004084…
21	BANGUN SARA…	-	081333023649		AN.SRI ANDAY…	-	0.0		AN.SRI ANDAY…
23	GARUDA PERKA…	-				JAKARTA	0.0		
24	ADDITONE/ADD…	-	031-5662090,5…			-	0.0		
25	AVIAN LANCAR	PRIYO SUDAR…	0341-323121, 0…		2992156/57(blk…	MALANG	0.0		

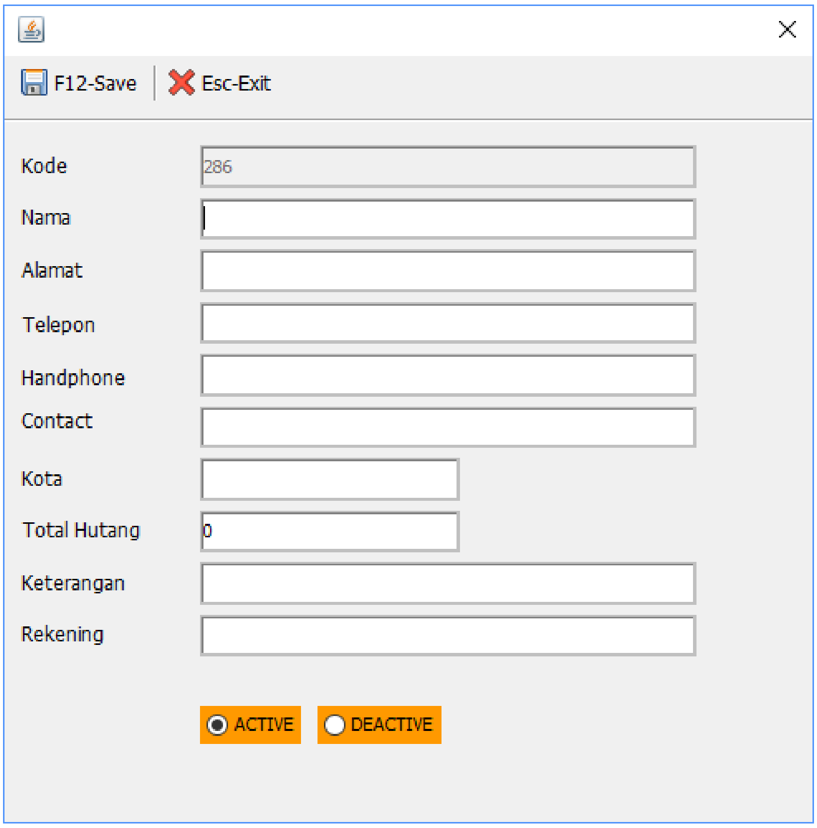
Gambar 4.14. Form Master Supplier

```
public class Master_Supplier extends javax.swing.JDialog {

    private ResultSet hasil;
    private Connect connection;
    private PreparedStatement PS;
    private ArrayList<ListSupplier> list;
    private ListSupplier listSupplier;
    private TableModel model;
    private MouseAdapter MA;
    private String comboBox;

    public Master_Supplier() {
        initComponents();
    }
    public Master_Supplier(java.awt.Frame parent, boolean modal, Connect
    connection) {
        super(parent, modal);
        initComponents();
        this.connection = connection;
        tampilTabel("1");
    }

    private void tampilTabel(String param) {
        String datax = "";
        try {
            datax = "SELECT * "
            + "from supplier "
            + "" + (param.equals("-1") ? "" : (param.equals("1") ||
            param.equals("0") ? "where aktif_supplier=" +
            Integer.parseInt(param) : "where nama_supplier like '%" + param
            + "%' ") );
            hasil = connection.ambilData(datax);
            setModel(hasil);
        } catch (Exception e) {
            System.out.println("Error tampil tabel");
        }
    }
```

```
private void updateData(int kodeSupplier) {
    PS = null;
    try {
        String sql = "Update supplier set nama_supplier=?,
        alamat_supplier=?,"
        + "kota_supplier=?,telepon_supplier=?,handphone_supplier=?,
        contact_supplier=?,"
        + "keterangan_supplier=?,rekening_supplier=?,aktif_supplier=? "
        + "where kode_Supplier=?";
        PS = connection.Connect().prepareStatement(sql);
        PS.setString(1, listSupplier.getNama_supplier());
        PS.setString(2, listSupplier.getAlamat_supplier());
        PS.setString(3, listSupplier.getKota_supplier());
        PS.setString(4, listSupplier.getTelepon_supplier());
        PS.setString(5, listSupplier.getHp_supplier());
        PS.setString(6, listSupplier.getContact_supplier());
        PS.setString(7, listSupplier.getKeterangan());
        PS.setString(8, listSupplier.getRekening());
        PS.setInt(9, listSupplier.getStatus());
        PS.setInt(10, listSupplier.getKode_supplier());
        PS.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Master_Supplier_Line_50_" + e.toString());
    } finally { tampilTabel("-1"); }
 }

private void insertData(ListSupplier listSupplier) {
    PS = null;
    if (listSupplier.getKode_supplier() != 0) {
        try {
            String sql = "insert into supplier values
            (?,?,?,?,?,?,?,?,?,?,?)";
            PS = connection.Connect().prepareStatement(sql);
            PS.setInt(1, listSupplier.getKode_supplier());
            PS.setString(2, listSupplier.getNama_supplier());
            PS.setString(3, listSupplier.getAlamat_supplier());
            PS.setString(4, listSupplier.getKota_supplier());
            PS.setString(5, listSupplier.getKota_supplier());
            PS.setString(6, listSupplier.getTelepon_supplier());
            PS.setString(7, listSupplier.getHp_supplier());
            PS.setString(8, listSupplier.getContact_supplier());
            PS.setString(9, listSupplier.getRekening());
            PS.setInt(10, listSupplier.getStatus());
            PS.setString(11, listSupplier.getKeterangan());
            PS.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Master_Supplier_Line_75_" +
            e.toString());
        } finally { tampilTabel("-1"); }
    }
 }

private void deleteData(ListSupplier listSupplier) {
    PS = null;
    try {
        String sql = "delete from supplier where kode_supplier=?";
        PS = connection.Connect().prepareStatement(sql);
        PS.setInt(1, listSupplier.getKode_supplier());
        PS.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Master_Supplier_Line_75_" + e.toString());
    } finally { tampilTabel("-1"); }
 }
```

```
    private void setModel(ResultSet hasil) {
        try {
            list = new ArrayList<>();
            while (hasil.next()) {
                this.listSupplier = new ListSupplier();
                this.listSupplier.setKode_supplier(hasil.getInt
                ("kode_supplier"));
                this.listSupplier.setNama_supplier(hasil.getString
                ("nama_supplier"));
                this.listSupplier.setAlamat_supplier(hasil.getString
                ("alamat_supplier"));
                this.listSupplier.setKota_supplier(hasil.getString
                ("kota_supplier"));
                this.listSupplier.setTelepon_supplier(hasil.getString
                ("telepon_supplier"));
                this.listSupplier.setHp_supplier(hasil.getString
                ("handphone_supplier"));
                this.listSupplier.setContact_supplier(hasil.getString
                ("contact_supplier"));
                this.listSupplier.setRekening(hasil.getString
                ("rekening_supplier"));
                this.listSupplier.setStatus(hasil.getInt
                ("aktif_supplier"));
                this.listSupplier.setKeterangan(hasil.getString
                ("keterangan_supplier"));
                list.add(listSupplier);
                listSupplier = null;
            }
            model = new modelTabelSupplier(list);
            jTable2.setModel(model);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private int getNewId() {
        String sql = "SELECT kode_supplier FROM supplier ORDER BY
        kode_supplier DESC LIMIT 1"; int id = 0;
        try {
            hasil = connection.ambilData(sql);
            while (hasil.next()) { id = hasil.getInt("kode_supplier") + 1;}
        } catch (SQLException e) {} return id;
    }

    private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {
        listSupplier = new ListSupplier();
        Master_Supplier_TambahEdit tp = new Master_Supplier_TambahEdit(new
        Awal(rootPaneCheckingEnabled), rootPaneCheckingEnabled,
        listSupplier, getNewId());
        tp.setLocationRelativeTo(this); tp.setVisible(true);
        insertData(listSupplier);
    }

    private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
        this.listSupplier = list.get(jTable2.getSelectedRow());
    }

    private void jLabel21MouseClicked(java.awt.event.MouseEvent evt) {
        if (jTable2.getSelectedRow() < 0) {
            JOptionPane.showMessageDialog(null, "Pilih Supplier yang akan
            diedit");
        } else { Master_Supplier_TambahEdit tp = new
            Master_Supplier_TambahEdit(new Awal(rootPaneCheckingEnabled),
            rootPaneCheckingEnabled, listSupplier, true);
            tp.setLocationRelativeTo(this);
            tp.setVisible(true);
            updateData(listSupplier.getKode_supplier());
        }
    }
    private void jComboBox4ActionPerformed(java.awt.event.ActionEvent evt)
    {
        comboBox = jComboBox4.getSelectedItem().toString();
        switch (jComboBox4.getSelectedIndex()) {
            case 0:
                tampilTabel("1");
                break;
            case 1:
                tampilTabel("0");
                break;
            default:
                tampilTabel("-1");
                break;
        }
    }

    private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {
        deleteData(listSupplier);
    }

    private void jTextField14KeyReleased(java.awt.event.KeyEvent evt) {
        tampilTabel(jTextField14.getText());
    }

    private javax.swing.JComboBox<String> jComboBox4;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    private javax.swing.JTable jTable2;
    private javax.swing.JTextField jTextField14;
```

4.4.4. Form Master Supplier TambahEdit



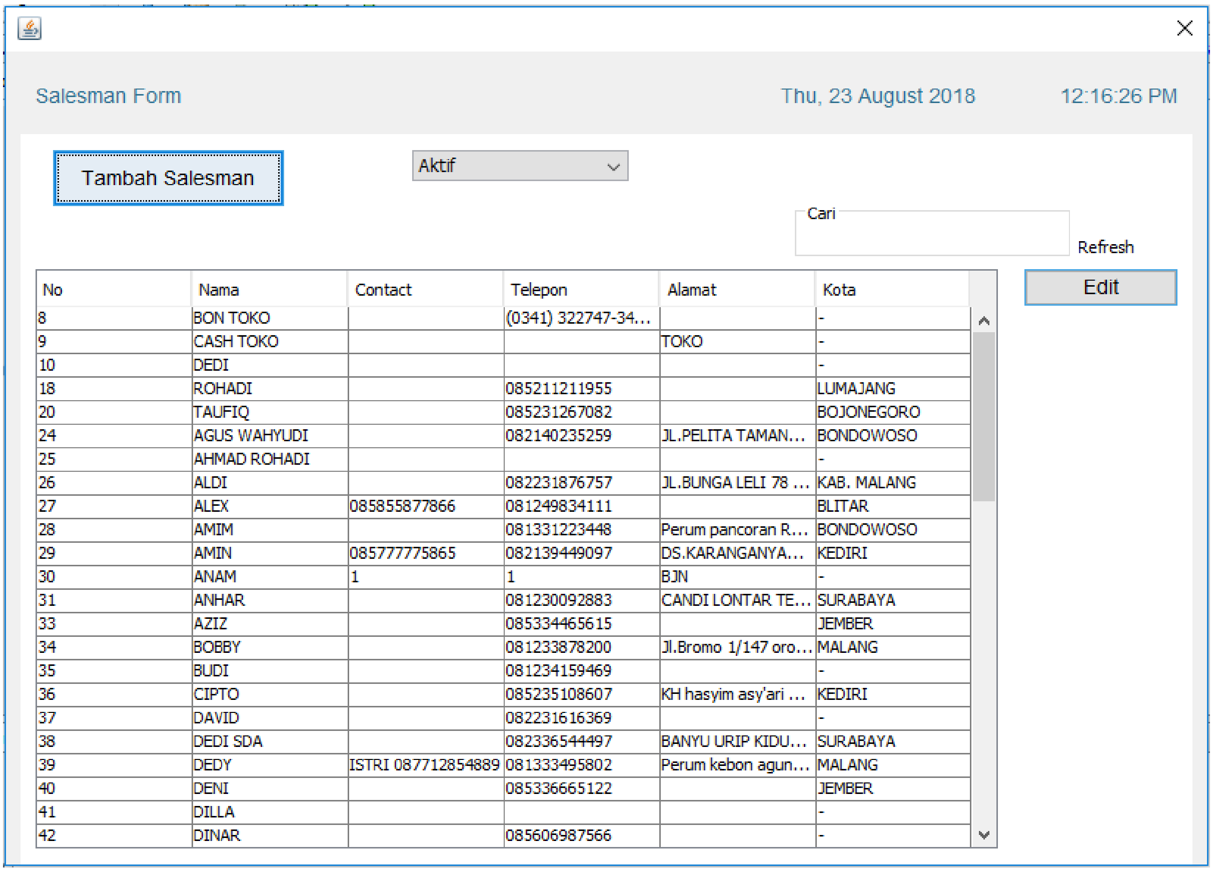
Kode	286
Nama	
Alamat	
Telepon	
Handphone	
Contact	
Kota	
Total Hutang	0
Keterangan	
Rekening	

● ACTIVE ○ DEACTIVE

Gambar 4.15. Form Master Supplier TambahEdit

```
public class Master_Supplier_TambahEdit extends javax.swing.JDialog {
    private ListSupplier listSupplier; private boolean isEdit = false;
    private int id;

    public Master_Supplier_TambahEdit() {
        initComponents();
    }

    Master_Supplier_TambahEdit(java.awt.Frame parent, boolean modal,
    ListSupplier listSupplier, boolean isEdit) {
        super(parent, modal); initComponents(); this.isEdit = isEdit;
        this.listSupplier = listSupplier; setContent();
    }

    Master_Supplier_TambahEdit(java.awt.Frame parent, boolean modal,
    ListSupplier listSupplier, int id) {
        super(parent, modal); initComponents(); this.id = id;
        this.listSupplier = listSupplier;
        jTextField30.setText(String.valueOf(id));
    }

    private void setContent() {
        jTextField30.setText(String.valueOf(listSupplier
        .getKode_supplier()));
        jTextField8.setText(listSupplier.getNama_supplier());
        jTextField33.setText(listSupplier.getAlamat_supplier());
        jTextField9.setText(listSupplier.getTelepon_supplier());
        jTextField10.setText(listSupplier.getContact_supplier());
        jTextField37.setText(listSupplier.getKota_supplier());
        jTextField11.setText(String.valueOf(listSupplier
        .getTotal_hutang()));
        jTextField38.setText(String.valueOf(listSupplier.getKeterangan()));
        jTextField13.setText(String.valueOf(listSupplier.getRekening()));
        if (listSupplier.getStatus() == 1) { aktif.setSelected(true); }
        else { nonAktif.setSelected(true); }
    }

    private void jLabel64MouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void jLabel68MouseClicked(java.awt.event.MouseEvent evt) {
        if (!isEdit) {
            this.listSupplier.setKode_supplier(this.id);
        }
        this.listSupplier.setNama_supplier(jTextField8.getText());
        this.listSupplier.setAlamat_supplier(jTextField33.getText());
        this.listSupplier.setTelepon_supplier(jTextField9.getText());
        this.listSupplier.setHp_supplier(jTextField36.getText());
        this.listSupplier.setContact_supplier(jTextField10.getText());
        this.listSupplier.setKota_supplier(jTextField37.getText());
        this.listSupplier.setTotal_hutang(Double.parseDouble
        (jTextField11.getText()));
        this.listSupplier.setKeterangan(jTextField38.getText());
        this.listSupplier.setRekening(jTextField13.getText());
        this.listSupplier.setStatus(aktif.isSelected() ? 1 : 0);
        this.dispose();
    }

    private javax.swing.JRadioButton aktif;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JLabel jLabel46;
    private javax.swing.JLabel jLabel50;
    private javax.swing.JLabel jLabel51;
    private javax.swing.JLabel jLabel52;
    private javax.swing.JLabel jLabel54;
    private javax.swing.JLabel jLabel57;
    private javax.swing.JLabel jLabel58;
    private javax.swing.JLabel jLabel59;
    private javax.swing.JLabel jLabel64;
    private javax.swing.JLabel jLabel65;
    private javax.swing.JLabel jLabel66;
    private javax.swing.JLabel jLabel68;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField11;
    private javax.swing.JTextField jTextField13;
    private javax.swing.JTextField jTextField30;
    private javax.swing.JTextField jTextField33;
    private javax.swing.JTextField jTextField36;
    private javax.swing.JTextField jTextField37;
    private javax.swing.JTextField jTextField38;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    private javax.swing.JRadioButton nonAktif;
```

4.4.5. Form Master Salesman



Salesman Form

Thu, 23 August 2018

12:16:26 PM

Aktif

No	Nama	Contact	Telepon	Alamat	Kota
8	BON TOKO		(0341) 322747-34...		-
9	CASH TOKO			TOKO	-
10	DEDI				-
18	ROHADI		085211211955		LUMAJANG
20	TAUFIQ		085231267082		BOJONEGORO
24	AGUS WAHYUDI		082140235259	JL.PELITA TAMAN...	BONDOWOSO
25	AHMAD ROHADI				-
26	ALDI		082231876757	JL.BUNGA LELI 78 ...	KAB. MALANG
27	ALEX	085855877866	081249834111		BLITAR
28	AMIM		081331223448	Perum pancoran R...	BONDOWOSO
29	AMIN	085777775865	082139449097	DS.KARANGANYA...	KEDIRI
30	ANAM	1	1	BJN	-
31	ANHAR		081230092883	CANDI LONTAR TE...	SURABAYA
33	AZIZ		085334465615		JEMBER
34	BOBBY		081233878200	Jl.Bromo 1/147 oro...	MALANG
35	BUDI		081234159469		-
36	CIPTO		085235108607	KH hasyim asy'ari ...	KEDIRI
37	DAVID		082231616369		-
38	DEDI SDA		082336544497	BANYU URIP KIDU...	SURABAYA
39	DEDY	ISTRI 087712854889	081333495802	Perum kebon agun...	MALANG
40	DENI		085336665122		JEMBER
41	DILLA				-
42	DINAR		085606987566		-

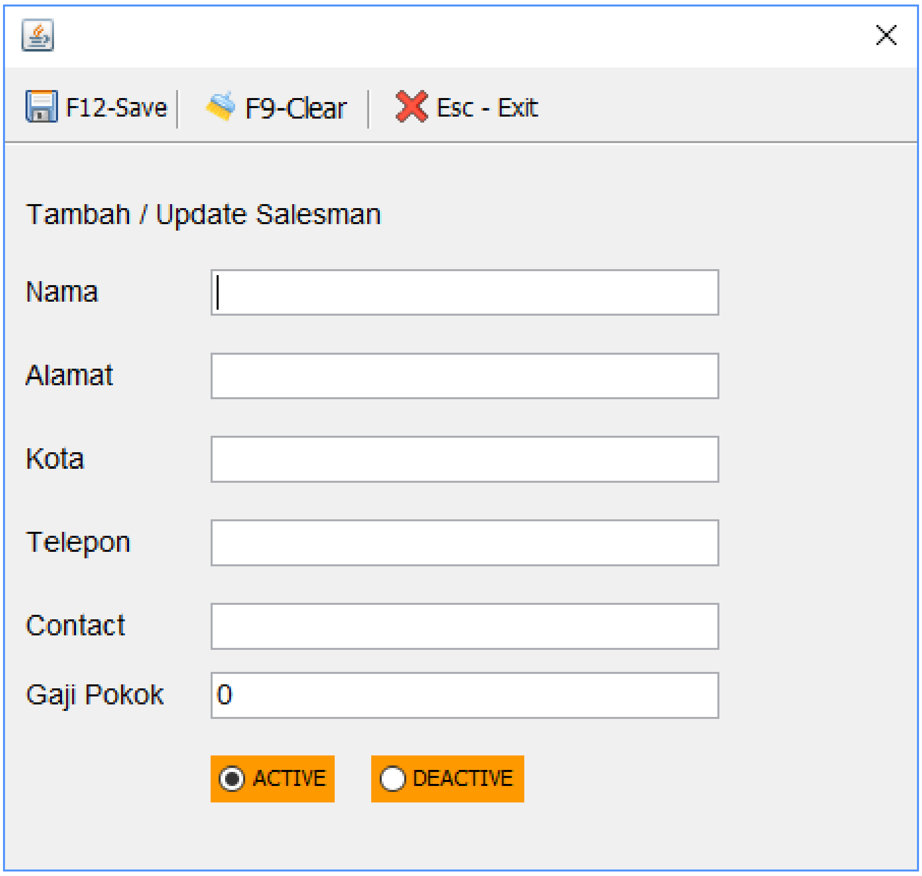
Gambar 4.16. Form Master Salesman

```
public class Master_Salesman extends javax.swing.JDialog {

    private ArrayList<ListSalesman> list;
    private ListSalesman listSalesman;
    private DefaultTableModel tabel;
    private ResultSet hasil;
    private PreparedStatement PS;
    private Connect connection;
    private Clock clock;
    private int dd;

    public Master_Salesman() {
        initComponents();
    }

    public Master_Salesman(java.awt.Frame parent, boolean modal, Connect
    connection) {
        super(parent, modal);
        initComponents();
        clock = new Clock(lblWaktu, 1); clock = new Clock(lblTanggal, 0);
        this.connection = connection;
        jTable9.setModel(new DefaultTableModel(new String[]{"No", "Nama",
        "Contact", "Telepon", "Alamat", "Kota"}, 0));
        tampilTabel("1");
    }

    private void deleteTabel() {
        for(int i = 0;i < tabel.getRowCount(); i++) { tabel.removeRow(0); }
    }
    public String tampilTabel(String param) {
        String data = "SELECT kode_salesman, nama_salesman,
        contact_salesman, telepon_salesman, alamat, kota_salesman,
        aktif_sales, gaji_salesman "
        + "FROM salesman "
        + (param.equals("-1") ? "" : (param.equals("1") ||
        param.equals("0") ? "WHERE aktif_sales = " +Integer.parseInt(param)
        : "WHERE nama_salesman like '%" + param + "%'"));
        try {
            hasil = connection.ambilData(data); setModel(hasil);
        } catch (Exception e) {
            System.out.println("Master_Salesman_Line_43_" + e.toString());
            System.out.println(data);
        } return data;
    }

    public void setModel(ResultSet hasil) {
        try {
            list = new ArrayList<>();
            tabel = new DefaultTableModel(new String[]{"No", "Nama",
            "Contact", "Telepon", "Alamat", "Kota"}, 0);
            while (hasil.next()) {
                listSalesman = new ListSalesman();
                listSalesman.setNo(hasil.getInt("kode_salesman"));
                listSalesman.setNama_salesman(hasil.getString
                ("nama_salesman"));
                listSalesman.setContact_salesman(hasil.getString
                ("contact_salesman"));
                listSalesman.setTelepon_salesman(hasil.getString
                ("telepon_salesman"));
                listSalesman.setAlamat_salesman(hasil.getString("alamat"));
                listSalesman.setKota_salesman(hasil.getString
                ("kota_salesman"));
                listSalesman.setStatus(hasil.getInt("aktif_sales"));
                listSalesman.setGaji(hasil.getDouble("gaji_salesman"));
                list.add(listSalesman);
                tabel.addRow(new Object[]{
                    listSalesman.getNo(),
                    listSalesman.getNama_salesman(),
                    listSalesman.getContact_salesman(),
                    listSalesman.getTelepon_salesman(),
                    listSalesman.getAlamat_salesman(),
                    listSalesman.getKota_salesman()
                });
            listSalesman = null;
            }
        } catch (SQLException e) {
            System.out.println("Master_Salesman_Line_59_" + e.toString());
        } finally {
            jTable9.setModel(tabel);
        }
    }

    private int getNewId() {
        String sql = "SELECT kode_salesman FROM salesman ORDER BY
        kode_salesman DESC LIMIT 1";
        int id = 0;
        try {
            hasil = connection.ambilData(sql);
            while (hasil.next()) {
                id = hasil.getInt("kode_salesman") + 1;
            }
        } catch (SQLException e) {} return id;
    }
```

```
private void insertData(ListSalesman listSalesman) {
    PS = null;
    if (listSalesman.getNo() != 0) {
        try {
            String sql = "insert into salesman values
            (?,?,?,?,?,?,?,?,?,?)";
            PS = connection.Connect().prepareStatement(sql);
            PS.setInt(1, listSalesman.getNo());
            PS.setString(2, listSalesman.getNama_salesman());
            PS.setString(3, listSalesman.getAlamat_salesman());
            PS.setString(4, listSalesman.getKota_salesman());
            PS.setString(5, listSalesman.getTelepon_salesman());
            PS.setString(6, listSalesman.getContact_salesman());
            PS.setInt(7, listSalesman.getStatus());
            PS.setString(8, ""); PS.setString(9, "");
            PS.setDouble(10, listSalesman.getGaji());
            PS.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Master_Salesman_Line_105_" +
            e.toString());
        }
    }
}

private void updateData(ListSalesman listSalesman) {
    PS = null;
    try {
        String sql = "update salesman set nama_salesman=?, alamat=?,
        kota_salesman=?,telepon_salesman=?,contact_salesman=?,"
        + "aktif_sales=?,gaji_salesman=?"
        + "where kode_salesman=?";
        PS = connection.Connect().prepareStatement(sql);
        PS.setString(1, listSalesman.getNama_salesman());
        PS.setString(2, listSalesman.getAlamat_salesman());
        PS.setString(3, listSalesman.getKota_salesman());
        PS.setString(4, listSalesman.getTelepon_salesman());
        PS.setString(5, listSalesman.getContact_salesman());
        PS.setInt(6, listSalesman.getStatus());
        PS.setDouble(7, listSalesman.getGaji());
        PS.setInt(8, listSalesman.getNo()); PS.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Master_Salesman_Line_126_" + e.toString());
    }
}

private void jButton33ActionPerformed(java.awt.event.ActionEvent evt) {
    listSalesman = list.get(jTable9.getSelectedRow());
    Master_Salesman_TambahSalesman ts = new
    Master_Salesman_TambahSalesman(new Awal
    (rootPaneCheckingEnabled), rootPaneCheckingEnabled,
    listSalesman, true); ts.setLocationRelativeTo(this);
    ts.setVisible(true); updateData(listSalesman); listSalesman = null;
    tampilTabel("1"); jTable9.clearSelection();
}

private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {
    listSalesman = new ListSalesman();
    Master_Salesman_TambahSalesman ts = new
    Master_Salesman_TambahSalesman(new Awal(rootPaneCheckingEnabled),
    rootPaneCheckingEnabled, listSalesman, getNewId());
    ts.setLocationRelativeTo(this); ts.setVisible(true);
    insertData(listSalesman); tampilTabel("1");
}
```

```
    private void jComboBox7PopupMenuWillBecomeInvisible(javax.swing.event
    .PopupMenuEvent evt) {
        dd = jComboBox7.getSelectedIndex();
        deleteTabel();
        switch (dd) {
            case 0:
                tampilTabel("1");
                break;
            case 1:
                tampilTabel("0");
                break;
            case 2:
                tampilTabel("-1");
                break;
            default:
                break;
        }
    }

    private void jTextField90KeyReleased(java.awt.event.KeyEvent evt) {
        tampilTabel(jTextField90.getText());
    }

    private javax.swing.JButton jButton29;
    private javax.swing.JButton jButton33;
    private javax.swing.JComboBox<String> jComboBox7;
    private javax.swing.JLabel jLabel184;
    private javax.swing.JLabel jLabel45;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane10;
    private javax.swing.JTable jTable9;
    private javax.swing.JTextField jTextField90;
    private javax.swing.JLabel lblTanggal;
    private javax.swing.JLabel lblWaktu;
```

4.4.6. Form Master Salesman Tambah Salesman



Tambah / Update Salesman

Nama

Alamat

Kota

Telepon

Contact

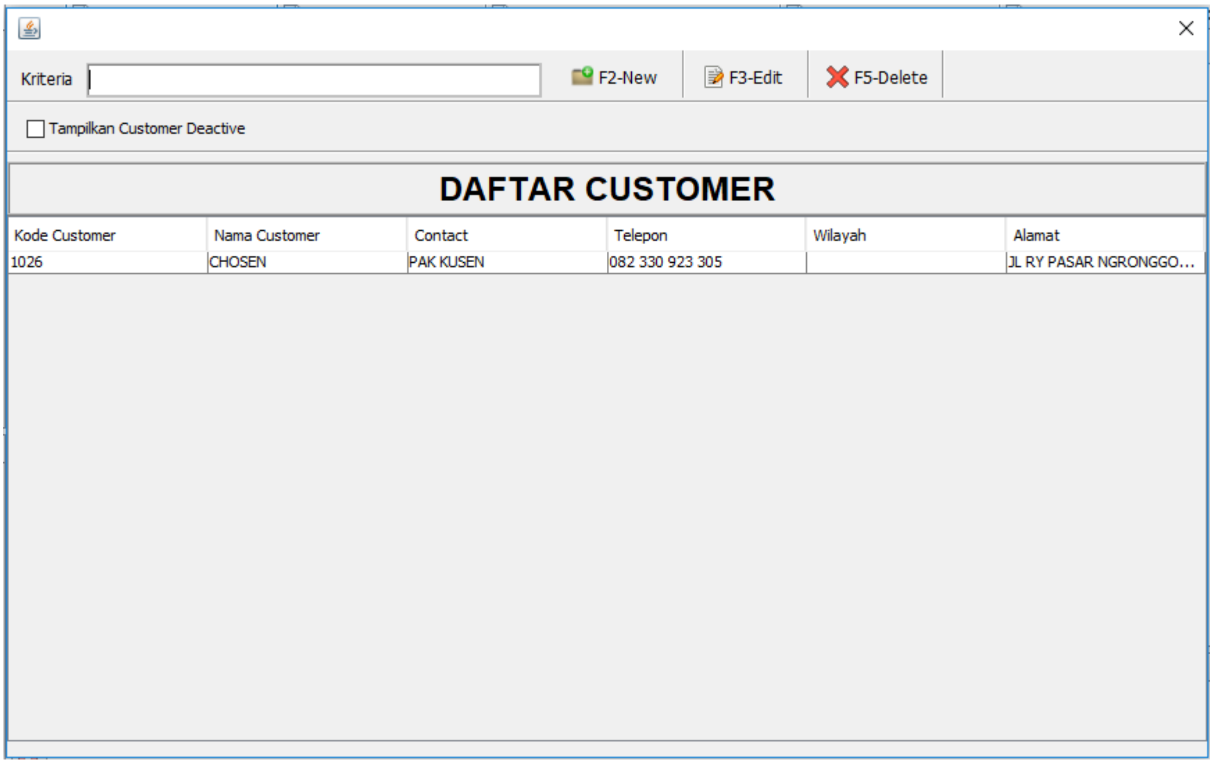
Gaji Pokok 0

● ACTIVE ○ DEACTIVE

Gambar 4.17. Form Master Saleman Tambah Salesman

```
public class Master_Salesman_TambahSalesman extends javax.swing.JDialog {
    private ListSalesman listSalesman; private boolean isEdit = false;
    private int id;

    public Master_Salesman_TambahSalesman() { initComponents(); }

    public Master_Salesman_TambahSalesman(java.awt.Frame parent, boolean
    modal, ListSalesman listSalesman, boolean isEdit) {
        super(parent, modal); initComponents();
        this.isEdit = isEdit;this.listSalesman = listSalesman;
        setContent();
    }

    public Master_Salesman_TambahSalesman(java.awt.Frame parent, boolean
    modal, ListSalesman listSalesman, int id) {
        super(parent, modal); initComponents(); this.id = id;
        this.listSalesman = listSalesman;
    }

    private void setContent() {
        jTextField27.setText(listSalesman.getNama_salesman());
        jTextField30.setText(listSalesman.getAlamat_salesman());
        jTextField33.setText(listSalesman.getKota_salesman());
        jTextField34.setText(listSalesman.getTelepon_salesman());
        jTextField35.setText(listSalesman.getContact_salesman());
        jTextField36.setText(String.valueOf(listSalesman.getGaji()));
        if (listSalesman.getStatus() == 1) {
            jRadioButton3.setSelected(true);
        } else { deactive.setSelected(true); }
    }
    private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
        if(!isEdit){ listSalesman.setNo(id); }
        listSalesman.setNama_salesman(jTextField27.getText());
        listSalesman.setAlamat_salesman(jTextField30.getText());
        listSalesman.setKota_salesman(jTextField33.getText());
        listSalesman.setTelepon_salesman(jTextField34.getText());
        listSalesman.setContact_salesman(jTextField35.getText());
        listSalesman.setStatus(jRadioButton3.isSelected() ? 1 : 0);
        listSalesman.setGaji(Double.parseDouble(jTextField36.getText()));
        this.dispose();
    }

    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JRadioButton deactive;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel63;
    private javax.swing.JLabel jLabel64;
    private javax.swing.JLabel jLabel65;
    private javax.swing.JLabel jLabel66;
    private javax.swing.JLabel jLabel67;
    private javax.swing.JLabel jLabel68;
    private javax.swing.JLabel jLabel69;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    private javax.swing.JTextField jTextField27;
    private javax.swing.JTextField jTextField30;
    private javax.swing.JTextField jTextField33;
    private javax.swing.JTextField jTextField34;
    private javax.swing.JTextField jTextField35;
    private javax.swing.JTextField jTextField36;
```

4.4.7. Form Master Customer



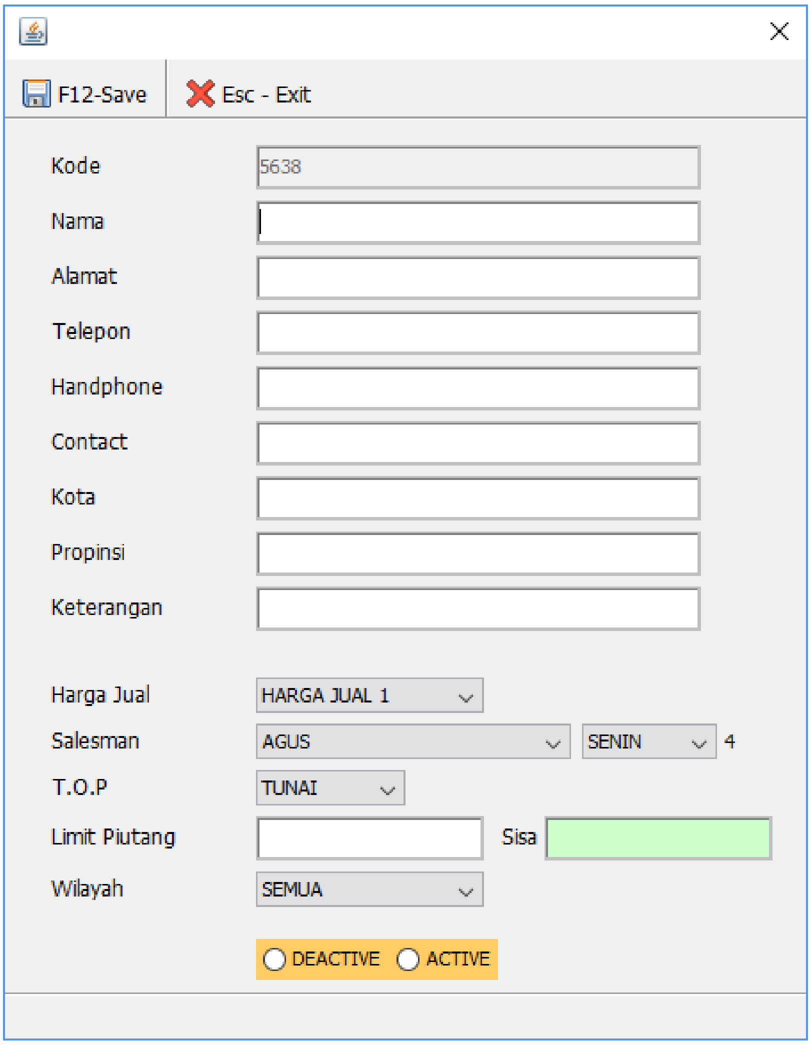
DAFTAR CUSTOMER

Kode Customer	Nama Customer	Contact	Telepon	Wilayah	Alamat
1026	CHOSEN	PAK KUSEN	082 330 923 305		JL RY PASAR NGRONGGO...

Gambar 4.18. Form Master Customer

```
public class Master_Customer extends javax.swing.JDialog {
    private Connect connection;
    private ArrayList<ListSalesman> list_Sales;
    private ArrayList<myArrlist> arrlist;
    private myArrlist myarrlist;
    private ListCustomer listCustomer;
    private ListSalesman listSalesman;
    private ResultSet hasil;
    private PreparedStatement PS;
    private modelTabelCustomer model;

    public Master_Customer() {
        initComponents();
    }

    public Master_Customer(java.awt.Frame parent, boolean modal, Connect
    connection) {
        super(parent, modal);
        initComponents();
        this.connection = connection;
        tampilTabel("1");
    }

    private void tampilTabel(String param) {
        int i = 0;
        try {
            String datax = "SELECT * "
            + "from customer c, salesman s "
            + "where " + (param.equals("1") || param.equals("0") ?
            "c.status_customer =" + Integer.parseInt(param) :
            "c.nama_customer like '%" + param + "%'") + " and
            c.kode_salesman = s.kode_salesman";
            hasil = connection.ambilData(datax); setModel(hasil);
        } catch (NumberFormatException e) {
            System.out.println("Error tampil tabel : " + e.getMessage());
        }
    }

    private ArrayList<ListSalesman> getListSalesman() {
        list_Sales = new ArrayList<>();
        try {
            String sql = "SELECT kode_salesman, nama_salesman from
            salesman";
            hasil = connection.ambilData(sql);
            while (hasil.next()) {
                listSalesman = new ListSalesman();
                listSalesman.setNo(hasil.getInt(1));
                listSalesman.setNama_salesman(hasil.getString(2));
                list_Sales.add(listSalesman);
                listSalesman = null;
            }
        } catch (SQLException err) {
            System.out.println("getListSalesman : " + err.getMessage());
        }
        return list_Sales;
    }

    private int getNewId() {
        String sql = "SELECT kode_customer FROM customer ORDER BY
        kode_customer DESC LIMIT 1"; int id = 0;
        try {
            hasil = connection.ambilData(sql);
            if (hasil.last()) {id = hasil.getInt("kode_customer") + 1;}
        } catch (SQLException e) {} return id;
    }

    private void setModel(ResultSet hasil) {
        try {
            myarrlist = new myArrlist();
            while (hasil.next()) {
                listCustomer = new ListCustomer();
                this.listCustomer.setKode_customer
                (hasil.getInt("kode_customer"));
                this.listCustomer.setNama_customer(hasil.
                getString("nama_customer"));
                this.listCustomer.setAlamat_customer(hasil
                .getString("alamat_customer"));
                this.listCustomer.setTelepon_customer(hasil
                .getString("telepon_customer"));
                this.listCustomer.setContact_customer
                (hasil.getString("contact_customer"));
                this.listCustomer.setKota_customer
                (hasil.getString("kota_customer"));
                this.listCustomer.setProvinsi_customer(hasil
                .getString("provinsi_customer"));
                this.listCustomer.setHari_tagihan(hasil
                .getString("hari_tagihan"));
                this.listCustomer.setStatus_customer(hasil
                .getInt("status_customer"));
                this.listCustomer.setKode_salesman(hasil
                .getInt("kode_salesman"));
                this.myarrlist.getLc().add(listCustomer);
                listCustomer = null;
                listSalesman = new ListSalesman();
                listSalesman.setNo(hasil.getInt("kode_salesman"));
                listSalesman.setNama_salesman(hasil.getString
                ("nama_salesman"));
                this.myarrlist.getLs().add(listSalesman);
                listSalesman = null;
            }
            model = new modelTabelCustomer(myarrlist.getLc());
            jTable7.setModel(model);
        } catch (SQLException e) {
            System.out.println("Master_Customer_Line_54_" +
            e.getMessage());
        }
    }

    private void insertData(ListCustomer listCustomer) {
        PS = null;
        if (listCustomer.getKode_customer() != 0) {
            try {
                String sql = "insert into customer
                values(?,?,?,?,?,?,?,?,?,?)";
                PS = connection.Connect().prepareStatement(sql);
                PS.setInt(1, listCustomer.getKode_customer());
                PS.setString(2, listCustomer.getNama_customer());
                PS.setString(3, listCustomer.getKota_customer());
                PS.setString(4, listCustomer.getAlamat_customer());
                PS.setString(5, listCustomer.getProvinsi_customer());
                PS.setString(6, listCustomer.getTelepon_customer());
                PS.setString(7, listCustomer.getContact_customer());
                PS.setString(8, listCustomer.getHari_tagihan());
                PS.setInt(9, listCustomer.getStatus_customer());
                PS.setInt(10, listCustomer.getKode_salesman());
                PS.executeUpdate(); tampilTabel("1");
            } catch (SQLException e) {
                System.out.println("Master_Customer_Line_84_" +
                e.toString());
            }
        }
    }

    private void updateData(Object[] list) {
        PS = null;
        this.listCustomer = (ListCustomer) list[0];
        this.listSalesman = (ListSalesman) list[1];
        try {
            String sql = "update customer set nama_customer = ?,
            kota_customer = ?, alamat_customer = ?, "
            + "provinsi_customer = ?, telepon_customer = ?,
            contact_customer = ?, "
            + "hari_tagihan = ?, status_customer = ?, kode_salesman = ? "
            + "where kode_customer = ?";
            PS = connection.Connect().prepareStatement(sql);
            PS.setString(1, listCustomer.getNama_customer());
            PS.setString(2, listCustomer.getKota_customer());
            PS.setString(3, listCustomer.getAlamat_customer());
            PS.setString(4, listCustomer.getProvinsi_customer());
            PS.setString(5, listCustomer.getTelepon_customer());
            PS.setString(6, listCustomer.getContact_customer());
            PS.setString(7, listCustomer.getHari_tagihan());
            PS.setInt(8, listCustomer.getStatus_customer());
            PS.setInt(9, listCustomer.getKode_salesman());
            PS.setInt(10, listCustomer.getKode_customer());
            PS.executeUpdate(); tampilTabel("1");
        } catch (SQLException err) {
            System.out.println(err.getMessage());
        }
    }
    private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
        listCustomer = new ListCustomer();
        Master_Customer_NewEdit mcne = new Master_Customer_NewEdit(new
        Awal(rootPaneCheckingEnabled), rootPaneCheckingEnabled,
        listCustomer, getNewId(), getListSalesman());
        mcne.setLocationRelativeTo(new Awal(rootPaneCheckingEnabled));
        mcne.setVisible(true); insertData(listCustomer);
    }

    private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
        listCustomer = (ListCustomer) myarrlist.getLc().get
        (jTable7.getSelectedRow());
        listSalesman = (ListSalesman) myarrlist.getLs().get
        (jTable7.getSelectedRow());
        Object[] temp = {listCustomer, listSalesman};
        Master_Customer_NewEdit mcne = new Master_Customer_NewEdit(new
        Awal(rootPaneCheckingEnabled), rootPaneCheckingEnabled,
        temp, true, getListSalesman());
        mcne.setLocationRelativeTo(new Awal(rootPaneCheckingEnabled));
        mcne.setVisible(true); updateData(temp);
    }

    private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {
        if (jTable7.getSelectedRow() >= 0) {
            int hapus = JOptionPane.showConfirmDialog
            (null, "Apakah Anda yakin Menghapus data ini ?", "Konfimasi
            Hapus Data", JOptionPane.YES_NO_OPTION, 3);
            if (hapus == JOptionPane.YES_OPTION) {
                try {
                    String sql = "DELETE from customer
                    where kode_customer = ?";
                    PS = connection.Connect().prepareStatement(sql);
                    PS.setInt(1, Integer.parseInt(jTable7.getValueAt
                    (jTable7.getSelectedRow(), 0).toString()));
                    PS.executeUpdate();
                } catch (SQLException err) {
                    System.out.println(err.getMessage());
                } finally {
                    JOptionPane.showMessageDialog(null, "Data sudah
                    dihapus", "", 1); tampilTabel("1");
                }
            }
        }
    }

    private void jCheckBox1MouseClicked(java.awt.event.MouseEvent evt) {
        if (jCheckBox1.isSelected()) { tampilTabel("0"); }
        else { tampilTabel("1"); }
    }

    private void jTextField5KeyReleased(java.awt.event.KeyEvent evt) {
        tampilTabel(jTextField5.getText());
    }

    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane7;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    private javax.swing.JSeparator jSeparator5;
    private javax.swing.JSeparator jSeparator6;
    private javax.swing.JTable jTable7;
    private javax.swing.JTextField jTextField5;
```

4.4.8. Form Master Customer NewEdit

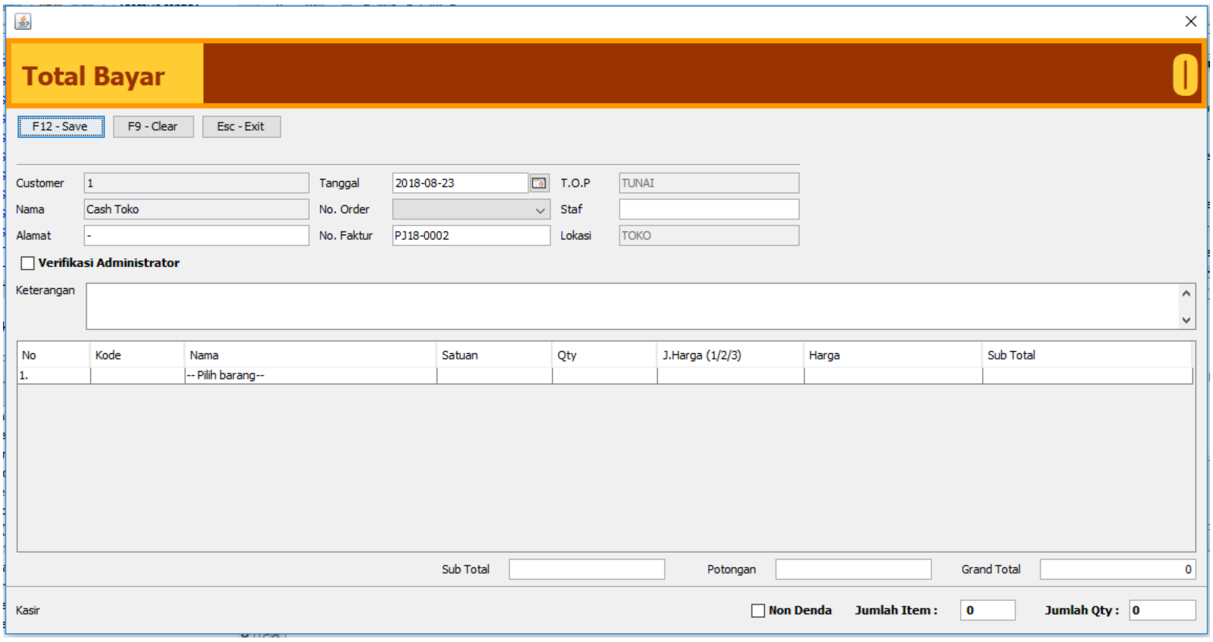


Kode	5638		
Nama			
Alamat			
Telepon			
Handphone			
Contact			
Kota			
Propinsi			
Keterangan			
Harga Jual	HARGA JUAL 1		
Salesman	AGUS	SENIN	4
T.O.P	TUNAI		
Limit Piutang		Sisa	
Wilayah	SEMUA		
	○ DEACTIVE ○ ACTIVE		

Gambar 4.19. Form Master Customer NewEdit

```
public class Master_Customer_NewEdit extends javax.swing.JDialog {
    private ListCustomer listCustomer;
    private ListSalesman listSalesman, listSalesman2;
    private ArrayList<ListSalesman> list_Sales;
    private boolean isEdit = false;
    private int id;

    public Master_Customer_NewEdit() {
        initComponents();
    }

    public Master_Customer_NewEdit(java.awt.Frame parent, boolean modal,
    Object[] list, boolean isEdit, ArrayList<ListSalesman> list_Sales) {
        super(parent, modal);
        initComponents();
        this.isEdit = isEdit;
        listCustomer = (ListCustomer) list[0];
        listSalesman = (ListSalesman) list[1];
        this.list_Sales = list_Sales;
        setComboSalesman();
        setContent();
    }
    public Master_Customer_NewEdit(java.awt.Frame parent, boolean modal,
    ListCustomer listCustomer, int id, ArrayList<ListSalesman> list_Sales)
{
        super(parent, modal);
        initComponents();
        this.id = id;
        this.listCustomer = listCustomer;
        this.list_Sales = list_Sales;
        jTextField5.setText(String.valueOf(id));
        setComboSalesman();
    }

    private void setComboSalesman(){
        for(int x = 0; x < list_Sales.size(); x++){
            listSalesman2 = list_Sales.get(x);
            jComboBox3.addItem(listSalesman2.getNama_salesman());
            listSalesman2 = null;
        }
    }

    private void setContent(){
        jTextField5.setText(String.valueOf(listCustomer
        .getKode_customer()));
        jTextField6.setText(listCustomer.getNama_customer());
        jTextField7.setText(listCustomer.getAlamat_customer());
        jTextField8.setText(listCustomer.getTelepon_customer());
        jTextField10.setText(listCustomer.getContact_customer());
        jTextField11.setText(listCustomer.getKota_customer());
        jTextField12.setText(listCustomer.getProvinsi_customer());
        if(listCustomer.getStatus_customer() == 0){
            jRadioButton1.setSelected(true);
        } else { jRadioButton2.setSelected(true); }
        jComboBox4.setSelectedItem(listCustomer.getHari_tagihan());
        jComboBox3.setSelectedItem(listSalesman.getNama_salesman());
    }

    private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
        if(!isEdit){ listCustomer.setKode_customer(id); }
        listCustomer.setNama_customer(jTextField6.getText());
        listCustomer.setAlamat_customer(jTextField7.getText());
        listCustomer.setTelepon_customer(jTextField8.getText());
        listCustomer.setContact_customer(jTextField10.getText());
        listCustomer.setKota_customer(jTextField11.getText());
        listCustomer.setProvinsi_customer(jTextField12.getText());
        listCustomer.setStatus_customer(jRadioButton1.isSelected()
        ? 0 : 1);
        listCustomer.setHari_tagihan(jComboBox4.getSelectedItem()
        .toString());
        listCustomer.setKode_salesman(Integer.parseInt(jLabel1.getText()));
        this.dispose();
    }

    private void jComboBox3ItemStateChanged(java.awt.event.ItemEvent evt) {
        if (evt.getStateChange() == ItemEvent.SELECTED) {
            jLabel1.setText(String.valueOf(list_Sales.get(jComboBox3
            .getSelectedIndex()).getNo()));
        }
    }
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox3;
    private javax.swing.JComboBox<String> jComboBox4;
    private javax.swing.JComboBox<String> jComboBox5;
    private javax.swing.JComboBox<String> jComboBox6;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JLabel jLabel26;
    private javax.swing.JLabel jLabel27;
    private javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel29;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField11;
    private javax.swing.JTextField jTextField12;
    private javax.swing.JTextField jTextField13;
    private javax.swing.JTextField jTextField14;
    private javax.swing.JTextField jTextField15;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
```

4.4.9. Form Toko Transaksi



Customer	1	Tanggal	2018-08-23	T.O.P	TUNAI
Nama	Cash Toko	No. Order		Staf	
Alamat	-	No. Faktur	PJ18-0002	Lokasi	TOKO

□ Verifikasi Administrator

Keterangan

No	Kode	Nama	Satuan	Qty	J.Harga (1/2/3)	Harga	Sub Total
1.		-- Pilih barang--					

Sub Total | Potongan | Grand Total 0

Kasir | □ Non Denda | Jumlah Item : 0 | Jumlah Qty : 0

Gambar 4.20. Form Toko Transaksi

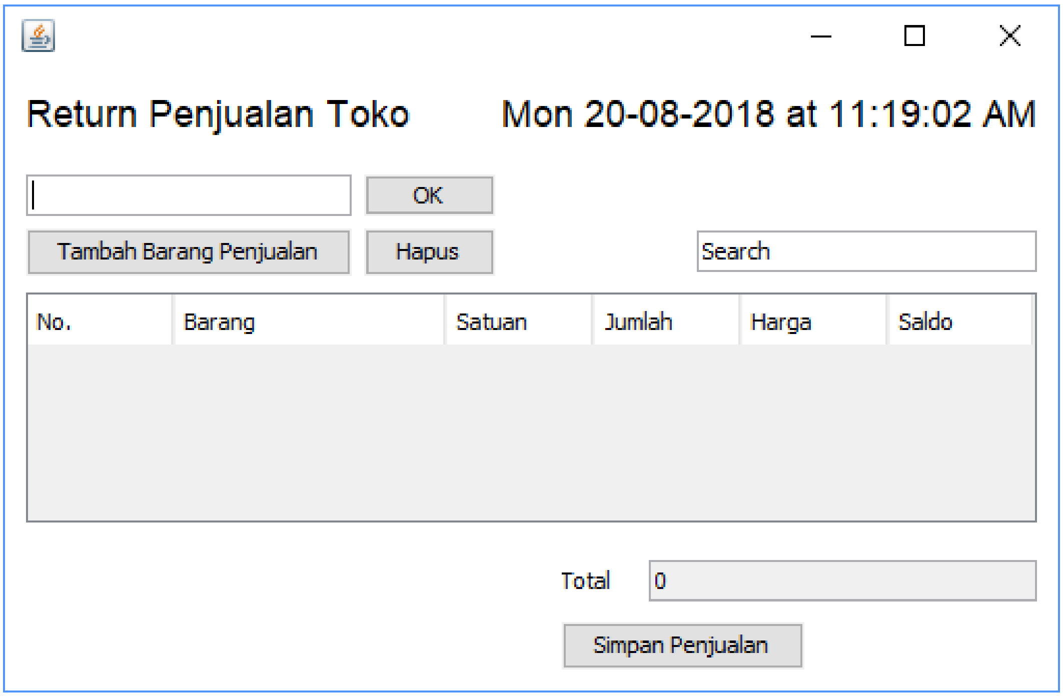
```
public class Toko_Transaksi2 extends javax.swing.JDialog {
    KeyboardFocusManager keyboardFocusManager = new
    DefaultKeyboardFocusManager();
    private ArrayList<ListTokoTransaksi> list;
    private ListTokoTransaksi listTokoTransaksi;
    private DefaultTableModel tabel;
    private Connect connection, connection1;
    private PreparedStatement PS, PSx, PS1;
    private TableModel model;
    private ResultSet hasil, hasil1;
    private ResultSet rs, rs1, rs2, rsx = null;
    private boolean merahAktif = false;
    private String id;
    private double harga, jumlah;
    private double totalBiaya = 0;
    private double subtotal = 0;
    private String dd;
    private double knvrsi = 0;
    private int tempid;
    private ArrayList<Integer> kdbarang;

    public Toko_Transaksi2() {
        initComponents();
        panelMerah.setVisible(false);
        connection = new Connect();
        kdbarang = new ArrayList<>();
        AutoCompleteDecorator.decorate(comBarang);
        AutoCompleteDecorator.decorate(comSatuan);
        AturlebarKolom();
        setTanggal();
        loadNumberTable();
        loadComTableBarang();
        String nofak = noFakturPenjualan();
        noFak.setText(nofak);
        int nocus = noCustomerHariIni();
        noCus.setText(String.valueOf(nocus));
    }
    private void hapussemuatabel() {
        double Hapus = JOptionPane.showConfirmDialog(null, "Apakah anda
        yakin mau menghapus semua data di tabel", "konfirmasi",
        JOptionPane.YES_NO_OPTION);
        if (Hapus == 0) {
            DefaultTableModel model = (DefaultTableModel)
            jTableTransaksi.getModel();
            for (int i = jTableTransaksi.getRowCount() - 1; i > -1; i--)
                { model.removeRow(i); }
            model.addRow(new Object[]{"1", "", "-- Pilih barang --", "",
            "0", "", "0", "0", "0"});
        } HitungSemua();
    }

    private void HitungSemua() {
        double subtotalfix = 0, grandtotal = 0, discount = 0, jumlahItem =
        0, jumlahQty = 0;
        if (!textDiscount.getText().equals("")) {
            discount = Double.parseDouble(textDiscount.getText()
            .toString());
        }
        if (jTableTransaksi.getRowCount() >= 1) {
            for (int i = jTableTransaksi.getRowCount() - 1; i > -1; i--) {
                double x = Double.parseDouble(jTableTransaksi.getValueAt
                (i, 7).toString()); subtotalfix += x;
            }
        } subTotalFix.setText(String.valueOf(subtotalfix));
        double i = jTableTransaksi.getRowCount();
        for (int j = 0; j < i; j++) {
            jumlahQty += Double.parseDouble(jTableTransaksi.getValueAt
            (j, 4).toString()); textJumlahItem.setText(String.valueOf(j));
        } textJumlahQty.setText(String.valueOf(jumlahQty));
        if (discount > subtotalfix) {
            JOptionPane.showMessageDialog(null, "Discount tidak boleh lebih
            besar daripada subtotal."); discount = 0;
            textDiscount.setText(String.valueOf(discount));
        } else {
            grandtotal = subtotalfix - discount;
            textGrandTotal.setText(String.valueOf(grandtotal));
            jTotal.setText(String.valueOf(grandtotal));
        }
    }

    private void AturlebarKolom() {
        TableColumn column;
        jTableTransaksi.setAutoResizeMode(javax.swing.JTable
        .AUTO_RESIZE_OFF);
        column = jTableTransaksi.getColumnModel().getColumn(0);
        column.setPreferredWidth(70);
        column = jTableTransaksi.getColumnModel().getColumn(1);
        column.setPreferredWidth(90);
        column = jTableTransaksi.getColumnModel().getColumn(2);
        column.setPreferredWidth(240);
        column = jTableTransaksi.getColumnModel().getColumn(3);
        column.setPreferredWidth(110);
        column = jTableTransaksi.getColumnModel().getColumn(4);
        column.setPreferredWidth(100);
        column = jTableTransaksi.getColumnModel().getColumn(5);
        column.setPreferredWidth(140);
        column = jTableTransaksi.getColumnModel().getColumn(6);
        column.setPreferredWidth(170);
        column = jTableTransaksi.getColumnModel().getColumn(7);
        column.setPreferredWidth(200);
        column = jTableTransaksi.getColumnModel().getColumn(8);
        column.setPreferredWidth(0);
        column.setMaxWidth(0);
    }
    private void loadNumberTable() {
        double baris = jTableTransaksi.getRowCount();
        for (int a = 0; a < baris; a++) {
            String nomor = String.valueOf(a + 1);
            jTableTransaksi.setValueAt(nomor + ".", a, 0);
        }
    }
    private void loadComTableBarang() {
        String sql;
        try {
            sql = "select kode_barang, nama_barang from barang";
            sql += " order by nama_barang asc";
            PS = connection.Connect().prepareStatement(sql);
            rs = PS.executeQuery();
            while (rs.next()) {
                comBarang.addItem(rs.getString(2));
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Eror_Line_188_" +
            e.getMessage());
        }
    }

    private void loadComSatuanBarang(String kodeBarang) { String kali = "";
        try {
            String sql = "select k.nama_konversi,b.harga_jual_2_barang *
            bk.jumlah_konversi from konversi k, barang_konversi bk,
            barang b where b.kode_barang "
            + "= bk.kode_barang and bk.kode_konversi = k.kode_konversi and
            b.kode_barang = '" + kodeBarang + "'";
            PS = connection.Connect().prepareStatement(sql);
            rs = PS.executeQuery(); comSatuan.removeAllItems();
            while (rs.next()) { comSatuan.addItem(rs.getString(1)); }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Eror_Line_204_" + e);
        }
    }

    private void loadHargaSatuanBarang() {
        String satuan = comSatuan.getSelectedItem().toString();
        String namaBarang = comBarang.getSelectedItem().toString();
        String kali = ""; String bagi = "";
        try {
            String sql = "select k.nama_konversi,b.harga_jual_2_barang *
            bk.jumlah_konversi,bl.kode_barang/bk.jumlah_konversi from
            konversi k, barang_konversi bk, barang b, barang_lokasi bl
            where b.kode_barang "
            + "= bk.kode_barang and bk.kode_konversi = k.kode_konversi and
            b.nama_barang = '" + namaBarang + "' and k.nama_konversi='" +
            satuan + "' and b.kode_barang = bl.kode_barang and
            bl.kode_lokasi=4"; System.out.println(sql);
            PS = connection.Connect().prepareStatement(sql);
            rs = PS.executeQuery();
            if (rs.next()) { String name = rs.getString(1);
                kali = rs.getString(2); bagi = rs.getString(3);
            }
            jTableTransaksi.setValueAt(kali,
            jTableTransaksi.getSelectedRow(), 5);
            jTableTransaksi.setValueAt(bagi,
            jTableTransaksi.getSelectedRow(), 8);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Eror_Line_228_" +
            e.getMessage());
        }
    }
    private void setTanggal() {
        try {
            Calendar ca = new GregorianCalendar();
            String day = ca.get(Calendar.DAY_OF_MONTH) + "";
            String month = ca.get(Calendar.MONTH) + 1 + "";
            String year = ca.get(Calendar.YEAR) + "";
            String hours = ca.get(Calendar.HOUR_OF_DAY) + "";
            String minutes = ca.get(Calendar.MINUTE) + "";
            String seconds = ca.get(Calendar.SECOND) + "";
            if (day.length() == 1) { day = "0" + day; }
            if (month.length() == 1) { month = "0" + month; }
            dd = year + "-" + month + "-" + day + " " + hours + ":" +
            minutes + ":" + seconds;
            Date date = new SimpleDateFormat
            ("yyyy-MM-dd hh:mm:ss").parse(dd);
            jDateChooser1.setDate(date);
        } catch (ParseException ex) {
            Logger.getLogger(Toko_Transaksi2.class
                    .getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void bExitTransaksiActionPerformed(java.awt.event.ActionEvent
    evt) { this.dispose(); }

    private void bClearTransaksiActionPerformed(java.awt.event.ActionEvent
    evt) { hapussemuatabel(); }

    private void bSaveTransaksiActionPerformed(java.awt.event.ActionEvent
evt) {
        for (int x = 0; x < jTableTransaksi.getRowCount() - 1; x++) {
            if (!jTableTransaksi.getValueAt(x, 6).toString().equals("")) {
                if (!String.valueOf(Double.parseDouble(jTableTransaksi
                .getValueAt(x, 5).toString()) * knvrsi)
                .equals(String.valueOf(Double.parseDouble(jTableTransaksi
                .getValueAt(x, 6).toString())))) {
                    JOptionPane.showMessageDialog(null, "Masukan Transaksi
                    Dengan Benar!\n"
                    + "Pilih Barang --> Pilih Satuan --> Masukan Jumlah
                    --> Enter");
                    DefaultTableModel model = (DefaultTableModel)
                    jTableTransaksi.getModel();
                    for (int i = jTableTransaksi.getRowCount() - 1;
                    i > -1; i--) { model.removeRow(i); }
                    model.addRow(new Object[]{"1", "0",
                    "-- Pilih barang --", "0", "0", "0", "0", "0", "0"});
                    HitungSemua();
                }
                if (Double.parseDouble(textGrandTotal.getText()
                .toString()) < 0) {
                    JOptionPane.showMessageDialog(null, "Potongan tidak
                    boleh lebih besar dari jumlah bayar.");
                } else if (Double.parseDouble(textGrandTotal.getText()
                  .toString()) == 0) {
                    JOptionPane.showMessageDialog(null, "Masukan
                    Transaksi!");
                } else {
                    jTextField3.setText("0");
                    if (!textGrandTotal.getText().toString().equals("0")) {
                        jTextField2.setText(textGrandTotal.getText()
                        .toString());
                        double kembalian = 0;
                        double grandtotal = Double.parseDouble(jTextField2
                        .getText().toString());
                        Double.parseDouble(jTextField2.getText()
                        .toString());
                        jTextField4.setText(String.valueOf(kembalian));
                        jComboBox1.addItem("Cash");dPembayaran.show();
                        dPembayaran.setLocationRelativeTo(null);
                    } else {
                        JOptionPane.showMessageDialog(null, "Silahkan
                        Masukan Item terlebih dahulu!");
                    }
                }
            } else {
                JOptionPane.showMessageDialog(null, "Masukan Transaksi
                Dengan Benar!\n"
                + "Pilih Barang --> Pilih Satuan --> Masukan Jumlah -->
                Enter");
                DefaultTableModel model = (DefaultTableModel)
                jTableTransaksi.getModel();
                for (int i = jTableTransaksi.getRowCount() - 1;
                i > -1; i--) { del.removeRow(i); }
                model.addRow(new Object[]{"1", "0", "-- Pilih barang --",
                "0", "0", "0", "0", "0", "0"});
                HitungSemua();
            }
        }
    }

    private void bBatalActionPerformed(java.awt.event.ActionEvent evt) {
        dPembayaran.dispose();
    }

    private void SimpanTokoPenjualan() {
        String nofak = noFakturPenjualan();
        double total = Double.parseDouble(jTotal.getText());
        PS = null;
        try {
            String sql = "insert into toko_penjualan
            (no_faktur_toko_penjualan,tgl_toko_penjualan,kode_pegawai,
            total) values (?,?,?,?)";
            PS = connection.Connect().prepareStatement(sql);
            PS.setString(1, nofak);
            PS.setString(2, dd);
            PS.setInt(3, 1);
            PS.setDouble(4, total);
            PS.executeUpdate();
            simpanTokoPenjualanDetail(nofak);
        } catch (SQLException e) {
            System.out.print("Toko Transaksi : " + e.toString());
        }
    }

    private void jTableTransaksiKeyPressed(java.awt.event.KeyEvent evt) {
        DefaultTableModel model = (DefaultTableModel)
        jTableTransaksi.getModel();
        int selectedRow = jTableTransaksi.getSelectedRow();
        double baris = jTableTransaksi.getRowCount();
        double jumlah = 0;
        double harga = 0;
        double qty = 0;
        TableModel tabelModel;
        tabelModel = jTableTransaksi.getModel();
        jumlah = Double.parseDouble(tabelModel.getValueAt(jTableTransaksi
        .getSelectedRow(), 4).toString());
        harga = Double.parseDouble(tabelModel.getValueAt(jTableTransaksi
        .getSelectedRow(), 6).toString());
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            if(Double.parseDouble(tabelModel.getValueAt(jTableTransaksi
            .getSelectedRow(), 4).toString()) > Double.parseDouble
            (tabelModel.getValueAt(jTableTransaksi.getSelectedRow(),
            8).toString())) {
                JOptionPane.showMessageDialog(null, "Stock tersedia untuk
                satuan (" + comSatuan.getSelectedItem() + ") : " +
                Double.parseDouble(tabelModel.getValueAt(jTableTransaksi.
                getSelectedRow(), 8).toString()), "", 2);
                jTableTransaksi.setValueAt(Double.parseDouble(tabelModel
                .getValueAt(jTableTransaksi.getSelectedRow(), 8)
                .toString()), selectedRow, 4);
            } else {
                jumlah = Double.parseDouble(tabelModel.getValueAt
                (jTableTransaksi.getSelectedRow(), 4).toString());
                harga = Double.parseDouble(tabelModel.getValueAt
                (jTableTransaksi.getSelectedRow(), 6).toString());
                subtotal = jumlah * harga; tabelModel.setValueAt(subtotal,
                jTableTransaksi.getSelectedRow(), 7);
                if (tabelModel.getValueAt(jTableTransaksi.getSelectedRow(),
                7).toString().equals("0.0")) {
                    JOptionPane.showMessageDialog(null, "Data Terakhir
                    Tidak Boleh kosong", "", 2);
                } else {
                    if (Double.parseDouble(tabelModel.getValueAt
                    (jTableTransaksi.getRowCount() - 1, 7).toString())
                    != 0) {
                        addRow(model);
                    }
                }
                HitungSemua();
            }
        } else if (evt.getKeyCode() == KeyEvent.VK_DELETE) {
            if (jTableTransaksi.getRowCount() - 1 == -1) {
                JOptionPane.showMessageDialog(null, "Data didalam tabel
                telah tiada.", "", 2);
            } else {
                removeSelectedRows(jTableTransaksi); HitungSemua();
            }
        } else if (evt.getKeyCode() == KeyEvent.VK_INSERT) {
            if (jTableTransaksi.getRowCount() - 1 == -1) { addRow(model); }
        } loadNumberTable();
    }

    private void addRow(DefaultTableModel model) {
        model.addRow(new Object[]{"1", "0", "-- Pilih barang --", "0", "0",
        "0", "0", "0", "0"});
    }

    private static String rptabel(String b) {
        b = b.replace(",", "");
        b = NumberFormat.getNumberInstance(Locale.getDefault())
        .format(Double.parseDouble(b)); return b;
    }

    private void removeSelectedRows(JTable table) {
        int Hapus = 1;
        Hapus = JOptionPane.showConfirmDialog(null, "Apakah anda yakin mau
        menghapus baris ?", "konfirmasi", JOptionPane.YES_NO_OPTION);
        if (Hapus == 0) {
            DefaultTableModel model = (DefaultTableModel)
            this.jTableTransaksi.getModel();
            int[] rows = table.getSelectedRows();
            for (int i = 0; i < rows.length; i++) {
                model.removeRow(rows[i] - i);
            }
        }
    }
```

```
private double DapatKodeKonversi(String kodeBarang) {
    double kodeBarangKonversi = 0;
    try {
        String sql = "select kode_barang_konversi from barang_konversi
        where kode_barang ='" + kodeBarang + "'";
        PS = connection.Connect().prepareStatement(sql);
        rs = PS.executeQuery();
        if (rs.next()) { kodeBarangKonversi = rs.getInt(1); }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Eror
        DapatKodeKonversi" + e);
    } return kodeBarangKonversi;
}

private void simpanTokoPenjualanDetail(String nofak) {
    PS = null; int i = jTableTransaksi.getRowCount() - 1;
    try {
        for (int j = 0; j < i; j++) {
            String kodeBarang = jTableTransaksi.getValueAt(j, 1)
            .toString();
            double jlh = knvrsi * Double.parseDouble(jTableTransaksi
            .getValueAt(j, 4).toString());
            double hrgaFix = Double.parseDouble(jTableTransaksi
            .getValueAt(j, 6).toString());
            double hrga2 = Double.parseDouble(jTableTransaksi
            .getValueAt(j, 5).toString());
            double kodeBarangKonversi = DapatKodeKonversi(kodeBarang);
            String data2 = "insert into toko_penjualan_detail
            (no_faktur_toko_penjualan,kode_barang,jumlah_barang,
            harga_barang,kode_barang_konversi) values (?,?,?,?,?)";
            PS = connection.Connect().prepareStatement(data2);
            PS.setString(1, nofak);
            PS.setString(2, kodeBarang);
            PS.setDouble(3, jlh);
            PS.setDouble(4, hrga2);
            PS.setDouble(5, kodeBarangKonversi);
            PS.executeUpdate();
        }
    } catch (Exception e) {
        System.out.println("Toko_Retur/simpanTokoPenjualanDetail- "
        + e);
    }
}

private String selectLastData() {
    String lastNo = "";
    try {
        String data = "SELECT no_faktur_toko_penjualan FROM
        toko_penjualan ORDER BY id_toko_penjualan DESC LIMIT 1";
        hasil = connection.ambilData(data);
        if (hasil.next()) {
            String nomor = hasil.getString("no_faktur_toko_penjualan");
            int noLama = Integer.parseInt(nomor.substring
            (nomor.length() - 4)); noLama++;
            String no = Integer.toString(noLama);
            if (no.length() == 1) { no = "000" + no; }
            else if (no.length() == 2) { no = "00" + no; }
            else if (no.length() == 3) { no = "0" + no; } lastNo = no;
        } else { lastNo = "0001"; }
    } catch (Exception e) {
        System.out.print("TokoPenjualan/selectLastData - " + e);
    } return lastNo;
}
```

```
    private int selectLastCustomer(String tanggal) {
        int lastNo = 0;
        try {
            String data = "select count(id_toko_penjualan) from
            toko_penjualan"
            + " where tgl_toko_penjualan like '%" + tanggal + "%' ";
            hasil = connection.ambilData(data);
            if (hasil.next()) {
                int nomor = hasil.getInt(1); nomor = nomor + 1;
                lastNo = nomor;
            } else { lastNo = 1; }
        } catch (Exception e) {
            System.out.println("TokoPenjualan/selectLastCustomer - " + e);
        } return lastNo;
    }

    private String noFakturPenjualan() {
        String year = Integer.toString(Calendar.getInstance()
        .get(Calendar.YEAR));
        String no = "PJ" + year.substring(year.length() - 2) + "-" +
        selectLastData(); return no;
    }

    private int noCustomerHariIni() {
        SimpleDateFormat dt = new SimpleDateFormat("yyyy-MM-dd");
        String tanggal = dt.format(jDateChooser1.getDate());
        int no = selectLastCustomer(tanggal); return no;
    }

    private void comSatuanActionPerformed(java.awt.event.ActionEvent evt) {
        String sat = ""; String bar = ""; String kali = "";
        String bagi = "";
        jTableTransaksi.setValueAt(0, jTableTransaksi.getSelectedRow(), 7);
        if (comSatuan.getSelectedItem() == null) {
        } else {
            if (comBarang.getSelectedItem() == null) {}
            else {
                sat = comSatuan.getSelectedItem().toString();
                bar = comBarang.getSelectedItem().toString();
                try {
                    String sql = "select k.nama_konversi,
                    b.harga_jual_2_barang*bk.jumlah_konversi,
                    bl.jumlah/bk.jumlah_konversi, bk.jumlah_konversi
                    from konversi k, barang_konversi bk, barang b,
                    barang_lokasi bl where bl.kode_lokasi=4 and
                    b.kode_barang = bl.kode_barang and b.kode_barang "
                    + "= bk.kode_barang and bk.kode_konversi =
                    k.kode_konversi and b.nama_barang = '" + bar + "'
                    and k.nama_konversi='" + sat + "'";
                    PSx = connection.Connect().prepareStatement(sql);
                    rsx = PSx.executeQuery();
                    while (rsx.next()) {
                        String name = rsx.getString(1);
                        kali = rsx.getString(2);
                        bagi = String.valueOf(Math.floor(rsx.getInt(3)));
                        knvrsi = rsx.getInt(4);
                    }
                    jTableTransaksi.setValueAt(kali,
                    jTableTransaksi.getSelectedRow(), 6);
                    jTableTransaksi.setValueAt(bagi,
                    jTableTransaksi.getSelectedRow(), 8);
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null,
                    "Eror_Line_1282_" + e);
                }
            }
        }
    }
    private void textDiscountKeyReleased(java.awt.event.KeyEvent evt) {
        textGrandTotal.setText(String.valueOf(Double.parseDouble
        (subTotalFix.getText().toString()) - Double.parseDouble
        (textDiscount.getText().toString())));
    }

    private boolean check(int id) {
        for (int x = 0; x < jTableTransaksi.getRowCount() - 1; x++) {
            if (Integer.parseInt(jTableTransaksi.getValueAt
            (x, 1).toString()) == id) {
                JOptionPane.showMessageDialog(rootPane, "Barang : " +
                comBarang.getSelectedItem().toString() + "\nsudah ada!");
                comBarang.setSelectedIndex(0); return false;
            }
        } return true;
    }

    private void comBarangItemStateChanged(java.awt.event.ItemEvent evt) {
        if (evt.getStateChange() == ItemEvent.SELECTED) {
            int selectedRow = jTableTransaksi.getSelectedRow();
            boolean pass = true;
            if (selectedRow != -1) {
                try {
                    String sql = "select b.kode_barang, b.nama_barang,
                    b.harga_jual_1_barang, "
                    + "b.harga_jual_2_barang, b.harga_jual_3_barang,
                    b.harga_rata_rata_barang, "
                    + "bl.jumlah from barang b, barang_lokasi bl, lokasi l
                    where b.kode_barang = bl.kode_barang and "
                    + "bl.kode_lokasi = l.kode_lokasi and
                    l.nama_lokasi='toko' and b.nama_barang = '" + comBarang
                    .getSelectedItem() + "'";
                    PS = connection.Connect().prepareStatement(sql);
                    rs = PS.executeQuery();
                    if (rs.next()) {
                        id = rs.getString(1);
                        jumlah = rs.getInt(7);
                        harga = rs.getInt(4);
                        tempid = Integer.parseInt(id);
                        if (jTableTransaksi.getRowCount() > 1) {
                            if (check(tempid)) { pass = true;}
                            else { pass = false; }
                        }
                        if (pass) {
                            jTableTransaksi.setValueAt(id, selectedRow, 1);
                            jTableTransaksi.setValueAt(0, selectedRow, 4);
                            jTableTransaksi.setValueAt(harga,
                            selectedRow, 5);
                            jTableTransaksi.setValueAt(0, selectedRow, 7);
                            loadComSatuanBarang(jTableTransaksi
                            .getValueAt(selectedRow, 1).toString());
                            jTableTransaksi.setValueAt(comSatuan
                            .getSelectedItem(), selectedRow, 3);
                        }
                    }
                } catch (SQLException e) {
                    JOptionPane.showMessageDialog(null, "Error : " +
                    e.getMessage());
                }
            }
        }
    }
    private void bSimpanMouseClicked(java.awt.event.MouseEvent evt) {
        if (jTextField3.getText().toString().equals("0")) {
            JOptionPane.showMessageDialog(null, "Bayar tidak boleh nol!");
        } else if (jTextField3.getText() == null) {
            JOptionPane.showMessageDialog(null, "Bayar tidak boleh
            kosong!");
        } else if (Double.parseDouble(jTextField3.getText().toString())
        < Double.parseDouble(jTextField2.getText().toString())) {
            JOptionPane.showMessageDialog(null, "Uang yang dibayar kurang
            dari yang harus dibayar!");
        } else {
            SimpanTokoPenjualan(); dPembayaran.dispose();
            this.dispose(); Toko_Transaksi2 x = new Toko_Transaksi2();
            x.setLocationRelativeTo(this); x.setVisible(true);
        }
    }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)
    {
        for (int x = 0; x < jTableTransaksi.getRowCount() - 1; x++) {
            if (!jTableTransaksi.getValueAt(x, 6).toString().equals("")) {
                if (!String.valueOf(Double.parseDouble(jTableTransaksi
                .getValueAt(x, 5).toString()) * knvrsi)
                .equals(String.valueOf(Double.parseDouble
                (jTableTransaksi.getValueAt(x, 6).toString())))) {
                    System.out.println(jTableTransaksi.getValueAt(x, 5) + "
                    *" + knvrsi + " : " +jTableTransaksi.getValueAt(x, 6));
                    JOptionPane.showMessageDialog(null, "Masukan Transaksi
                    Dengan Benar!\n"
                    + "Pilih Barang --> Pilih Satuan --> Masukan Jumlah -->
                    Enter");
                    DefaultTableModel model = (DefaultTableModel)
                    jTableTransaksi.getModel();
                    for(int i=jTableTransaksi.getRowCount()-1;i > -1;i--) {
                        model.removeRow(i);
                    }
                    model.addRow(new Object[]{"1", "0", "-- Pilih barang –
                    ", "0", "0", "0", "0", "0", "0"});
                    HitungSemua();
                }
                if (Double.parseDouble(textGrandTotal.getText().toString())
                < 0) {
                    JOptionPane.showMessageDialog(null, "Potongan tidak
                    boleh lebih besar dari jumlah bayar.");
                } else if (Double.parseDouble(textGrandTotal.getText()
                .toString()) == 0) {
                    JOptionPane.showMessageDialog(null, "Masukan
                    Transaksi!");
                } else {
                    jTextField3.setText("0");
                    if (!textGrandTotal.getText().toString().equals("0")) {
                        jTextField2.setText(textGrandTotal.getText()
                        .toString());
                        double kembalian = 0;
                        double grandtotal = Double.parseDouble(jTextField2
                        .getText().toString());
                        Double.parseDouble(jTextField2.getText()
                        .toString());
                        jTextField4.setText(String.valueOf(kembalian));
                        jComboBox1.addItem("Cash"); dPembayaran.show();
                        dPembayaran.setLocationRelativeTo(null);
                    } else {
                        JOptionPane.showMessageDialog(null, "Silahkan
                        Masukan Item terlebih dahulu!");
                    }
                }
            } else {
                JOptionPane.showMessageDialog(null, "Masukan Transaksi
                Dengan Benar!\n"
```

```
                        + "Pilih Barang --> Pilih Satuan --> Masukan Jumlah
                        --> Enter");
                DefaultTableModel model = (DefaultTableModel)
                jTableTransaksi.getModel();
                for(int i = jTableTransaksi.getRowCount()-1; i > -1;i--) {
                    model.removeRow(i);
                }
                model.addRow(new Object[]{"1", "0", "-- Pilih barang --",
                "0", "0", "0", "0", "0", "0"}); HitungSemua();
            }
        }
    }

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
    { hapussemuatabel(); }

    private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt)
    { this.dispose(); }

    private void jTextField3KeyReleased(java.awt.event.KeyEvent evt) {
        jTextField4.setText(String.valueOf(Double.parseDouble
        (jTextField3.getText().toString()) -
        Double.parseDouble(jTextField2.getText().toString())));
    }

    private javax.swing.JButton bBatal;
    private javax.swing.JButton bClearTransaksi;
    private javax.swing.JButton bExitTransaksi;
    private javax.swing.JButton bSaveTransaksi;
    private javax.swing.JButton bSimpan;
    private javax.swing.JComboBox<String> comBarang;
    private javax.swing.JComboBox<String> comSatuan;
    private javax.swing.JDialog dPembayaran;
    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JCheckBox jCheckBox2;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JComboBox jComboBox4;
    private com.toedter.calendar.JDateChooser jDateChooser1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel100;
    private javax.swing.JLabel jLabel101;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JLabel jLabel26;
    private javax.swing.JLabel jLabel27;
    private javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel29;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JLabel jLabel31;
    private javax.swing.JLabel jLabel96;
    private javax.swing.JLabel jLabel97;
    private javax.swing.JLabel jLabel98;
    private javax.swing.JLabel jLabel99;
    private javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JMenuItem jMenuItem3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JPanel jPanel7;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JScrollPane jScrollPane5;
    private javax.swing.JSeparator jSeparator18;
    private javax.swing.JSeparator jSeparator19;
    private javax.swing.JTable jTableTransaksi;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JLabel jTotal;
    private javax.swing.JTextField noCus;
    private javax.swing.JTextField noFak;
    private javax.swing.JPanel panelMerah;
    private javax.swing.JMenu save;
    private javax.swing.JTextField subTotalFix;
    private javax.swing.JTextField textDiscount;
    private javax.swing.JTextField textGrandTotal;
    private javax.swing.JTextField textHargaJual1;
    private javax.swing.JTextField textHargaJual2;
    private javax.swing.JTextField textHargaJual3;
    private javax.swing.JTextField textHargaJualF;
    private javax.swing.JTextField textJumlahItem;
    private javax.swing.JTextField textJumlahQty;
    private javax.swing.JTextField textNoJual;
    private javax.swing.JTextField textQty;
    private javax.swing.JTextField textStaff;
    private javax.swing.JTextField textTanggal;
    private javax.swing.JTextField textToko;
    private javax.swing.JTextField textTunai;
    private javax.swing.JTextField txtAlamat;
    private javax.swing.JTextField txtNama;
```

4.4.10. Form Toko Penjualan Return



Return Penjualan Toko

Mon 20-08-2018 at 11:19:02 AM

No.	Barang	Satuan	Jumlah	Harga	Saldo

Total 0

Gambar 4.21. Form Toko Penjualan Return

```
public class Toko_ReturPenjualan extends javax.swing.JFrame {
    private HashMap konversi, idReturn, jmlBarang, hashCode;
    private DefaultTableModel tabelBarang, tabelRetur;
    private ResultSet hasil;
    public Statement stmt;
    private Connect connection;
    private String code_barang, code_konversi;
    private String noNota = "";
    private PreparedStatement PS;

    public Toko_ReturPenjualan() {
        initComponents();
        this.setLocationRelativeTo(null);
        connection = new Connect();
        konversi = new HashMap(); idReturn = new HashMap();
        jmlBarang = new HashMap(); hashCode = new HashMap();
        tabelBarang = new DefaultTableModel(new
        String[]{"No.","Kode","Nama Barang"},0);
        jTable2.setModel(tabelBarang);
        jTable2.getColumnModel().getColumn(0).setPreferredWidth(5);
        jTable2.getColumnModel().getColumn(1).setPreferredWidth(5);
        tabelRetur = new DefaultTableModel(new
        String[]{"No.","Barang","Satuan","Jumlah","Harga","Saldo"},
        0);
        jTable3.setModel(tabelRetur);
        jTable3.getColumnModel().getColumn(0).setPreferredWidth(1);
        jTable3.getColumnModel().getColumn(2).setPreferredWidth(5);
        jTable3.getColumnModel().getColumn(3).setPreferredWidth(5);
        jTable3.getColumnModel().getColumn(4).setPreferredWidth(5);
        jTable3.getColumnModel().getColumn(5).setPreferredWidth(5);
        showDate();
        isiTabelReturn("*");
    }

    public void setPlaceHolder(javax.swing.JTextField a, String b)
    {
        if(b == null) {
            b = "Search";}
        if(a.getText().equals(b)) {
            a.setText("");}
        else if(a.getText() == null || a.getText().equals("")) {
            a.setText(b);}
    }

    public void showDate() {
        java.util.Timer t = new java.util.Timer();
        t.schedule(new TimerTask() {
            @Override
            public void run() {
                java.util.Date d = new java.util.Date( );
                java.text.SimpleDateFormat dt = new
                  java.text.SimpleDateFormat ("E dd-MM-yyyy 'at'
                  hh:mm:ss a");
                vdatetime.setText(dt.format(d));
            }
        }, 500, 500);
    }
```

```
private String tampilTabelDataBarang(String search){
     String data = "";
     try{
          data = "SELECT TPD.kode_barang, B.nama_barang,
              TPD.jumlah_barang "
          + "FROM barang B, toko_penjualan_detail TPD "
          + "WHERE no_faktur_toko_penjualan = '"+noNota+"' AND "
          + "TPD.kode_barang = B.kode_barang "
          +(search.equalsIgnoreCase("*") ? "" : "AND
          B.nama_barang LIKE '%" +search+ "%' " )+" "
          + "ORDER BY TPD.kode_barang";
          hasil = connection.ambilData(data);
          setModelTabelBarang(hasil);

     } catch(Exception e){
          System.out.println("Error/Toko_ReturPenjualan/
               tampilTabelDataBarang -> "+e);
     } return data;
}

public void setModelTabelBarang(ResultSet hasil){
     try{
          int no = 1;
          while (hasil.next()){
              String kode = hasil.getString("kode_barang");
              String nama = hasil.getString("nama_barang");
              tabelBarang.addRow(new Object[]{no,kode,nama});
              jmlBarang.put(kode,
              hasil.getString("jumlah_barang"));
              no++;
          }
     } catch(Exception e){
          System.out.println("Error /Toko_ReturPenjualan/
          setModel -> "+e);
     }
}

private void deleteTabel(DefaultTableModel NamaTabel){
   int baris = NamaTabel.getRowCount();
   for (int i = 0; i < baris; i++) {
       NamaTabel.removeRow(0);
   }
}

private void isiPilihBarang(String kode){
   try{
          String data = "SELECT TPD.kode_barang, B.nama_barang,
          B.harga_jual_2_barang, TPD.kode_barang_konversi,
          K.nama_konversi "
         + "FROM barang B, barang_konversi BK,
          toko_penjualan_detail TPD, konversi K "
         + "WHERE B.kode_barang = TPD.kode_barang AND
          TPD.kode_barang_konversi = BK.kode_barang_konversi "
          + "AND BK.kode_konversi = k.kode_konversi AND
          TPD.kode_barang = '"+kode+"' ";
          hasil = connection.ambilData(data);
          while (hasil.next()){
            code_konversi =
              hasil.getString("kode_barang_konversi");
              vNamaBarang.setText(hasil.getString
              ("nama_barang"));
```

```
             vHarga.setText(String.valueOf(Math.round(
             hasil.getInt("harga_jual_2_barang"))));
             jLabel8.setText("*Jumlah Maksimal Return : "
             +jmlBarang.get(kode));
             code_barang = kode;
             vSatuan.removeAllItems();
             vSatuan.addItem(hasil.getString("nama_konversi"));}
     } catch(Exception e){
         System.out.println("Toko_Retur/isiPilihBarang "+e);
     }
}

private String selectLastDataDetailReturn(String year){
   String lastNo = "";
   try{
       String data = "SELECT no_faktur_toko_penjualan_return "
         + "FROM toko_penjualan_return "
         + "ORDER BY id_toko_penjualan_return DESC LIMIT 1";
         hasil = connection.ambilData(data);
         if(hasil.next()){
         String nomor = hasil.getString
             ("no_faktur_toko_penjualan_return");
             if(nomor.substring(2, 4)
             .equalsIgnoreCase(year)){
                   int noLama = Integer.parseInt
                   (nomor.substring(nomor.length() - 5))
                 noLama++;
                 String no = Integer.toString(noLama);
                 if(no.length() == 1){ no = "0000"+no; }
                   else if(no.length() == 2){ no = "000"+no; }
                   else if(no.length() == 3){ no = "00"+no; }
                   else if(no.length() == 4){ no = "0"+no; }
                   lastNo = no;
             } else{ lastNo = "00001"; }
         } else { lastNo = "00001"; }
     } catch(Exception e){
         System.out.println("TokoRetur/selectLastData - "+e);
     } return lastNo;
}

private void selectKonversi(String kode_barang){
   vSatuan.removeAllItems();
   try{
         String data = "SELECT k.kode_konversi, k.nama_konversi"
         + " FROM konversi k, barang_konversi bk"
         + " WHERE bk.kode_konversi = k.kode_konversi AND
         bk.kode_barang = '"+kode_barang+"' ";
         hasil = connection.ambilData(data);
         while(hasil.next()){
             String kode = hasil.getString("kode_konversi");
             String nama = hasil.getString("nama_konversi");
             konversi.put(nama, kode);
             vSatuan.addItem(nama);
         }
     } catch(Exception e){
         System.out.println("Toko_Return/selectKonversi - "+e);
     }
}
```

```
private void isiTabelReturn(String search){
     ResultSet nama_barang, harga_barang;
     try{
          String data = "SELECT TKR.id_toko_keranjang,
          B.nama_barang, K.nama_konversi, TKR.jumlah_barang,
          B.harga_jual_2_barang, TKR.harga_barang "
          + "FROM barang B, konversi K, barang_konversi BK,
          toko_keranjang_return TKR "
          + "WHERE TKR.kode_barang = B.kode_barang AND
          TKR.kode_barang_konversi = BK.kode_barang_konversi "
          + "AND BK.kode_konversi = K.kode_konversi "
          + (search.equalsIgnoreCase("*") ? "" : "AND
          B.nama_barang LIKE '%"+search+"%' ")+" "
          + "ORDER BY TKR.id_toko_keranjang";
          hasil = connection.ambilData(data);
          setModelTabelReturn(hasil);
     } catch(Exception e){
          System.out.println("Toko_Return/isiTabelBarang - "+e);
     }
}

private void setModelTabelReturn(ResultSet hasil){
     try{
          int no = 1;
          while(hasil.next()){
              String barang = hasil.getString("nama_barang");
              String satuan = hasil.getString("nama_konversi");
              String jumlah = hasil.getString("jumlah_barang");
              String harga = String.valueOf
              (Math.round(hasil.getInt("harga_jual_2_barang")));
              String total = hasil.getString("harga_barang");
              tabelRetur.addRow(new Object[]
              {no,barang,satuan,jumlah,harga,total});
              idReturn.put(no,hasil.getString
              ("id_toko_keranjang"));
              no++;
          } setTotalReturn();
     } catch(Exception e){
          System.out.println("Toko_Return/setModelTabelReturn -
          "+e);
     }
}

private void setTotalReturn(){
     try{
          int total = 0;
          String data = "SELECT harga_barang "
          + "FROM toko_keranjang_return";
          hasil = connection.ambilData(data);
          while(hasil.next()){
              total = total + hasil.getInt("harga_barang"); }
          jTextField5.setText(Integer.toString(total));
     } catch(Exception e){
          System.out.println("Toko_Return/setTotalReturn - "+e);
     }
}
```

```
private void inputTokoKeranjangReturn(String codeBarang){
     int total = Integer.parseInt(vJumlah.getText()) *
     Integer.parseInt(vHarga.getText());
     try{
          String data = "INSERT INTO toko_keranjang_return
          (kode_barang, jumlah_barang, harga_barang,
          kode_barang_konversi) "
          + "VALUES ('"+codeBarang+"', '"+vJumlah.getText()+"',
          '"+total+"', '"+code_konversi+"')";
          connection.simpanData(data);
          dPilihBarang.dispose();
     } catch(Exception e){
          System.out.println("Toko_Retur/inputTokoKeranjangReturn
          - "+e);
     }
}

private String noFakturTPReturn(){
     String year = Integer.toString(Calendar.getInstance()
     .get(Calendar.YEAR));
     String lastY = year.substring(year.length()-2);
     String no = "RJ"+ lastY +"-
     "+selectLastDataDetailReturn(lastY);
     return no;
}

private void simpanTokoDetailReturn(String noReturn){
     PS = null;
     String kode_pegawai = "1";
     java.util.Date d = new java.util.Date();
     java.text.SimpleDateFormat dt = new
     java.text.SimpleDateFormat ("yyyy-MM-dd hh:mm:ss");
     String date = dt.format(d);
     try{
          String data = "INSERT INTO toko_penjualan_return
          (no_faktur_toko_penjualan_return, "
          + "tgl_toko_penjualan_return, kode_pegawai) "
        + "VALUES (?, ?, ?)";
          PS = connection.Connect().prepareStatement(data);
          PS.setString(1, noReturn); PS.setString(2, date);
          PS.setString(3, kode_pegawai); PS.executeUpdate();
          simpanTokoPenjualanDetailReturn(noReturn);
     } catch(Exception e){
          System.out.println("Toko_Retur/inputTokoDetailReturn -
          "+e);
     }
}

private void simpanTokoPenjualanDetailReturn(String noReturn){
     PS = null;
     int i = jTable3.getRowCount();
     try{
          for(int j = 0; j < i; j++){
              String namaB = jTable3.getValueAt(j, 1).toString();
              String kodeB = selectNoBarang(namaB);
              String namaS = jTable3.getValueAt(j, 2).toString();
              String kodeS = selectKodeKonversi(namaS);
              String data2 = "INSERT INTO
              toko_penjualan_detail_return"
              + "(no_faktur_toko_penjualan_return, kode_barang, "
              + "jumlah_barang, harga_barang,
              kode_barang_konversi) "
            + "VALUES (?, ?, ?, ?, ?)";
```

```
                 PS = connection.Connect().prepareStatement(data2);
                 PS.setString(1, noReturn);
                 PS.setString(2, kodeB);
                 PS.setString(3, jTable3.getValueAt(j, 3)
                 .toString());
                 PS.setString(4, jTable3.getValueAt(j, 5)
                 .toString());
                 PS.setString(5, kodeS);
                 PS.executeUpdate();
             } JOptionPane.showMessageDialog(null, "Return Berhasil
             Disimpan");
             vNota.setText("Nomor Nota"); vNota.setEditable(true);
             vOk.setText("OK"); deleteKeranjangReturn();
             deleteTabel(tabelRetur); jTextField5.setText("0");
         } catch(Exception e){
             System.out.println("Toko_Retur/
             simpanTokoPenjualanDetailReturn - "+e);
         }
    }

    private void deleteKeranjangReturn(){
         try{
             String data = "DELETE FROM toko_keranjang_return";
             connection.simpanData(data);
         } catch(Exception e){
             System.out.println("Toko_Retur/deleteKeranjangReturn -
             "+e);
         }
    }

    private String selectNoBarang(String namaB){
         String noB = "";
         try{
             String data = "SELECT kode_barang "
            + "FROM barang "
            + "WHERE nama_barang = ?";
             PS = connection.Connect().prepareStatement(data);
             PS.setString(1, namaB);
             hasil = PS.executeQuery();
             while(hasil.next()){
                 noB = hasil.getString("kode_barang"); }
         } catch(Exception e){
             System.out.println("Toko_Retur/selectNoBarang - "+e);
         } return noB;
    }

    private String selectKodeKonversi(String namaK){
         String noK = "";
         try{
             String data = "SELECT kode_konversi "
             + "FROM konversi "
            + "WHERE nama_konversi = ?";
             PS = connection.Connect().prepareStatement(data);
             PS.setString(1, namaK); hasil = PS.executeQuery();
             while(hasil.next()){
                 noK = hasil.getString("kode_konversi"); }
         } catch(Exception e){
             System.out.println("Toko_Retur/selectNoBarang - "+e);
         } return noK;
    }
```

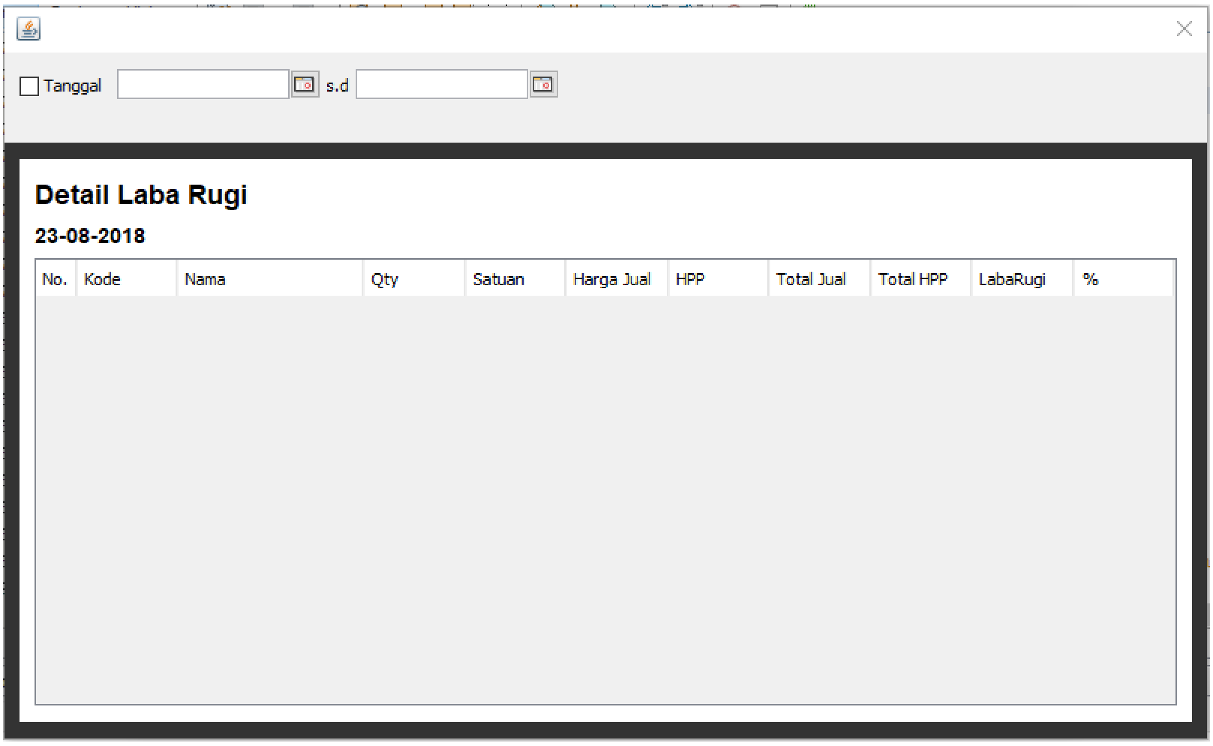
```
     private void hapusBarang(String kode){
          try{
               String data = "DELETE FROM toko_keranjang_return "
               + "WHERE id_toko_keranjang = '"+kode+"' ";
               connection.simpanData(data);
               JOptionPane.showMessageDialog(null, "Barang berhasil
               dihapus");
               deleteTabel(tabelRetur); isiTabelReturn("*");
          } catch(Exception e){
               System.out.println("Toko_Return/hapusBarang - "+e);
          }
     }

     private void hitungTotalHarga(){
          int harga = Integer.parseInt(vHarga.getText().toString());
          int jumlah = Integer.parseInt(vJumlah.getText()
          .toString());
          int total = harga*jumlah;
          vTotalHarga.setText(Integer.toString(total));
     }

     private String selectNoNota(String nomor){
          String noNota = "";
          try{
               String data = "SELECT no_faktur_toko_penjualan "
              + "FROM toko_penjualan "
              + "WHERE no_faktur_toko_penjualan = '"+nomor+"' ";
               hasil = connection.ambilData(data);
               if(hasil.next()){
                   noNota = hasil.getString
                   ("no_faktur_toko_penjualan");
               } else{
                 JOptionPane.showMessageDialog(null, "Nomor Nota
                   Tidak ditemukan");
               }
          } catch(Exception e){
               System.out.println("Toko_Retur/selectNoNota "+e);
          } return noNota;
    }


     private void vNamaBarangMouseClicked(java.awt.event.
     MouseEvent evt) {
          deleteTabel(tabelBarang); tampilTabelDataBarang("*");
          dPilihDataBarang.show();
          dPilihDataBarang.setLocationRelativeTo(null);
     }

     private void btnBatalActionPerformed(java.awt.event.
     ActionEvent evt) { dPilihBarang.dispose(); }

     private void vSearchFocusGained(java.awt.event.
     FocusEvent evt) { setPlaceHolder(vSearch, null); }

    private void vSearchFocusLost(java.awt.event.FocusEvent evt) {
        setPlaceHolder(vSearch, null);
    }

     private void vSearchReturFocusLost(java.awt.event.FocusEvent
     evt) { setPlaceHolder(vSearchRetur, null); }
```

```
private void vSearchReturFocusGained(java.awt.event.FocusEvent
evt) { setPlaceHolder(vSearchRetur, null); }

private void btnTambahBarangReturActionPerformed(java.awt.event
.ActionEvent evt) {
     if(!vNota.getText().equals("Nomor Nota")){
          dPilihBarang.show();
          dPilihBarang.setLocationRelativeTo(null);
     } else {
          JOptionPane.showMessageDialog(null, "Silahkan masukan
          Nomor Nota");
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
     int row = jTable3.getSelectedRow();
     if(row < 0){
          JOptionPane.showMessageDialog(null, "Silahkan Pilih
          Barang yg akan dihapus");
     } else{
          row = row +1;
          hapusBarang(idReturn.get(row).toString());
     }
}

private void btnSimpanPenjualan1ActionPerformed(java.awt.event
.ActionEvent evt) {
     if(jTable3.getRowCount() == 0){
          JOptionPane.showMessageDialog(null, "Tidak ada Barang
          yang Return");
     } else{ simpanTokoDetailReturn(noFakturTPReturn()); }
}

private void vNotaFocusGained(java.awt.event.FocusEvent evt) {
     if(vNota.getText().equals("Nomor Nota")){
          vNota.setText("");
     }
}

private void vNotaFocusLost(java.awt.event.FocusEvent evt) {
     if(vNota.getText().isEmpty()){vNota.setText("Nomor Nota");}
}

private void vOkActionPerformed(java.awt.event.ActionEvent
evt) {
     if(vOk.getText().equals("OK")){
          String nomor_nota = vNota.getText();
          if(nomor_nota.equals("Nomor Nota")){
              JOptionPane.showMessageDialog(null, "Silahkan isi
              Nomor Nota");
          } else {
              noNota = selectNoNota(nomor_nota);
              if(!noNota.equalsIgnoreCase("")){
                     vNota.setEditable(false);
                     vOk.setText("Edit");
              }
          }
     } else {
          vNota.setEditable(true);
          vOk.setText("OK");
     }
}
```

```
    private void vSearchReturKeyReleased(java.awt.event.KeyEvent
    evt) {    deleteTabel(tabelRetur);
              isiTabelReturn(vSearchRetur.getText());
    }

    private void vNotaKeyReleased(java.awt.event.KeyEvent evt) {
         if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
              String nomor_nota = vNota.getText();
              if(nomor_nota.equals("Nomor Nota")){
                JOptionPane.showMessageDialog(null, "Silahkan isi
                  Nomor Nota");
              } else{
                  noNota = selectNoNota(nomor_nota);
                  if(!noNota.equalsIgnoreCase("")){
                        vNota.setEditable(false);
                        vOk.setText("Edit"); }
              }
         }
    }

    private void jTable3KeyReleased(java.awt.event.KeyEvent evt) {
         if (evt.getKeyCode() == KeyEvent.VK_DELETE) {
              int row = jTable3.getSelectedRow();
              if(row < 0){
                  JOptionPane.showMessageDialog(null, "Silahkan Pilih
                  Barang yg akan dihapus");
              } else {
                  row = row +1;
                  hapusBarang(idReturn.get(row).toString());
              }
         }
    }

    private void btnSimpanActionPerformed(java.awt.event
    .ActionEvent evt) {
         if(code_barang != null  &&
         !vJumlah.getText().equalsIgnoreCase("") &&
         !vTotalHarga.getText().equalsIgnoreCase("")){
              int jumlahStok = Integer.parseInt(jmlBarang.get
              (code_barang).toString());
              int jml = Integer.parseInt(vJumlah.getText());
              if(jml <= jumlahStok && jml > 0){
                  inputTokoKeranjangReturn(code_barang);
                  hashCode.put(vNamaBarang.getText(), code_barang);
                  konversi.put(vSatuan.getSelectedItem(),
                  code_konversi);
                  vNamaBarang.setText(""); vSatuan.removeAllItems();
                  vJumlah.setText("0");
                  vHarga.setText(""); vTotalHarga.setText("");
                  code_barang = null; deleteTabel(tabelRetur);
                  isiTabelReturn("*");
              } else {
                  JOptionPane.showMessageDialog(null, "Jumlah harus
                  Kurang dari atau sama dengan Jumlah maksimal
                  return");
              }
         } else {
              JOptionPane.showMessageDialog(null, "Silahkan Pilih
              Barang, Jumlah Barang & Total Harga");
         }
   }
```

```
     private void vSearchKeyReleased(java.awt.event.KeyEvent evt){
        deleteTabel(tabelBarang);
        tampilTabelDataBarang(vSearch.getText().toString());
    }
    private void vJumlahKeyReleased(java.awt.event.KeyEvent evt){
         char vchar = evt.getKeyChar();
         if(!(Character.isDigit(vchar)) || (vchar ==
         KeyEvent.VK_BACK_SPACE) || (vchar == KeyEvent.VK_DELETE)){
            evt.consume();
         }
         if(!vJumlah.getText().equalsIgnoreCase("") &&
         !vHarga.getText().equalsIgnoreCase("")){
              hitungTotalHarga();
         }
    }
    private void jTable2MouseClicked(java.awt.event.MouseEvent
     evt) {
        int row = jTable2.getSelectedRow();
        if(row < 0){
              JOptionPane.showMessageDialog(null, "Silahkan Pilih
              Barang");
        } else {
              String kode = jTable2.getModel().getValueAt(row, 1)
              .toString();
              isiPilihBarang(kode); dPilihDataBarang.dispose();
              hitungTotalHarga();
        }
    }

    private javax.swing.JButton btnBatal;
    private javax.swing.JButton btnSimpan;
    private javax.swing.JButton btnSimpanPenjualan1;
    private javax.swing.JButton btnTambahBarangRetur;
    private javax.swing.JDialog dPilihBarang;
    private javax.swing.JDialog dPilihDataBarang;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jReturPenjualan;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JTable jTable2;
    private javax.swing.JTable jTable3;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField vHarga;
    private javax.swing.JTextField vJumlah;
    private javax.swing.JTextField vNamaBarang;
    private javax.swing.JTextField vNota;
    private javax.swing.JButton vOk;
    private javax.swing.JComboBox vSatuan;
    private javax.swing.JTextField vSearch;
    private javax.swing.JTextField vSearchRetur;
    private javax.swing.JTextField vTotalHarga;
    private javax.swing.JLabel vTotalStok;
    private javax.swing.JLabel vdatetime;
```

4.4.11. Form Toko Laporan



Detail Laba Rugi

23-08-2018

No.	Kode	Nama	Qty	Satuan	Harga Jual	HPP	Total Jual	Total HPP	LabaRugi	%

Gambar 4.22. Form Toko Laporan

```
public class Toko_Laporan extends javax.swing.JDialog {

    private SimpleDateFormat dt = new SimpleDateFormat("yyyy-MM-dd");
    private SimpleDateFormat dt2 = new SimpleDateFormat("dd-MM-yyyy");
    private DefaultTableModel laporanPenjualan, laporanReturn;
    private ResultSet hasil;
    private Connect connection;
    private PreparedStatement PS;
    private String tgl0, tgl1, tgl2;
    ArrayList<ListTokoLaporanToko> list;
    ListTokoLaporanToko laporanToko;

    public Toko_Laporan() {
        initComponents();

    }

    public Toko_Laporan(java.awt.Frame parent, boolean modal, Connect
    connection) {
        super(parent, modal);
        initComponents();
        this.setLocationRelativeTo(null);
        this.connection = connection;
        tgl0 = getCurrentDate();
        getData(true, tgl0, tgl0);
        jLabel3.setText(getCurrentDate2());
    }

    private String getCurrentDate() {
        String date = "";
        java.util.Date d = new java.util.Date();
        date = dt.format(d);
        return date;
    }
```

```
//tanggal dengan format dd-MM-yyyy
    private String getCurrentDate2(){
        String date2 = "";
        java.util.Date d = new java.util.Date();
        date2 = dt2.format(d);
        return date2;
    }

    private void getData(boolean currentDate, String tgl1, String tgl2) {
        list = new ArrayList<>();
        try {
            laporanPenjualan = new DefaultTableModel(new String[]{"No.",
            "Kode", "Nama", "Qty", "Satuan", "Harga Jual", "HPP", "Total
            Jual", "Total HPP", "LabaRugi", "%"}, 0);
            String sql = "SELECT PD.no_faktur_toko_penjualan,
            B.nama_barang, P.tgl_toko_penjualan, PD.jumlah_barang,
            K.nama_konversi, B.harga_jual_2_barang, PD.harga_barang\n"
            + "from toko_penjualan_detail PD, toko_penjualan P, Barang B,
            barang_konversi BK, konversi K "
            + "where P.no_faktur_toko_penjualan =
            PD.no_faktur_toko_penjualan AND "
            + "PD.kode_barang_konversi = BK.kode_barang_konversi AND "
            + "BK.kode_konversi = k.kode_konversi and "
            + "PD.kode_barang = B.kode_barang and "
            + (currentDate == true ? " CAST(P.tgl_toko_penjualan as date) =
            '" + tgl1 + "' "
            : "P.tgl_toko_penjualan BETWEEN '" + tgl1 + "' AND '" + tgl2 +
            "' ") + " ";
            hasil = connection.ambilData(sql); setData(hasil);
            String sql2 = "SELECT PDR.no_faktur_toko_penjualan_return,
            B.nama_barang, PR.tgl_toko_penjualan_return, "
            + "PDR.jumlah_barang, K.nama_konversi, B.harga_jual_2_barang,
            PDR.harga_barang "
            + "FROM toko_penjualan_detail_return PDR, barang B, konversi K,
            toko_penjualan_return PR "
            + "WHERE PDR.kode_barang = B.kode_barang AND
            PDR.kode_barang_konversi = K.kode_konversi "
            + "AND PR.no_faktur_toko_penjualan_return =
            PDR.no_faktur_toko_penjualan_return AND "
            + (currentDate == true ? "CAST(PR.tgl_toko_penjualan_return as
            date) = '"+tgl1+"' " :
            "PR.tgl_toko_penjualan_return BETWEEN '" + tgl1 + "' AND '" +
            tgl2 + "' ")+" ";
            hasil = connection.ambilData(sql2); setData(hasil);
        } catch (Exception e) {
            System.out.println(e.toString() + "_line_62");
        } finally { showData(); }
    }

    private void setData(ResultSet hasil) {
        try {
            while (hasil.next()) {
                laporanToko = new ListTokoLaporanToko();
                laporanToko.setNo_faktur(hasil.getString(1));
                laporanToko.setNamabarang(hasil.getString(2));
                laporanToko.setTanggal(hasil.getString(3));
                laporanToko.setTgl(hasil.getDate(3));
                laporanToko.setJumlah(hasil.getInt(4));
                laporanToko.setNamakonversi(hasil.getString(5));
                laporanToko.setHargajual(hasil.getInt(6));
                laporanToko.setTotaljual(hasil.getInt(7));
                list.add(laporanToko); laporanToko = null;
            }
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
    }
    private void showData() {
        int no = 0;
        Collections.sort(list, new arraysort());
        for (int i = 0; i < list.size(); i++) {
            laporanToko = list.get(i);
            laporanPenjualan.addRow(new Object[]{
                ++no,
                laporanToko.getNo_faktur(),
                laporanToko.getNamabarang(),
                laporanToko.getJumlah(),
                laporanToko.getNamakonversi(),
                laporanToko.getHargajual(),
                0, laporanToko.getTotaljual(),
                0, 0, 0
            });
        }
        jTable1.setModel(laporanPenjualan);
        jTable1.getColumnModel().getColumn(0).setPreferredWidth(30);
        jTable1.getColumnModel().getColumn(2).setPreferredWidth(140);
    }

    private void deleteTabel(){
        int baris = laporanPenjualan.getRowCount();
        for (int i = 0; i < baris; i++) {
            laporanPenjualan.removeRow(0);
        }
    }

    private void jDateChooser1PropertyChange(java.beans.PropertyChangeEvent
    evt) {
        if (jCheckBox1.isSelected()) {
            if (jDateChooser1.getDate() != null && jDateChooser2
            .getDate() != null) {
                String tanggal1 = dt.format(jDateChooser1.getDate());
                String tanggal2 = dt.format(jDateChooser2.getDate());
                getData(false, tanggal1, tanggal2);
                jLabel3.setText(dt2.format(jDateChooser1.getDate()) + " s/d
                " + dt2.format(jDateChooser2.getDate()));
            }
        }
    }

    private void jDateChooser2PropertyChange(java.beans.PropertyChangeEvent
    evt) {
        if (jCheckBox1.isSelected()) {
            if (jDateChooser1.getDate() != null && jDateChooser2
            .getDate() != null) {
                String tanggal1 = dt.format(jDateChooser1.getDate());
                String tanggal2 = dt.format(jDateChooser2.getDate());
                getData(false, tanggal1, tanggal2);
                jLabel3.setText(dt2.format(jDateChooser1.getDate()) + " s/d
                " + dt2.format(jDateChooser2.getDate()));
            }
        }
    }
    private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent
    evt) {
        if(!jCheckBox1.isSelected()){
            tgl0 = getCurrentDate(); getData(true, tgl0, tgl0);
            jLabel3.setText(getCurrentDate2());
        } else{
            if (jDateChooser1.getDate() != null && jDateChooser2
            .getDate() != null) {
                tgl1 = dt.format(jDateChooser1.getDate());
                tgl2 = dt.format(jDateChooser2.getDate());
                getData(false, tgl1, tgl2);
                jLabel3.setText(dt2.format(jDateChooser1.getDate()) + " s/d
                " + dt2.format(jDateChooser2.getDate()));
            }
        }
    }


    private javax.swing.JCheckBox jCheckBox1;
    private com.toedter.calendar.JDateChooser jDateChooser1;
    private com.toedter.calendar.JDateChooser jDateChooser2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
```

BAB V

PENUTUP

5. 1. Kesimpulan

Dengan berakhirnya Praktik Kerja Lapangan yang telah kami lakukan selama dua bulan. Dapat disimpulkan bahwa Praktik Kerja Lapangan menjadi sarana pengimplementasian dari ilmu yang telah kami dapatkan selama proses perkuliahan, walaupun tidak semua ilmu yang didapatkan dapat kami implementasikan pada tempat penelitian. Praktik Kerja Lapangan juga sangat membantu kami untuk benar-benar memahami dunia kerja sebenarnya yang akan kami hadapi pasca kelulusan.

5. 2. Saran

Praktik Kerja Lapangan merupakan sarana pengembangan diri mahasiswa serta pengenalan dunia kerja kepada mahasiswa. Kami berharap Praktik Kerja Lapangan tetap selalu diadakan karena mengingat banyak manfaat yang diperoleh mahasiswa, khususnya mahasiswa Teknik Informatika STIKI Malang. Sehingga mahasiswa dapat mengenal dan merasakan bagaimana terjun langsung ke dalam dunia kerja dan mengimplementasikan apa yang telah diperoleh dari proses perkuliahan.