

Menu Dan Panduan Penggunaan

RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN BARANG MENGGUNAKAN METODE QUICK MODELLING (METODE SECARA CEPAT) PADA ADVANCE DI BEKASI

Perkembangan zaman saat ini, menjadikan teknologi sangat berkembang dengan pesat, salah satunya adalah teknologi komputer. Kehadiran komputer sendiri mempunyai pengaruh positif yang sangat signifikan terutama di dunia kerja. Pengolahan data yang baik tentu akan menghasilkan informasi yang cepat, tepat dan akurat serta dapat mengatasi masalah-masalah yang terjadi, masalah yang sering timbul ketika pengumpulan data masih dilakukan dengan cara pengolahan data yang masih dilakukan dengan cara manual. Advance merupakan salah satu perusahaan penjualan yang masih menggunakan sistem manual dalam pencatatan stok barang mereka. Pencatatan dilakukan dengan manual dengan cara mendata semua barang yang ada. Metode pengembangan sistem pada aplikasi ini menggunakan metode model secara cepat (*Quick Modelling*) yang terdiri dari beberapa tahapan. Adapun tahapan tersebut yaitu analisis, desain, pengkodean, pengujian dan pemeliharaan sistem. Perancangan yang dibuat oleh penulis ini dimaksudkan untuk membantu admin/sales dalam pencatatan persediaan barang. Dalam sistem ini admin dapat mengetahui atau melihat transaksi, data barang dan stok barang yang dibutuhkan melalui sistem yang sudah dibuat ini.

I. Latar Belakang Masalah

Perkembangan zaman saat ini, menjadikan teknologi sangat berkembang dengan pesat, salah satunya adalah teknologi komputer, hampir semua orang memiliki komputer, PC ataupun Laptop kini sudah menjadi elemen penting yang selalu menunjang segala aktifitas manusia. Sistem informasi diyakini banyak pihak memberi kontribusi terhadap kebutuhan untuk membangun keunggulan kompetitif melalui biaya yang rendah, kualitas yang lebih baik dan peningkatan pelayanan kepada konsumen. Kehadiran komputer sendiri mempunyai pengaruh positif yang sangat signifikan terutama di dunia kerja. Komputer mempermudah manusia untuk membuat dan mengumpulkan data, mulai dari data personal sampai kepada data perusahaan. Pengolahan data yang baik tentu akan menghasilkan informasi yang cepat, tepat dan akurat serta dapat mengatasi masalah-masalah yang terjadi, masalah yang sering timbul ketika pengumpulan data masih dilakukan dengan cara pengolahan data yang masih dilakukan dengan cara manual. Advance merupakan perusahaan yang bergerak dibidang penjualan alat kesehatan. Barang yang dijual di perusahaan ini jumlahnya cukup banyak, akan tetapi masih melakukan pencatatan stok item barangnya harus bolak-balik untuk menghitung secara manual. Dalam penyimpanan data sehingga hal tersebut dapat mengurangi keefisienan kinerja perusahaan tersebut karena banyak data yang harus disimpan, menyebabkan penumpukan kertas-kertas (berkas) data yang ditulis, banyaknya data yang hilang atau

rusak membuat lamanya proses pencatatan dan penyusunan Pengelolaan data persediaan barang di counter Advance Bekasi yang diharapkan dapat memberikan kontribusi bagi toko dalam upaya peningkatan efisiensi pengelolaan persediaan barang. Untuk memecahkan mengatasi masalah tersebut maka diusulkan untuk membangun sebuah sistem informasi persediaan barang, guna mencapai efektifitas maupun efisiensi. Dan dimaksudkan juga untuk mempermudah kinerja karyawan dan mampu memberikan informasi dan laporan yang cepat, tepat dan akurat.

1.1 Tujuan dan Manfaat

Tujuan dari penulisan ini adalah:

1. Dapat memberikan kemudahan dalam kegiatan pencatatan barang.
2. Untuk mengetahui sistem berjalan pada Advance Bekasi, terkait pengolahan data mulai dari pengadaan barang, pengolahan stok barang sampai kepada pengeluaran barang..
3. Dimaksudkan juga sebagai referensi dasar untuk membantu memberikan solusi dari permasalahan yang ada.

Sedangkan manfaat dari penulisan ini adalah sebagai berikut:

1. Manfaat bagi penulis:
Sebagai salah satu syarat untuk bidang B dalam Laporan Beban Kerja Dosen
2. Manfaat untuk objek penelitian:
 - a. Untuk menjadi bahan evaluasi terhadap sistem yang sedang berjalan.
 - b. Menghasilkan suatu informasi yang cepat, tepat dan akurat dalam membantu dan mempermudah proses pengolahan data persediaan barang barang
3. Manfaat untuk pembaca:
Memberikan pemahaman mengenai konsep Rancang bangun sistem informasi persediaan barang dengan metode *Quick Modelling* (Model Secara Cepat) dan juga dapat digunakan sebagai bahan referensi/acuan penelitian bagi penulis selanjutnya.

1.2 Metode Penelitian

1.2.1 Metode Pengembangan Perangkat Lunak

Metode yang penulis gunakan adalah metode *Quick Modelling* dalam metode ini memiliki 4 tahapan. Menurut Sukanto dan Shalahuddin dalam (Firmansyah Y , Udi, 2018), “Quick Modelling atau sering disebut dengan *Metode Secara Cepat* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan *bestpractice* atau cara yang sudah teruji baik. ”

1.3.2 Pemodelan Secara Cepat (*Quick modeling*)

Tahapan pemodelan untuk sistem yang dibangun dibagi menjadi 4 bagian antara lain, desain aliran data sistem menggunakan *Use Case Diagram*, representasi relasi menggunakan *Entity Relationship Diagram (ERD)*, desain *database* sistem yang dibangun, dan desain Antarmuka pengguna sistem yang dibangun dengan 5 tahapan yang terdiri dari :

1. Analisis

Penulis mencoba melakukan analisis terhadap kebutuhan sistem yang sudah berjalan untuk mengidentifikasi kendala yang dihadapi dalam perancangan sistem informasi persediaan barang sehingga bisa diputuskan langkah yang dapat diambil dalam melakukan pendataan transaksi penjualan maupun dan barang yang sering memakan waktu yang banyak dan kurang terkontrol.

2. Desain

Merancang suatu desain sebuah sistem yang meliputi berbagai macam data yang akan digunakan yang didapat dari tempat penelitian.

3. Pengkodean

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*Support*)

Pada tahap ini, penulis akan mengidentifikasi adanya suatu kemungkinan permasalahan yang akan timbul dari program yang dijalankan.

Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. \

1.2.2 Metode Pengumpulan Data

Dalam pengumpulan data, penulis menggunakan beberapa metode dalam penelitian, diantaranya adalah :

1. Metode Pengamatan (*Observation*)

Penulis melakukan pengamatan secara langsung di counter Advance MM Bekasi guna mendapatkan informasi berkaitan dengan persediaan barang di counter Advance MM Bekasi.

2. Metode Wawancara (*Interview*)

Dalam mendapatkan informasi yang akurat, lengkap, dan jelas, penulis melakukan wawancara secara langsung dengan Leader Counter Advance MM Bekasi.

3. Metode Studi Pustaka (*Library Research*)

Selain melakukan kegiatan diatas penulis juga melakukan studi pustaka, pengumpulan data dilakukan melalui studi literatur atau penelitian serupa terdahulu diinternet dan lainnya yang berhubungan dengan materi yang penulis bahas dalam penulisan ini.

1.3 Ruang Lingkup

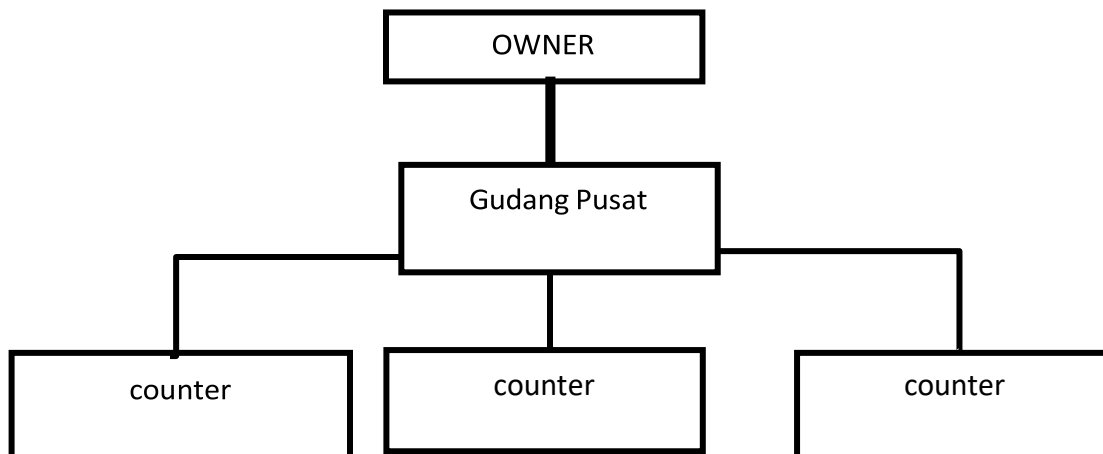
Dalam penulisan ini, batasan masalah pada counter Advance Bekasi dilakukan agar penulisan tidak menyimpang dan terarah dari tujuan semula, maka pembatasan akan dibatasi mulai dari penerimaan barang, pengecekan barang, pengeluaran barang dan proses pemantauan barang agar stok data akurat dengan stok aktual dan menghasilkan *output* berupa dokumen laporan.

II. Analisa Sistem

2.1 Tinjauan Perusahaan

Advance adalah merek untuk peralatan sehat yang telah berdiri sejak 1994. Menjadi pemimpin pasar dalam produk gaya hidup sehat, *ADVANCE* menawarkan berbagai produk gaya hidup sehat dengan teknologi, desain, dan fungsi inovatif. Advance memiliki jaringan distribusi yang luas di seluruh Indonesia, dan hingga hari ini kami memiliki lebih dari 70 ruang pamer yang dirancang kontemporer di kota-kota besar di seluruh negeri dan didukung oleh tenaga pemasaran profesional kami untuk melayani kebutuhan pelanggan kami. Menjadi merek terkemuka dan tepercaya dalam produk air & gaya hidup sehat di Indonesia. Inovasi berkelanjutan untuk produk gaya hidup bernilai tambah tinggi Pengembangan berkelanjutan dari promosi pemasaran kreatif Pengiriman keunggulan layanan yang berkelanjutan dan kepuasan pelanggan SSB adalah perusahaan yang dinamis dan inovatif yang berfokus pada kesehatan dan berbasis pada tiga core bisnis - produk Air, Kesehatan, dan Kebugaran. Terlepas dari permulaan kami yang sederhana, kami telah menetapkan visi - untuk menjadi merek terkemuka dan tepercaya di Indonesia untuk solusi menuju kehidupan yang lebih baik. Untuk menjadi yang terdepan dalam bisnis ini, kami memanfaatkan kekuatan utama kami dalam mengidentifikasi produk yang dapat dipasarkan dengan nilai tambah tinggi dan menggabungkannya dengan inisiatif pemasaran yang kuat yang akan membawa nilai lebih besar pada minat dan kepuasan pelanggan kami. Kami akan berusaha untuk meningkatkan jangkauan layanan dan produk kami sebagai bagian dari misi kami untuk memberikan kepuasan pelanggan total. DAVID HARIPIN – CHAIRMAN

Dibawah ini adalah struktur organisasi yang terdapat pada advance store bekasi menggunakan Lini dan Staff sebagai berikut:



Gambar III.1. Struktur Organisasi

Sumber: Advance counter bekasi

Deskripsi tugas dan wewenang dari bagian-bagian organisasi yang terkait adalah sebagai berikut:

1. Pemilik Toko/*Owner*
 - a. Membuat program kerja, memberikan pengarahan serta perintah kerja pada pegawai.
 - b. Menilai hasil kerja pegawai, menyusun rencana kerja perusahaan, mengangkat dan memberhentikan pegawai.
 - c. Bertanggung jawab sepenuhnya atas aktivitas usaha toko.

d. Bertugas melakukan pembelian barang..

2. Pegawai (Bagian Gudang)

- a. Mengontrol stok barang.
- b. Menerima barang dari supplier untuk di stok ke gudang.
- c. Melakukan perawatan dan pengawasan kualitas barang.
- d. Melakukan retur bila terdapat barang rusak.

3. Pegawai Counter (Sales)

- a. Membuka dan menutup toko.
- b. Menjalankan perintah atasan dan melayani pelanggan.
- c. Melakukan penjualan/transaksi.

2.2 Prosedur Sistem Berjalan

Pada sub-bab ini penulis akan menjelaskan prosedur sistem persediaan barang yang terjadi pada Advance Store Bekasi. Prosedur tersebut dimulai dengan prosedur penjualan barang dan pembelian barang.

Berikut ini adalah prosedur-prosedur yang terdapat dalam persediaan barang:

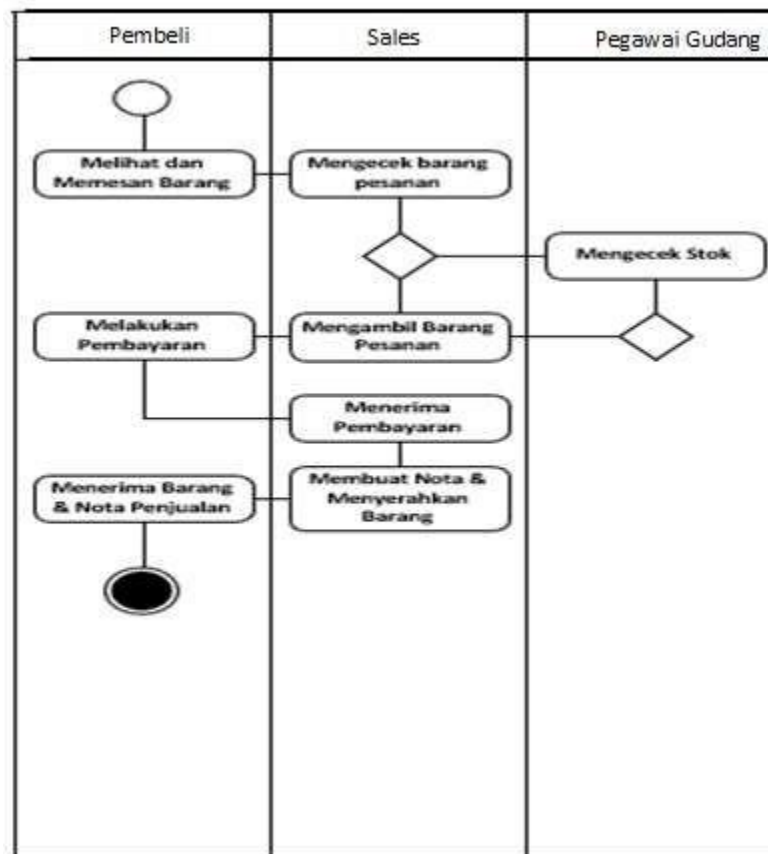
1. Prosedur Pembelian Barang

Pegawai bagian gudang mengecek stok barang, jika stok yang terdapat barang yang dibutuhkan masih tersedia maka bagian gudang akan memberikannya pada counter namun jika kebutuhan stok kurang, pegawai bagian gudang akan membuat list barang kosong lalu diserahkan kepada pemilik. Pemilik menerima list permintaan tersebut dan lalu membuat pemesanan barang kepada supplier.. Setelah sampai faktur penjualan dan barang tersebut diterima oleh pegawai bagian gudang.

2. Proses Penjualan Barang

Pelanggan datang ke counter kemudian memesan barang kepada pegawai bagian counter, jika barang tidak tersedia (*indent*) maka pegawai counter akan menghubungi bagian gudang untuk mengambil barang, jika barang tersedia maka pegawai counter pun mengambil barang pesanan dari pembeli tersebut, lalu pembeli memberikan pembayaran sesuai barang yang telah dipesan. Setelah pembayaran selesai dilakukan maka pegawai counter akan membuat bukti transaksi/sales order kepada pembeli sebagai bukti pembayaran yang sah.

2.3 Activity Diagram



Gambar III.2. Activity Diagram

2.4 Spesifikasi Bentuk Dokumen Masukan

Dokumen *input* merupakan semua bentuk dokumen yang masuk ke sebuah sistem perusahaan yang berasal dari lingkungan perusahaan. Dokumen-dokumen *input* sistem berjalan pada Advance Store Bekasi sebagai berikut:

1. Nama Masukan: Laporan Rekap Stok Periodik

Fungsi	: Bukti penerimaan barang oleh pegawai counter(sales)
Sumber	: Bagian Gudang
Tujuan	: Pegawai counter(sales)
Media	: Kertas
Frekuensi	: Setiap melakukan pengiriman barang ke counter
Bentuk	: Lampiran A-1

3.5 Spesifikasi Bentuk Dokumen Keluaran

1. Nama Keluaran: Nota Penjualan

Fungsi	: Bukti transaksi penjualan barang ke pelanggan
Sumber	: Pegawai counter(sales)
Tujuan	: Pelanggan
Media	: Kertas
Frekuensi	: Setiap ada penjualan barang
Bentuk	: Lampiran B-1

3.6 Permasalahan Pokok

Masalah-masalah yang ada pada sistem berjalan pada Advance Store Bekasi sebagai berikut:

1. Sistem persediaan barang yang masih manual yang mengakibatkan lambatnya pencatatan data pada proses transaksi penjualan serta pembelian dalam toko.
2. Perlunya melakukan laporan/catatan penjualan barang yang terjadi setiap hari ketika counter sudah tutup agar tidak terjadi selisih persediaan barang.
3. Semua arsip dan laporan masih dicatat dalam buku atau kertas sehingga tidak efisien dan ada kemungkinan dokumen hilang ataupun rusak.

3.7 Pemecahan Masalah

Pemecahan-pemecahan masalah diatas secara garis besar sebagai berikut:

1. Menggunakan sebuah sistem yang telah terintegrasi dengan komputer yaitu sistem informasi persediaan barang untuk memudahkan pencatatan barang pada proses dalam setiap transaksi di toko tersebut.
2. Semua data akan disimpan dalam komputer, sehingga akan memudahkan dalam mencari data serta mencegah hilangnya data tersebut. Sehingga pekerjaan yang dilakukan menjadi lebih efektif dan efisien.

III. Rancang Bangun Sistem

3.1 Tahapan Perancangan Sistem

3.1.1 Analisis Kebutuhan

Berdasarkan proses persediaan barang pada counter advance bekasi, tahapanyang dilakukan adalah analisa kebutuhan, berikut ini merupakan spesifikasi kebutuhan (dari sistem persediaan yang ada:

A. Kebutuhan Pengguna

Dalam aplikasi persediaan barang ada empat pengguna yang dapat berinteraksi satu sama dalam sebuah lingkungan sistem yaitu: userr, pegawai (gudang), pemilik, dan pegawai counter (sales). Keempat pengguna tersebut memiliki karakteristik interaksi dengan sistem yang berbeda-beda dan memiliki kebutuhan informasi yang juga berbeda.

Akses user:

- A1. Dapat melakukan *login*.
- A2. Dapat untuk melihat, menambah, mengubah dan menghapus data master

pegawai.

A3. Dapat untuk melihat, menambah, mengubah dan menghapus data master *user*.A4.

Dapat untuk melihat, menambah dan menghapus data master pelanggan.

A5. Memperbaharui sistem persediaan barang. A6.

Mendapat akses untuk melihat semua laporan.

Akses Pegawai Counter (sales):

B1. Dapat mendaftar menjadi *user*.

B2. Dapat melakukan *login*.

B3. Mendapat akses untuk melakukan transaksi penjualan.

B4. Mendapat akses untuk melakukan retur penjualan.

Akses Pegawai (Gudang):

C1. Dapat melakukan *login*.

C2. Dapat mendaftar menjadi *user*.

C3. Mendapat akses untuk melakukan retur pembelian. C4.

Mendapat akses untuk mengelola data master barang.

C5. Mendapat akses untuk melihat stok barang.

Akses Pemilik

D1. Dapat melakukan *login*.

D2. Memiliki hak akses untuk melihat semua laporan.

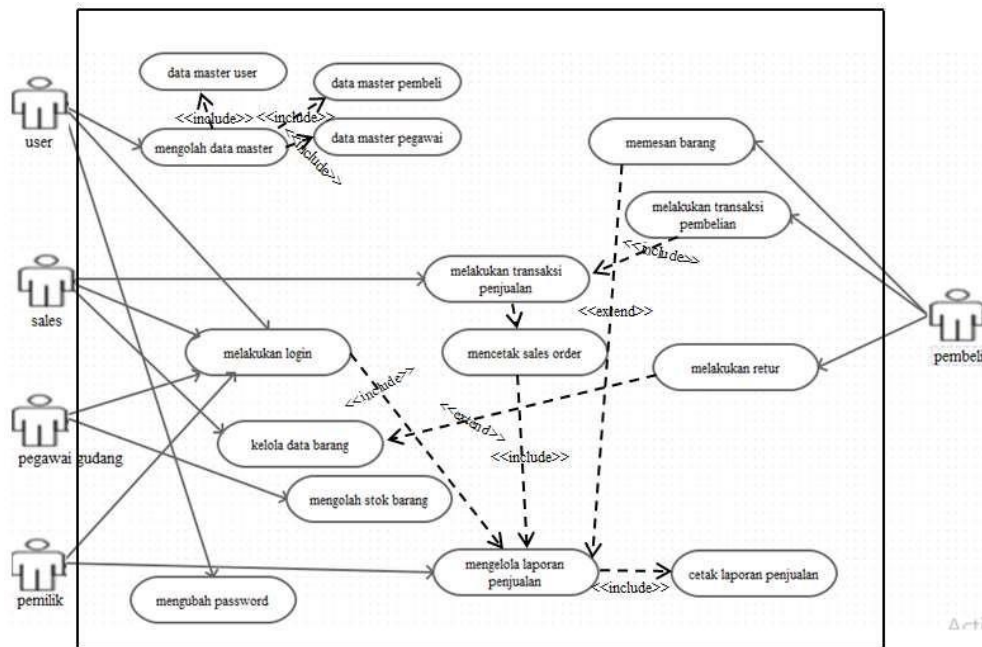
B. Kebutuhan Sistem

1. User harus melakukan *login* terlebih dahulu untuk dapat masuk kedalam halaman utama dengan memasukkan *username* serta *password*. Dan semua *user* harus menjaga *username* dan *password* mereka agar tetap terjaga privasinya.
2. *User* melakukan *logout* setelah selesai menggunakan aplikasi.
3. Pemakai harus terdaftar menjadi *user* supaya bisa menjalankan fasilitas-fasilitas dalam sistem, seperti transaksi penjualan dan pembelian barang.

IV. Rancangan *Diagram Use Case*

Use Case Diagram menunjukkan hubungan interaksi antara *use case*, *actor*, dan pekerja. Diagram ini menggambarkan model lengkap tentang apa saja yang perusahaan lakukan, siapa saja yang ada di dalam yang berperan didalamnya dan siapa orang yang berperan diluarnya. Hal ini menggambarkan ruang lingkup organisasi, sehingga dapat dilihat apa atau siapa saja yang adadiluar

organisasi dan sampai dimana adanya pembatasan.



Gambar IV.1 Use Case Diagram

<i>Use Case Name</i>	<i>Login</i>
<i>Requirements</i>	Untuk dapat memasuki halaman menu utama
<i>Goal</i>	Untuk dapat masuk ke halaman menu utama
<i>Pre-Conditions</i>	Harus membuat <i>username</i> dan <i>password</i> terlebih dahulu yang tersambung dengan sistem
<i>Post-Conditions</i>	Sistem melakukan validasi data dan user bisa masuk ke halaman menu utama
<i>Failed end Conditions</i>	User tidak dapat login ke dalam sistem
<i>Actors</i>	User
<i>Main Flow/Basic Path</i>	1. Memasukan <i>username</i> , <i>password</i> & level akses 2. Validasi data

	3. Masuk ke halaman utama 4. Menampilkan menu utama
Alternate Flow/Invariant A	1. Memasukan username serta password 2. Validasi data 3. Username serta password salah 4. Kembali ke menu login dan muncul pesan username atau password salah 5. Memasukan kembali username serta password 6. Validasi data yang dimasukkan 7. Masuk ke halaman utama
Invariant B	-

Tabel IV.1. Deskripsi Use Case Login

Use Case Name	Transaksi Penjualan
Requirements	Sales melakukan transaksi penjualan
Goal	Melakukan transaksi penjualan
Pre-Conditions	Sales mengisi pesanan pembeli ke dalam sistem
Post-Conditions	Mencetak sales order yang diberikan kepada pelanggan
Failed end Conditions	Data penjualan tak berhasil disimpan
Actors	Pembeli, Pegawai counter (sales), Sistem
Main Flow/Basic Path	1. Memesan barang 2. Halaman utama lalu halaman transaksi penjualan 3. Mencari data barang berdasarkan kode barang 4. Data barang tersedia lalu input jumlah pembelian 5. Menerima struk pembayaran 6. Simpan transaksi penjualan 7. Menampilkan Sales order 8. Mencetak Sales order

	9. Menerima Sales order dan barang
Alternate Flow/Invariant A	1. Memesan barang 2. Membuka transaksi penjualan 3. Mencari data barang 4. Menampilkan data barang 5. Stok barang sedang kosong 6. Melakukan transaksi pembelian
Invariant B	-

Tabel IV.2. Deskripsi Use Case Transaksi Penjualan

Use Case Name	Retur Penjualan
Requirements	Dapat melakukan proses retur penjualan
Goal	Mengeluarkan bukti retur sebagai tanda ganti barang yang rusak
Pre-Conditions	Masuk ke menu retur penjualan
Post-Conditions	Memberikan retur penjualan
Failed end Conditions	Gagal melakukan retur penjualan
Actors	Pembeli, Pegawai (Sales), Sistem
Main Flow/Basic Path	1. Melaporkan kerusakan barang 2. Masuk ke menu retur 3. Menginput kerusakan barang yang sudah dibeli 5. Memberikan Kartu garansi 6. Mengecek tanggal transaksi 7. Menampilkan bukti retur 8. Mencetak bukti retur 9. Menerima bukti retur
Alternate Flow/Invariant A	1. Melaporkan kerusakan barang 2. Masuk ke menu retur

	3. Menginput kerusakan barang yang sudah dibeli 5. Memberikan Kartu garansi 6. Mengecek tanggal transaksi 7. Gagal melakukan transaksi karna sudah lewat masa garansi 8. Menampilkan tulisan “Gagal melakukan retur” 9. Kembali ke menu retur
<i>Invariant B</i>	-

Tabel IV.3. Deskripsi *Use Case* Retur Penjualan

<i>Use Case Name</i>	Data Barang
<i>Requirements</i>	Pegawai dapat mengelola data barang
<i>Goal</i>	Untuk bisa mengelola detail data barang
<i>Pre-Conditions</i>	Pegawai mengisi kode barang
<i>Post-Conditions</i>	Data barang bisa dilakukan pengelolaan
<i>Failed end Conditions</i>	Pegawai tak dapat melihat data barang
<i>Actors</i>	Pegawai (Gudang),Pegawai counter(Sales), Sistem
<i>Main Flow/Basic Path</i>	1. Masuk ke menu data barang 2. Mencari data berdasarkan kode barang 3. Menampilkan data barang 4. Mengolah data barang 6. Menyimpan perubahan data barang 7. Pemberitahuan data berhasil tersimpan
<i>Alternate Flow/Invariant A</i>	1. Masuk ke menu data barang 2. Mencari data barang dengan memasukkan kode barang 3. Stok barang yang tidak ditemukan dan akan muncul pesan “Barang tidak ditemukan” 4. Mengisi kembali dengan kode barang yang valid

	5. Menampilkan data barang
<i>Invariant B</i>	-

Tabel IV.4. Deskripsi *Use Case* Data Barang

<i>Use Case Name</i>	Stok Barang
<i>Requirements</i>	Dapat melihat stok barang
<i>Goal</i>	Mengetahui stok barang yang tersedia
<i>Pre-Conditions</i>	Mengisi dalam pencarian berdasarkan kode barang
<i>Post-Conditions</i>	Bisa melihat detail tentang laporan stok barang
<i>Failed end Conditions</i>	Sistem tidak dapat menampilkan stok barang
<i>Actors</i>	Pegawai counter (sales), Sistem
<i>Main Flow/Basic Path</i>	1. Masuk ke menu stok barang 2. Memasukkan kode barang 3. Data stok barang tersedia 4. Menampilkan stok barang 5. Mencetak halaman stok barang 6. Laporan stok barang
<i>Alternate Flow/Invariant A</i>	1. Masuk ke stok barang 2. Mencari data stok barang berdasarkan kode barang 3. Data stok barang yang tidak ditemukan dan akan muncul pesan “Barang tidak ditemukan” 4. Mengisi kembali dengan kode barang yang valid 5. Data stok barang yang diminta muncul
<i>Invariant B</i>	-

Tabel IV.4. Deskripsi *Use Case* Stok Barang

<i>Use Case Name</i>	Laporan Penjualan
<i>Requirements</i>	Dapat melihat laporan tentang penjualan yang terjadi

<i>Goal</i>	Dapat melihat laporan dari transaksi penjualan
<i>Pre-Conditions</i>	Mengisi tanggal ketika laporan dibuat
<i>Post-Conditions</i>	Dapat melihat laporan dari transaksi penjualan
<i>Failed end Conditions</i>	Gagal melihat laporan dari transaksi penjualan
<i>Actors</i>	Pemilik, Sistem
<i>Main Flow/Basic Path</i>	1. Masuk ke menu laporan 2. Menampilkan form laporan penjualan 3. Memasukan tanggal laporan 4. Menampilkan laporan 5. Mencetak laporan 6. Menerima laporan penjualan
<i>Alternate Flow/Invariant A</i>	-
<i>Invariant B</i>	-

Tabel IV.5. Deskripsi Use Case Laporan Penjualan

<i>Use Case Name</i>	Data User
<i>Requirements</i>	Admin dapat melihat master data user
<i>Goal</i>	Mengelola segala yang berhubungan dengan data user
<i>Pre-Conditions</i>	Memasukan data user
<i>Post-Conditions</i>	Dapat mengakses dan mengelola data user
<i>Failed end Conditions</i>	Tidak dapat melihat data user
<i>Actors</i>	User, Sistem

<i>Main Flow/Basic Path</i>	1. Masuk ke menu data user 2. Menampilkan <i>form</i> data user 3. Mencari data user 4. Menampilkan data user 5. Mengelola data user 6. Menyimpan perubahan data user 7. Pemberitahuan penyimpanan perubahan data user
<i>Aternate Flow/Invariant A</i>	1. Masuk ke menu data user 2. Mengetikan kode/nama user 3. Muncul pesan “Data tidak ditemukan” 4. Kembali mengetik kode user 5. Menampilkan data user yang dicari
<i>Invariant B</i>	

Tabel IV.6. Deskripsi *Use Case* Data User

<i>Use Case Name</i>	Data pembeli
<i>Requirements</i>	Sales mendapat akses mengelola data pembeli
<i>Goal</i>	Mengelola data pembeli
<i>Pre-Conditions</i>	Sales memasukan nama p pembeli
<i>Post-Conditions</i>	Dapat mengelola apapun tentang data pelanggan
<i>Failed end Conditions</i>	Sales tak dapat melihat data pembeli
<i>Actors</i>	Pegawai counter(sales), Sistem
<i>Main Flow/Basic Path</i>	1. Masuk ke menu data pembeli 2. Menampilkan form data pembeli 3. Mengolah data pembeli 4. Menginput data pembeli 5. Menyimpan perubahan data pembeli 6. Pemberitahuan penyimpanan data pembeli baru

<i>Alternate Flow/Invariant A</i>	1. Masuk ke menu data pembeli 2. Mengetikan nama pembeli 3. Muncul pesan “Data tidak ditemukan” 4. Kembali mengetik kode pembeli 5. Menampilkan data pembeli yang dicari
<i>Invariant B</i>	-

Tabel IV.7. Deskripsi *Use Case* Data Pembeli

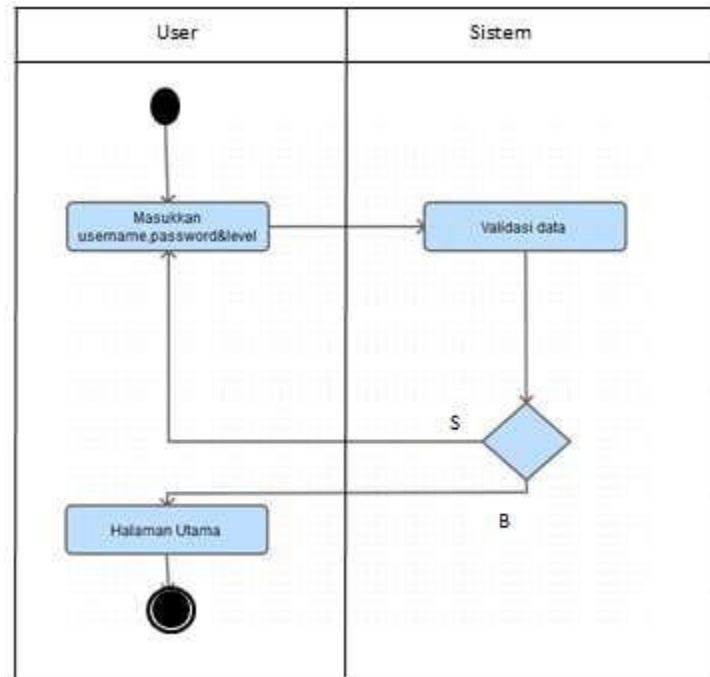
<i>Use Case Name</i>	Data Pegawai
<i>Requirements</i>	Dapat melihat data dan mengelola pegawai
<i>Goal</i>	Mengelola beserta melihat data pegawai
<i>Pre-Conditions</i>	Masuk ke menu data pegawai
<i>Post-Conditions</i>	Dapat mengelola data pegawai
<i>Failed end Conditions</i>	Tidak dapat melihat data pegawai
<i>Actors</i>	User, Sistem
<i>Main Flow/Basic Path</i>	1. Masuk ke menu data pegawai 2. Menampilkan form data pegawai 3. Memasukkan kode pegawai 4. Menampilkan data pegawai 5. <i>Menginput</i> data pegawai baru 6. Menyimpan data pegawai yang telah dikelola 8. Pemberitahuan penyimpanan perubahan data pegawai
<i>Alternate Flow/Invariant A</i>	1. Masuk ke menu data pegawai 2. Mengetikan kode pegawai 3. Muncul pesan “Data tidak ditemukan” 4. Kembali mengetik kode pegawai 5. Menampilkan data pegawai yang dicari
<i>Invariant B</i>	-

Tabel IV.8. Deskripsi *Use Case* Data Pegawai

4.1 Rancangan Diagram Aktivitas

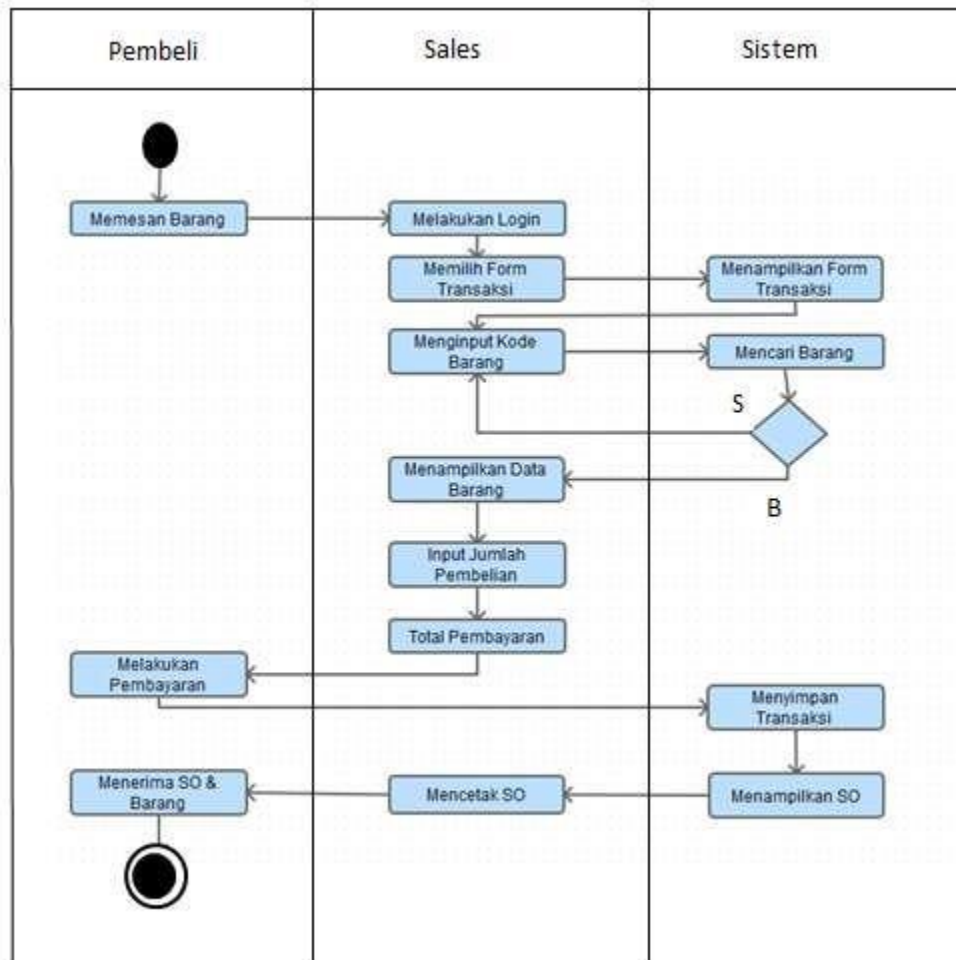
Activity Diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk menjelaskan aktivitas yang dibentuk dalam sebuah proses, sehingga dapat digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

1. *Activity Diagram Login*



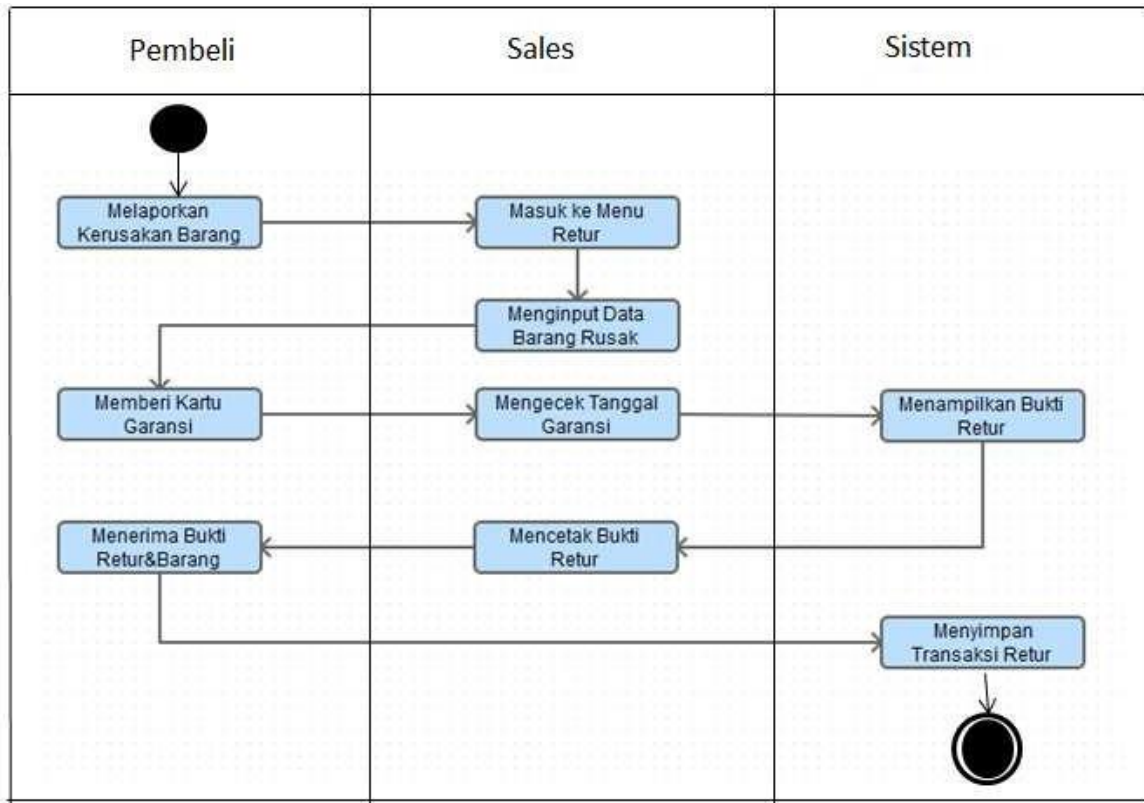
Gambar IV.2. *Activity Diagram Login*

2. *Activity Diagram Transaksi Penjualan*

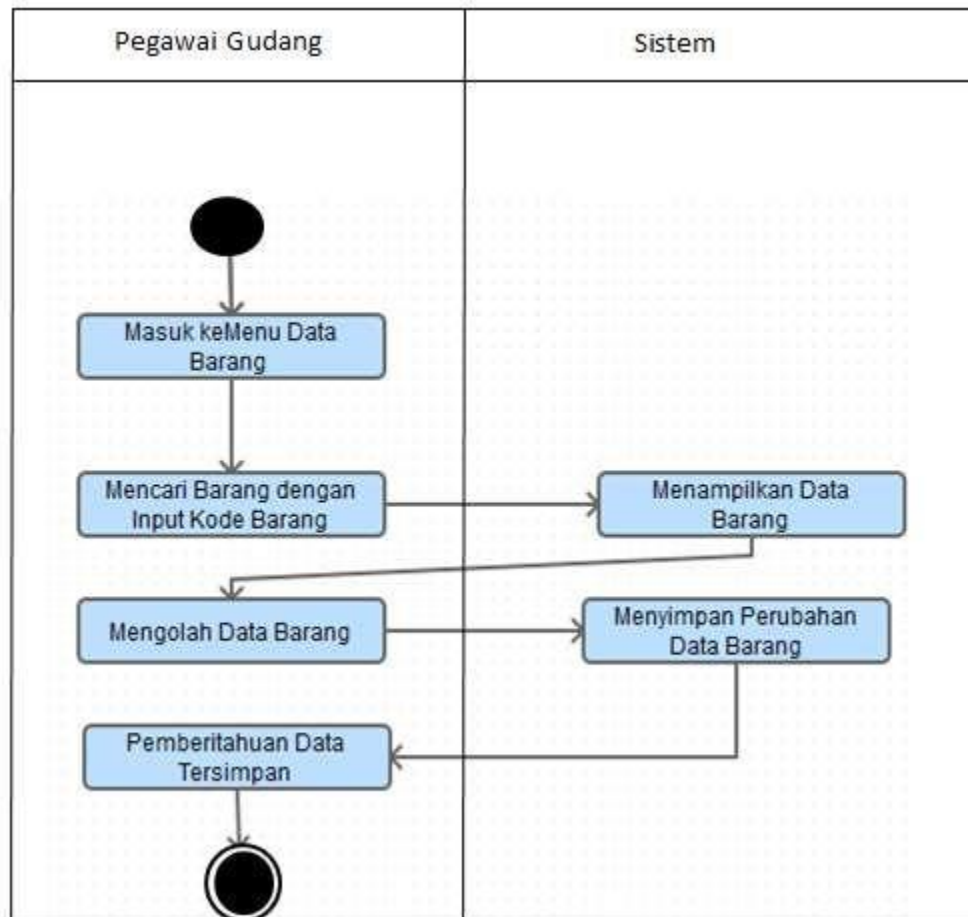


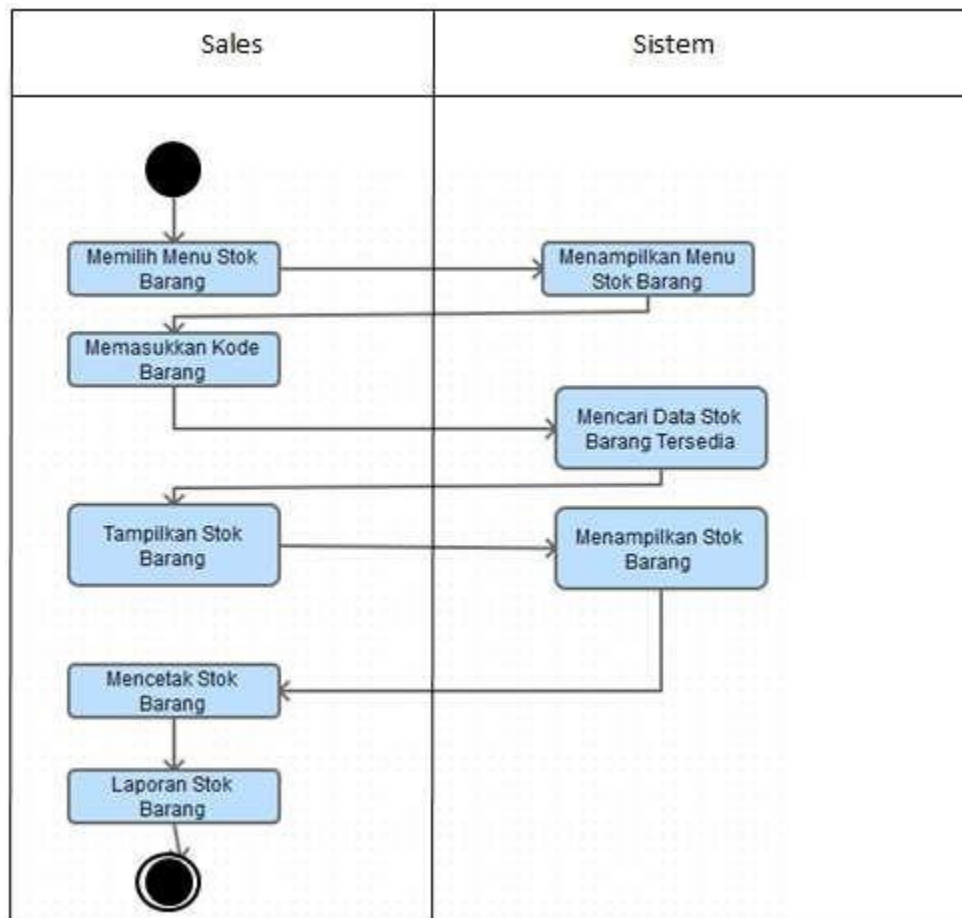
Gambar IV.3. *Activity Diagram Transaksi Penjualan*

3. Activity Diagram Retur Penjualan

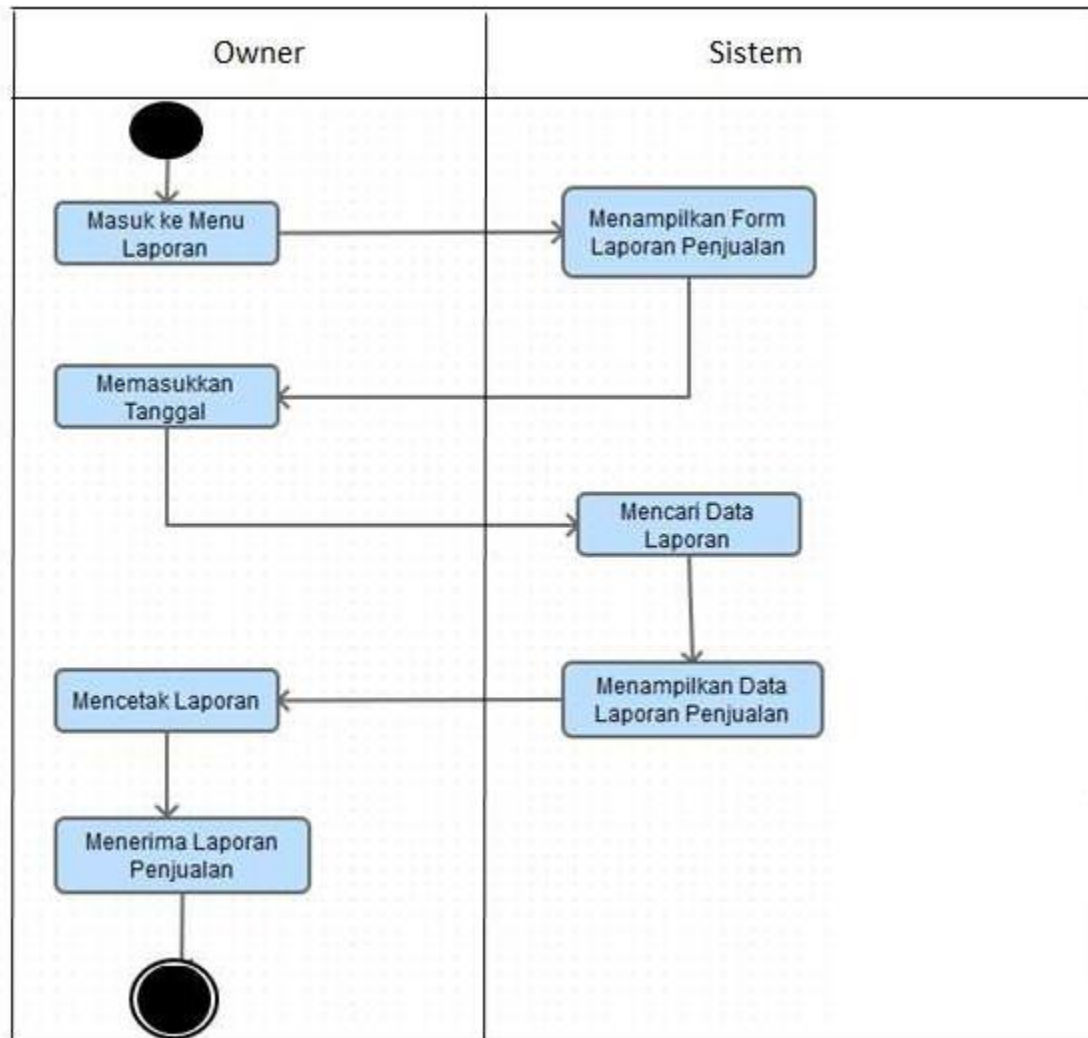


Gambar IV.4. Activity Diagram Retur Penjualan

4. *Activity Diagram Data Barang*Gambar IV.5. *Activity Diagram Data Barang*

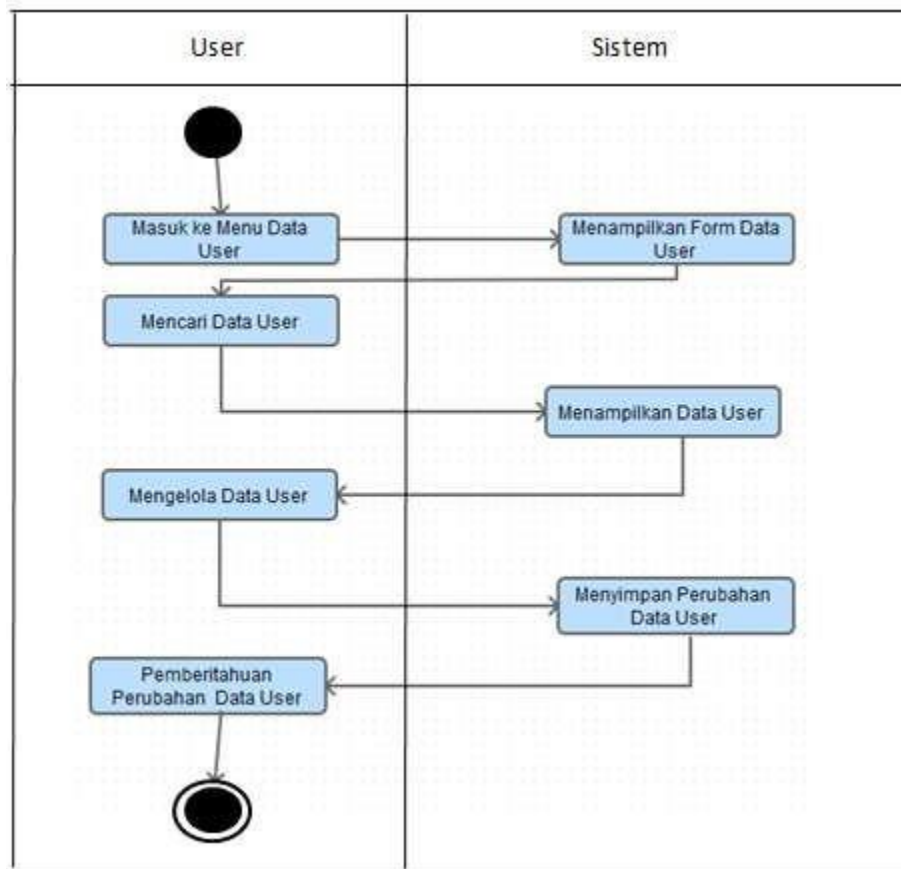
5. *Activity Diagram Stok Barang*Gambar IV.6. *Activity Diagram Stok Barang*

6. Activity Diagram Laporan Penjualan



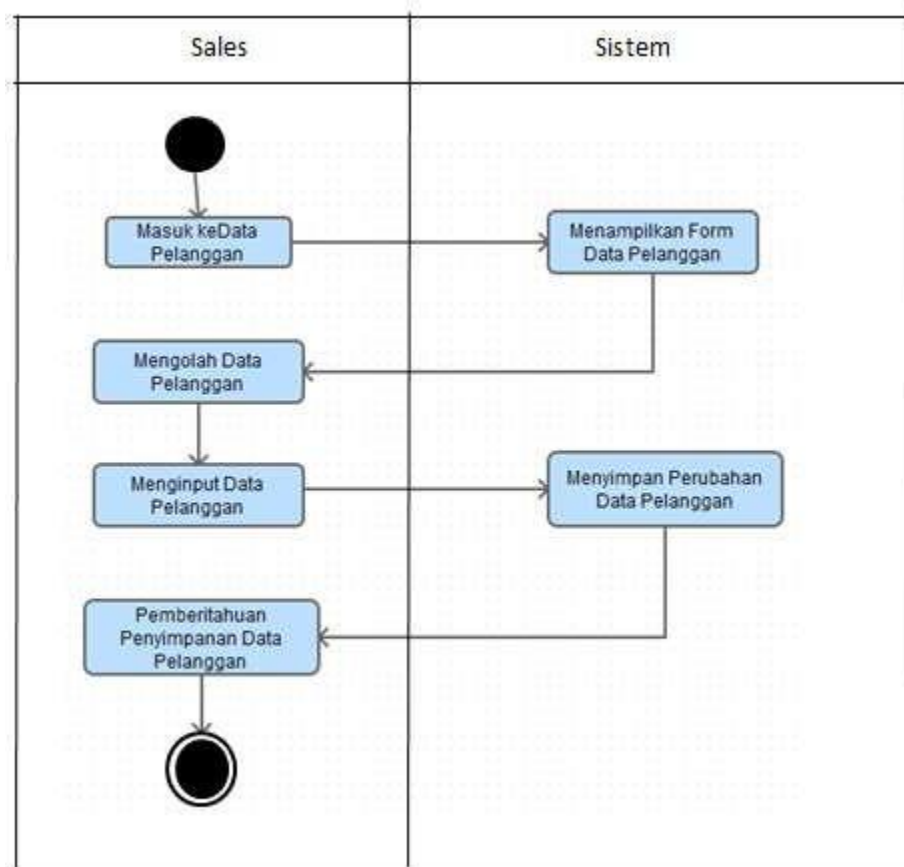
Gambar IV.7. Activity Diagram Laporan Penjualan

7. Activity Diagram Data User



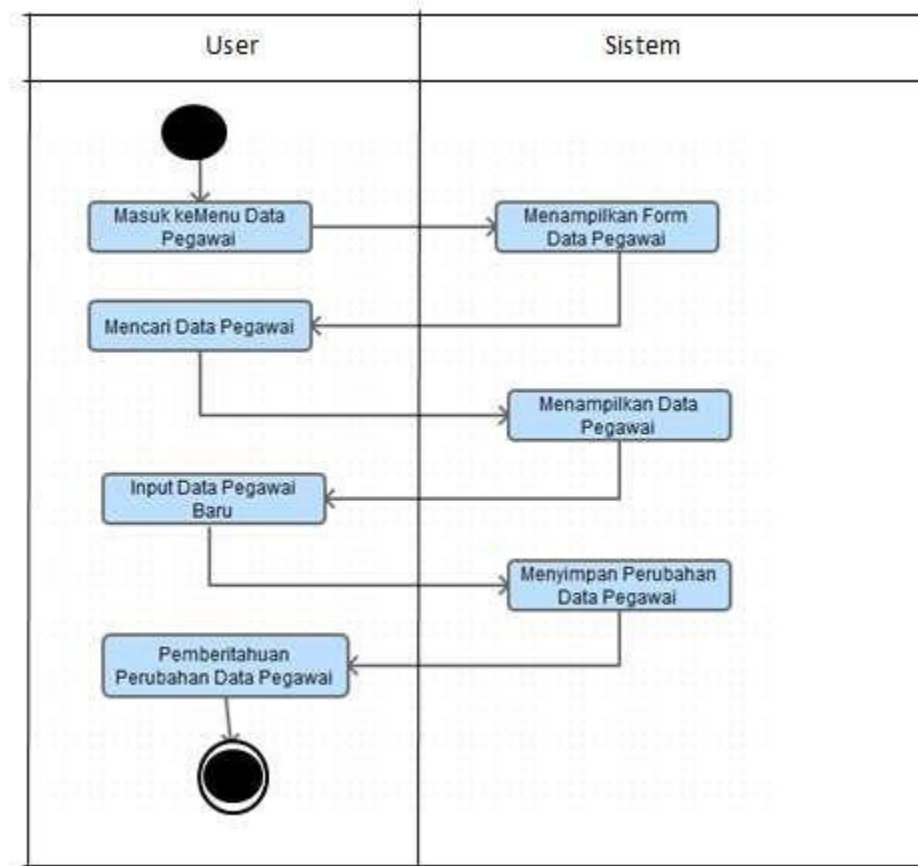
Gambar IV.8. Activity Diagram Data User

8. Activity Diagram Data Pelanggan



Gambar IV.9. Activity Diagram Data Pelanggan

9. Activity Diagram Data Pegawai



Gambar IV.10. Activity Diagram Data Pegawai

4.1.1 Rancangan Dokumen Sistem Usulan

A. Dokumen Masukan

1. Nama Dokumen : Penjualan
- Fungsi : Untuk mengetahui barang yang akan dibeli
- Sumber : Pegawai counter (Sales)
- Tujuan : Pegawai Gudang
- Media : Kertas
- Jumlah : 1 Lembar

- | | | |
|--|-----------|--------------------------------------|
| | Frekuensi | : Setiap terjadi transaksi pembelian |
| | Bentuk | : Lihat Lampiran A1 |
- B. Dokumen Keluaran
1. Nama Dokumen : Sales Order

Fungsi	: Untuk mengetahui barang yang dijual di toko
Sumber	: Pegawai counter (Sales)
Tujuan	: Pelanggan
Media	: Kertas
Jumlah	: 1 Lembar
Frekuensi	: Setiap terjadi transaksi penjualan
Bentuk	: Lihat Lampiran D-1

 2. Nama Dokumen : Retur Penjualan

Fungsi	: Untuk mengetahui daftar barang yang diretur
Sumber	: Pegawai counter (Sales)
Tujuan	: Pegawai Gudang
Media	: Kertas
Jumlah	: 1 Lembar
Frekuensi	: Setiap terjadi retur atas barang yang rusak
Bentuk	: Lihat Lampiran D-2

 4. Nama Dokumen : Laporan Penjualan

Fungsi	: Untuk mengetahui data-data hasil penjualan barang
Sumber	: Pegawai counter (Sales)
Tujuan	: Pemilik
Media	: Kertas
Jumlah	: 1 Lembar
Frekuensi	: 1 Bulan
Bentuk	: Lihat Lampiran D-3

5. Nama Dokumen : Laporan Stok Barang
 Fungsi : Untuk mengetahui data stok barang
 Sumber : Pegawai (Bagian Gudang)
 Tujuan : Pemilik dan Sales
 Media : Kertas
 Jumlah : 1 Lembar
 Frekuensi : 1 Bulan
 Bentuk : Lihat Lampiran D-4

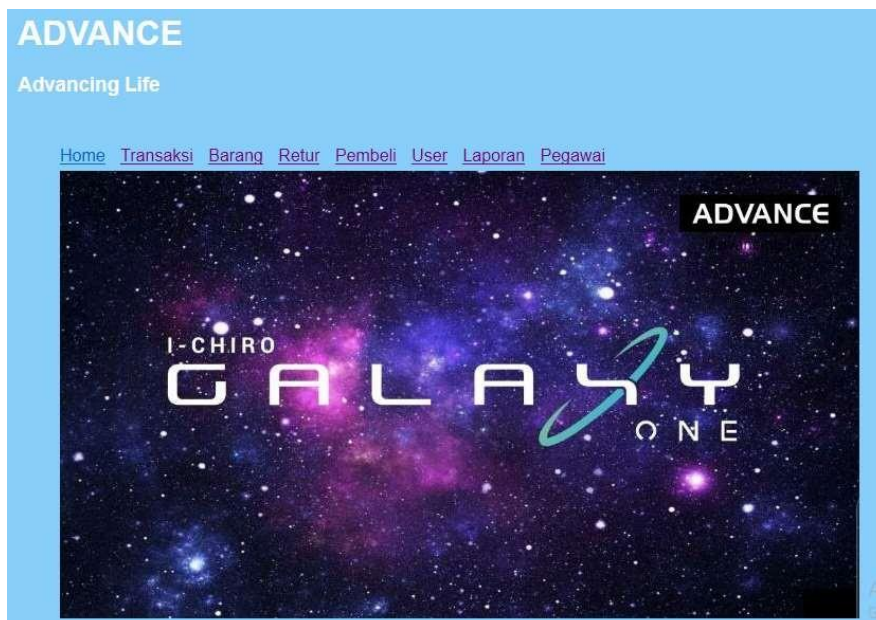
4.1.5 Rancangan Prototype

1. Halaman *Login*



Gambar IV.11. Halaman *Login*

2. Halaman Utama (*Home*)



Gambar IV.12. Halaman Utama (*Home*)

3. Transaksi Penjualan

The screenshot shows the 'Transaksi Penjualan' (Sales Transaction) form within the ADVANCE system. The form is titled 'ADVANCE Advancing Life' and includes a 'Home' link. The form fields are as follows:

Kode Barang :	Kode
Nama Barang :	Nama
Tanggal :	mm/dd/yyyy
Kode User :	Kode
QTY :	Jumlah
Satuan :	Jumlah
Harga Jual :	harga
Diskon :	diskon
Subtotal :	Subtotal
Bayar :	Bayar

At the bottom of the form are two buttons: 'submit' and 'batal'. The footer of the page reads 'Copyright © 2020'.

Gambar IV.13. Transaksi Penjualan

4. Barang

The screenshot shows the 'Barang' (Goods) form within the ADVANCE system. The form is titled 'ADVANCE Advancing Life' and includes a 'Home' link. The form fields are as follows:

Kode barang :	Kode
Nama Barang :	Nama Barang
Stok :	Jumlah stok
Satuan :	Satuan
Harga Jual :	Harga

At the bottom of the form are two buttons: 'submit' and 'batal'. The footer of the page reads 'Copyright © 2020'.

Gambar IV.14. Barang

5. Retur Penjualan

The screenshot shows the 'Retur Barang Penjualan' (Sales Return) form within the ADVANCE system. The form is titled 'ADVANCE Advancing Life' and includes a 'Home' link. The form fields are as follows:

Kode Pembeli :	Kode
Tanggal Retur :	mm/dd/yyyy
Kode User :	Kode

At the bottom of the form are two buttons: 'submit' and 'batal'. The footer of the page reads 'Copyright © 2020'.

Gambar IV.15. Retur Penjualan

6. Pembeli

The screenshot shows a web interface for the 'ADVANCE' system, with the tagline 'Advancing Life'. A 'Home' link is visible. The main heading is 'Registrasi Pembeli' (Buyer Registration), which is highlighted with a red dashed border. Below this, there is a form with the following fields: 'Nama' (Name), 'Alamat' (Address), and 'No Telepon' (Phone Number). Each field has a corresponding input box. At the bottom of the form are two buttons: 'submit' and 'batal' (cancel). The footer of the page reads 'Copyright © 2020'.

Gambar IV.16. Pembeli

7. User

The screenshot shows a web interface for the 'ADVANCE' system, with the tagline 'Advancing Life'. A 'Home' link is visible. The main heading is 'Registrasi User' (User Registration), which is highlighted with a red dashed border. Below this, there is a form with the following fields: 'Nama' (Name), 'Jenis Kelamin' (Gender) with a dropdown menu, 'Username', 'Password', and 'Level' with a dropdown menu. Each field has a corresponding input box. At the bottom of the form are two buttons: 'submit' and 'batal' (cancel). The footer of the page reads 'Copyright © 2020'.

Gambar IV.17. User

8. Laporan

The screenshot shows a web interface for the 'ADVANCE' system, with the tagline 'Advancing Life'. A 'Home' link is visible. The main heading is 'Laporan Penjualan' (Sales Report), which is highlighted with a red dashed border. Below this, there is a form with the following fields: 'Per Periode' (For Period), 'Tanggal Awal' (Start Date) with a date picker, and 'Sampai Dengan' (Up to) with a date picker. At the bottom of the form are two buttons: 'cetak' (print) and 'batal' (cancel). The footer of the page reads 'Copyright © 2020'.

Gambar IV.18. Laporan

9. Pegawai

Gambar IV.19. Pegawai

ADVANCE

Advancing Life

[Home](#)

Registrasi Data Pegawai

Nama:

Nama

Tanggal Lahir

mm/dd/yyyy

Jenis Kelamin

Pilih Jenis Kelamin

Alamat :

Alamat

No Telepon :

No Telepon

Level :

Pilih Level User

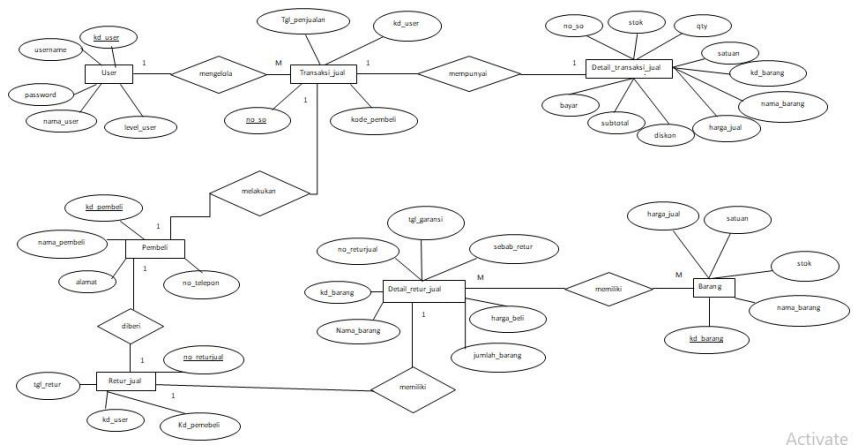
submit

batal

Copyright © 2020

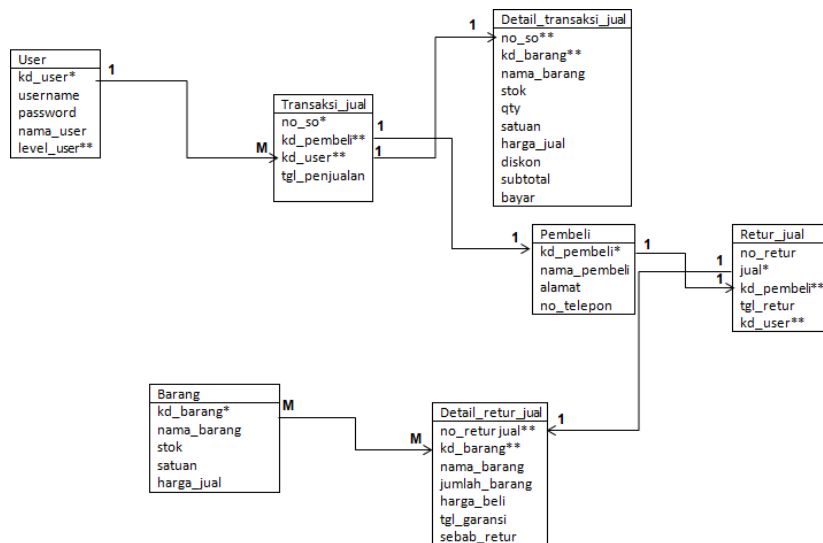
4.2 Perancangan Perangkat Lunak

4.2.1 Entity Relationship Diagram (ERD)



Gambar IV.20. Entity Relationship Diagram

4.2.2 Logical Record Structure (LRS)



Gambar IV.21. Logical Record Structure

4.2.3 Spesifikasi File

1. Spesifikasi File User

Nama File : User

Akronim : user

Fungsi : Untuk menyimpan data user

Tipe File : File Master

Organisasi File: *Index Sequential*

Akses File : *Random*

Media : *Hard disk*

Panjang Record: 75 byte

Kunci field : kd_user

Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	Kode User	kd_user	Varchar	5	Primary Key
2	Username	username	Varchar	20	
3	Password	password	Varchar	15	
4	Nama User	nama_user	Char	25	
5	Level User	level_user	Enum ('1', '2', '3')	10	

Tabel IV.9. Spesifikasi File User

2. Spesifikasi File Pembeli

Nama File : Pembeli
 Akronim : pembeli
 Fungsi : Untuk menyimpan data pembeli
 Tipe File : File Master
 Organisasi File: *Index Sequential*
 Akses File : *Random*
 Media : *Hard disk*
 Panjang Record: 112 *byte*
 Kunci field : kd_pembeli
 Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	Kode Pembeli	kd_pembeli	<i>Varchar</i>	5	<i>Primary Key</i>
2	Nama Pembeli	nama_pembeli	<i>Char</i>	20	
3	Alamat	Alamat	<i>Varchar</i>	75	
4	No Telepon	no_telepon	<i>Int</i>	12	

Tabel IV.10. Spesifikasi File Pembeli

3. Spesifikasi File Transaksi Penjualan

Nama File : Transaksi Penjualan
 Akronim : transaksi_jual
 Fungsi : Untuk menerangkan tentang transaksi penjualan
 Tipe File : File Transaksi
 Organisasi File: *Index Sequential*
 Akses File : *Random*
 Media : *Hard disk*
 Panjang Record: 40 *byte*
 Kunci field : no_nota
 Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	No SO	no_so	<i>Int</i>	30	<i>Primary Key</i>

2	Tanggal Penjualan	tgl_penjualan	Date		
3	Kode Pelanggan	kd_pembeli	Varchar	5	<i>Foreign Key</i>
4	Kode User	kd_user	Int	5	<i>Foreign Key</i>

Tabel IV.11. Spesifikasi File Transaksi Penjualan

4. Spesifikasi File Detail Transaksi Penjualan

Nama File : Detail Transaksi Penjualan

Akronim : Detail_transaksi_jual

Fungsi : Untuk lebih menerangkan data-data yang ada dalam transaksi penjualan

Tipe File : File Transaksi

Organisasi File: Index Sequential

Akses File : Random

Media : Hard disk

Panjang Record: 143 byte

Kunci field : -

Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	No Sales Order	no_so	Int	25	Foreign Key
2	Kode Barang	kd_barang	Varchar	10	Foreign Key
3	Nama Barang	nama_barang	Varchar	50	
4	Stok	Stok	Int	5	
5	Qty	Qty	Int	5	
6	Satuan	Satuan	Char	8	
7	Harga Jual	harga_jual	Int	10	
8	Diskon	Diskon	Int	10	
9	Subtotal	Subtotal	Int	10	
10	Bayar	Bayar	Int	10	

Tabel IV.12. Spesifikasi File Detail Transaksi Penjualan

5. Spesifikasi File Barang

Nama File : Barang

Akronim : barang

Fungsi : Untuk menyimpan data barang
 Tipe File : File Transaksi
 Organisasi File: Index Sequential
 Akses File : Random
 Media : Hard disk
 Panjang Record: 80 byte
 Kunci field : kd_barang
 Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	Kode Barang	kd_barang	Varchar	5	Primary Key
2	Nama Barang	nama_barang	Varchar	50	
3	Stok	stok	Int	5	
4	Satuan	satuan	Char	10	
5	Harga Jual	harga_jual	Int	10	

Tabel IV.13. Spesifikasi File Barang

6. Spesifikasi File Retur Penjualan

Nama File : Retur Penjualan
 Akronim : Retur_jual
 Fungsi : Untuk menerangkan tentang transaksi pembelian
 Tipe File : File Transaksi
 Organisasi File: *Index Sequential*
 Akses File : *Random*
 Media : *Hard disk*
 Panjang Record: 45 byte
 Kunci field : -
 Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	No Retur Penjualan	no_retur_jual	<i>Int</i>	25	<i>Primary Key</i>
2	Kode Pembeli	kd_pemebeli	<i>Int</i>	5	<i>Foreign Key</i>
3	Tanggal Retur	tgl_retur	<i>Date</i>		
4	Kode User	kd_user	<i>Varchar</i>	15	<i>Foreign Key</i>

Tabel IV.14. Spesifikasi File Retur Penjualan

7. Spesifikasi File Detail Retur Penjualan

Nama File : Detail Retur Penjualann

Akronim : Detail_retur_jual

Fungsi : Untuk menerangkan detail tentang ada yang ada di retur penjualan

Tipe File : File Transaksi

Organisasi File: Index Sequential

Akses File : Random

Media : Hard disk

Panjang Record: 105 byte

Kunci field : -

Software : Xampp

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	No Retur	no_retur_jual	<i>Int</i>	15	<i>Foreign Key</i>
2	Kode Barang	kd_barang	<i>Varchar</i>	5	<i>Foreign Key</i>
3	Nama Barang	nama_barang	<i>Varchar</i>	30	
4	Jumlah Barang	jumlah_barang	<i>Int</i>	5	
5	Harga	harga_beli	<i>Int</i>	10	
6	Tanggal Garansi	tgl_garansi	<i>Date</i>		
7	Sebab Retur	sebab_retur	<i>Char</i>	40	

Tabel IV.15. Spesifikasi File Detail Retur Penjualan

9. Spesifikasi File Data Pegawai

Nama File : Data Pegawai

Akronim : pegawai

Fungsi : Untuk menyimpan data pelanggan

Tipe File : File Master

Organisasi File: *Index Sequential*

Akses File : *Random*

Media : *Hard disk*

Panjang Record: 155 *byte*

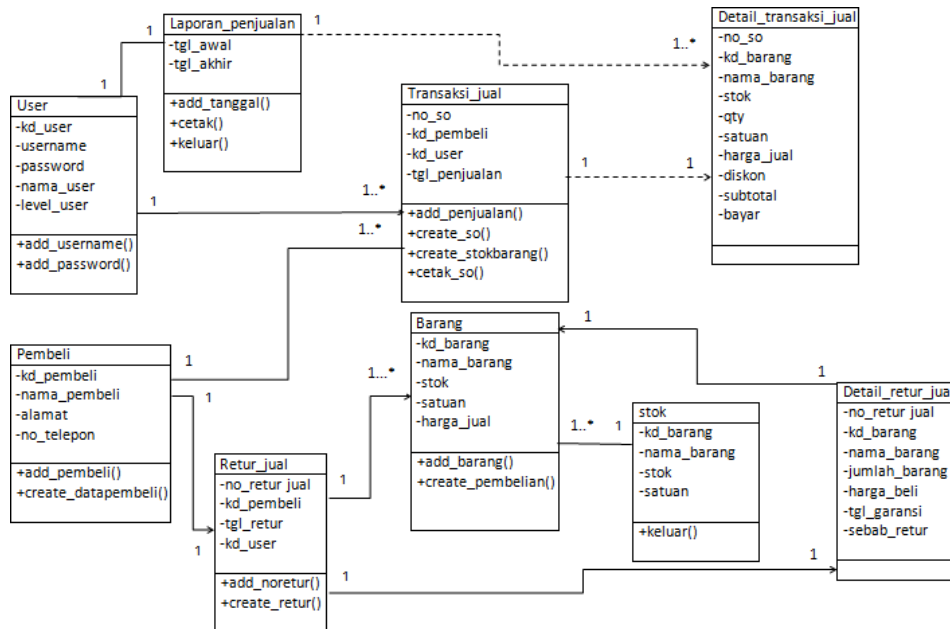
Kunci field : *kd_pegawai*

Software : *Xampp*

No	Elemen Data	Nama Field	Tipe Data	Size	Keterangan
1	Kode Pegawai	kd_pegawai	<i>Int</i>	5	<i>Primary Key</i>
2	Nama	Nama	<i>Varchar</i>	18	
3	Tanggal Lahir	tgl_lahir	<i>Date</i>		
4	Jenis Kelamin	jenis_kelamin	<i>Char</i>	10	
5	Alamat	Alamat	<i>Varchar</i>	100	
6	No Telepon	no_telepon	<i>Int</i>	12	
7	Bagian	Bagian	<i>Char</i>	10	

Tabel IV.16. Spesifikasi File Data Pegawai

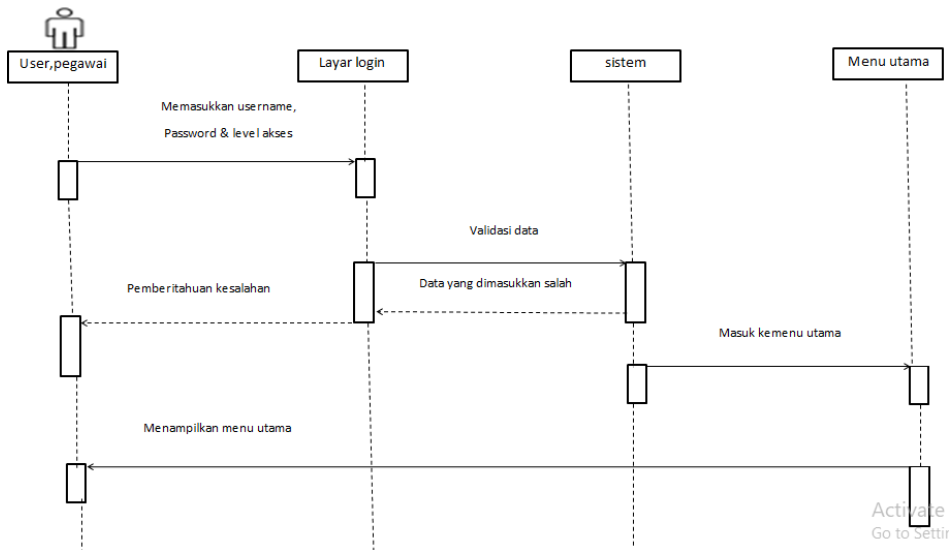
4.2.4 Class Diagram/Class Model



Gambar IV.22. Class Diagram

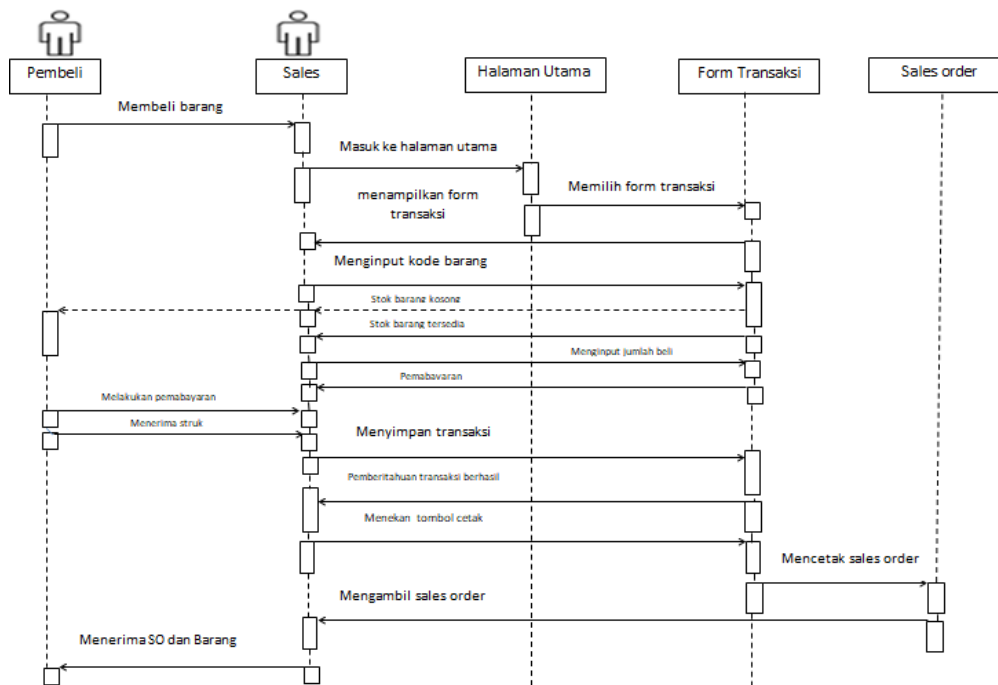
4.2.5 Sequence Diagram

1. Sequence Diagram Login



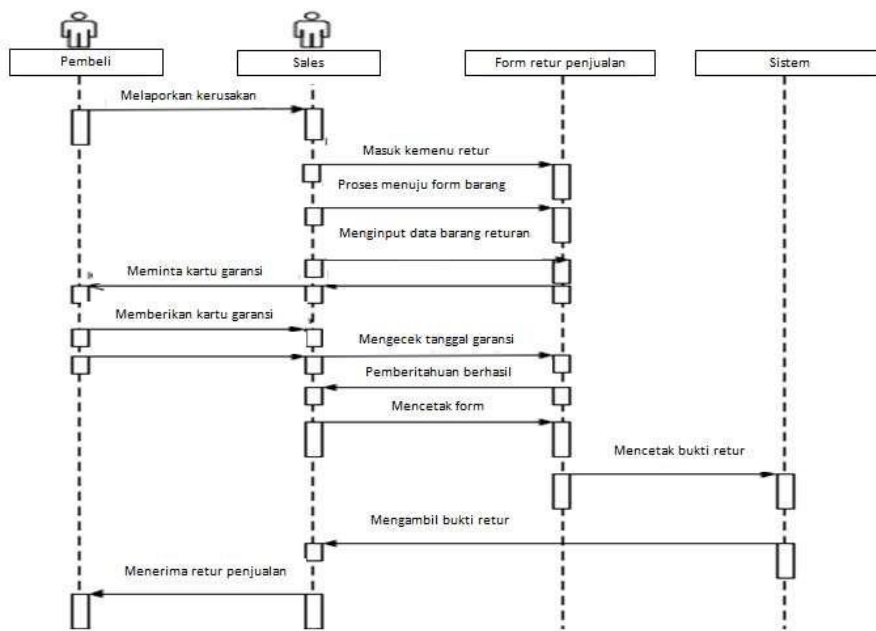
Gambar IV.23. Sequence Diagram Login

2. Sequence Diagram Transaksi Penjualan



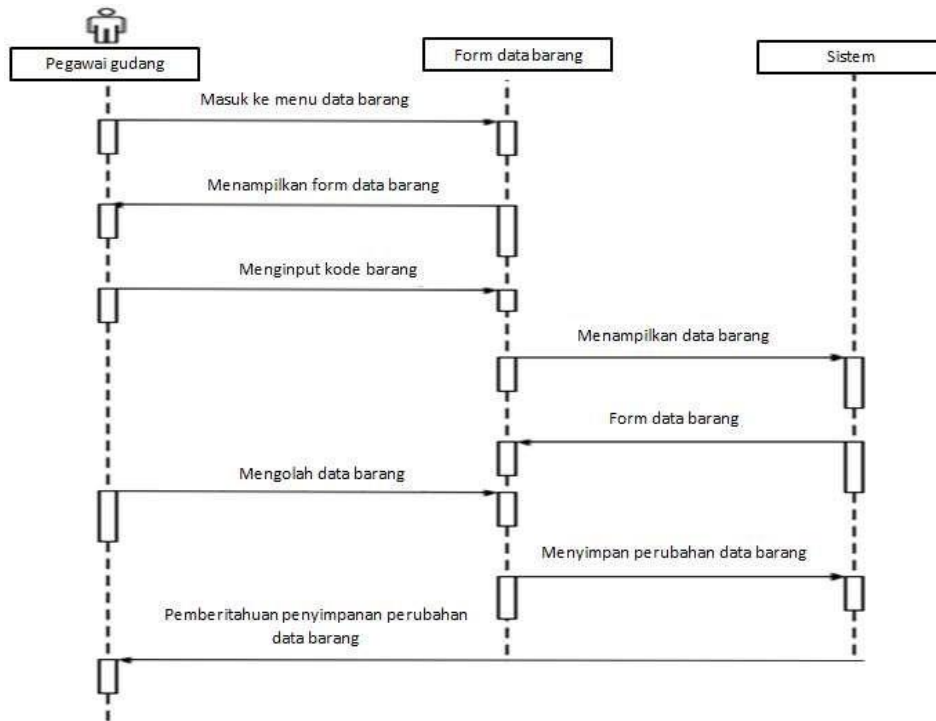
Gambar IV.24. Sequence Diagram Transaksi Penjualan

3. Sequence Diagram Retur Penjualan



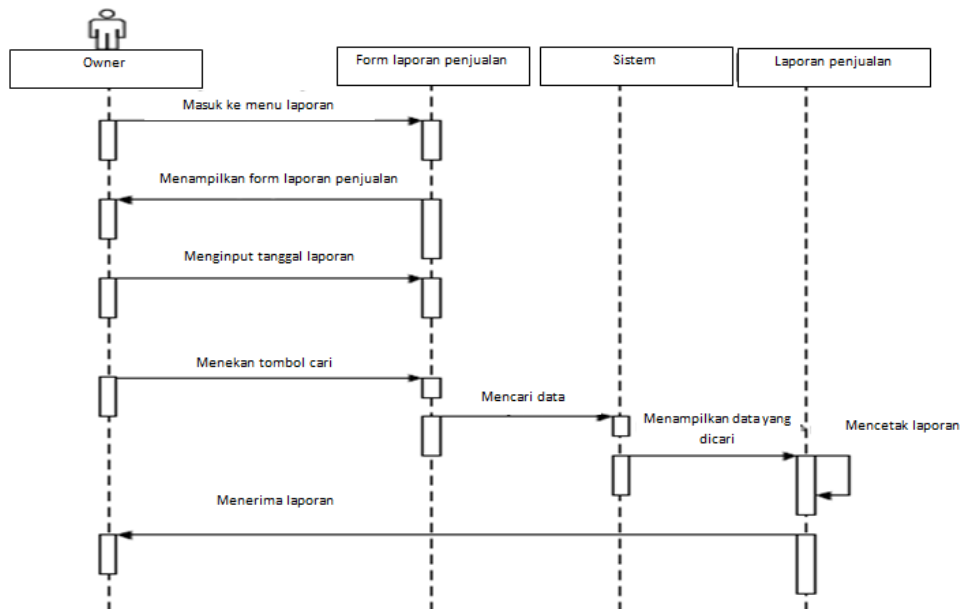
Gambar IV.25. Sequence Diagram Retur Penjualan

4. Sequence Diagram Data Barang



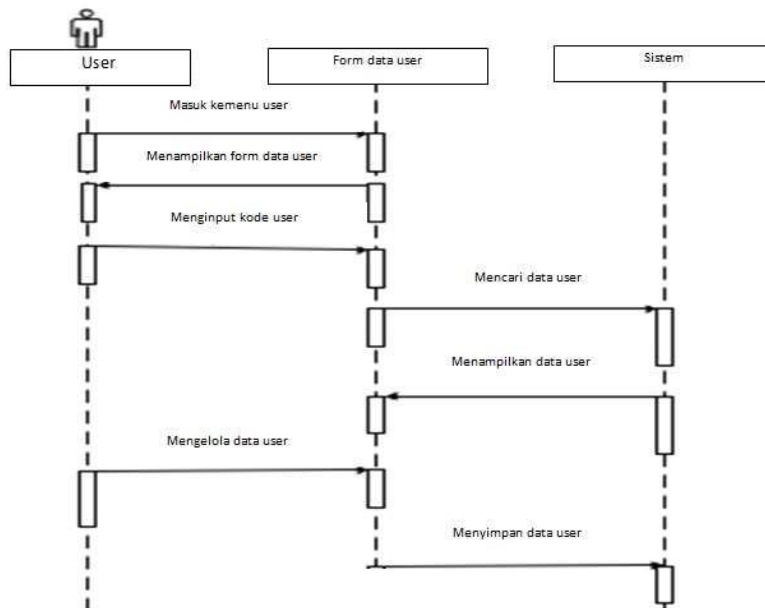
Gambar IV.26. Sequence Diagram Data Barang

5. Sequence Diagram Laporan Penjualan



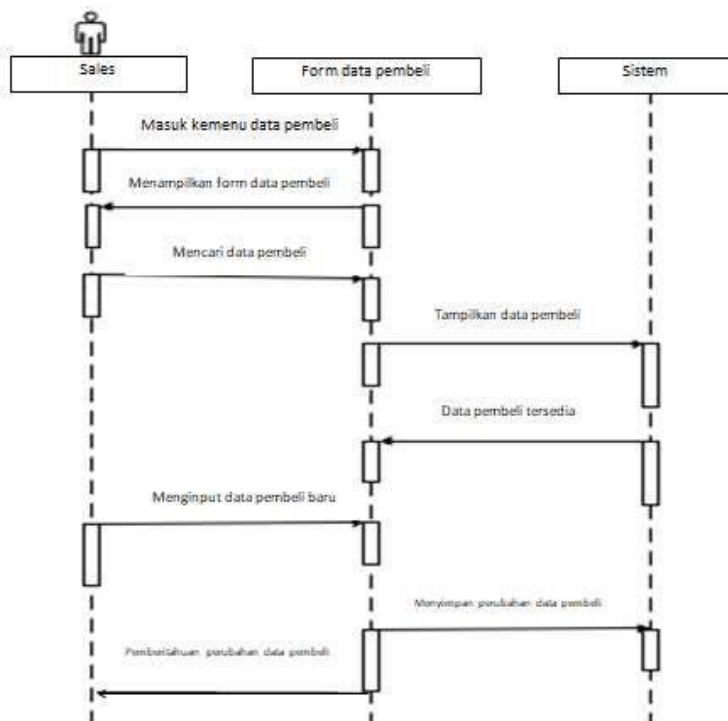
Gambar IV.27. Sequence Diagram Laporan

6. Sequence Diagram Data User



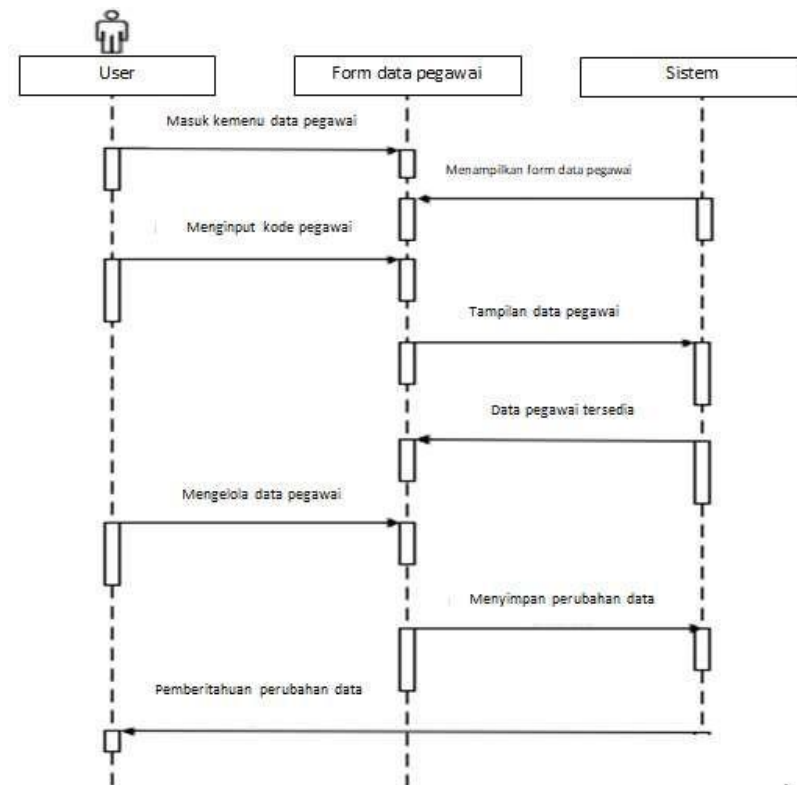
Gambar IV.28. Sequence Diagram Data User

7. Sequence Diagram Data Pembeli



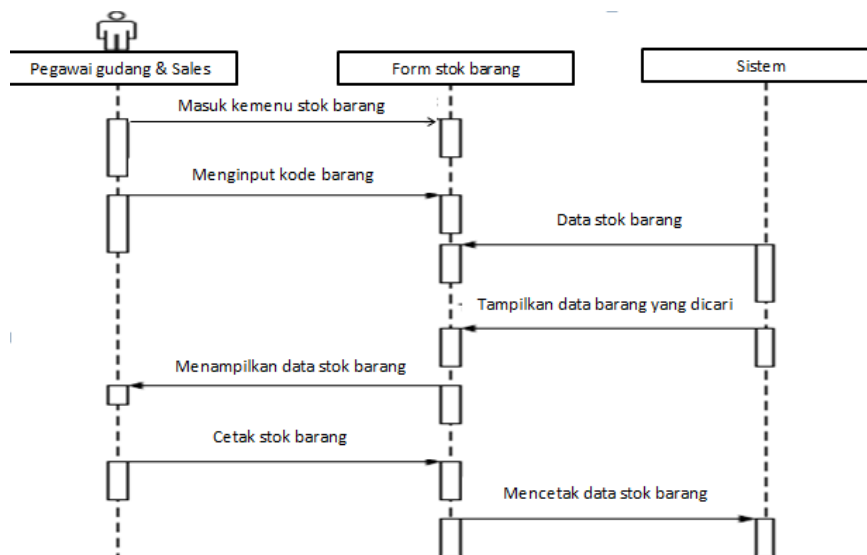
Gambar IV.29. Sequence Diagram Data pembeli

8. Sequence Diagram Data Pegawai



Gambar IV.30. Sequence Diagram Data Pegawai

9. Sequence Diagram Stok Barang



Gambar IV.31. Sequence Diagram Stok Barang



Sumber: Advance store Bekasi

Lampiran D.1 – Sales Order

Sales Order

ADVANCE

No : XXXXXXX

Nama : XXXXXXX

Date : XXXXXXX

Alamat : XXXXXXX

No Telepon : XXXXXXX

No	Kode Barang	Nama Barang	Qty	Harga Satuan	Total Harga
X	XXXXXXX	XXXXXXX	X	Rp. XXXXXXX	Rp. XXXXXXX
X	XXXXXXX	XXXXXXX	X	Rp. XXXXXXX	Rp. XXXXXXX

ADVANCING

Diskon : Rp. XXXXXXX
Subtotal : Rp. XXXXXXX
Bayar : Rp. XXXXXXX

Sales

Customer

XXXX

XXXXXX

Lampiran D.2 – Retur Barang

Retur Penjualan

ADVANCE

No : XXXXXXX

Nama : XXXXXXX

Tanggal : XXXXXXX

Sebab Retur : XXXXXXX

No	Kode Barang	Nama Barang	Qty	Harga	Total Harga
X	XXXXXXX	XXXXXXX	X	Rp. XXXXXXX	Rp. XXXXXXX
X	XXXXXXX	XXXXXXX	X	Rp. XXXXXXX	Rp. XXXXXXX

Sales

Customer

Tanggal Cetak

XX-XX-XXXX



XXXX

XXXXXXX

Lampiran D.3 – Laporan Penjualan Barang

ADVANCE

LAPORAN PENJUALAN

XX-XX-XXXX s/d XX-XX-XXXX

No	Tanggal	No SO	Nama Barang	Qty	Harga	Diskon	Total Harga
X	XXXXXXX	XXXXX	XXXXXXX	X	Rp. XXXX	Rp. XXXX	Rp. XXXX
X	XXXXXXX	XXXXX	XXXXXXX	X	Rp. XXXX	Rp. XXXX	Rp. XXXX

Tanggal Cetak

XX-XX-XXXX

Lampiran Stok Barang

ADVANCE

STOK BARANG

XX-XX-XXXX s/d XX-XX-XXXX

No	Kode Barang	Nama Barang	Qty	Satuan	Harga
X	XXXXXXX	XXXXX	X	Rp. XXXX	Rp. XXXX
X	XXXXXXX	XXXXX	X	Rp. XXXX	Rp. XXXX

Tanggal Cetak

XX-XX-XXXX

Source Code JSP

===Database Inventory

== Table structure for table Inventory

```
|-----  
|Column|Type|Null|Default  
|-----  
|/**id**/|int(11)|No|  
|label|varchar(50)|No|  
|Nama_barang|int(11)|No|
```

== Dumping data for table course

== Table structure for table inventory

```
|-----  
|Column|Type|Null|Default  
|-----  
|stock_barang_id|int(11)|No|  
|Laporan_penjualan_id|int(11)|No|  
|Sales_order|float|No|  
|description|varchar(150)|No|
```

== Dumping data for table sales order

== Table structure for table Laporan Stock

```
|-----  
|Column|Type|Null|Default  
|-----  
|/**id**/|int(11)|No|  
|Qty_qty|varchar(50)|No|  
|Harga_Satuan|varchar(50)|No|  
|Kode_barang|char(1)|Harga_Satuan|  
|Total_Harga|Retur_penjualan||
```

```
|Qty|varchar(50)|Transaksi|
|Retur_barang|text|No|
== Dumping data for table Inventory
== Table structure for table user
|-----
|Column|Type|Null|Default
|-----
|/**id**//|int(11)|No|
|username|varchar(50)|Nama|
|Jenis_Kelamin|varchar(50)|User|Password
|Level created|date|No|
== Dumping data for table user
```

Source Code Control Servlet

```
package control;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import Persediaan barang alat kesehatan.; //mengakses parameter
import Transaksi;
import Barang;
import Pembeli;
import User;
import Laporan;
import Pegawai;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.annotation.WebServlet;

/**
```

```

*
* @author michael
*/

@WebServlet(name = " RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN
BARANG MENGGUNAKAN METODE QUICK MODELLING (METODE SECARA
CEPAT) PADA ADVANCE DI BEKASI Servlet", urlPatterns = {"/Laporan penjualan &
Laporan stock barangTest1Servlet"})

public class pendaftaran Siswa Baru1Servlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException, ClassNotFoundException, SQLException
    {
        response.setContentType("text/html;charset=UTF-8");

Sales Order= new Transaksi(); //--Inisialisasi Object—

        String URL = "jdbc:mysql://localhost:3306/ ADVANCE DI BEKASI ";
        String USERNAME = "root";
        String PASSWORD = "";

        Connection koneksi = null;
        PreparedStatement pstmt = null;
        PreparedStatement pstmt2 = null;
        int result = 0;
        try (PrintWriter out = response.getWriter()) {

```

```

        /* TODO output your page here. You may use following sample code. */
        Class.forName("com.mysql.jdbc.Driver");
        koneksi = DriverManager.getConnection(URL, USERNAME, PASSWORD);

        String aksi = request.getParameter("aksi");
        if (aksi != null) {

            }
            break;
        default:
            break;
        }
    }
}
}
}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Laporan Stock
Barang1Servlet.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {

```

```

        Logger.getLogger(pembelian1Servlet.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Laporan
Penjualan1Servlet.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Stock barang1Servlet.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```