

BAB III

METODOLOGI

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Ruang Laboratorium Jurusan Teknologi Informasi Universitas Muhammadiyah Yogyakarta. Adapun waktu penelitian Sistem Informasi Manajemen SG Mandiri ini dilaksanakan dari bulan Mei 2016 - Agustus 2016.

3.2. Alat dan Bahan

Di dalam penelitian ini mempergunakan alat untuk mendukung perancangan, pengembangan, dan implementasi sistem informasi. Alat yang digunakan dikategorikan ke dalam 2 golongan, yaitu perangkat keras dan perangkat lunak.

3.2.1. *Software*

Untuk membangun sistem informasi manajemen SG Mandiri, pada penelitian ini dibutuhkan beberapa *software*, yaitu:

- a. Visual Studio 2013 sebagai software pengembangan aplikasi.
- b. SQL Server 2014 sebagai database yang digunakan aplikasi
- c. Balsamiq Mockups 3 sebagai software perancangan user interface
- d. Microsoft Visio 2013 untuk merancang gambaran sistem

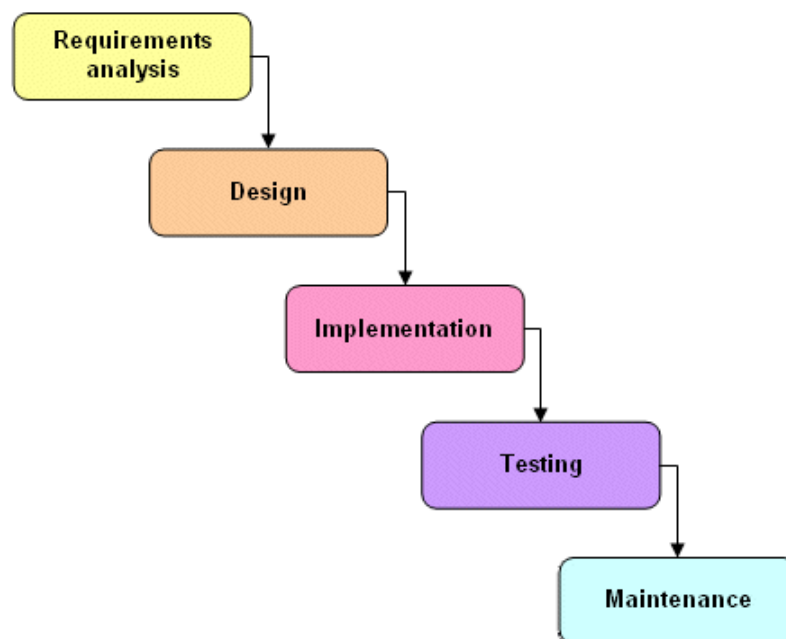
3.2.2. *Hardware*

Selain perangkat lunak (*software*), dibutuhkan pula perangkat keras (*hardware*) yang digunakan untuk mendukung sistem informasi manajemen SG Mandiri, yaitu:

- a. Laptop, spesifikasi 32/64 *bit architecture*, Sistem Operasi Windows 8/10, 2 GB RAM, .NET Framework 3.5
- b. *Personal Computer* (PC) atau laptop dengan spesifikasi minim 32 bit *architecture*, RAM 2 GB , .NET Framework 3.5 untuk akses client di *device* berbeda.
- c. *Printer Kasir DotMatrix* EPSON TM-220D untuk mencekat nota

3.3. **Langkah Penelitian**

Dalam penelitian ini, pengembangan aplikasi menggunakan metode pengembangan *software* model *waterfall*. Model ini merupakan turunan dari metode *SDLC* (*Software Developing Life Cycle*) dan merupakan model klasik yang mengusung pengembangan perangkat lunak yang sistematis/sekuensial. Inti dari model *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Metode *waterfall* ini memiliki 5 tahapan, yaitu *requirement analysis*, *system design*, *implementation*, *testing*, dan *maintenance*.



Gambar 3.1 Model *Waterfall*

Penjelasan metode *Waterfall* yang terdapat pada Gambar 3.1, yaitu:

3.3.1. Analisis Kebutuhan

Di atau analisis kebutuhan, seluruh kebutuhan software harus bisa didapatkan, termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

Pada tahap ini dilakukan open interview yang dilakukan dengan pemilik usaha secara langsung dan diketahui bahwa dibutuhkan aplikasi untuk membantu sekretaris untuk mencatat dan memantau transaksi yang dilakukan, mencatat presensi pegawai sehingga pembagian dan pemantauan gaji lebih mudah, serta pengaturan dan pemantauan kredit yang dilakukan pegawai dan pelanggan.

3.3.2. Perancangan Sistem

Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan. Berdasarkan analisis kebutuhan yang sudah dilakukan, perancangan sistem dilakukan dengan membangun *use case diagram*, *class diagram*, *activity diagram* pada setiap proses dengan menggunakan *software Microsoft Visio 2013*. Setelah kebutuhan sistem sudah jelas, lalu dikembangkan rancangan antar muka dengan memanfaatkan *software balsamiq*.

3.3.3. Implementasi

Dalam tahap implementasi ini desain yang sudah disusun di realisasikan dengan menggunakan kode kode program. Proses tersebut menghasilkan sebuah arsitektur keseluruhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin sudah ditransformasikan kedalam satu atau lebih program yang dapat dijalankan. Tahapan ini telah menentukan alur software hingga pada tahap algoritma detail. Pembangunan dan implementasi desain sistem dilakukan dengan menggunakan bahasa pemrograman C# ASP.NET dengan pola arsitektur MVC, software yang digunakan untuk pengembangan aplikasi adalah *visual studio 2013*.

3.3.4. Testing

Pada tahap ini unit program akan diintegrasikan dan diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah aplikasi selesai diuji, maka sistem sudah siap disampaikan

kepada *client* dan dipakai dalam proses yang diperlukan. Pada pengujian sistem yang dilakukan kali ini menggunakan metode *black box testing*.

3.3.5. Maintenance

Tahap maintenance atau pemeliharaan diperlukan pada saat *software* selesai dibangun dan diimplementasikan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat masih bisa terus berkembang. Ketika *software* dijalankan, bisa ditemui *error* atau *bug* yang tidak ditemukan sebelumnya, atau bisa juga kebutuhan untuk penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya. Pada penelitian kali ini tidak melibatkan maintenance karena proses yang dilakukan hanya sampai pada tahap *testing* saja.

3.4 Analisis Kebutuhan

3.4.1. Analisis kebutuhan sistem

Berdasarkan proses yang sudah dijabarkan sebelumnya maka dapat disusun kebutuhan sistem yang akan diterapkan di UD. SG Mandiri. Adapun kebutuhan untuk sistem tersebut meliputi:

a. Sistem transaksi jual dan beli

Transaksi yang dilakukan dapat dicatat, dicetak, dan disimpan secara terstruktur sehingga pemantauan dapat dilakukan secara lebih mudah. Sistem penjadwalan tidak perlu dikomputerisasi dikarenakan frekuensinya yang jarang dilakukan dan tidak dibutuhkan pemantauan lebih lanjut.

b. Sistem pengurusan presensi dan gaji

Presensi karyawan yang dilakukan setiap hari dapat dicatat dan setiap satu minggu dilakukan rekap untuk menentukan nominal gaji yang diterima. Sistem yang dilakukan secara manual akan lebih dipermudah dengan cara komputerisasi.

c. Sistem pengontrolan kredit

Data kredit yang dilakukan oleh pegawai atau pelanggan tetap dapat dicatat, dilihat, dan dipantau melalui arsip penyimpanan yang di dalamnya terdapat daftar pemohon kredit beserta barang dan nominalnya di dalam rupiah serta angsuran yang telah dilakukan pemohon.

3.4.2. Analisis kebutuhan perangkat lunak

1. Kebutuhan Input

- a. Data barang, jumlah barang, data pelanggan
- b. Data pegawai, presensi
- c. Data barang kredit, pemohon kredit, dan angsuran

2. Kebutuhan Proses

- a. Proses mencatat, mencetak, menyimpan, dan merekap data transaksi pembelian dan penjualan
- b. Proses menambah, mengatur, menghapus, dan merekap data pegawai beserta presensi hariannya.
- c. Proses menambah, mengatur, menghapus, dan merekap data kredit dan angsuran yang telah tercatat.

3. Kebutuhan Output

- a. Laporan dari masing-masing data rekap transaksi pembelian dan penjualan, pegawai dan presensi, kredit dan angsuran per-pemohon.
- b. Nota transaksi pembelian dan penjualan.

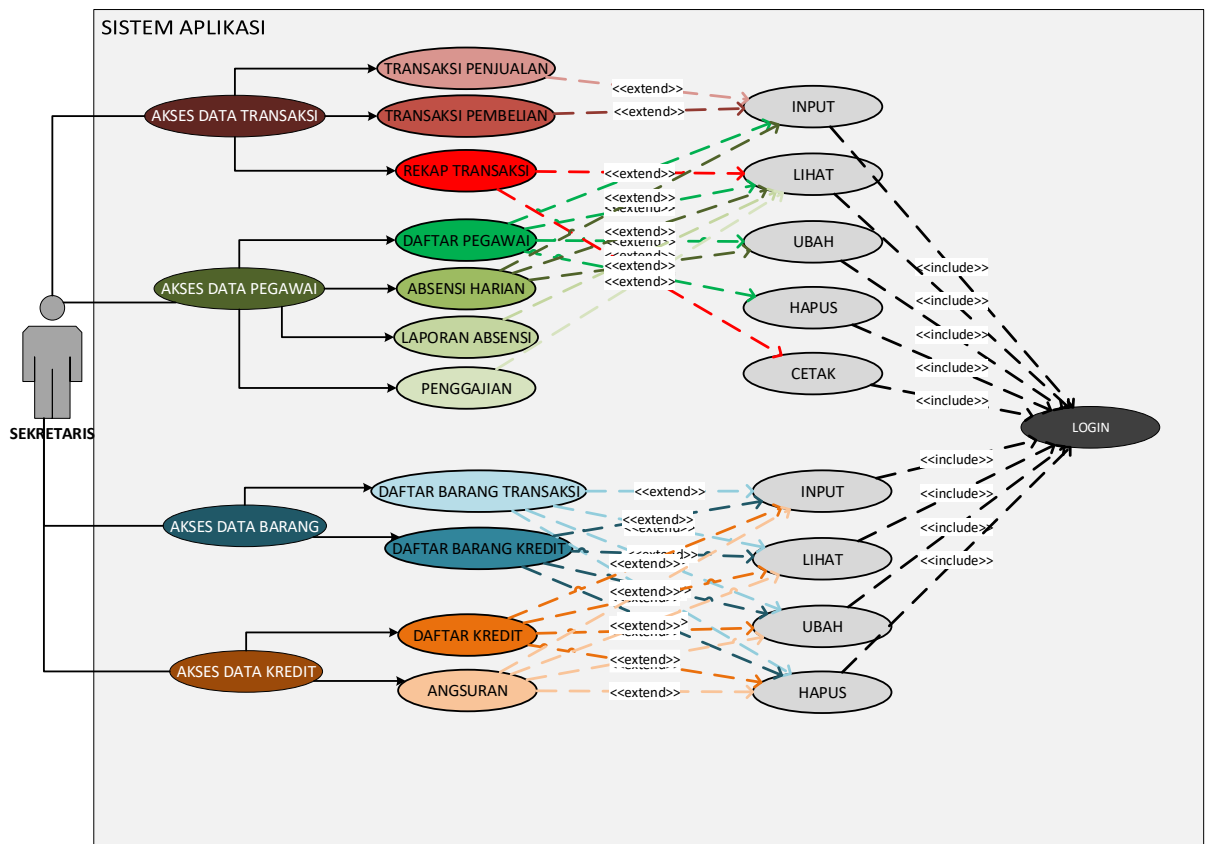
3.5 Rancangan Aplikasi

3.5.1. Rancangan Sistem Informasi

Metode perancangan yang digunakan adalah *United Markup Language* (UML). Seperti yang diketahui, model UML yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*.

Use Case Diagram

Rancangan Use Case diagram yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Use Case Diagram Aplikasi

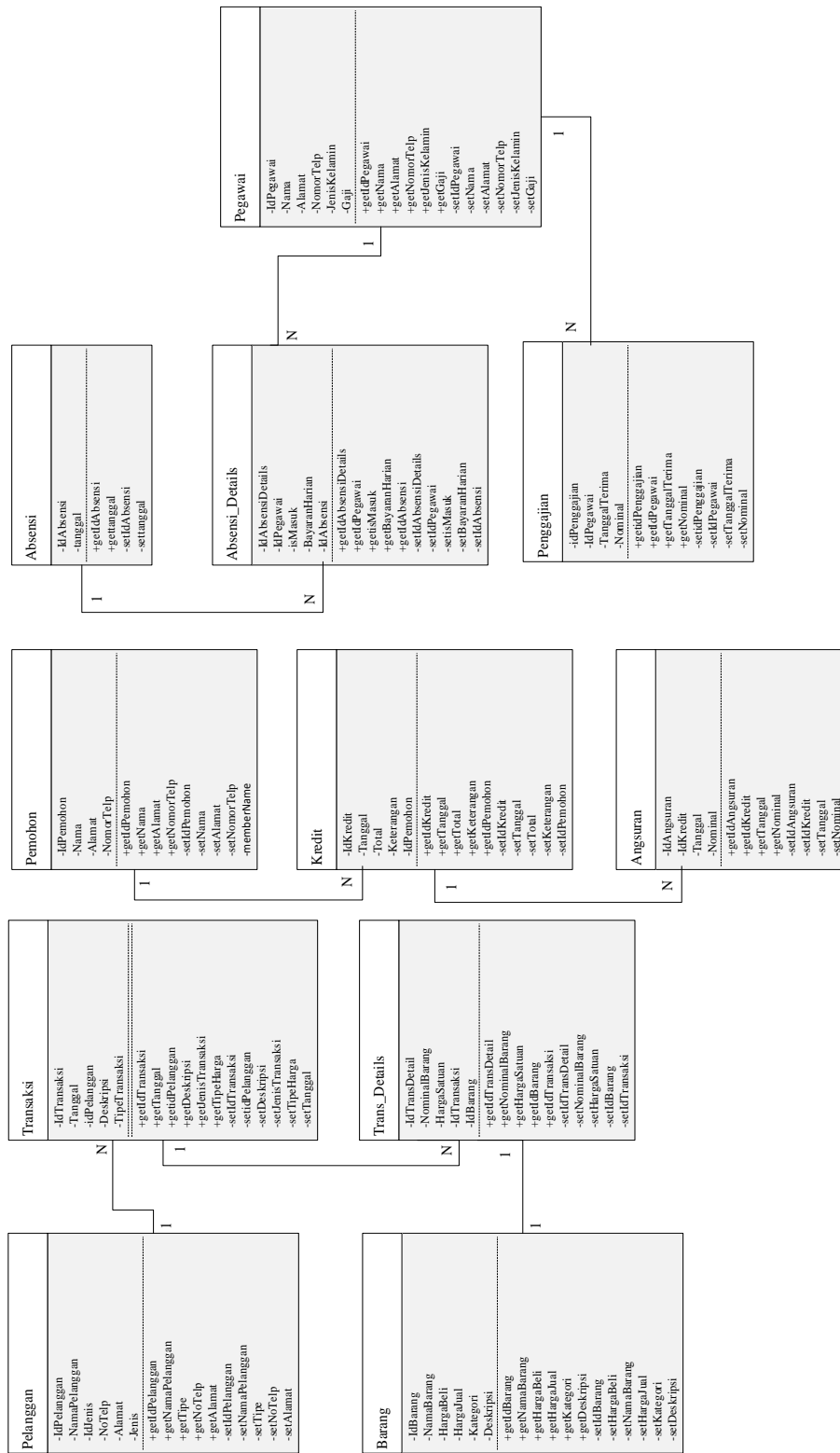
1. Di dalam sistem ini mempunyai satu aktor untuk menjalankan aplikasi, yaitu sekretaris. Aktor dapat melakukan akses ke semua data dan menjalankan fitur-fitur yang ada dengan melakukan proses Log In sebelumnya.
2. Sekretaris dapat mengakses data transaksi yang ada pada fitur transaksi penjualan yang dapat melakukan input data, transaksi pembelian yang juga dapat memasukkan data, dan rekap transaksi yang dapat melihat dan mencetak data.
3. Sekretaris dapat mengakses data pegawai pada fitur daftar pegawai yang dapat memasukkan, melihat, mengubah, dan menghapus data. Dapat mengakses data presensi harian untuk memasukkan, melihat, dan mengubah

data. Lalu dapat mengakses data laporan presensi dan penggajian yang masing-masing dapat melihat data.

4. Sekretaris dapat mengakses data barang pada fitur daftar barang transaksi dan daftar barang kredit yang masing-masing dapat memasukkan, melihat, mengubah, dan menghapus data.
5. Sekretaris dapat mengakses data kredit pada fitur daftar kredit dan data angsuran yang masing masing dapat memasukkan, melihat, mengubah, dan menghapus data.

Class Diagram

Rancangan Class diagram yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.3.



Gambar 3.3 Class Diagram Aplikasi

Penjelasan gambar Class Diagram:

1. Seluruh kelas yang ada di dalam proyek sistem ini terinisialisasi menjadi model tabel yang didasarkan pada database yang dibangun.
2. Kelas Pelanggan, Barang, Pemohon, Pegawai, dan Presensi merupakan kelas yang berdiri sendiri dan diasosiasi oleh kelas lain.
3. Kelas Transaksi berasosiasi dengan kelas Pelanggan, dimana satu pelanggan dapat berhubungan dengan lebih dari satu data transaksi.
4. Kelas Trans_Details merupakan turunan dari kelas Transaksi. Dimana Transaksi memiliki detail transaksi seperti tanggal, pelanggan, tipe transaksi, dan deskripsi. Pada kelas Trans_Details memiliki detail barang yang disertakan dalam transaksi tersebut, seperti data barang, harga, dan nominal barang.
5. Kelas Kredit berasosiasi dengan kelas Pemohon, dimana satu pemohon dapat berhubungan dengan lebih dari satu data dari kelas Kredit.
6. Kelas Angsuran merupakan kelas yang berasosiasi dengan kelas Kredit, dimana satu data dari kelas Kredit dapat berhubungan dengan lebih dari satu data dari kelas Angsuran.
7. Kelas Presensi_Details merupakan turunan dari kelas Presensi dan berasosiasi dengan kelas Pegawai. Sebagai contohnya adalah kelas Presensi memiliki data tanggal kehadiran karyawan dan kelas Presensi_Details memiliki data-data karyawan yang hadir pada tanggal tersebut.
8. Kelas Penggajian merupakan kelas lanjutan dari Presensi, kelas ini memiliki data pegawai, tanggal terima gaji, dan nominal yang didapatkan. Data ini

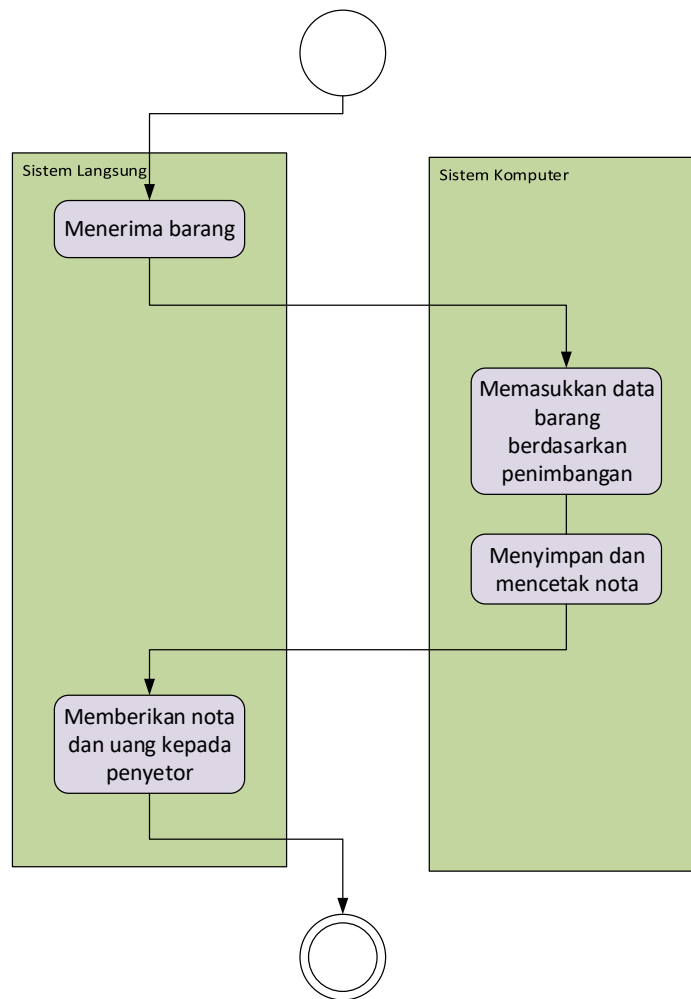
diambil dari gabungan data dari kelas Presensi dan Presensi_Details, lalu diambil secara pemrograman.

Activity Diagram

Berdasarkan *Use Case* Diagram yang telah dibuat sebelumnya maka dapat dirancang *Activity* Diagram dengan aktor “Sekretaris”. Diagram ini merupakan rancangan alur aktifitas yang dilakukan aktor untuk memenuhi proses terkait, lingkungan dalam diagram ini dibedakan menjadi 2 yaitu sistem langsung yang maksudnya adalah kegiatan yang dilakukan secara manual tanpa melibatkan komputer dan sistem komputer yang memerlukan penggunaan komputer di dalam prosesnya.

1. *Activity Diagram* Pembelian

Rancangan *activity diagram* pembelian yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.4.

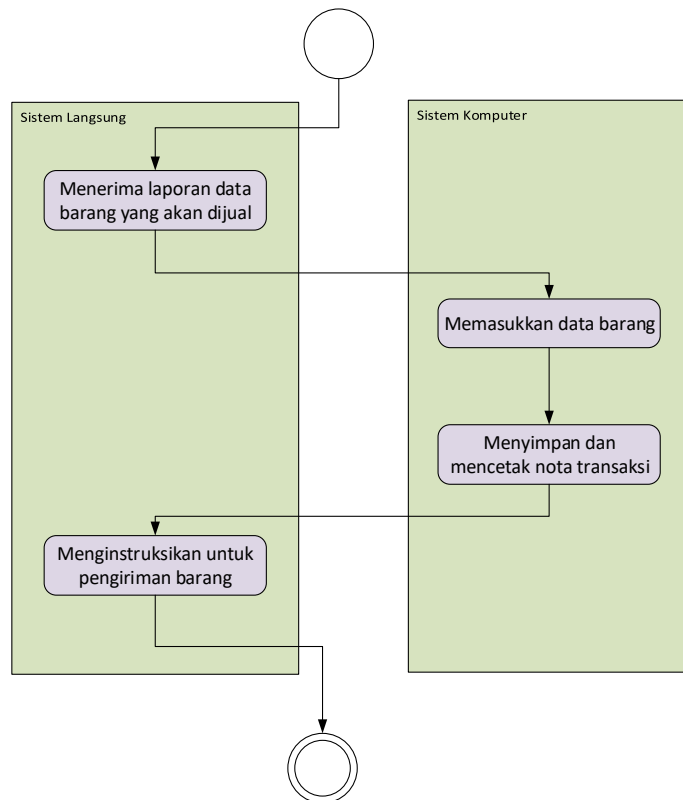


Gambar 3.4 Activity Diagram Pembelian

1. Sekretaris pertama-tama menerima barang dari pelanggan.
2. Sekretaris memasukkan data-data barang yang diterima ke dalam sistem informasi di dalam komputer.
3. Sekretaris menyimpan data dan mencetak nota dari sistem informasi pada komputer.
4. Sekretaris memberikan nota dan uang sebesar yang tertera pada sistem kepada penjual.

2. Activity Diagram Penjualan

Rancangan *activity diagram* penjualan yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.5.

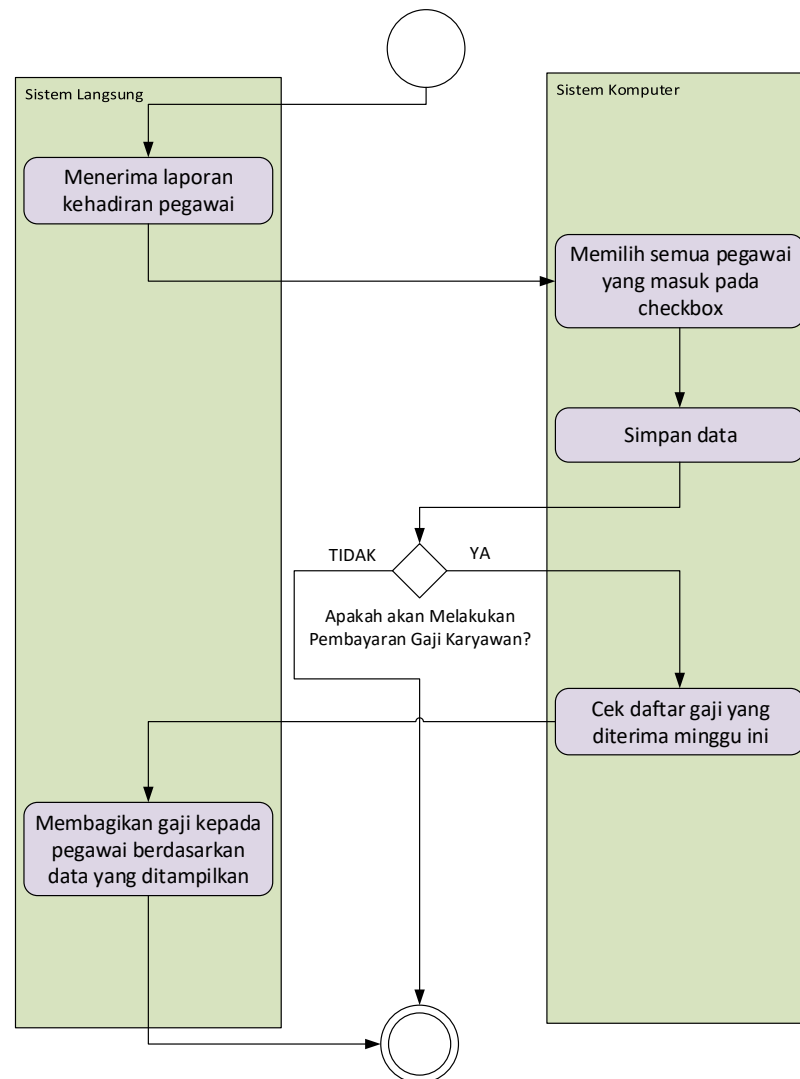


Gambar 3.5 Activity Diagram Penjualan

1. Alur sistem penjualan dimulai dengan sekretaris menerima laporan data barang yang ada disetorkan atau dijual dan memasukkannya ke dalam sistem informasi yang ada di komputer.
2. Data yang sudah dimasukkan kemudian disimpan dan dicetak nota transaksi yang dilakukan.
3. Sekretaris kemudian menginstruksikan kepada pegawai lain untuk pengiriman barang.

3. Activity Diagram Kehadiran dan Gaji

Rancangan *activity diagram* kehadiran dan penggajian yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.6.



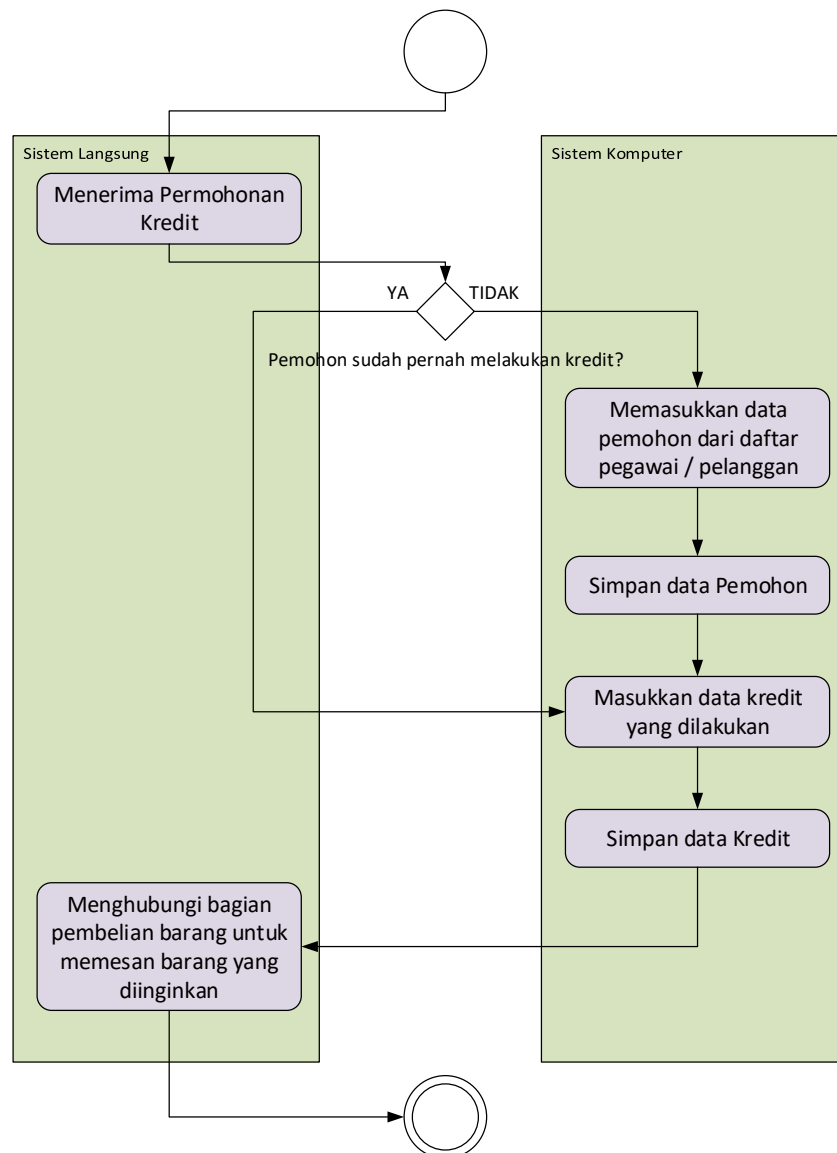
Gambar 3.6 Activity Diagram Kehadiran dan Gaji

1. Pegawai melaporkan kehadirannya kepada sekretaris, lalu sekretaris mencatat kehadiran pegawai tersebut.
2. Data presensi dimasukkan kedalam sistem informasi dan menyimpannya.

3. Setiap satu minggu sekali dilakukan pembayaran gaji pegawai berdasarkan banyak hari masuknya pada hari sabtu, jika akan melakukan pembayaran maka sekretaris dapat membuka bagian penggajian dan akan ditampilkan besaran gaji yang diberikan.
4. Sekretaris membagikan gaji kepada tiap pegawai sebesar masing-masing bagian.

4. *Activity Diagram* Permohonan Kredit

Rancangan *activity diagram* permohonan kredit yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.7.



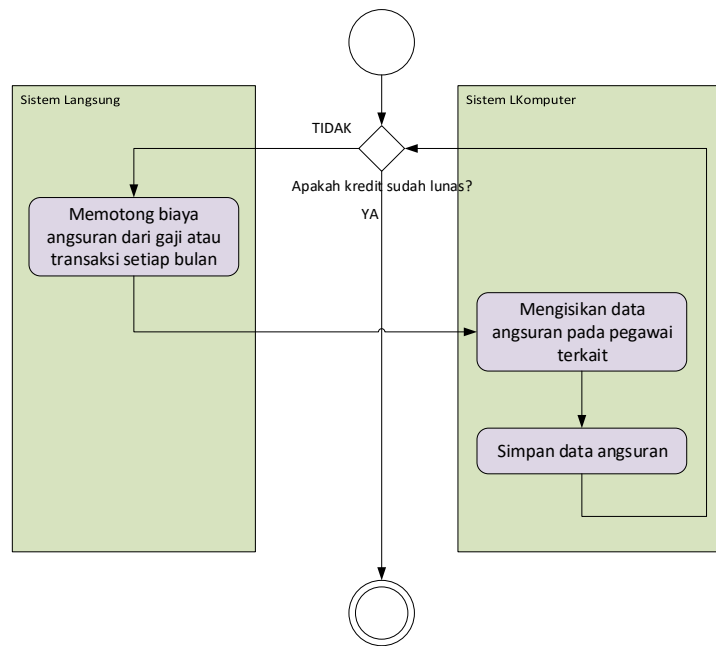
Gambar 3.7 Activity Diagram Permohonan Kredit

Penjelasan:

1. Bagi pegawai yang mau melakukan kredit dapat melaporkan kepada sekretaris dan datanya didaftarkan pada sistem informasi termasuk data barang yang ingin dikreditkan.
2. Sekretaris menghubungi bagian pemesanan barang untuk mendapatkan barang yang diinginkan.

5. Activity Diagram Angsuran Kredit

Rancangan *activity diagram* angsuran kredit yang digunakan dalam pembangunan aplikasi ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Activity Diagram Angsuran Kredit

Penjelasan:

1. Untuk angsuran kredit bagi pemohon kredit yang sudah mendaftarkan kreditnya, sekretaris memotong gaji pegawai setiap pegawai sebesar kesepakatan kedua belah pihak.
2. Sekretaris mencatat dan menyimpan data angsuran pada aplikasi sistem informasi.

3.5.2. Rancangan Basis Data

Untuk pengembangan aplikasi pemodelan basidata yang digunakan adalah *ER Diagram*. *ER Diagram* merupakan diagram yang digunakan untuk merancang basis data dan memperlihatkan relasi antar entitas atau objek dengan atributnya.

Tujuan *ER Diagram* dalam pengembangan aplikasi adalah dapat memberikan gambaran umum tentang sistem yang dikembangkan sehingga memudahkan dalam merancang basis data. Gambaran *ER Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.9. Penjelasan Gambar 3.9 adalah:

1. Terdapat 13 entitas pada 3 proses yang berbeda dan tiap entitas memiliki atribut dan relasi.
2. Entitas Jenis berhubungan dengan entitas Pelanggan dengan kardinalitas *many to one*, artinya satu atau lebih data Jenis dapat dimiliki oleh satu pelanggan.
3. Entitas Pelanggan berhubungan dengan entitas Transaksi dengan kardinalitas *one to many*, artinya satu data Pelanggan dapat berhubungan dengan satu atau lebih data Transaksi.
4. Entitas Transaksi berhubungan dengan entitas Transaksi_Details dengan kardinalitas *one to many*, yang artinya satu data Transaksi dapat berhubungan dengan satu atau lebih data Transaksi_Details.
5. Entitas Transaksi_Details memiliki hubungan dengan entitas Barang dengan kardinalitas *optional one to one*, yang berarti satu data Transaksi_Details bisa jadi berhubungan dengan satu data Barang.
6. Entitas Barang berhubungan dengan entitas Kategori dengan kardinalitas *many to one*, yang berarti satu atau lebih data barang dapat berhubungan dengan satu data kategori.

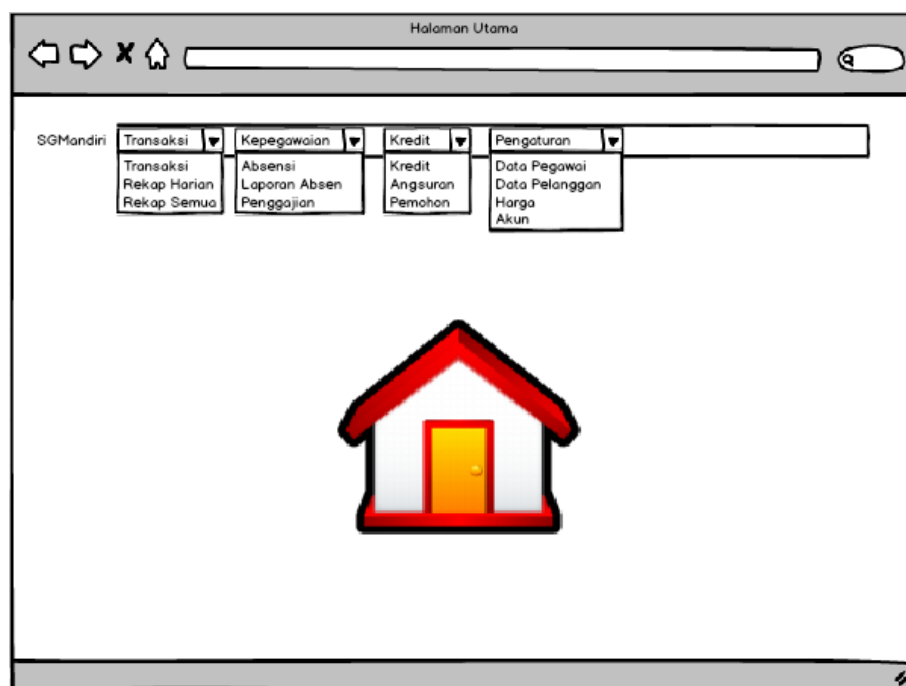
7. Entitas Presensi berhubungan dengan entitas Pegawai dengan melalui relasi yang memiliki atribut sendiri dan dirangkap menjadi anak tabel presensi yang disebut dengan entitas asosiatif dengan kardinalitas *one to many*.
8. Entitas asosiatif Presensi berhubungan dengan entitas Pegawai dengan kardinalitas *many to many*, artinya satu data dari entitas asosiatif Presensi dapat berhubungan dengan satu atau lebih data Pegawai dan juga berlaku sebaliknya.
9. Entitas Pegawai berhubungan dengan entitas Penggajian dengan kardinalitas *one to many*, yang berarti satu data Pegawai dapat memiliki satu atau lebih data Penggajian.
10. Entitas Pemohon berhubungan dengan entitas Kredit dengan kardinalitas *one to many*, yang berarti satu data Pemohon dapat berhubungan dengan satu atau lebih data Kredit.
11. Entitas Kredit berhubungan dengan entitas Angsuran dengan kardinalitas *one to many*, yang artinya satu data Kredit dapat berhubungan dengan satu atau lebih data Angsuran.
12. Entitas Users tidak berhubungan dengan entitas lain dan berlaku sebagai penyimpanan data pengguna aplikasi.

3.5.3. Rancangan Antarmuka (*User Interface*)

Sebelum mengembangkan sistem seutuhnya, salah satu yang penting untuk dirancang adalah tampilan dan fungsi aplikasi. Rancangan antar muka akan dijadikan dasar bagi pengembang untuk mengembangkan aplikasi.

a. Tampilan Utama

Pada tampilan utama akan ditampilkan keterangan tentang perusahaan SG Mandiri dan beberapa info tentang aplikasi seperti pada Gambar 3.10.



Gambar 3.10 Tampilan Utama

Di paling atas akan terdapat menu navigasi pada setiap halaman yang dibuka, menu ini berguna untuk mempermudah pengguna untuk berpindah antar halaman. Fungsi setiap menu dan submenu pada navigasi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel Keterangan Navigasi Tampilan Utama

No	Menu	Sub Menu	Keterangan
1	Transaksi	Transaksi	Masuk ke dalam fitur transaksi dimana data transaksi dapat disimpan dan dicetak.
		Rekap Harian	Data-data transaksi yang dilakukan dalam satu hari ditampilkan dan dapat dilihat rinciannya.

		Rekap Semua	Data-data transaksi yang telah dilakukan ditampilkan dan dapat dilihat rinciannya berdasarkan filter nama atau jenisnya.
2	Kepegawaian	Presensi	Presensi karyawan yang dilakukan setiap hari oleh sekretaris.
		Laporan Kehadiran	Data-data presensi dikumpulkan dan dapat dilihat rinciannya di dalam halaman laporan kehadiran.
		Penggajian	Menampilkan besaran gaji yang diterima karyawan setiap minggunya berdasarkan banyaknya masuk kerja dalam seminggu.
3	Kredit	Kredit	Membuat data kredit pada pemohon yang telah terdaftar.
		Angsuran	Menambahkan / mengubah data angsuran yang dilakukan pada kredit tertentu.
		Pemohon	Mendaftarkan pemohon dalam database untuk dapat dilakukan kredit.
4	Pengaturan	Data Pegawai	Ditampilkan semua daftar karyawan yang tercatat, dan dapat menambah, mengubah, serta menghapus data karyawan.
		Data Pelanggan	Ditampilkan semua daftar pelanggan yang tercatat, dan dapat menambah, mengubah, serta menghapus data pelanggan.
		Harga	Ditampilkan semua daftar barang beserta harganya yang tercatat, dan dapat menambah, mengubah, serta menghapus data barang.
		Akun	Ditampilkan semua daftar pengguna aplikasi yang tercatat, dan dapat menambah, mengubah, serta menghapus data barang.

b. Tampilan Transaksi

Halaman transaksi digunakan untuk menghitung dan memasukkan data transaksi yang dilakukan ke dalam basis data. Gambaran rancangan halaman transaksi dapat dilihat pada Gambar 3.11.

The screenshot shows a web application window titled "TRANSAKSI". At the top, there is a navigation menu with options: "Transaksi", "Kepegawaian", "Laporan Transaksi", "Kredit", and "Pengaturan". Below the menu, the form contains the following fields:

- Nama Penjual:** A dropdown menu with "Jumakir" selected.
- Tanggal:** A date picker showing "22/01/2016".
- Keterangan:** A text input field.
- Jenis:** A dropdown menu with "Jual" selected, and a sub-menu showing "Rumah", "Bangkalan", and "Lain2".
- Barang:** A dropdown menu with "Besi A" selected.
- Nominal:** A text input field followed by "kg".

Below these fields is a "Tambah" button. Underneath is a table with the following data:

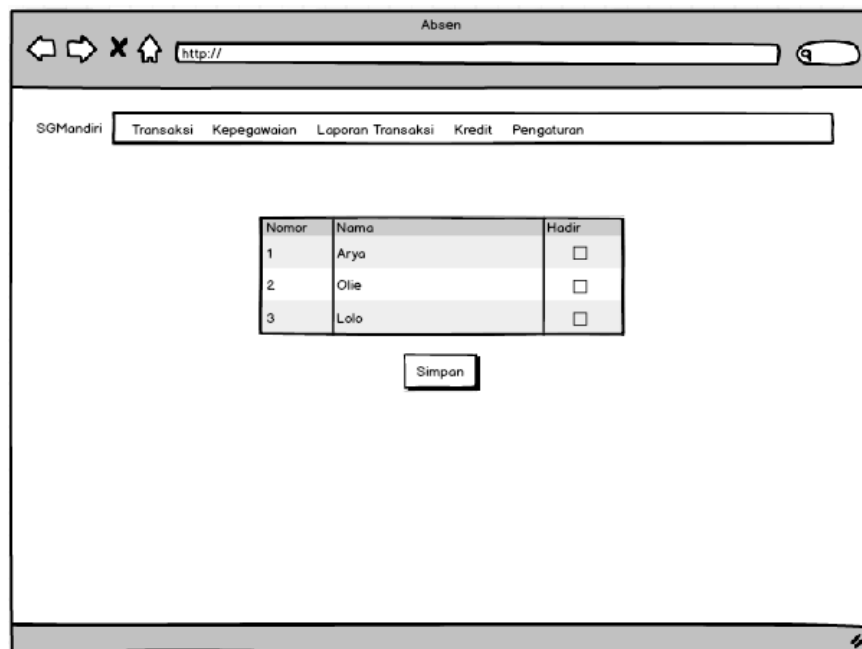
Nama Barang	Berat	@	Total	remove
Besi A	5	13000	65000	remove
Besi B	4	12000	48000	remove
Total			113000	

At the bottom of the form is a "Simpan & Cetak" button.

Gambar 3.11 Tampilan Transaksi

c. Tampilan Presensi Pegawai

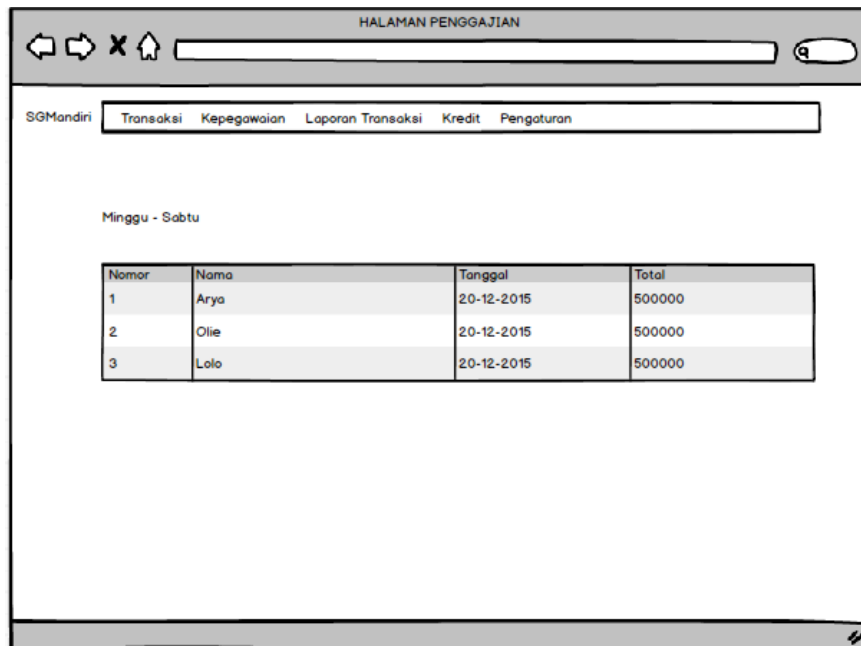
Presensi pegawai dilakukan setiap hari oleh sekretaris / pengguna aplikasi dan memilih karyawan mana saja yang masuk dengan cara mencentang di bagian kolom masuk jika karyawan tersebut masuk pada hari tersebut. Rancangan halaman presensi pegawai dapat dilihat pada Gambar 3.12.



Gambar 3.12 Tampilan Presensi Pegawai

d. Tampilan Penggajian

Gaji dihitung dengan cara perkalian jumlah masuk dalam kurun waktu satu minggu dengan nominal gaji harian yang dimiliki pegawai tersebut. Halaman ini ditujukan untuk membantu sekretaris dalam perhitungan gaji karyawan. Rancangan halaman penggajian dapat dilihat pada Gambar 3.13.

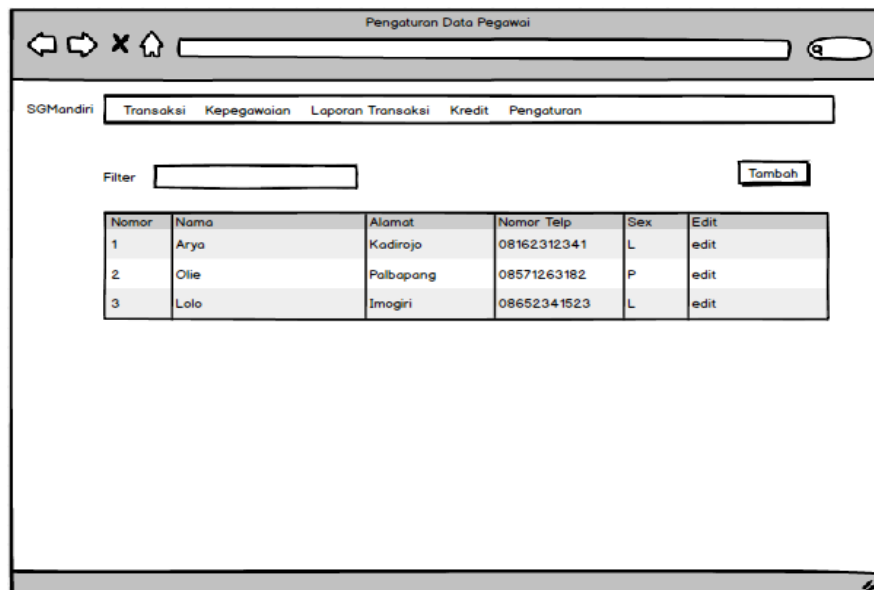


Nomor	Nama	Tanggal	Total
1	Arya	20-12-2015	500000
2	Olie	20-12-2015	500000
3	Lolo	20-12-2015	500000

Gambar 3.13 Tampilan Penggajian

e. Tampilan Data Pegawai

Data setiap pegawai ditampilkan dan dapat diubah jika ada perubahan data, ditambah jika ada karyawan baru. Rancangan halaman data pegawai dapat dilihat pada Gambar 3.14.

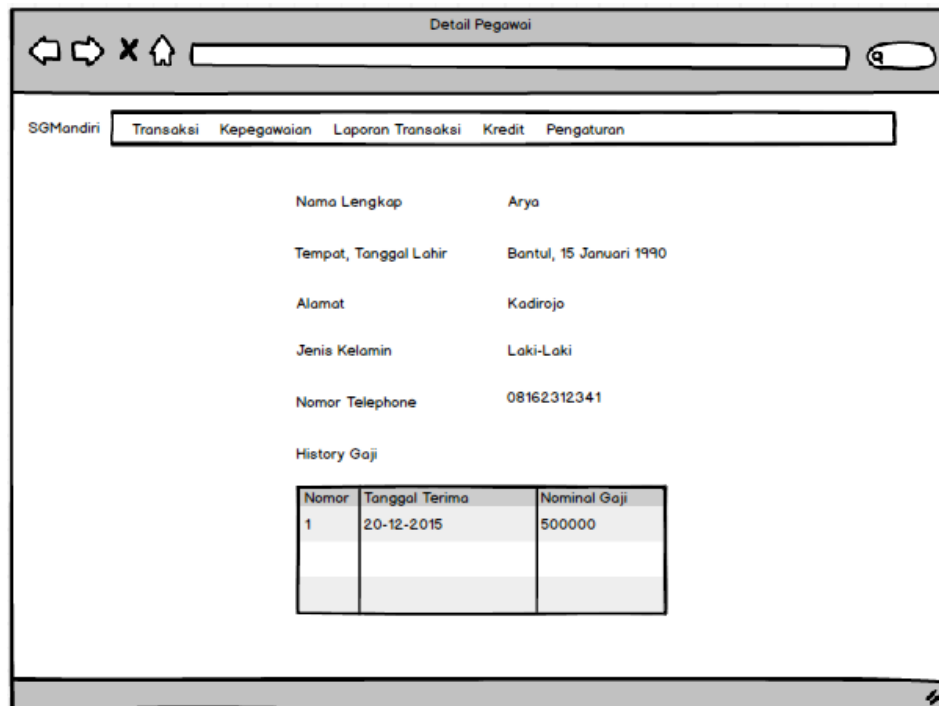


Nomor	Nama	Alamat	Nomor Telp	Sex	Edit
1	Arya	Kadirojo	08162312341	L	edit
2	Olie	Palbapang	08571263182	P	edit
3	Lolo	Imogiri	08652341523	L	edit

Gambar 3.14 Tampilan Data Pegawai

f. Tampilan Detail Data Pegawai

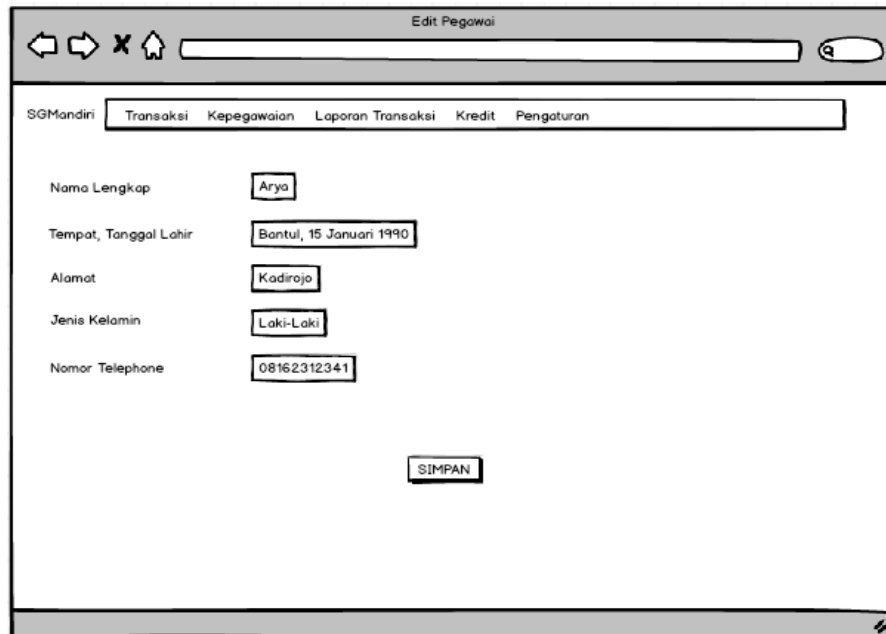
Rincian data pegawai dapat dilihat pada halaman detail data pegawai termasuk riwayat gaji yang diterima. Rancangan halaman detail data pegawai dapat dilihat pada Gambar 3.15.



Gambar 3.15 Tampilan Detail Data Pegawai

g. Tampilan Edit Data

Halaman edit data digunakan untuk mengubah data yang sudah ada. Rancangan halaman edit dapat dilihat pada Gambar 3.16.



The screenshot shows a web browser window titled "Edit Pegawai". The address bar contains "SGMandiri" and a search icon. Below the address bar is a navigation menu with tabs: "Transaksi", "Kepegawaian", "Laporan Transaksi", "Kredit", and "Pengaturan". The main content area contains a form with the following fields and values:

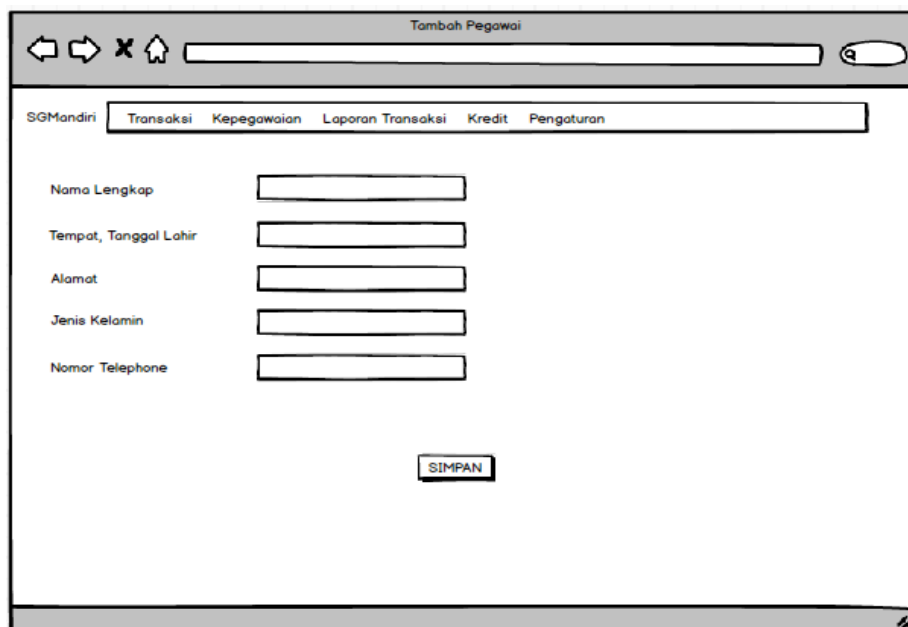
Nama Lengkap	Arya
Tempat, Tanggal Lahir	Bantul, 15 Januari 1990
Alamat	Kadirojo
Jenis Kelamin	Laki-Laki
Nomor Telephone	08162312341

At the bottom center of the form is a button labeled "SIMPAN".

Gambar 3.16 Tampilan Edit Data

h. Tampilan Tambah Data

Halaman tambah data digunakan untuk menambah data yang belum tercatat sebelumnya. Rancangan halaman tambah data dapat dilihat pada Gambar 3.17.



The screenshot shows a web browser window titled "Tambah Pegawai". The address bar contains "SGMandiri" and a search icon. Below the address bar is a navigation menu with tabs: "Transaksi", "Kepegawaian", "Laporan Transaksi", "Kredit", and "Pengaturan". The main content area contains a form with the following fields:

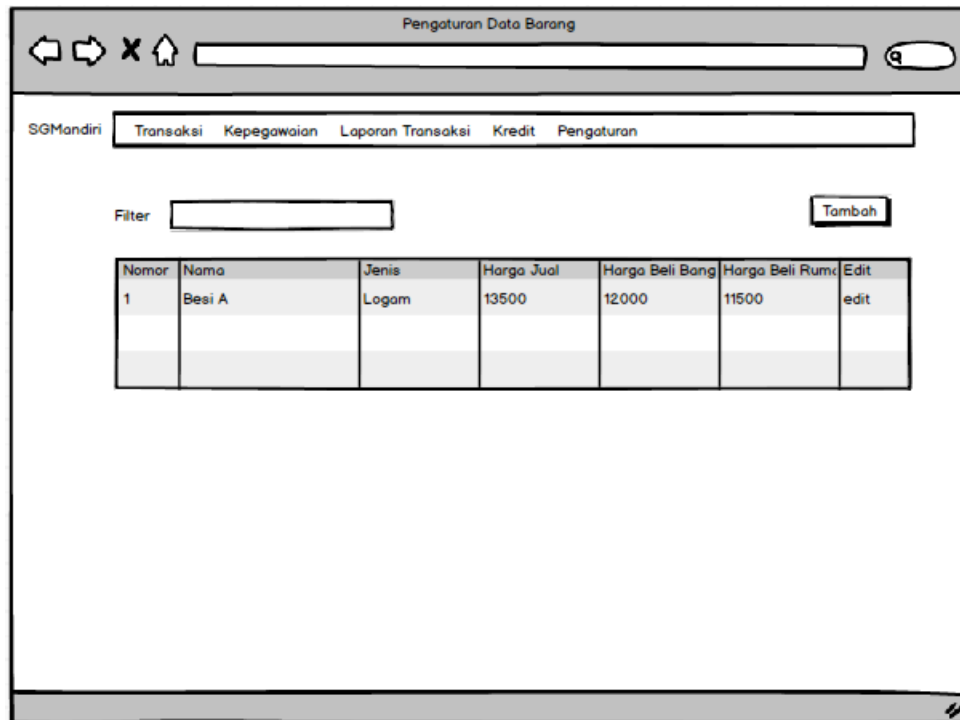
Nama Lengkap	<input type="text"/>
Tempat, Tanggal Lahir	<input type="text"/>
Alamat	<input type="text"/>
Jenis Kelamin	<input type="text"/>
Nomor Telephone	<input type="text"/>

At the bottom center of the form is a button labeled "SIMPAN".

Gambar 3.17 Tampilan Tambah Data

i. Tampilan Data Barang

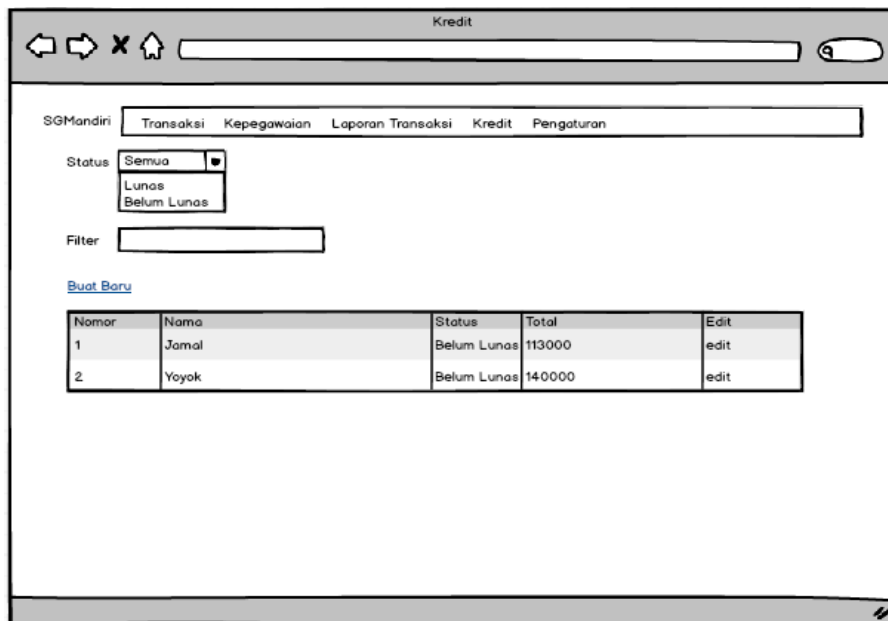
Daftar data barang yang telah tercatat ditampilkan dan dapat ditambahkan baru jika ada barang tambahan, serta mengubah data bila ada data yang berubah. Rancangan halaman data barang dapat dilihat pada Gambar 3.18.



Gambar 3.18 Tampilan Data Barang

j. Tampilan Kredit

Halaman kredit akan menampilkan daftar kredit yang sudah dilakukan, dapat menambahkan dan mengubah data kredit jika diperlukan. Rancangan halaman kredit dapat dilihat pada Gambar 3.19.

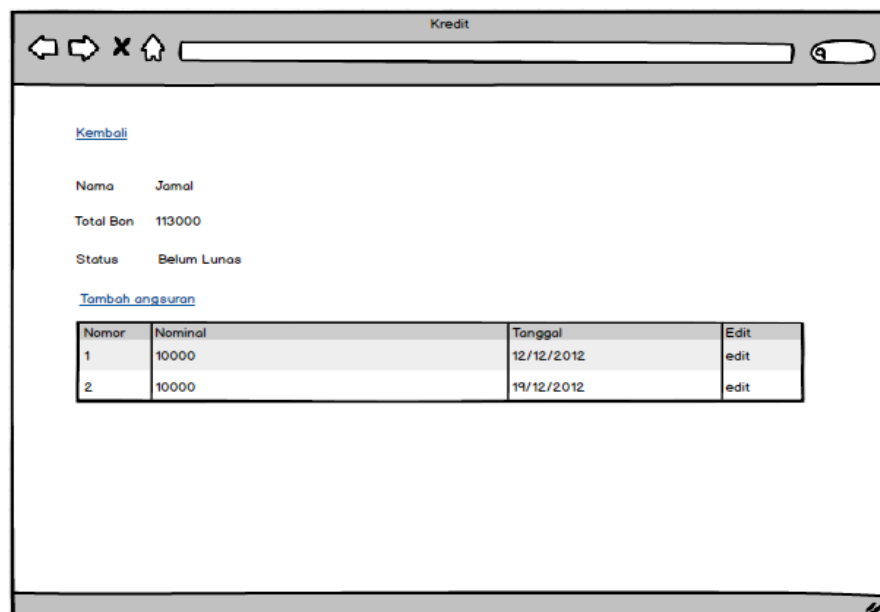


Gambar 3.19 Tampilan Kredit

k. Tampilan Angsuran Kredit

Angsuran yang dilakukan pada kredit tertentu dapat dilihat di halaman angsuran kredit, angsuran dapat ditambahkan atau diubah datanya bila diperlukan.

Rancangan halaman angsuran kredit dapat dilihat pada Gambar 3.20.



Gambar 3.20 Tampilan Angsurang Kredit