BAB IV

HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem

Tahap ini adalah tahap selanjutnya setelah dilakukannya analisis pada sistem yang berjalan. Adapun dari tahap ini adalah untuk memberikan gambaran mengenai sistem pembelian dan penjualan bahan bangunan yang akan diusulkan. Pada tahap ini juga dilakukan mengidentifikasi komponen-komponen sistem in*form*asi yang akan dirancang secara rinci dengan maksud untuk merancang sisitem in*form*asi yang akan diimplementasikan.

Perubahan sistem yang akan dilakukan ialah pada proses pengolahan data pembelian dan penjualan bahan bangunan berupa transkasi penjualan serta penginputan datanya secara komputerisasi, sedangkan untuk pembelian yaitu pembuatan data bahan bangunan yang akan dipesan kepada *supplier* serta penginputan data barang pembeliannya, semuanya dilakukan secara terkomputerisasi .

4.1.1. Tujuan Perancangan Sistem

Tujuan adanya perancangan sistem adalah untuk menghasilkan suatu rancangan sistem yang baik, karena dengan adanya rancangan sistem yang tepat maka akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. Karena jika perancangan kurang baik akan mengakibatkan sistem yang dibangun harus di rubah total atau sistem yang dibangun akan melebihi kebutuhan yang diperlukan.

4.1.2. Gambaran Umum Sistem yang Diusulkan

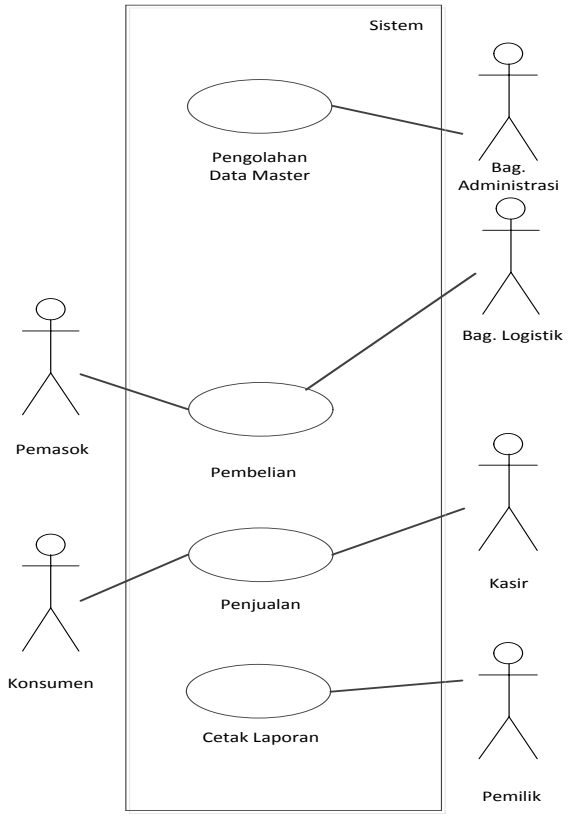
Sistem yang diusulkan adalah sebuah system yang digunakan untuk membantu mengatasi permasalahan yang sering muncul pada sistem in*form*asi yang telah ada dikarenakan akibat pengolahan data yang masih dilakukan semua secara manual seperti data penjualan dan pembelian bahan bangunan maka penulis mempunyai gambaran umum yang akan diusulkan yaitu untuk memperbaiki proses pengolahan data penjualan bahan bangunan maka dirancang suatu sistem in*form*asi penjualan dan pembelian yang terkomputerisasi agar dapat meningkatkan kinerja proses yang efektif dan efesien.

4.1.3. Perancangan Prosedur yang Diusulkan

Perancangan prosedur sistem in*form*asi pembelian dan penjualan bahan bangunan berbasis java yang diusulkan pada CV ALBET ini mencakup *use case diagram*, skenario *use case, activity diagram , sequence diagram, class diagram, deployment diagram.*

4.1.3.1. *Use Case Diagram* yang Diusulkan

Use Case Diagram sangat penting untuk mengorganisasi dan memodelkan perilaku suatu system yang dibutuhkan serta diharapkan pengguna. Adapun *use case diagram* yang diusulkan dari sistem in*form*asi pembelian dan penjulan pada CV ALBET adalah sebagai berikut:



Gambar 4. 1 *Use Case Diagram* yang Diusulkan

Dalam *use case diagram* diatas, dapat terlihat terdapat 5 proses inti yang akan diusulkan pada CV ALBET, yaitu sebagai berikut :

1. Proses pertama, yaitu aktivitas *login* yang terlebih dahulu harus dilakukan oleh pengguna sistem yaitu pemilik, bagian logistik, kasir dan bagian administrasi agar bisa berinteraksi dengan sistem.
2. Proses kedua, yaitu pengolahan seluruh data *master* seperti data pengguna, data pemasok, data kategori barang, data jenis barang, data satuan dan data barang yang dilakukan oleh bagian administrasi yang mempunyai hak akses penuh terhadap pengolahan data master pada sistem ini.

3. Proses ketiga, yaitu proses pembelian seperti pembuatan nota pembelian, penerimaan barang dan pembayaran pembelian yang dilakukan oleh kasir kepada pemasok serta sebagai bahan *input* untuk laporan pembelian yang akan bisa diakse oleh pemilik.
4. Proses keempat, yaitu proses penjualan dimana kasir melakukan proses transaksi penjualan kepada pelanggan, serta pencarian stok data barang yang nantinya akan digunakan dalam pengolahan data bahan *input* untuk laporan penjualan yang akan bias diakse oleh pemilik.
5. Proses kelima, yaitu proses cek laporan yang dilakukan oleh pemilik untuk menngecek laporan pembelian dan penjualan

4.1.3.2. *Scenario Use Case* yang Diusulkan

Scenario Use Case digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Berikut merupakan *scenario use case* pada yang diusulkan pada CV ALBET:

4.1.3.2.1. *Scenario Use Case Login*

Berikut merupakan *scenario use case login* :

Nama *Use Case* : *Login*

Aktor : Bagian Administrasi, Kasir, Pemilik

Tujuan : Proses *user* melakukan l*ogin* dengan mengisi *username* dan *password* kemudian di verifikasi oleh sistem.

Tabel 4. 1 *Scenario Use Case Login* yang Diusulkan

Nama *Use Case* :	*Login*
Tujuan :	Untuk melakukan *login*
Aktor :	Bagin Administrasi, Kasir, Pemilik
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu “File”	
2. Klik tombol “Login”	
	3. Menampilkan *form login*
4. Menginput *username dan password*	
	5. Sistem akan melakukan validasi *username* dan password (jika data *login* valid maka *user* bisa masuk ke dalam sistem, jika data login tidak valid makan *user* tidak bisa masuk ke dalam sistem)
	6. Masuk ke halaman utama *user*
Kondisi Akhir :	*User* masuk ke dalam system

4.1.3.2.2. *Scenario Use Case* Kelola Data *Master*

Berikut merupakan *scenario* kelola data master :

Aktor : Bagian Administrasi

Tujuan : Proses pengolahan data master sebagai mengelola data untuk melakukan proses lain seperti pembelian dan penjualan

Tabel 4. 2 *Scenario Use Case* Kelola Data *Master* yang Diusulkan

Nama *Use Case* :	Kelola Data *Master*
Tujuan :	Mengola data master diantaranya: data pengguna, data pemasok, data kategori barang, satuan barang dan barang.
Aktor :	Bagian Administrasi
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu "Data Master"	
	2. Menampilkan menu-menu yang ada.
3. Pilih ke *form* master	
	4. Menampilkan halaman *form* master.
5. Input data master	
6. Memilih tombol-tombol ekseskusi data seperti tambah, ubah, hapus, batal dan keluar.	
	7. Memvalidasi data master sesuai dengan pemilihan tombol ekseskusi seprtijika kita memilih tombol eksekusi tambah maka sistem akan menampilkan pesan "Data Berhasil Disimpan" jika data tersebut valid.
	8. Sistem akan menyimpan data master dan

	menampilkan data master ketika kita memilih tombol eksekusi tambah dan ubah. Begitu juga jika memilih tombol eksekusi lain makan sistem akan melakukan sesuai dengan tombol yang dipilih.
Kondisi Akhir :	Data master tersimpan dalam *database*.

4.1.3.2.3. *Scenario Use Case* Pembelian

Proses pembelian terdiri dari 3 proses, yaitu: pemesanan pembelian (*purchases order)*, penerimaan barang dan pembayaran pembelian barang.

Tabel 4. 3 Skenario *Use Case Purchase Order* yang Diusulkan

Nama *Use Case* :	Pesanan Pembelian Barang
Tujuan :	Untuk melakukan pemesanan pembelian barang.
Aktor :	Kasir
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu “Data Transaksi”	
	2. Menampilkan menu-menu yang ada.
3. Pilih “Pesanan Pembelian”	
	4. Menampilkan halaman *Form* Pesanan Pembelian.

5. Memeriksa stok barang minimum	
6. Input daftar barang yang akan di beli.	
7. Klik tombol “Simpan”.	
	8. Memvalidasi data, apabila ada data yang sama maka sistem akan menampilkan pesan “data tersebut sudah tersimpan di *database*”, bila berhasil sistem akan menampilkan pesan “Input Berhasil”. Bila simpan data gagal Sistem akan menampilkan pesan “Simpan data gagal”.
	9. Menyimpan data pesanan barang ke dalam *database* (po_barang dan pobarang_detail).
	10. Mencetak *Purchases Order*.
Kondisi Akhir :	Data pesanan barang yang akan dibeli tersimpan di dalam *database* (po_barang dan pobarang_detail) dan *purchases order* telah dicetak.

Tabel 4. 4 *Scenario Use Case* Penerimaan Barang yang Diusulkan

Nama *Use Case* :	Penerimaan Barang
Tujuan :	Untuk melakukan penerimaan pembelian barang
Aktor :	Kasir
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu "Data Transaksi"	
	2. Menampilkan menu-menu yang ada.
3. Pilih "Penerimaan Barang"	
	4. Menampilkan halaman *Form* Penerimaan Barang.
5. Pilih No PO barang 6. Input No. Bukti, No. Surat Jalan, Jumlah Barang yang diterima dari pemasok/*supplier*.	
7. Klik tombol "Simpan Penerimaan".	
	8. Memvalidasi data, apabila ada data yang sama maka sistem akan menampilkan pesan "data tersebut sudah tersimpan di *database*", bila berhasil sistem akan menampilkan

	pesan “Input Berhasil”. Bila simpan data gagal Sistem akan menampilkan pesan “Simpan data gagal”.
	9. Menyimpan data penerimaan barang ke dalam *database* (po_masuk dan pomasuk_detail).
Kondisi Akhir :	Data penerimaan barang yang dibeli tersimpan di dalam *database* (po_masuk dan pomasuk_detail).

Tabel 4. 5 Skenario *Use Case* Pembayaran Pembelian yang Diusulkan

Nama *Use Case* :	Pembayaran Pembelian Barang
Tujuan :	Untuk melakukan pembayaran pembelian barang
Aktor :	Kasir
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu “Data Transaksi”	
	2. Menampilkan menu-menu yang ada.
3. Pilih “Pembayaran Pembelian Barang”	
	4. Menampilkan *Form* Pembayaran Pembelian Barang.

5. Klik Tombol “Tambah”	
6. Pilih No. PO barang	
	7. Hitung Total Pembayaran
8. Pilih Metode Pembayaran	
9. Input jumlah bayar	
10. Klik tombol “Simpan”.	
	11. Memvalidasi data, apabila ada data yang sama maka sistem akan menampilkan pesan “data tersebut sudah tersimpan di *database*”, bila berhasil sistem akan menampilkan pesan “Input Berhasil”. Bila simpan data gagal Sistem akan menampilkan pesan “Simpan data gagal”.
	12. Menyimpan data pembayaran pembelian barang ke dalam *database* (pembayaran_beli dan pembayaranbeli_detail).
Kondisi Akhir :	Data pembayaran barang yang dibeli tersimpan di dalam *database*.

4.1.3.2.4. *Scenario Use Case* Penjualan

Proses penjualan terdiri dari 3 proses, yaitu: penjualan, dan pencetakan surat jalan

.

Tabel 4. 6 *Scenario Use Case* Penjualan yang Diusulkan

Nama *Use Case* :	Penjualan Barang
Tujuan :	Untuk melakukan penjualan
Aktor :	Kasir.
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu "Data Transaksi".	
	2. Menampilkan menu-menu yang ada.
3. Pilih "Penjualan".	
	4. Menampilkan halaman *Form* Penjualan.
5. Memeriksa stok barang.	
6. Input data konsumen (Nama Konsumen, Alamat, No. Telepon) dan data barang (kode barang, nama barang, jumlah) yang dipesan konsumen.	
	7. Hitung Total Pembayaran

8. Input Jumlah yang dibayar oleh konsumen.	
9. Klik tombol “Simpan”.	
	10. Memvalidasi data, apabila ada data yang sama maka sistem akan menampilkan pesan “data tersebut sudah tersimpan di *database*”, bila berhasil sistem akan menampilkan pesan “Input Berhasil”. Bila simpan data gagal Sistem akan menampilkan pesan “Simpan data gagal”.
	11. Menyimpan data barang yang dijual ke dalam *database* (penjualan dan penjualan_detail).
	12. Mencetak faktur penjualan.
Kondisi Akhir :	Data barang yang dijual tersimpan di dalam *database* (penjualan dan penjualan_detail) dan faktur penjualan telah dicetak.

Tabel 4. 7 *Scenario Use Case* Pengeluaran Barang yang Diusulkan

Nama *Use Case* :	Pengeluaran Barang
Tujuan :	Untuk melakukan pengeluaran barang
Aktor :	Kasir
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu "Data Transaksi"	
	2. Menampilkan menu-menu yang ada.
3. Pilih "Pengeluaran Barang"	
	4. Menampilkan halaman *Form* Pengeluaran Barang.
5. Pilih No. SO Barang	
6. *Input* nama sopir dan nomor kendaraan	
7. Input jumlah barang keluar	
8. Klik tombol "Simpan Transaksi"	
	9. Memvalidasi data, apabila ada data yang sama maka sistem akan menampilkan pesan "data tersebut sudah tersimpan di *database*", bila berhasil sistem akan menampilkan

	pesan "*Input* Berhasil". Bila simpan data gagal Sistem akan menampilkan pesan "Simpan data gagal".
	10. Menyimpan data pembayaran penjualan ke dalam *database* (barang_keluar dan barangkeluar_detail).
	11. Mencetak surat jalan.
Kondisi Akhir :	Data pembayaran barang yang dijual tersimpan di dalam *database* (barang_keluar dan barangkeluar_detail) dan mencetak surat jalan.

4.1.3.2.5. *Scenario Use Case* Cetak Laporan

Aktor : Pemilik

Tujuan Untuk mencetak laporan pembelian dan penjualan.

Tabel 4. 8 *Scenario Use Case* Cetak Laporan

Nama *Use Case* :	Cetak Laporan
Tujuan :	Untuk mencetak laporan pembelian dan penjualan.
Aktor :	Pemilik
Kondisi Awal :	Masuk ke menu utama
Skenario Utama	
Aksi Aktor	Aksi Sistem
1. Memilih menu "Laporan"	

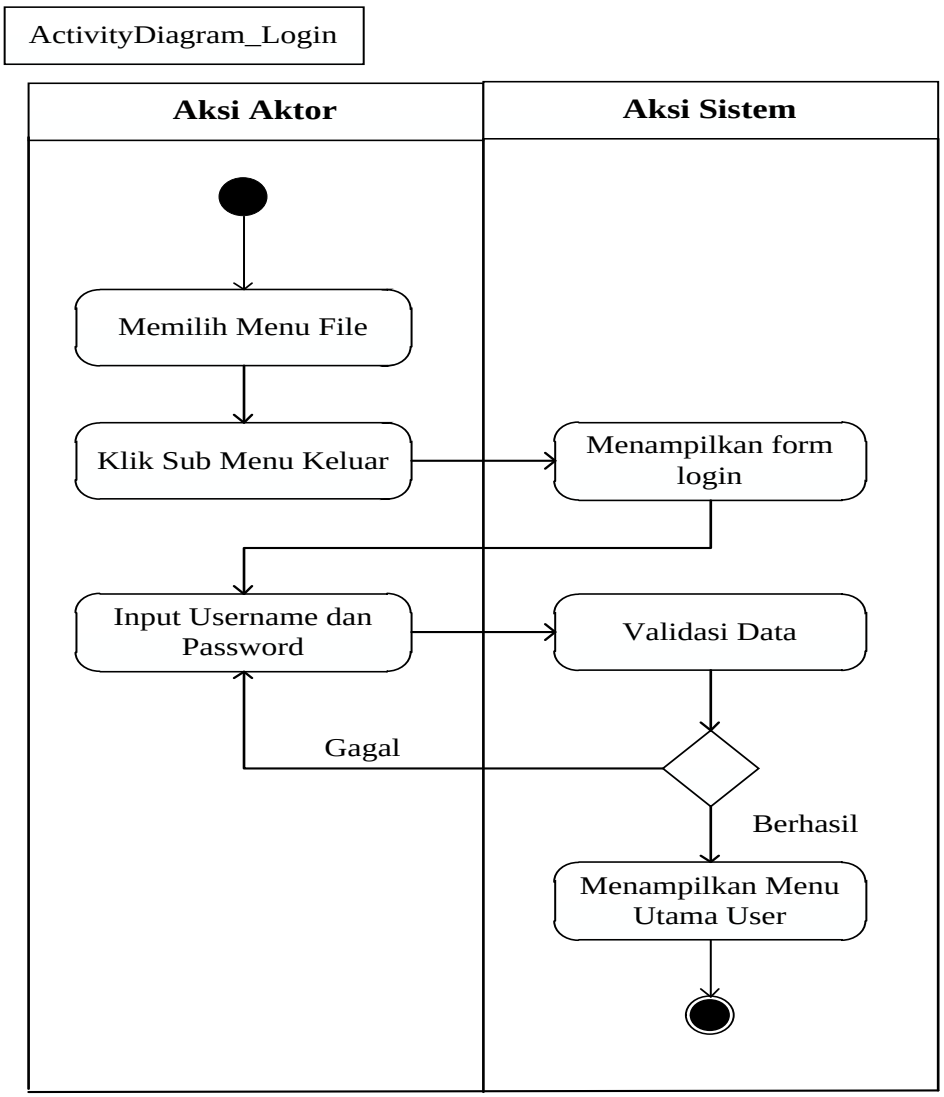
	2. Menampilkan menu-menu yang ada.
3. Pilih menu laporan.	
	4. Menampilkan halaman *Form* laporan.
5. Masukan tanggal cetak	
6. Klik tombol “Cetak”	
	7. Menampilkan dokumen laporan yang akan dicetak.
	8. Mencetak laporan
Kondisi Akhir :	Laporan seperti laporan pembelian dan laporan penjualan akan tercetak.

4.1.3.3. Activity Diagram yang Diusulkan

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun *activity diagram* yang diusulkan dari sistem in*form*asi pembelian dan penjualan bahan bangunan pada CV ALBET adalah sebagai berikut:

4.1.3.3.1. *Activity Diagram Login*

Ini merupakan proses *user* pengguna (bagian administrasi, kasir dan pemilik) melakukan login dengan mengisi *username* dan *password* kemudian di verifikasi oleh sistem.



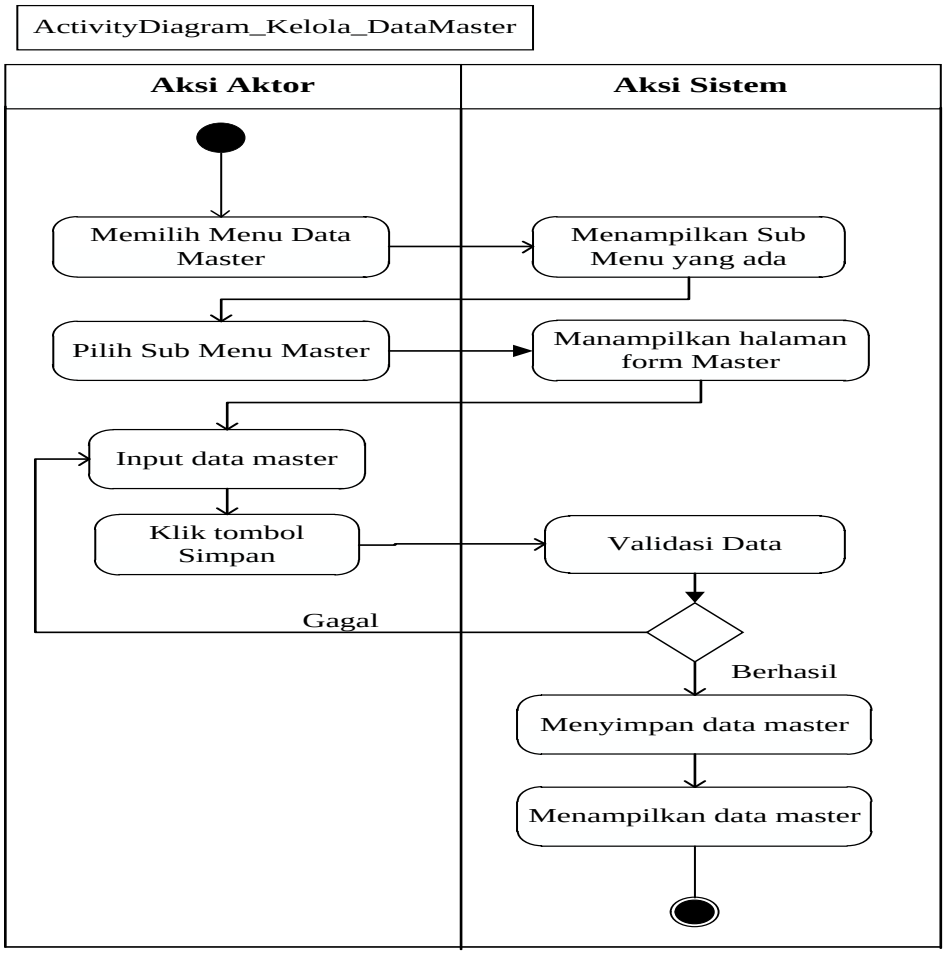
Gambar 4. 2 *Activity Diagram Login* yang diusulkan

Pada gambar 4.2 diatas, memperlihatkan aktivitas login, yang berfungsi untuk masuk kedalam sistem, dimana pada tahapan pertama, aktivitas *login* adalah dengan mengklik menu login, kemudian sistem akan menampilkan halaman login, kemudian aktivitas selanjutnya aktor mengisi (*input*) *username* dan *password*, kemudian mensubmitnya, aktivitas selanjutnya sistem akan mengecek (validasi) apakah *username* dan *password* sudah valid atau belum, jika sudah, maka aktor

akan masuk kedalam menu utama sistem, jika tidak, maka akan ada pesan bahwa *username* atau *password* yang dimasukan salah dan akan mengulangi aktivitas dimulai mengisi (*input*) *username* dan *password*.

4.1.3.3.2. *Activity Diagram* Kelola Data Master

Ini merupakan proses memasukan, mengubah, dan menghapus data master (barang, kategori, satuan, pengguna, pemasok) yang ada didalam sistem in*form*asi pembelian dan penjualan pada CV ALBET yang dilakukan oleh bagian administrasi :

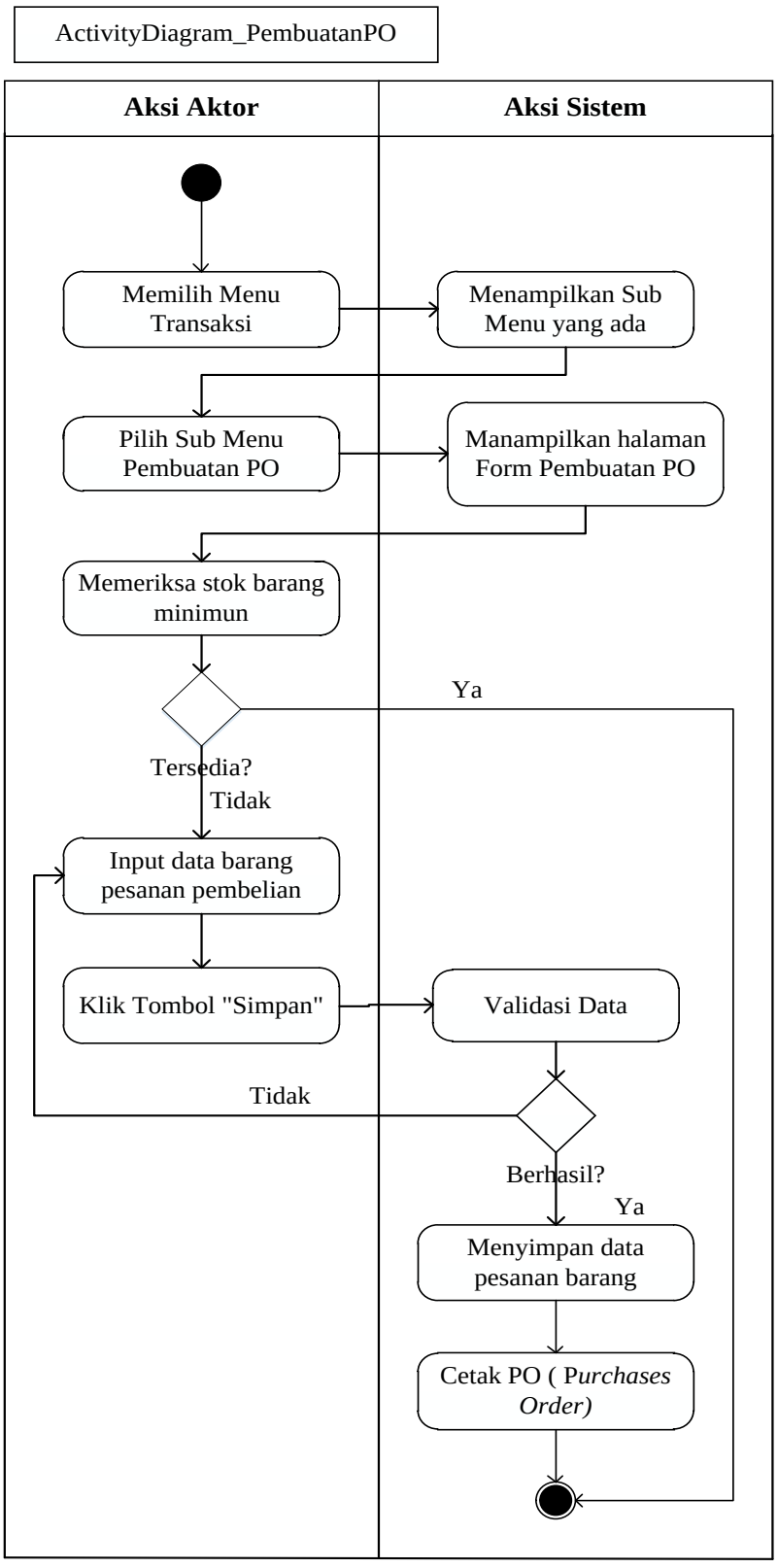


Gambar 4. 3 *Activity Diagram* Kelola Data Master yang diusulkan

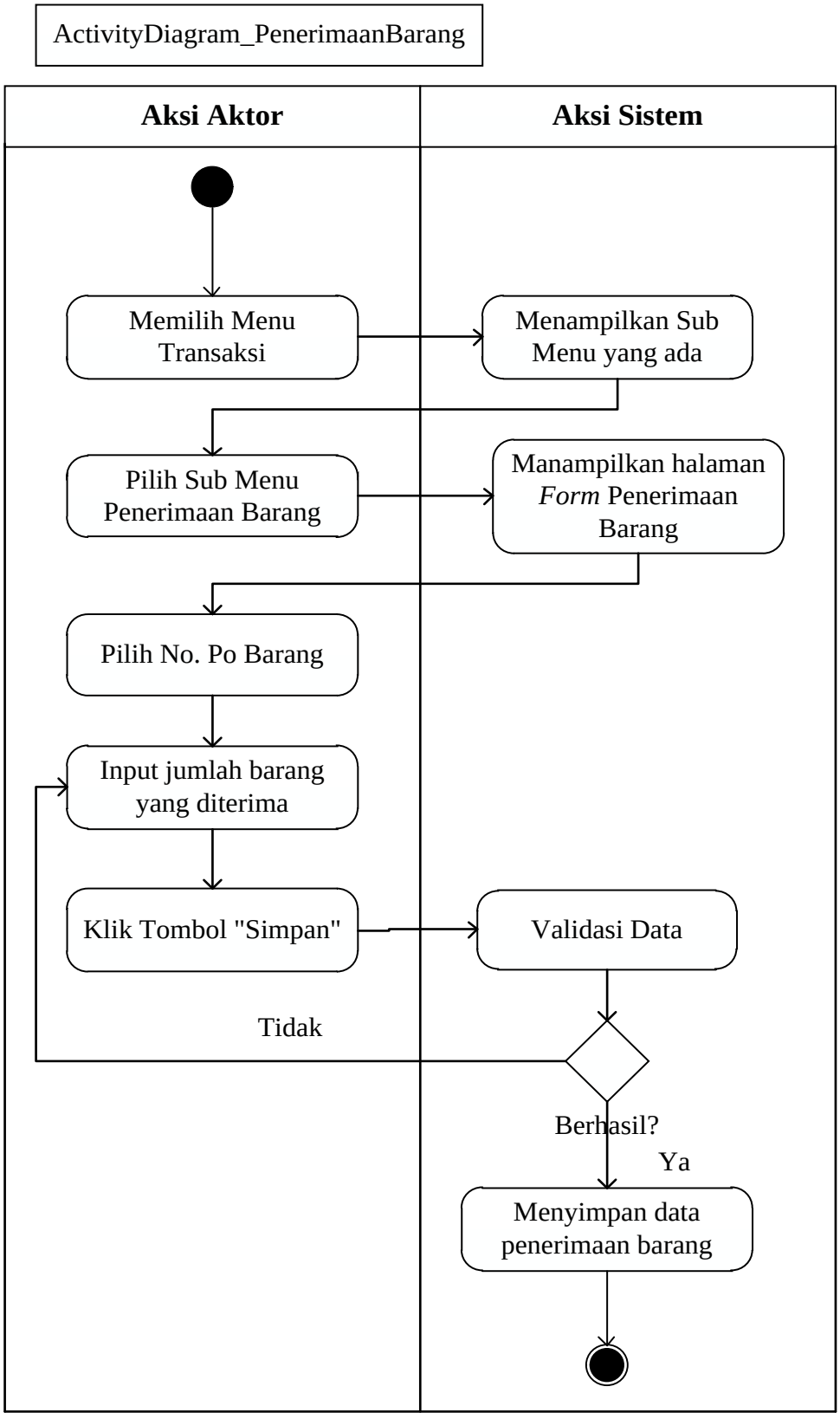
Pada gambar 4.3 diatas, memperlihatkan aktivitas kelola data master, yang berfungsi untuk mengola data master diantaranya: data pengguna, data pemasok, data kategori barang, satuan barang dan barang. Dimana pada tahapan pertama, aktivitas kelola data master adalah aktor melakukan aktivitas memilih menu "Data Master", selanjutnya sistem menampilkan menu-menu yang ada dan *form* pada menu yang dipilih. Setelah itu sistem akan menampilkan halaman *form* master, dimana aktor mengisi atau *input* data master dan memilih tombol-tombol ekseskusi data seperti tambah, ubah, hapus, batal dan keluar. Selanjutnya sistem akan memvalidasi data master sesuai dengan pemilihan tombol ekseskusi seprtijika kita memilih tombol eksekusi tambah maka sistem akan menampilkan pesan "Data Berhasil Disimpan" jika data tersebut valid. Sistem akan menyimpan data master dan menampilkan data master ketika kita memilih tombol eksekusi tambah dan ubah. Begitu juga jika memilih tombol eksekusi lain makan sistem akan melakukan sesuai dengan tombol yang dipilih. Dan akan ditampilkan dalam bentuk tabel.

4.1.3.3.3. *Activity Diagram* Pembelian

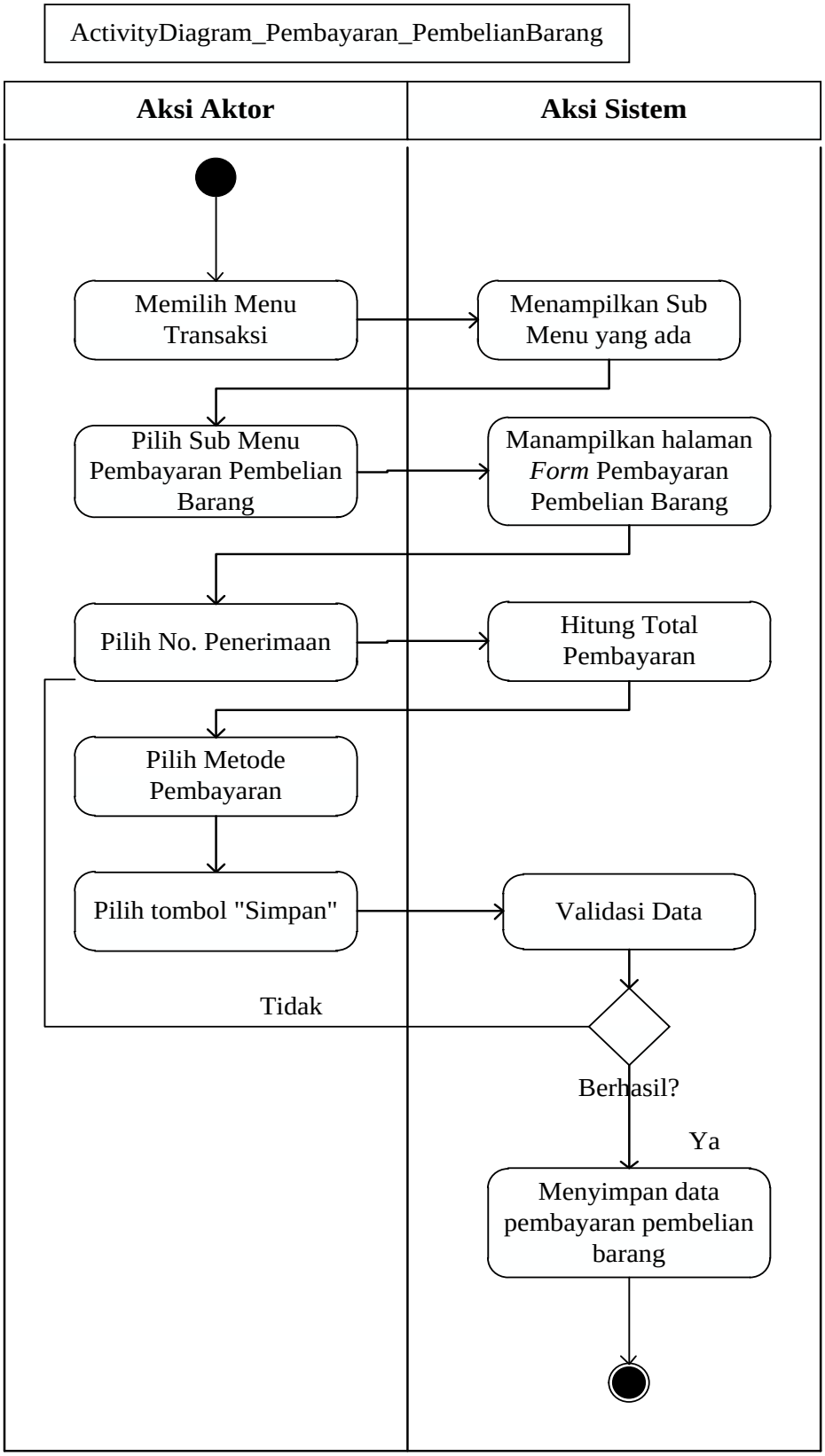
Ini merupakan proses kasir melakukan penginputan data pesanan pembelian (*purchases order*), penerimaan barang, pembayaran pembelian barang bahan bangunan dan retur pembelian kepada pemasok/*supplier.*



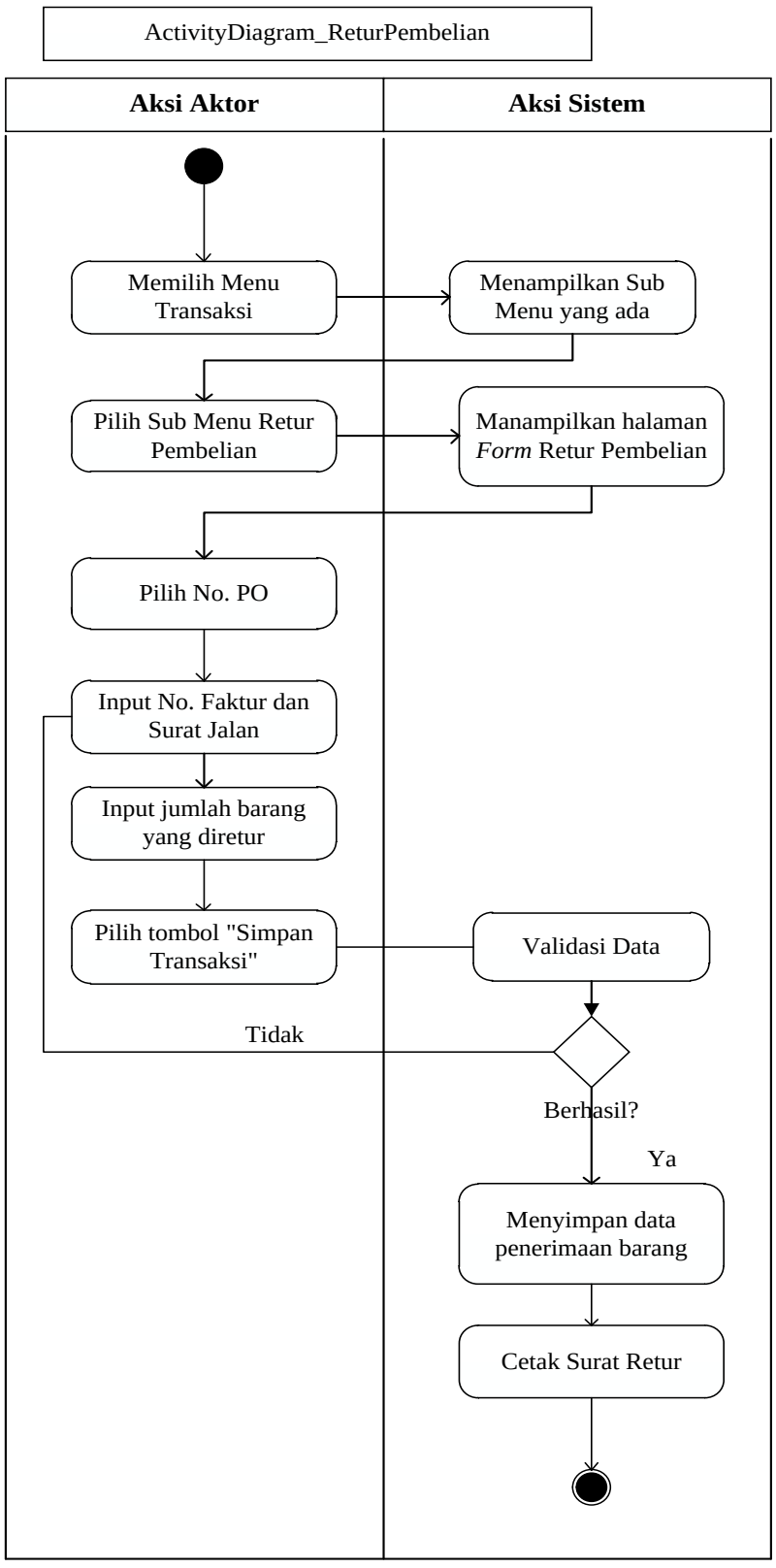
Gambar 4. 4 *Activity Diagram* Pesanan Pembelian yang Diusulkan



Gambar 4. 5 *Activity Diagram* Penerimaan Barang yang Diusulkan



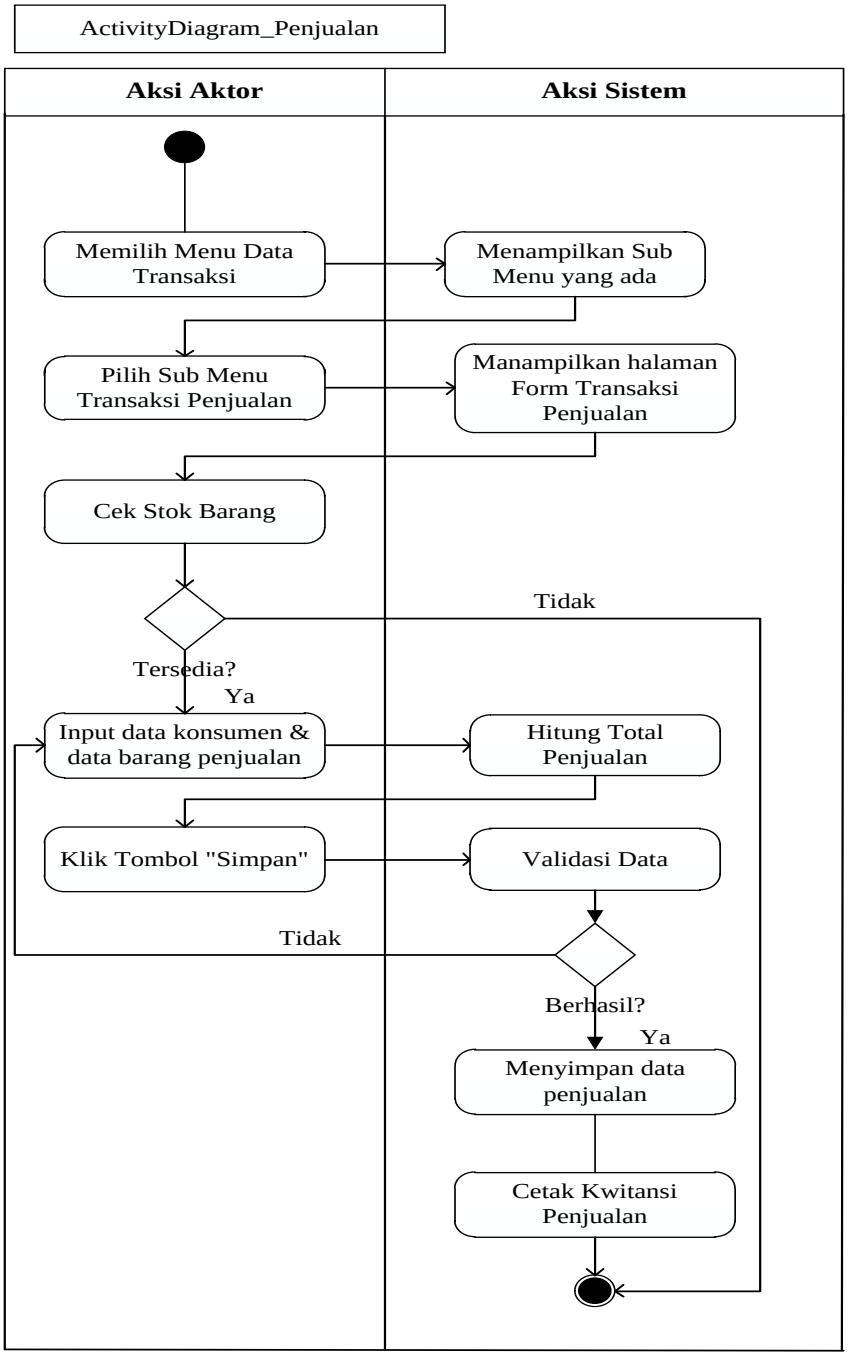
Gambar 4. 6 *Activity Diagram* Pembayaran Pembelian Barang yang Diusulkan



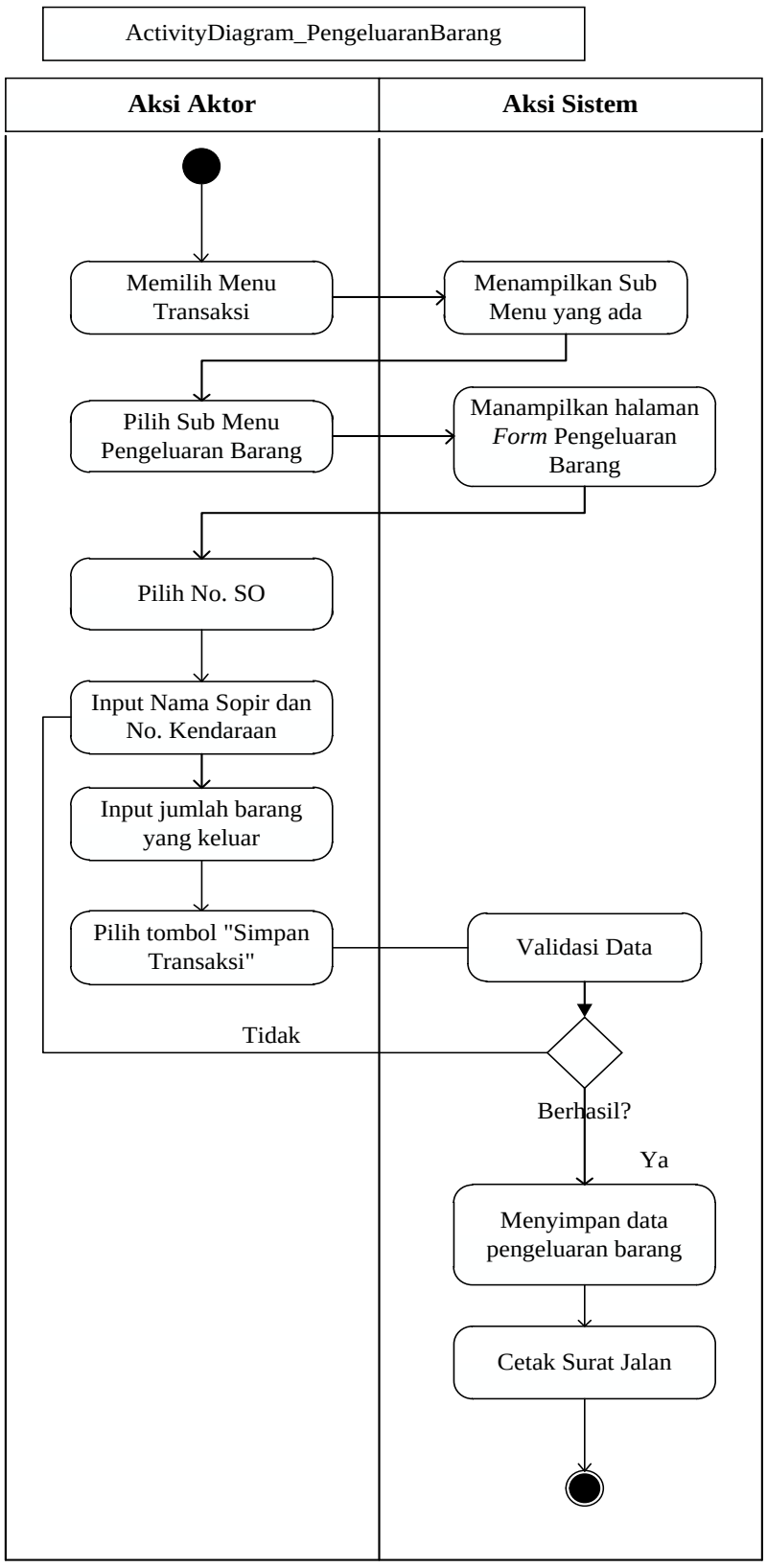
Gambar 4. 7 *Activity Diagram* Retur Pembelian yang Diusulkan

4.1.3.3.1. *Activity Diagram* Penjualan

Ini merupakan proses dimana kasir melakukan transaksi penjualan bahan bangunan pada CV ALBET, proses penjualan ini terdiri penjualan dan pengeluaran barang.



Gambar 4. 8 *Activity Diagram* Penjualan yang diusulkan



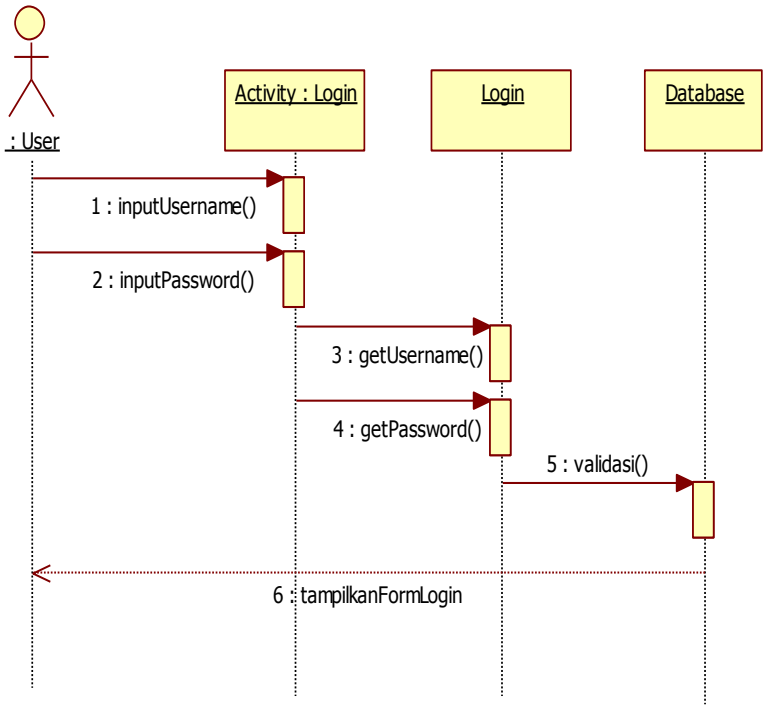
Gambar 4. 9 *Activity Diagram* Pengeluaran Barang yang diusulkan

4.1.3.4 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam sistem terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Adapun *sequence diagram* yang yang diusulkan pada sistem in*form*asi pembelian dan penjualan bahan bangunan pada CV ALBET adalah sebagai berikut:

1. *Sequence Diagram Login*

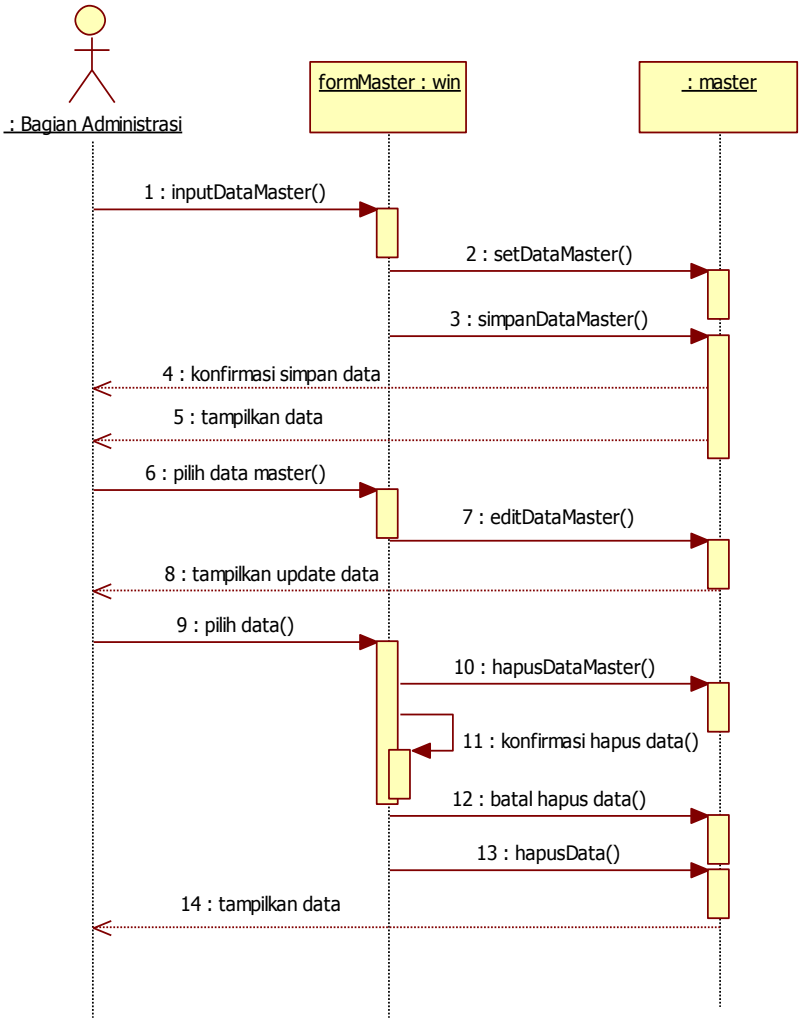
Sequence Diagram ini menjeaskan mengenai proses *login* sistem.



Gambar 4. 10 *Sequence Diagram* Proses *Login*

2. *Sequence Diagram* Kelola Data *Master*

Berikut ini *Sequence Diagram* untuk proses kelola data *master* pada sistem

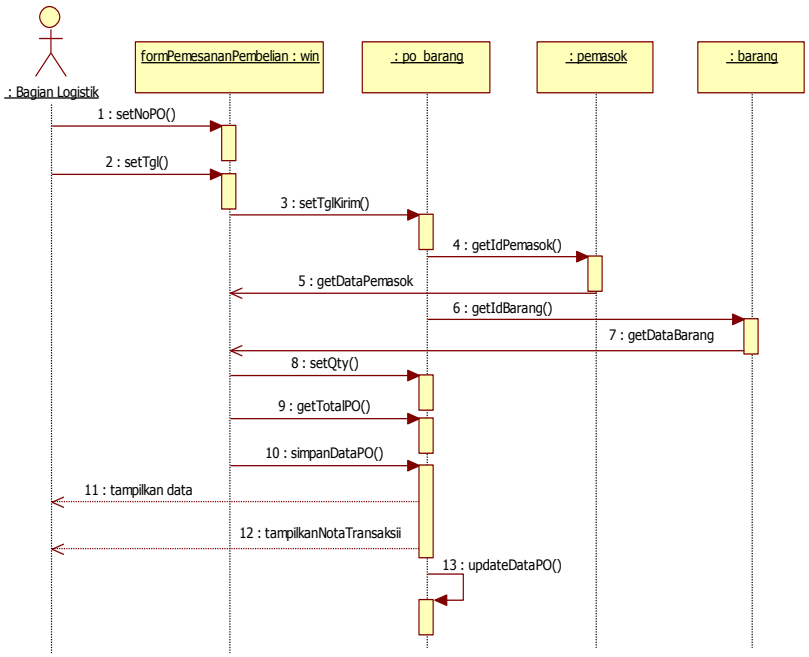


Gambar 4. 11 *Sequence Diagram* Kelola Data *Master*

3. *Sequence Diagram* Pembelian Barang

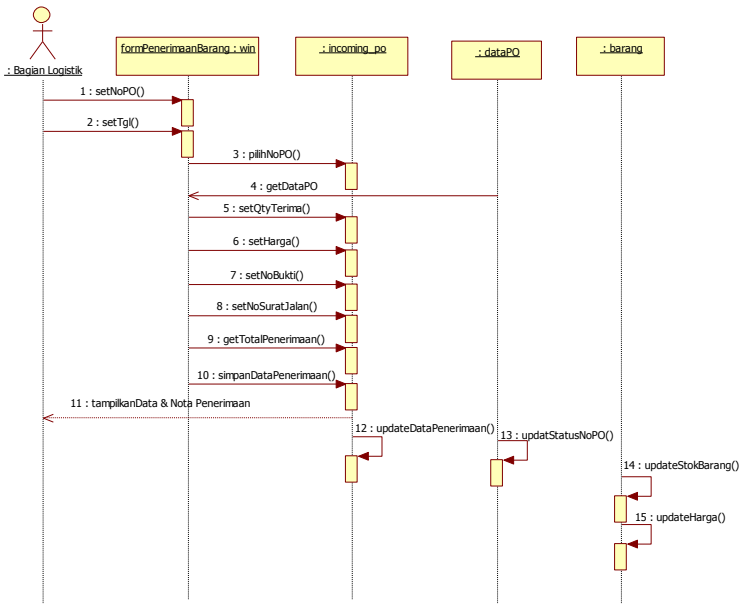
Berikut ini *Sequence Diagram* mengenai proses pembelian barang, proses ini terdiri dari 3 proses yaitu: Pemesanan Pembelian, Penerimaan Barang dan Pembayaran Pembelian Barang.

A. *Sequence Diagram* Pemesanan Pembelian



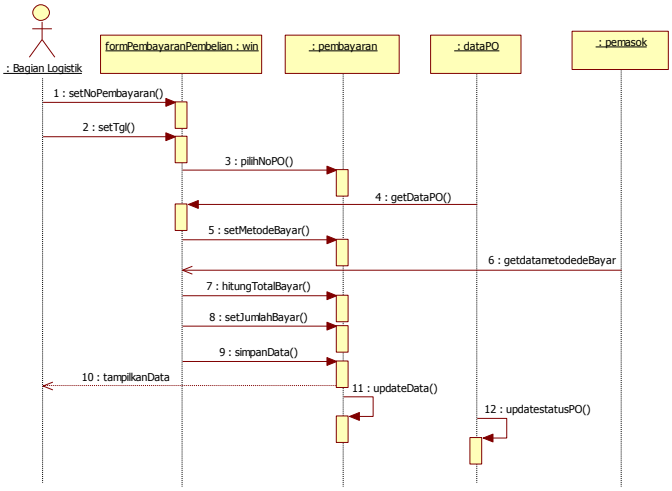
Gambar 4. 12 *Sequence Diagram* Pemesanan Pembelian

B. *Sequence Diagram* Penerimaan Barang



Gambar 4. 13 *Sequence Diagram* Penerimaan Barang

C. *Sequence Diagram* Pembayaran Pembelian Barang

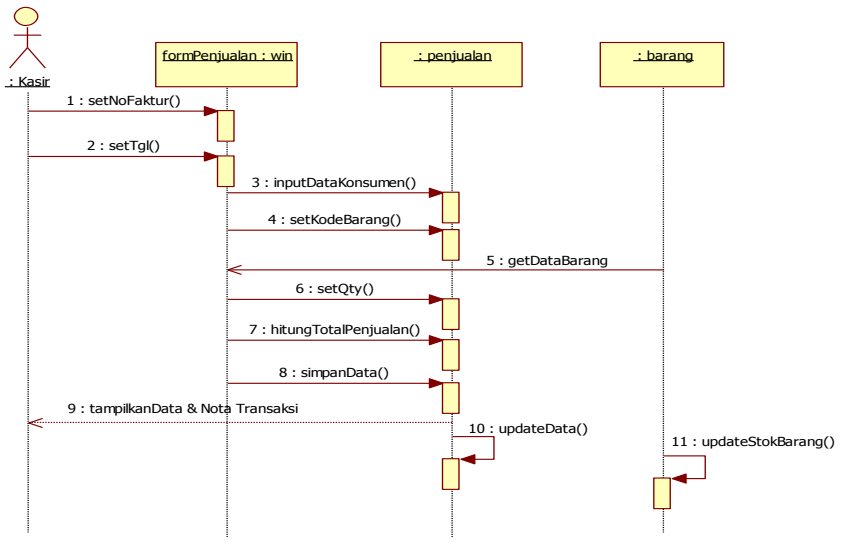


Gambar 4. 14 *Sequence Diagram* Pembayaran Pembelian Barang

4. *Sequence Diagram* Penjualan Barang

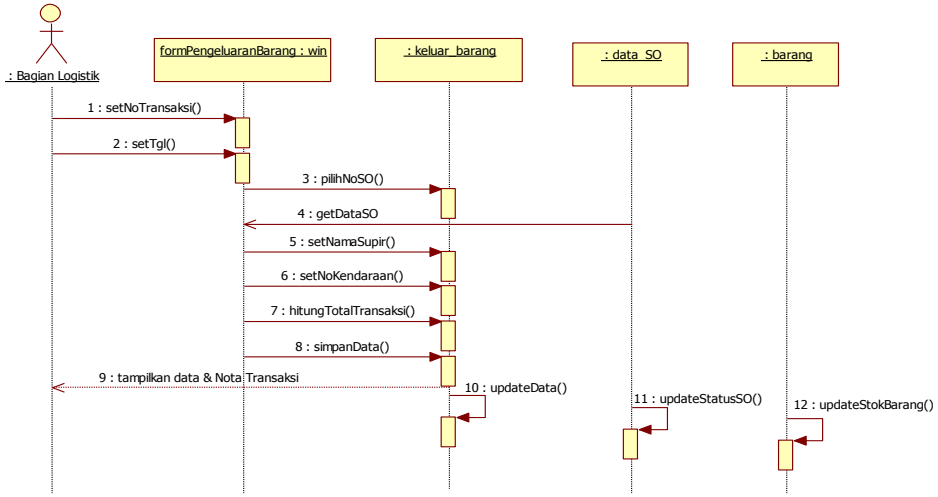
Berikut ini *Sequence Diagram* mengenai proses penjualan barang, proses ini terdiri dari 4 proses yaitu: Penjualan dan Pengeluaran Barang

A. *Sequence Diagram* Penjualan



Gambar 4. 15 *Sequence Diagram* Penjualan Langsung

B. *Sequence Diagram* Pengeluaran Barang

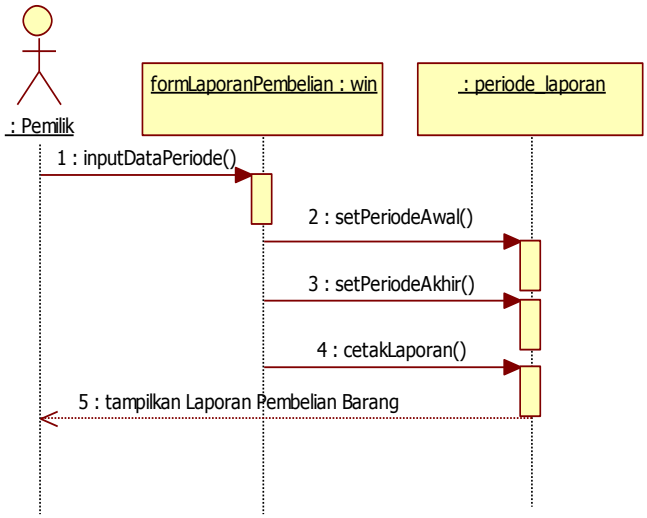


Gambar 4. 16 *Sequence Diagram* Pengeluaran Barang

5. *Sequence Diagram* Laporan

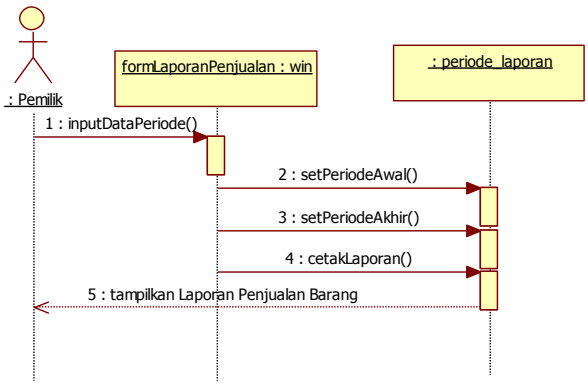
Berikut ini *Sequence Diagram* Laporan yang terdiri dari 2 jenis laporan yaitu: Laporan Transaksi Pembelian Barang dan Laporan Transaksi Penjualan Barang.

A. *Sequence Diagram* Laporan Pembelian Barang



Gambar 4. 17 Sequence Diagram Laporan Pembelian Barang

B. *Sequence Diagram* Laporan Penjualan Barang



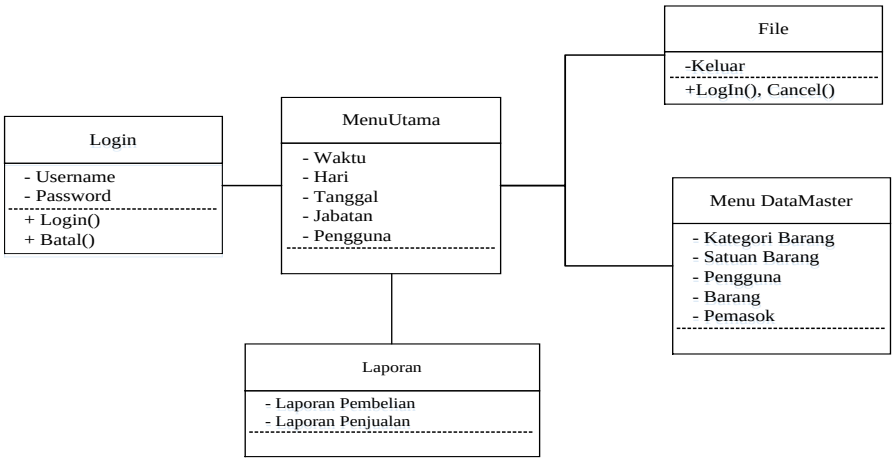
Gambar 4. 18 Sequence Diagram Laporan Penjualan Barang

4.1.3.5 *Class Diagram*

Diagram kelas atau *class diagram* adalah inti dari proses pemodelan objek. Pada diagram kelas disertai fitur-fitur yang menyertainya yaitu atribut (*attribute*) dan operasi (*operation*) yang menggambarkan perilaku suatu kelas.

4.1.3.5.1. *Class Diagram* Login, Menu Utama, *File*, Menu Laporan dan Menu Data Master

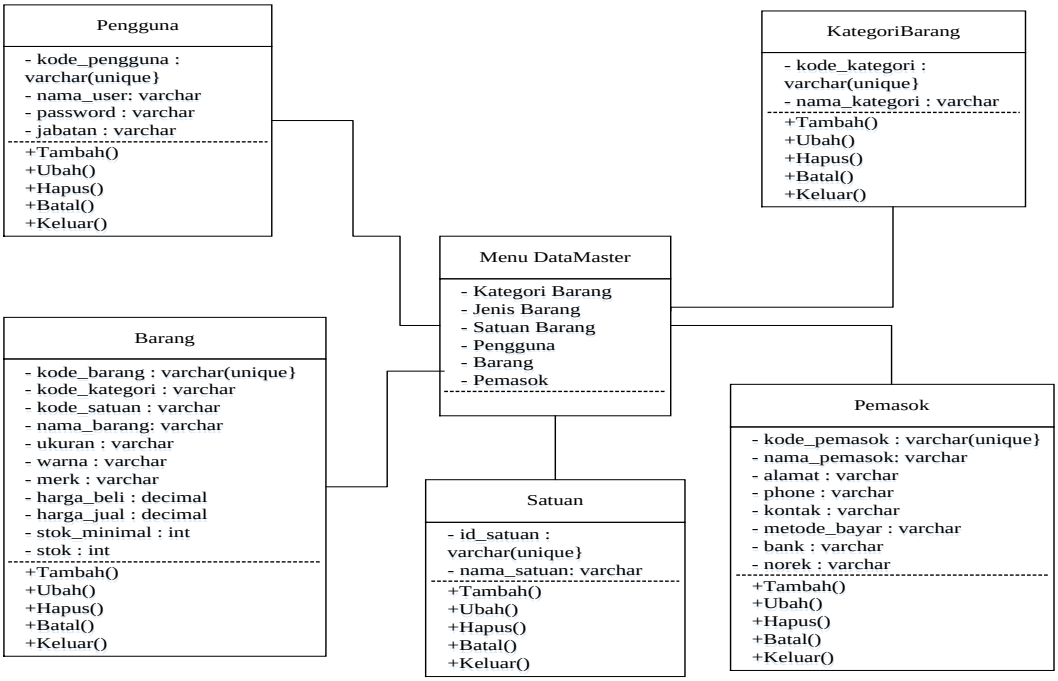
Pada *class diagram* dibawah ini menggambarkan hubungan antara proses login dan menu yang terdapat pada menu utama seperti menu file, menu data master dan menu Laporan serta atribut dan operasi apa saja yang dapat dilakukan pada kelas tersebut.



Gambar 4. 19 *Class Diagram* Login, Menu Utama, File dan Data Master

4.1.3.5.2. *Class Diagram* Data Master

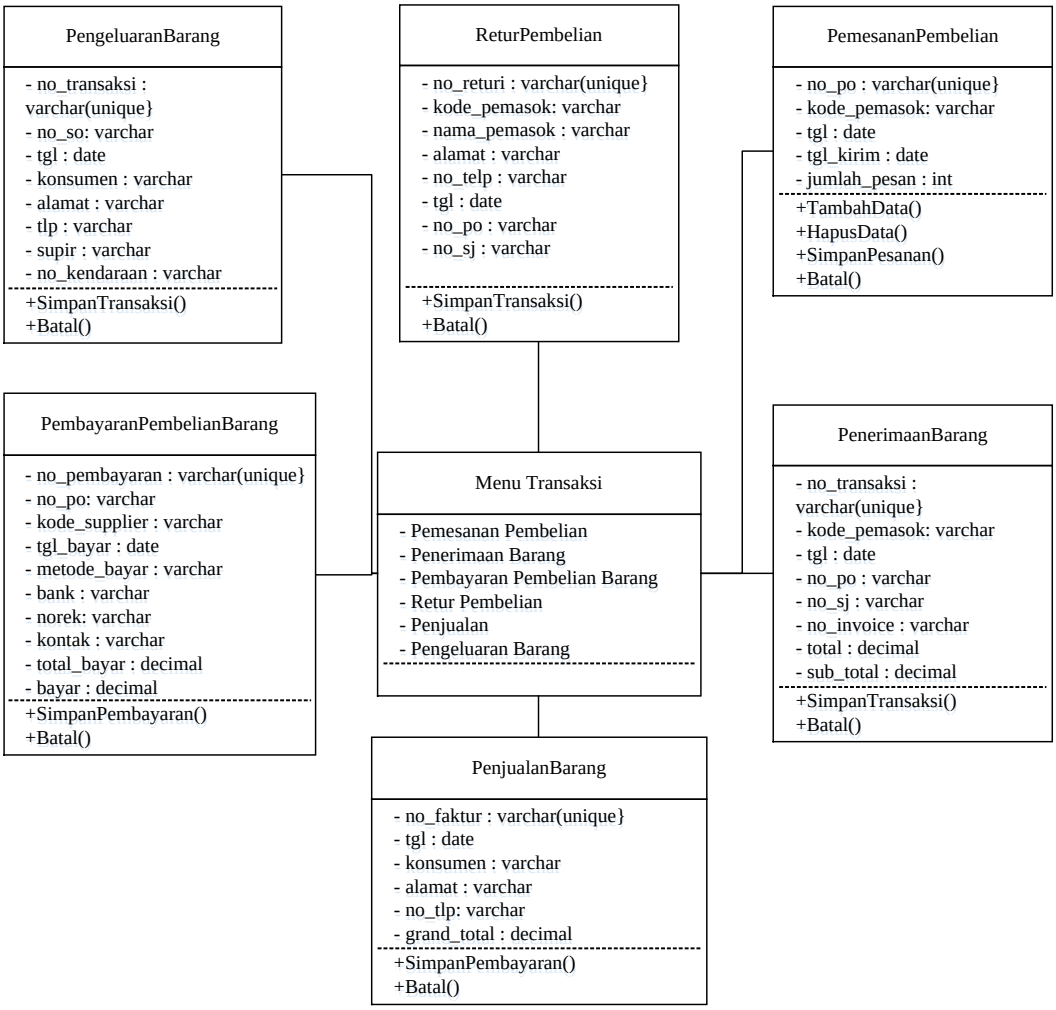
Pada *class diagram* dibawah ini menggambarkan kelas data master serta atribut dan operasi apa saja yang dapat dilakukan pada kelas tersebut.



Gambar 4. 20 *Class Diagram* Data Master yang diusulkan

4.1.3.5.3. *Class Diagram* Transaksi Pembelian

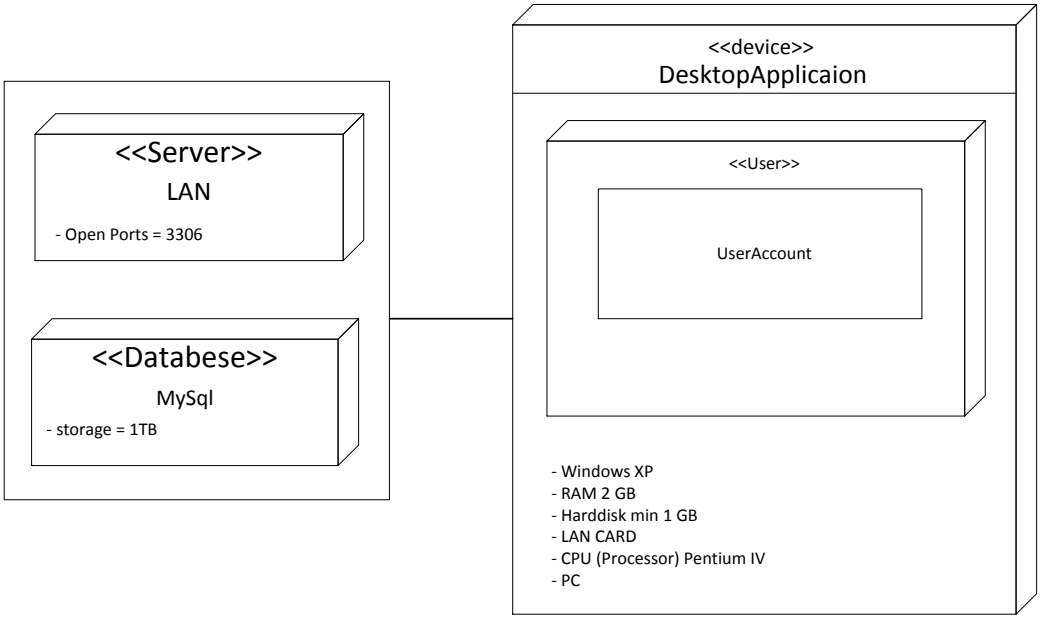
Pada *class diagram* dibawah ini menggambarkan kelas menu transaksi yang terdiri dari transaksi pembelian dan penjualan. Transaksi pembelian terdiri dari pemesanan, penerimaan dan pembayaran pembelian beserta detailnya serta atribut dan operasi apa saja yang dapat dilakukan pada kelas tersebut. Transaksi penjualan beserta detailnya serta atribut dan operasi apa saja yang dapat dilakukan pada kelas tersebut.



Gambar 4. 21 *Class Diagram* Transaksi yang diusulkan

4.1.3.6 *Deployment Diagram*

Deployment diagram memodelkan arsitektur system dengan mengidentifikasikan perangkat kerasnya. Perangkat keras itu bias berupa computer, piranti, network atau bahkan operator yang melakukan proses manual. *Deployment diagram* pada aplikasi ditunjukan pada gambar di bawah ini :



Gambar 4. 22 *Deployment Diagram* yang diusulkan

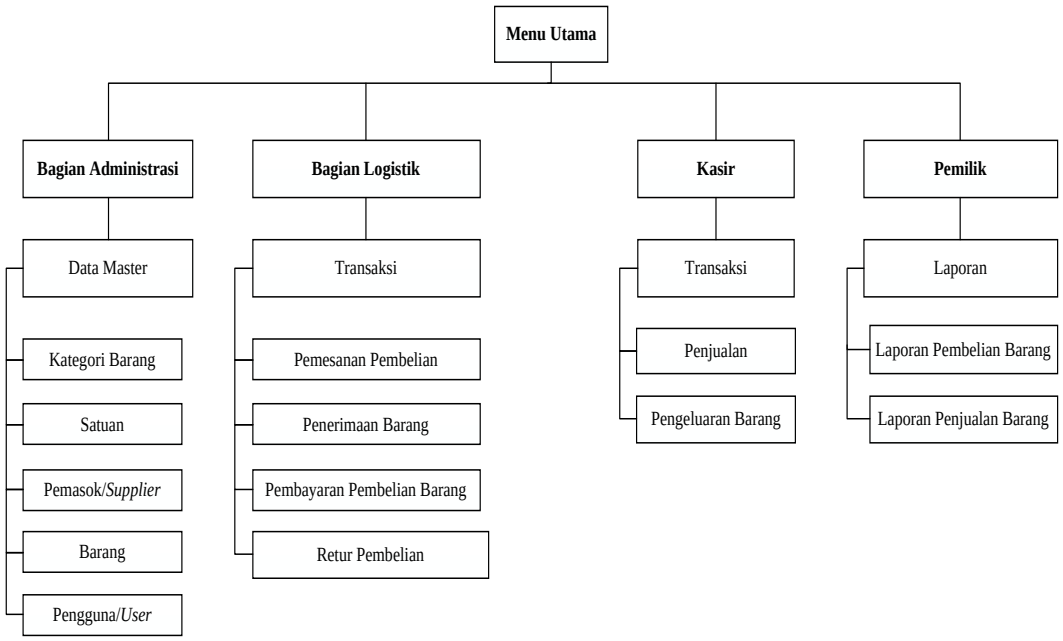
4.2 Perancangan Antar Muka

Agar sistem berinteraksi dengan para pengguna secara baik, maka perlu dirancang sebuah *interface* yang dapat memudahkan pengguna untuk mengoprasikannya. Sistem in*form*asi yang baik bukan hanya dinilai dari segi tampilanya semata, namun akan dinilai juga bagaimana pola aliran in*form*asi yang dibangun dalam bentuk sistem tersebut. Untuk mendukung proses pembentukan

tersebut, Secara umum perancangan antar muka suatu program meliputi : Struktur Menu, Perancangan *Input*, Perancangan *Output*.

4.2.1 Struktur Menu

Menu banyak digunakan dalam dialog karena merupakan jalur pemakai (*user interface*) yang mudah dipahami dan mudah digunakan. Menu berisi dengan beberapa *alternative* atau pilihan yang disajikan kepada *user*. *User* dapat memilih pilihan dimenu dengan cara menekan tombol angka atau huruf yang dihubungkan dengan pilihan tersebut. Jika pilihan dari menu terlalu banyak, maka dapat diorganisasikan secara berjenjang. Struktur menu dibawah ini menggambarkan hierarki dari Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan bahan – bahan bangunan pada CV ALBET. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini :



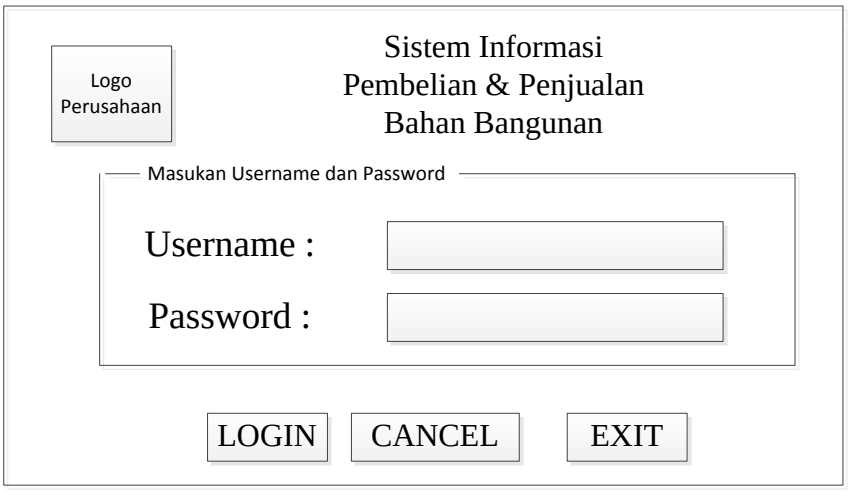
Gambar 4. 23 Struktur Menu

4.2.2 Perancangan Input

Perancangan *input* merupakan desain yang dirancang untuk menerima masukan dari pengguna sistem. Rancangan masukan ini harus dapat memberikan penjelasan bagi pemakainya, baik dari bentuk maupun dari masukan-masukan yang akan diisi.

1. *Design Form Login*

Sebelum masuk ke dalam menu, pengguna harus mengisi *Login* terlebih dahulu, yang rancangannya adalah sebagai berikut:



Gambar 4. 24 Rancangan *Login*

Form login ini diguanakan untuk otorisasi terhadap pengguna (*user). Form login* ini akan meminta *username* dan *password* untuk divalidasi sistem sebelum masuk ke menu utama, Apabila *username* atau *password* tidak sesuai maka sistem memberika pesan bahwa *username* dan *password* tidak valid dan akan meminta isi ulang. Selanjutnya apabila *username* dan p*assword* benar maka *user* akan dapat masuk ke *Form* Menu Utama.

2. *Design Form* Kategori Barang

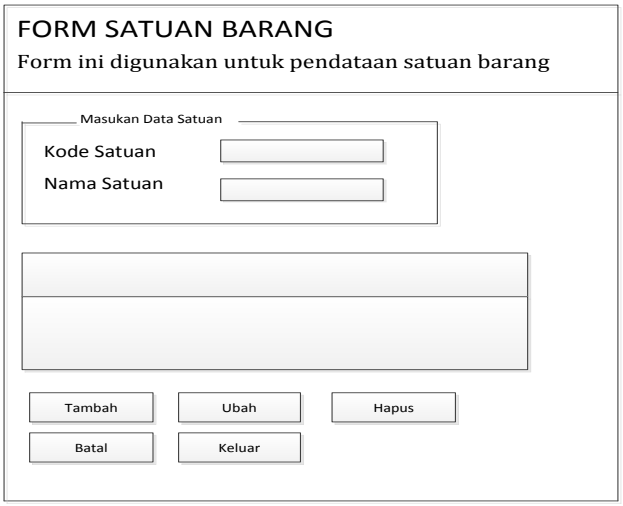
Untuk menginput data kategori bahan bangunan dan dapat mengedit kategori bahan bangunan dilakukan pada *form* kategori barang yang terdapat pada sub menu master, gambarnya dapat dilihat pada gambar 4.25.



Gambar 4. 25 Rancangan *Form* Kategori Barang

3. *Design Form* Satuan Barang

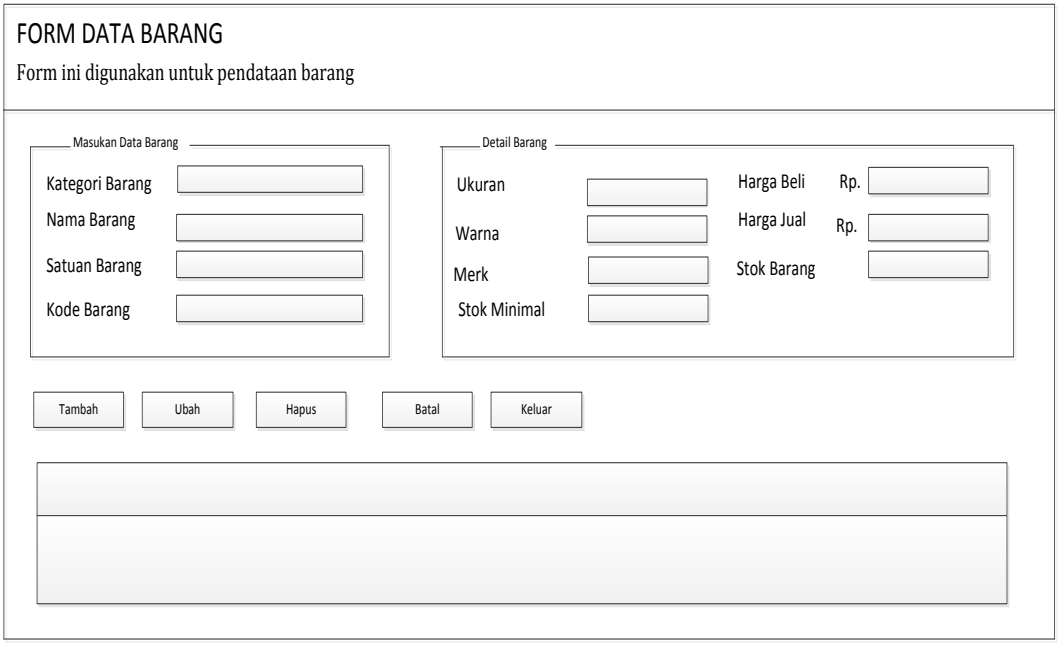
Untuk menginput data satuan bahan bangunan dan dapat mengedit satuan bahan bangunan dilakukan pada *form* satuan barang yang terdapat pada sub menu master, gambarnya dapat dilihat pada gambar 4.26.



Gambar 4. 26 Rancangan *Form* Satuan Barang

4. *Design Form* Barang

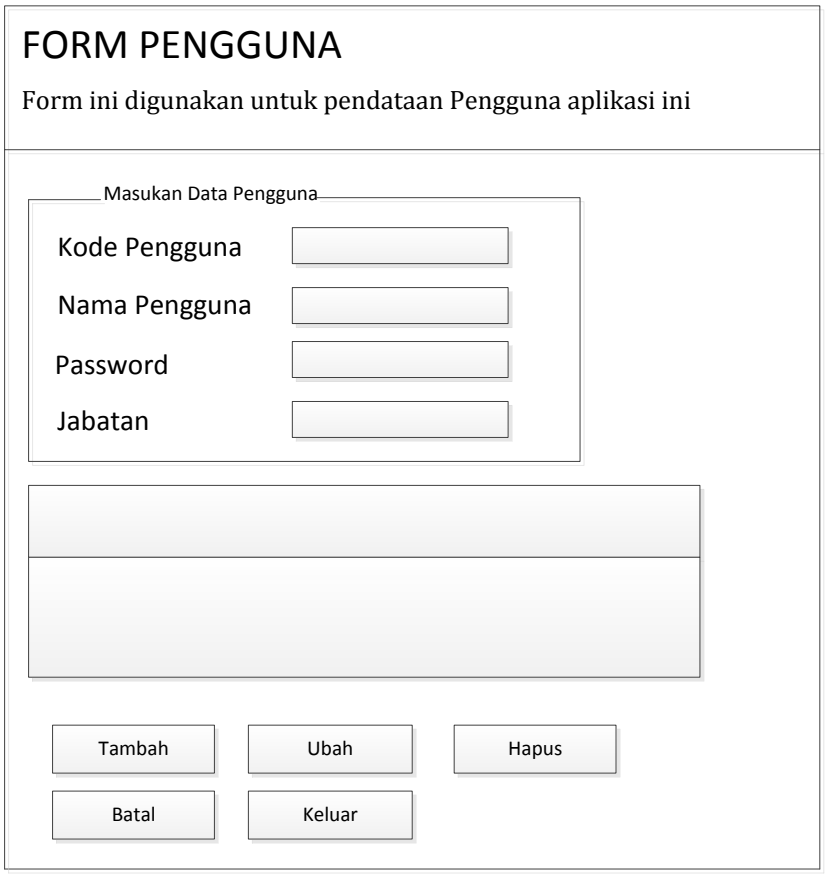
Untuk menginput data barang atau bahan bangunan dan dapat mengedit bahan bangunan dilakukan pada *form* barang yang terdapat pada sub menu master, gambarnya dapat dilihat pada gambar 4.27.



Gambar 4. 27 Rancangan *Form* Barang

5. *Design Form* Pengguna

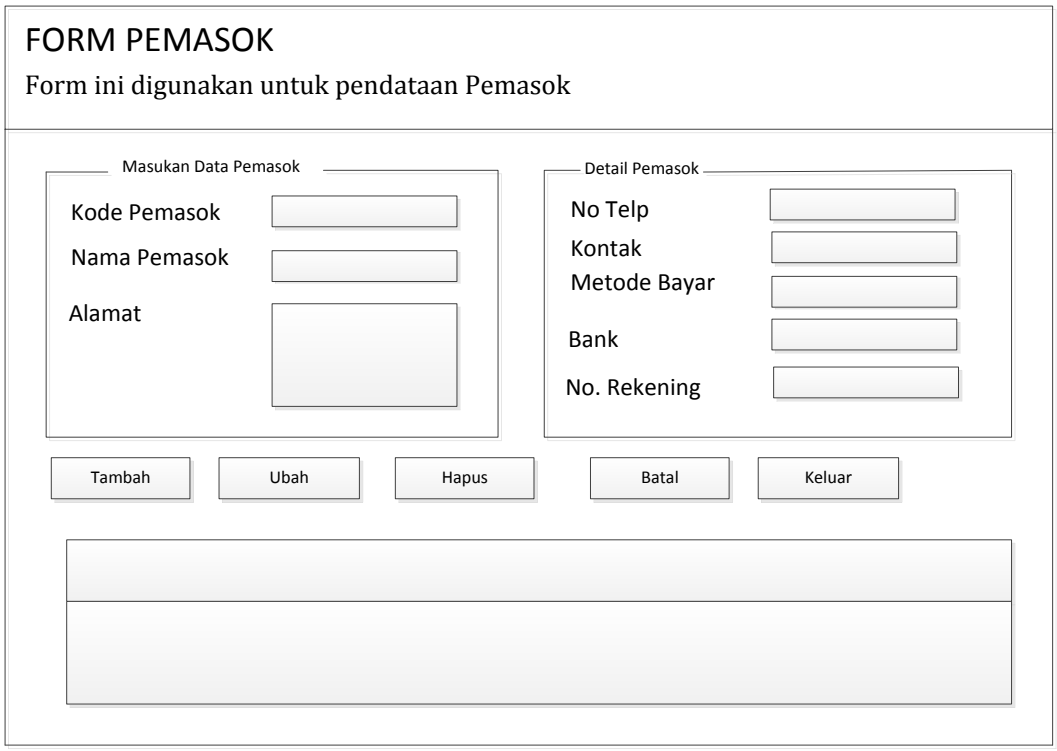
Untuk menginput data pengguna dan dapat mengedit pengguna dilakukan pada *form* pengguna yang terdapat pada sub menu master, gambarnya dapat dilihat pada gambar 4.28.



Gambar 4. 28 Rancangan *Form* Pengguna

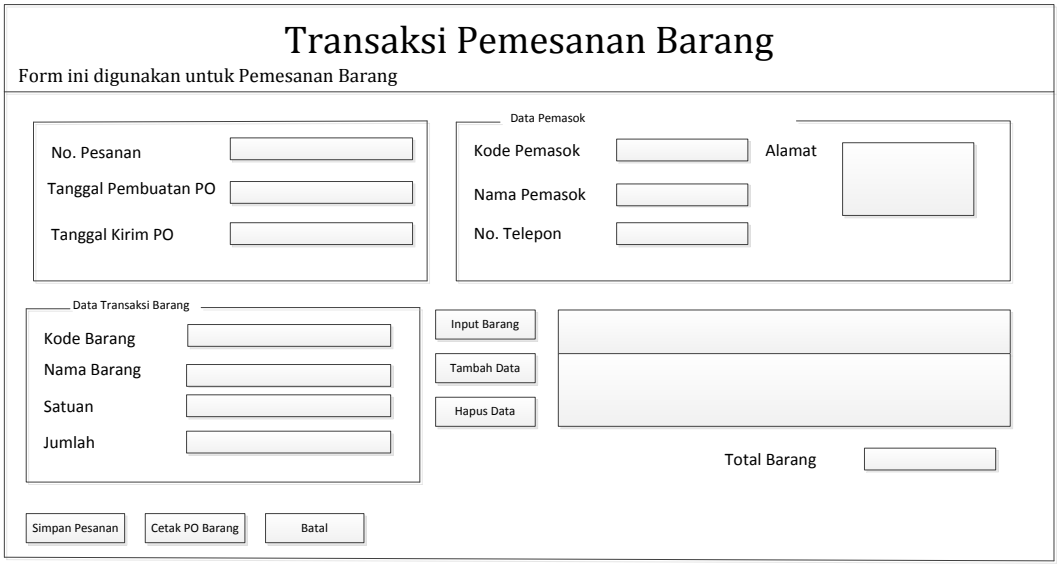
6. *Design Form* Pemasok

Untuk menginput data pemasok dan dapat mengedit pemasok dilakukan pada *form* pemasok yang terdapat pada sub menu master, gambarnya dapat dilihat pada gambar 4.29.



Gambar 4. 29 Rancangan *Form* Pemasok

7. *Design Form* Pemesanan Barang (Pembuatan PO)



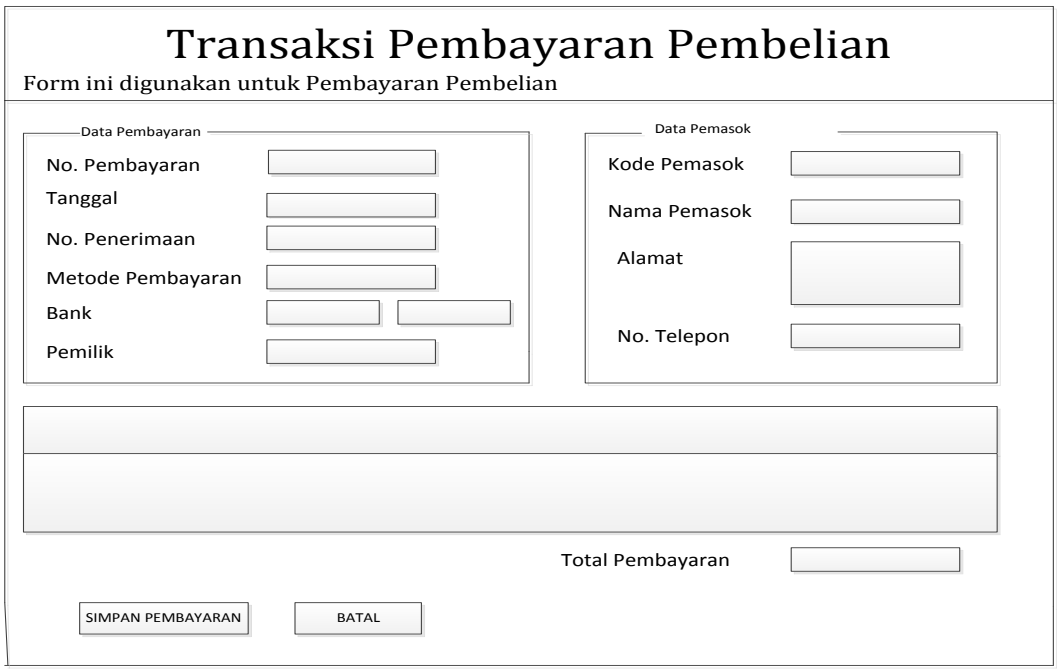
Gambar 4. 30 Rancangan *Form* Pemesanan Barang

8. *Design Form* Penerimaan Barang



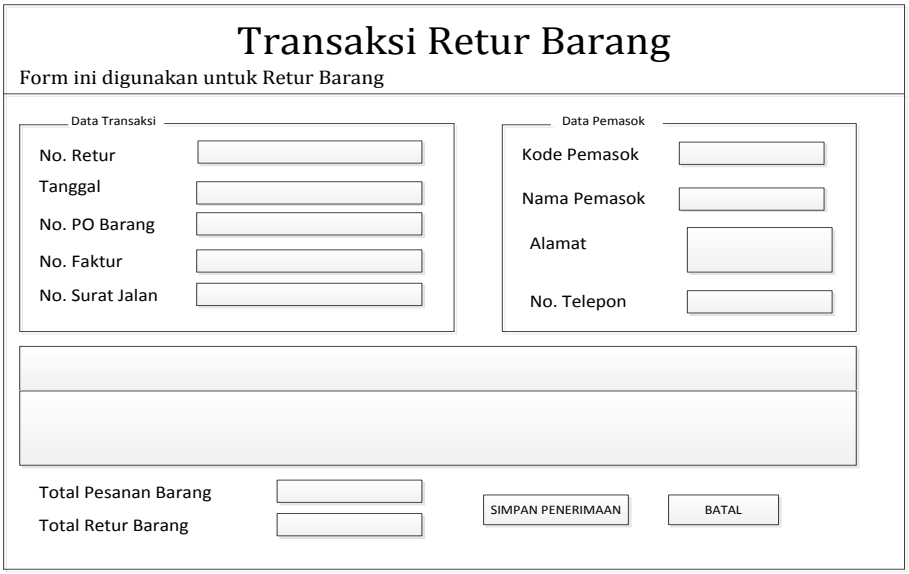
Gambar 4. 31 Rancangan *Form* Penerimaan Barang

9. *Design Form* Pembayaran Pembelian



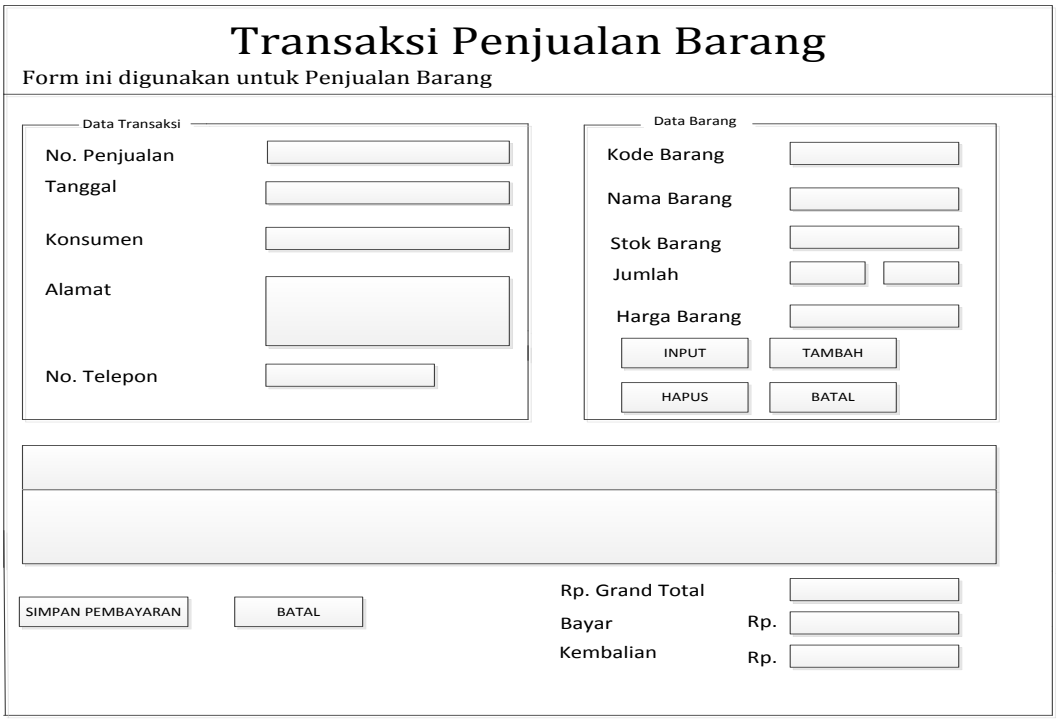
Gambar 4. 32 Rancangan *Form* Pembayaran Pembelian

10. *Design Form* Retur Pembelian



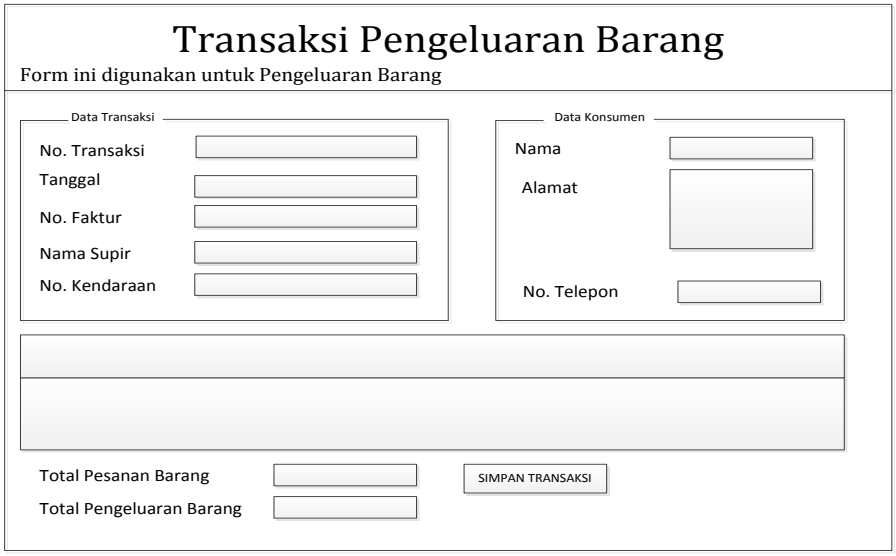
Gambar 4. 33 Rancangan *Form* Pembayaran Pembelian

11. *Design Form* Penjualan



Gambar 4. 34 Rancangan *Form* Penjualan

12. *Design Form* Pengeluaran Barang

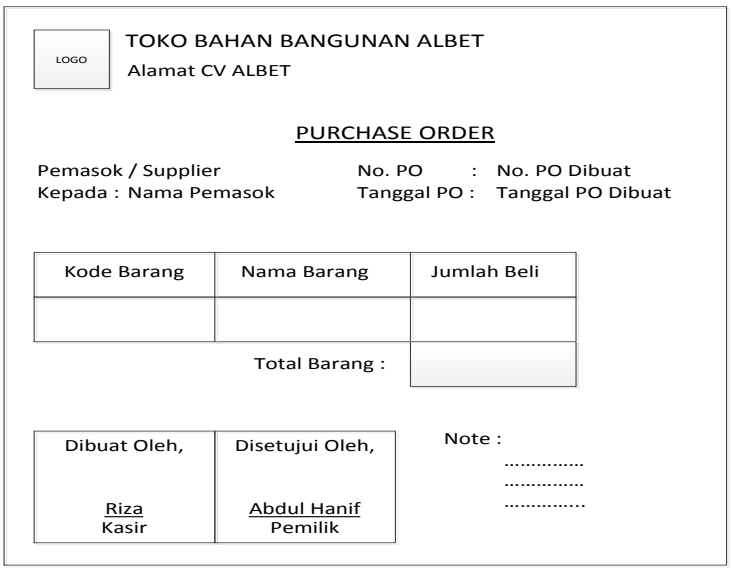


Gambar 4. 35 Rancangan *Form* Pegeluaran Barang

4.2.3. Perancangan Output

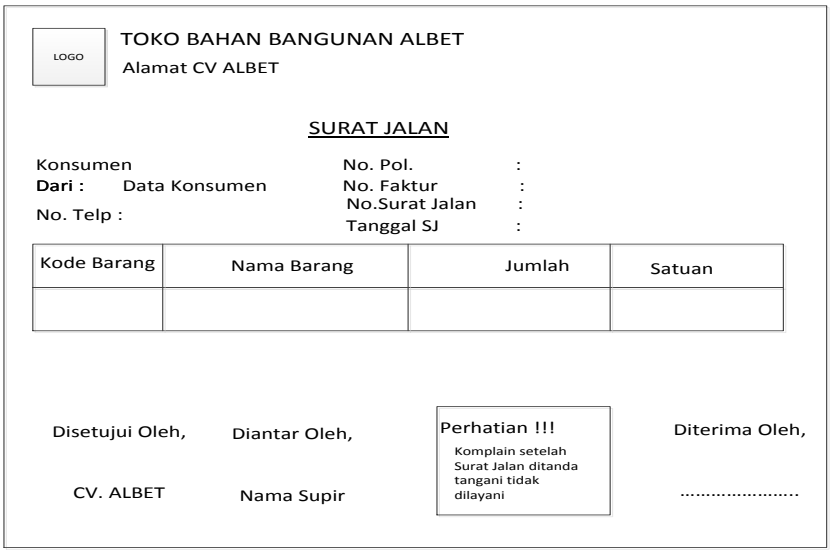
Perancangan *output* merupakan desain yang dirancang untuk melihat hasil (*output*) keluaran yang didapat pengguna.

1. *Design Purchase Order* (PO)



Gambar 4. 36 Rancangan *Purchase Order* (PO)

2. *Design* Surat Retur


LOGO

TOKO BAHAN BANGUNAN ALBET

Alamat CV ALBET

SURAT JALAN

Konsumen
Dari : Data Konsumen
No. Telp :

No. Pol. :
No. Faktur :
No.Surat Jalan :
Tanggal SJ :

Kode Barang	Nama Barang	Jumlah	Satuan

Disetujui Oleh,

CV. ALBET

Diantar Oleh,

Nama Supir

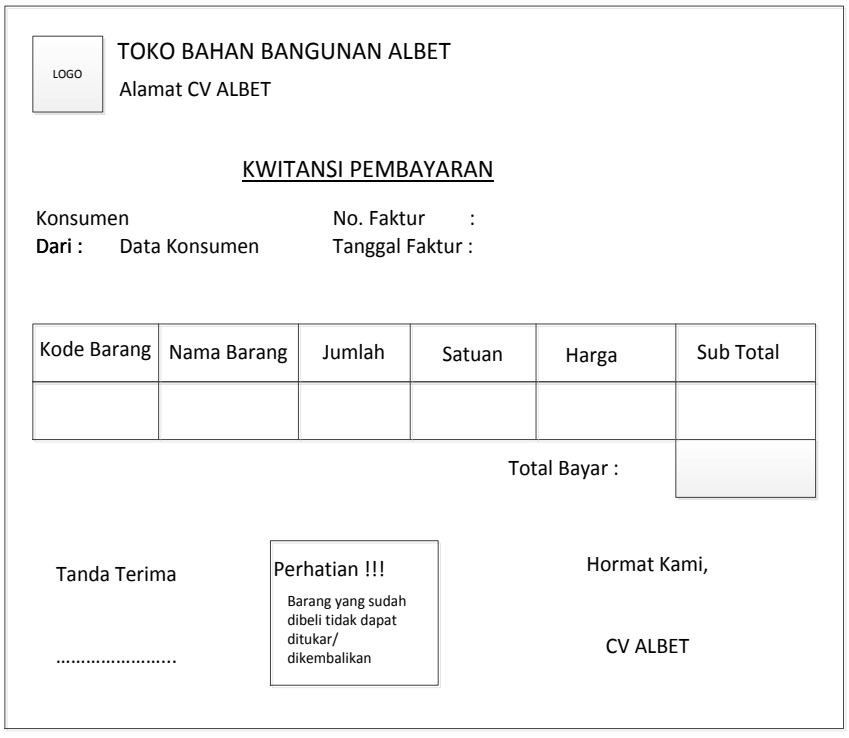
Perhatian !!!
Komplain setelah Surat Jalan ditanda tangani tidak dilayani

Diterima Oleh,

.....

Gambar 4. 37 Rancangan Surat Retur

3. *Design* Kwitansi Pembayaran


LOGO

TOKO BAHAN BANGUNAN ALBET

Alamat CV ALBET

KWITANSI PEMBAYARAN

Konsumen
Dari : Data Konsumen

No. Faktur :
Tanggal Faktur :

Kode Barang	Nama Barang	Jumlah	Satuan	Harga	Sub Total
				Total Bayar :	

Tanda Terima

.....

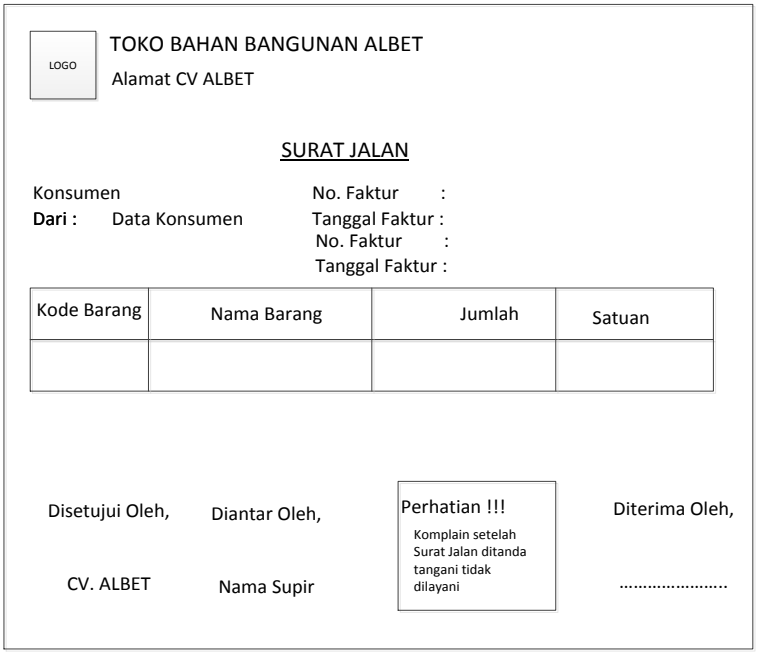
Perhatian !!!
Barang yang sudah dibeli tidak dapat ditukar/ dikembalikan

Hormat Kami,

CV ALBET

Gambar 4. 38 Rancangan Kwitansi Pembayaran

4. *Design* Surat Jalan



Gambar 4. 39 Rancangan Surat Jalan

5. *Design* Laporan Pembelian



Gambar 4. 40 Rancangan Laporan Pembelian

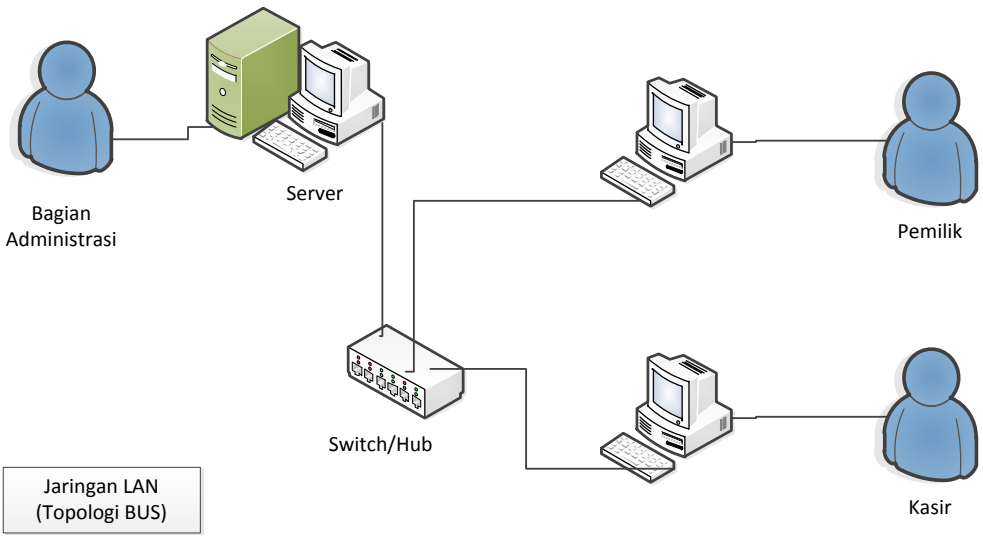
6. Design *Form* Laporan Penjualan



Gambar 4. 41 Rancangan Laporan Penjualan

4.3. Perancangan Arsitektur Jaringan

Arsitektur Jaringan yang akan diterapkan pada CV ALBET ini menggunakan t*opologi Bus*.Karena pada topologi ini terdiri pada satu jalur kabel utama dimana pada masing – masing ujungnya diberikan sebuah *Terminator.* Semua *Nodes* pada jaringan (*file server, workstation,* dan perangkat lainya) terkoneksi sebuah kabel utama (*backbone).* Arsitektur Jaringan yang akan digunakan terdapat pada gambar 4.42



Gambar 4. 42 Rancangan Arsitektur Jaringan

Adapun kelebihan dan kekurangan pada *topologi* ini

1. Kelebihan
   a. Mudah didalam mengkonfigurasikan komputer atau perangkat lain kedalam sebuah kabel utama.
   b. Tidak terlalu banyak menggunakan kabel dibandingkan dengan *topologi star*.
2. Kekurangan
   a. Seluruh jaringan akan mati jika ada kerusakan pada kabel utama (*backbone*).
   b. Membutuhkan terminator pada kedua sisi dari kabel utamanya.
   c. Sangat sulit mengidentifikasikan permasalahan jika jaringan sedang *down*.

4.4. Implementasi

Tahap implementasi sistem merupakan kegiatan yang dilakukan setelah tahap perancangan sistem selesai dilaksanakan. Tujuan yang ingin dicapai pada tahap ini adalah dapat dioperasikannya hasil perancangan sistem yang telah dibuat.

Pada tahap ini mengusulkan agar penerapan dilakukan secara *paralel*, yaitu dengan melakukan pengoperasian sistem yang baru secara bersamaan dengan sistem yang lama selama periode tertentu sampai sistem yang lama tidak dipakai lagi. Hal ini dimaksudkan sebagai *proteksi* kepada perusahaan atau instansi yang bersangkutan apabila terdapat kegagalan sistem yang baru nanti. Kegiatan-kegiatan yang dilakukan pada tahap implementasi ini yaitu sebagai berikut :

4.4.1. Batasan Implementasi

Dalam mengimplementasikan perangkat lunak pemantauan dan pengendalian aplikasi ini ada beberapa hal yang menjadi batasan implementasi yaitu :

1. Basis data yang digunakan dalam mengimplementasikan Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan Bahan- bahan Bangunan pada CV ALBET adalah SQL Server baik untuk basis data utama maupun extraksi basisdata yang disebarkan ke client.
2. Alamat IP dari komputer yang digunakan menggunakan alamat IP statis.

3. Dalam pembuatan aplikasi Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan Bahan – bahan Bangunan pada CV ALBET menggunakan bahasa pemrograman Java.

4.4.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan program aplikasi Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan Bahan – bahan Bangunan pada CV ALBET adalah sebagai berikut :

Dalam mengimplementasian aplikasi ini, perangkat lunak yang digunakan adalah sebagai berikut:

1. Windows 7 Ultimate 32-bit
2. Netbeans IDE 8.0
3. XAMPP version 3.2.1 yang terdiri dari :
    1. PHP Version 5.4.4
    2. MySQL 5.5.25a
    3. Apache 2.4.2

4.4.3. Implementasi Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk *Client* maupun *Server* memiliki spesifikasi sebagai berikut:

1. *Server* :
    1. Core i5-3317U, 1.70 Ghz
    2. RAM 4 GB DDR3.
    3. *Harddisk* 500 GB.

4. *Monitor, keyboard, mouse*

2. *Client* :

    1. Core2duo E7500
    2. RAM 1GB DDR2
    3. *Harddisk* 120 GB
    4. *Monitor, keyboard, mouse*

4.4.4. Implementasi Basis Data (Sintak SQL)

Implementasi basis data dilakukan dengan menggunakan bahasa SQL, dimana aplikasi pemrograman yang digunakan adalah MySQL, implementasi basis datanya dalam bahas SQL adalah sebagai berikut:

1. Tabel Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `kode_barang` varchar(7) NOT NULL,
  `id_kategori` varchar(2) NOT NULL,
  `id_satuan` varchar(2) NOT NULL,
  `nama_barang` varchar(25) NOT NULL,
  `ukuran` varchar(25) DEFAULT NULL,
  `merk` varchar(25) DEFAULT NULL,
  `warna` varchar(15) DEFAULT NULL,
  `harga_beli` decimal(15,0) DEFAULT NULL,
  `harga_jual` varchar(15) DEFAULT NULL,
```

```
  `stok_minimal` int(5) NOT NULL,
  `stok` int(5) DEFAULT NULL,
  PRIMARY KEY (`kode_barang`),
  KEY `barang_fk1` (`id_kategori`),
  KEY `barang_fk2` (`id_satuan`),
  KEY `kode_barang` (`kode_barang`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` varchar(2) NOT NULL,
  `nama_kategori` varchar(22) DEFAULT NULL,
  `keterangan` varchar(31) DEFAULT NULL,
  PRIMARY KEY (`id_kategori`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. Tabel Satuan

```
CREATE TABLE IF NOT EXISTS `satuan` (
  `id_satuan` varchar(2) NOT NULL,
  `nama_satuan` varchar(6) DEFAULT NULL,
  PRIMARY KEY (`id_satuan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. Tabel Pemasok /*Supplier*

```
CREATE TABLE IF NOT EXISTS `pemasok` (
  `kode_pemasok` varchar(7) NOT NULL,
  `nama_pemasok` varchar(28) DEFAULT NULL,
  `alamat` varchar(255) DEFAULT NULL,
  `telepon` varchar(15) DEFAULT NULL,
  `kontak` varchar(28) DEFAULT NULL,
  `metode_bayar` varchar(9) DEFAULT NULL,
  `nama_bank` varchar(15) DEFAULT NULL,
  `no_rekening` varchar(15) DEFAULT NULL,
  KEY `kode_pemasok` (`kode_pemasok`),
  KEY `nama_pemasok` (`nama_pemasok`),
  KEY `pemasok_fk1` (`nama_bank`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Tabel Pengguna (*User*)

```
CREATE TABLE IF NOT EXISTS `pengguna` (
  `kode_pengguna` varchar(5) NOT NULL,
  `nama_pengguna` varchar(11) DEFAULT NULL,
  `password` varchar(15) DEFAULT NULL,
  `jabatan` varchar(12) DEFAULT NULL,
  `status` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`kode_pengguna`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. Tabel PO Barang

```
CREATE TABLE IF NOT EXISTS `po_barang` (
  `no_po` varchar(15) NOT NULL,
  `kode_pemasok` varchar(7) DEFAULT NULL,
  `tanggal_po` date DEFAULT NULL,
  `tanggalkirim_po` date DEFAULT NULL,
  `jumlah_pesan` int(15) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`no_po`),
  KEY `po_fk1` (`kode_pemasok`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. Tabel PO Barang Detail

```
CREATE TABLE IF NOT EXISTS `pobarang_detail` (
  `no_po` varchar(15) DEFAULT NULL,
  `kode_barang` varchar(7) DEFAULT NULL,
  `nama_barang` varchar(25) DEFAULT NULL,
  `jumlah` int(15) DEFAULT NULL,
  `satuan` varchar(6) DEFAULT NULL,
  KEY `podetail_fk2` (`kode_barang`),
  KEY `podetail_fk1` (`no_po`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

8. Tabel PO Masuk

```
CREATE TABLE IF NOT EXISTS `po_masuk` (
  `no_transaksi` varchar(15) NOT NULL,
  `tgl` date DEFAULT NULL,
  `kode_pemasok` varchar(7) DEFAULT NULL,
  `no_po` varchar(15) DEFAULT NULL,
  `no_sj` varchar(25) DEFAULT NULL,
  `no_faktur` varchar(25) DEFAULT NULL,
  `total` decimal(10,0) DEFAULT NULL,
  `sub_total` decimal(10,0) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`no_transaksi`),
  KEY `pomasuk_fk1` (`no_po`),
  KEY `pomasuk_fk2` (`kode_pemasok`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Tabel PO Masuk Detail

```
CREATE TABLE IF NOT EXISTS `pomasuk_detail` (
  `no_transaksi` varchar(15) NOT NULL,
  `kode_barang` varchar(7) DEFAULT NULL,
  `jumlah` int(11) DEFAULT NULL,
```

```
  `harga` decimal(10,0) DEFAULT NULL,
  KEY `pomasuk_detail_fk1` (`no_transaksi`),
  KEY `pomasuk_detail_fk2` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

10. Tabel Pembayaran Beli

```
CREATE TABLE IF NOT EXISTS `pembayaran_beli` (
  `no_pembayaran` varchar(15) NOT NULL,
  `no_transaksi` varchar(15) DEFAULT NULL,
  `kode_pemasok` varchar(7) DEFAULT NULL,
  `tanggal_bayar` date DEFAULT NULL,
  `metode_bayar` varchar(9) DEFAULT NULL,
  `bank` varchar(15) DEFAULT NULL,
  `no_rekening` varchar(15) DEFAULT NULL,
  `kontak` varchar(28) DEFAULT NULL,
  `total_bayar` decimal(10,0) DEFAULT NULL,
  PRIMARY KEY (`no_pembayaran`),
  KEY `bayarbeli_ibfk1` (`kode_pemasok`),
  KEY `bayarbeli_ibfk2` (`no_transaksi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

11. Tabel Pembayaran Beli Detail

```
CREATE TABLE IF NOT EXISTS `pembayaranbeli_detail` (
```

```
  `no_pembayaran` varchar(15) DEFAULT NULL,
  `kode_barang` varchar(7) DEFAULT NULL,
  `jumlah` int(15) DEFAULT NULL,
  `harga` decimal(15,0) DEFAULT NULL,
  KEY `pembayarandetail_ibfk_1` (`no_pembayaran`),
  KEY `pembayarandetail_ibfk_2` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

12. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `no_retur` varchar(15) NOT NULL,
  `tgl` date DEFAULT NULL,
  `kode_pemasok` varchar(7) DEFAULT NULL,
  `nama_pemasok` varchar(28) DEFAULT NULL,
  `no_po` varchar(15) DEFAULT NULL,
  `no_sj` varchar(25) DEFAULT NULL,
  `no_bukti` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`no_retur`),
  KEY `retur_fk1` (`kode_pemasok`),
  KEY `retur_fk2` (`no_po`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Tabel Retur Detail

```
CREATE TABLE IF NOT EXISTS `retur_detail` (
  `no_retur` varchar(15) DEFAULT NULL,
  `kode_barang` varchar(7) DEFAULT NULL,
  `nama_barang` varchar(25) DEFAULT NULL,
  `jumlah_pesan` int(15) DEFAULT NULL,
  `jumlah_retur` int(15) DEFAULT NULL,
  KEY `returdetail_fk1` (`no_retur`),
  KEY `returdetail_fk2` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

14. Tabel Penjualan

```
CREATE TABLE IF NOT EXISTS `penjualan` (
  `no_faktur` varchar(15) NOT NULL,
  `tanggal` date DEFAULT NULL,
  `konsumen` varchar(50) DEFAULT NULL,
  `alamat` varchar(255) DEFAULT NULL,
  `no_telp` varchar(15) DEFAULT NULL,
  `grand_total` double(10,0) DEFAULT NULL,
  `status` tinyint(5) DEFAULT NULL,
  PRIMARY KEY (`no_faktur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

15. Tabel Penjualan Detail

```
CREATE TABLE IF NOT EXISTS `penjualan_detail` (
 `no_faktur` varchar(15) DEFAULT NULL,
 `kode_barang` varchar(7) DEFAULT NULL,
 `nama_barang` varchar(28) DEFAULT NULL,
 `harga` decimal(10,0) DEFAULT NULL,
 `jumlah` tinyint(5) DEFAULT NULL,
 `satuan` varchar(6) DEFAULT NULL,
 KEY `penjualan_detail_fk1` (`no_faktur`),
 KEY `penjualan_detail_fk2` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

16. Tabel Barang Keluar

```
CREATE TABLE IF NOT EXISTS `barang_keluar` (
 `no_transaksi` varchar(15) NOT NULL,
 `tanggal` date NOT NULL,
 `no_faktur` varchar(15) NOT NULL,
 `konsumen` varchar(50) NOT NULL,
 `alamat` varchar(255) NOT NULL,
 `no_telp` varchar(15) DEFAULT NULL,
 `sopir` varchar(25) NOT NULL,
 `no_kendaraan` varchar(10) NOT NULL,
 `total` decimal(10,0) NOT NULL,
```

```
  `status` tinyint(5) NOT NULL,
  PRIMARY KEY (`no_transaksi`),
  KEY `barangkeluar_fk1` (`no_faktur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

17. Tabel Barang Keluar Detail

```
CREATE TABLE IF NOT EXISTS `barangkeluar_detail` (
  `no_transaksi` varchar(15) DEFAULT NULL,
  `kode_barang` varchar(7) DEFAULT NULL,
  `nama_barang` varchar(28) DEFAULT NULL,
  `jumlah` tinyint(5) DEFAULT NULL,
  `satuan` varchar(6) DEFAULT NULL,
  KEY `barangkeluar_detail_fk1` (`no_transaksi`),
  KEY `barangkeluar_detail_fk2` (`kode_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.4.5. Implementasi Antar Muka

Implementasi antar muka merupakan tahapan dalam memenuhi kebutuhan pengguna (*user*) dalam berinteraksi dengan sistem. Implementasi antar muka dilakukan dengan membuat antar muka dalam bentuk *form*. Setiap halaman *form* yang dibuat akan dibentuk sebuah *file* yang berekstensi .java. File-file tersebut dapat diakses dan akan penghubung antara pengguna sistem. Berikut ini adalah implementasi antar muka struktur menu pada perancangan sistem in*form*asi

pembelian dan penjualan bahan bangunan berbasis java *desktop* pada CV ALBET Betung.

4.5.5.1. Implementasi Antar Muka Bagian Administrasi

Implementasi halaman yang terdapat pada halaman utama berdasarkan pengguna bagian administrasi yang dibuat berekstensi .java dapat dilihat pada tabel berikut.

Tabel 4. 9 Implementasi Halaman Pengguna Bagian Administrasi

Menu	Deskripsi	Nama File
Login	Halaman untuk masuk dan untuk keluar apliaksi program aplikasi	Login.java
Menu Utama	Halaman menu utama pada programa aplikasi	MenuUtama.java
Menu Master	Halaman untuk menginput data master pada program aplikasi.	BarangView.java KategoriView.java SatuanView.java Pemasok_View.java Pemasok_View.java
Menu Tentang	Halaman mengenai in*form*asi program aplikasi	Tentang_View.java

4.5.5.2. Implementasi Antar Muka Kasir

Implementasi halaman yang terdapat pada halaman utama berdasarkan pengguna kasir yang dibuat berekstensi .java dapat dilihat pada tabel berikut.

Tabel 4. 10 Implementasi Halaman Pengguna Bagian Logistik

Menu	Deskripsi	Nama File
Login	Halaman untuk masuk dan untuk keluar apliaksi program aplikasi	Login.java
Menu Utama	Halaman menu utama pada programa aplikasi	MenuUtama.java
Pembuatan PO	Halaman untuk menginput data transaksi pemesanan pembelian pada program aplikasi.	PembelianBarang.java
Penerimaan Barang	Halaman untuk menginput transaksi penerimaan barang pada program aplikasi.	PenerimaanBarang.java
Pembayaran Pembelian Barang	Halaman untuk menginput transaksi Pembayaran Pembelian Barang pada program aplikasi.	PembayaranPembelian_View.java
Penjualan	Halaman untuk menginput transaksi penjualan pada program aplikasi.	Penjualan_View.java
Pengeluaran Barang	Halaman untuk menginput transaksi Pengeluaran Barang pada program aplikasi.	BarangKeluar.java

Menu	Deskripsi	Nama File
Menu Tentang	Halaman mengenai in*form*asi program aplikasi	Tentang_View.java

4.5.5.3. Implementasi Antar Muka Pemilik

Implementasi halaman yang terdapat pada halaman utama berdasarkan pengguna pemilik yang dibuat berekstensi .java dapat dilihat pada tabel berikut.

Tabel 4. 11 Implementasi Halaman Pengguna Bagian Pemilik

Sub Menu	Deskripsi	Nama File
Login	Halaman untuk masuk dan untuk keluar apliaksi program aplikasi	Login.java
Menu Utama	Halaman menu utama pada programa aplikasi	MenuUtama.java
Laporan Pembelian Barang	Halaman untuk mencetak Laporan Pembelian Barang pada program aplikasi.	Laporan_Pembelian.java
Laporan Penjualan Barang	Halaman unutk mencetak Laporan Penjualan Barang pada program aplikasi.	LaporanPenjualan_View.java

4.4.6. Implementasi Instalasi Program

Untuk dapat menggunakan program aplikasi ini terlebih dahulu harus menginstall software pendukung seperti XAMPP. Berikut langkah untuk menginstall XAMPP.

Langkah awal untuk dapat menginstall XAMPP, tentunya Anda membutuhkan seperangkat komputer dengan *operating system* Windows 7. Berikut ini adalah langkah-langkah yang perlu dilakukan untuk menginstall XAMPP di windows 7 :

1. Silahkan download terlebih dahulu XAMPP versi 1.8.1, Klik dua-kali *file* XAMPP yang telah anda *download* (contoh yang dipakai di sini adalah xampp-win32-1.8.1-installer.exe, yaitu XAMPP versi 1.8.1).



Gambar 4. 43 *Installer*.exe XAMPP versi 1.8.1

2. Berikutnya klik tombol OK untuk pilihan bahasa, *English*.



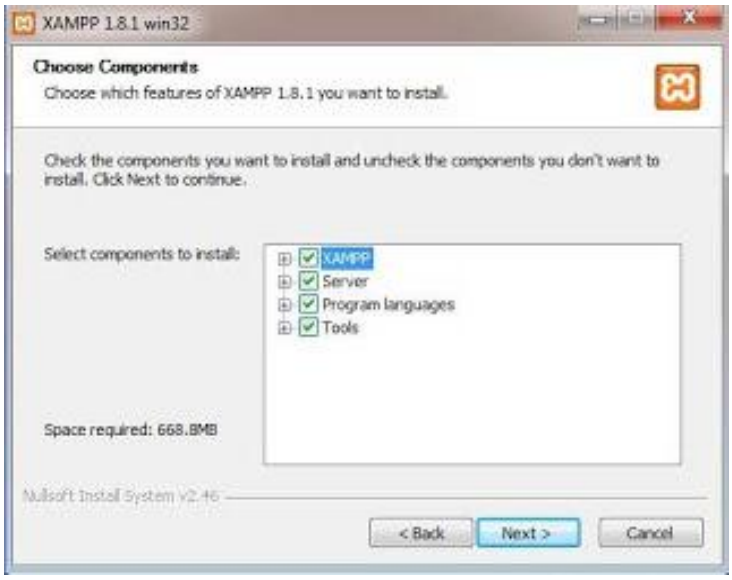
Gambar 4. 44 *Installer Language*

3. Klik tombol *Next* pada jendela *Setup Wizard* yang muncul.



Gambar 4. 45 *Setup Wizard*

4. Pada jendela yang muncul berikutnya, Jangan lupa klik 3 kotak pada *service section*. Yaitu anda harus menginstall apache as service, mysql as service dan filezilla as service. klik saja [Install], dimana *shortcut* untuk menjalankan XAMPP akan ada di *desktop* dan di menu *Start*.



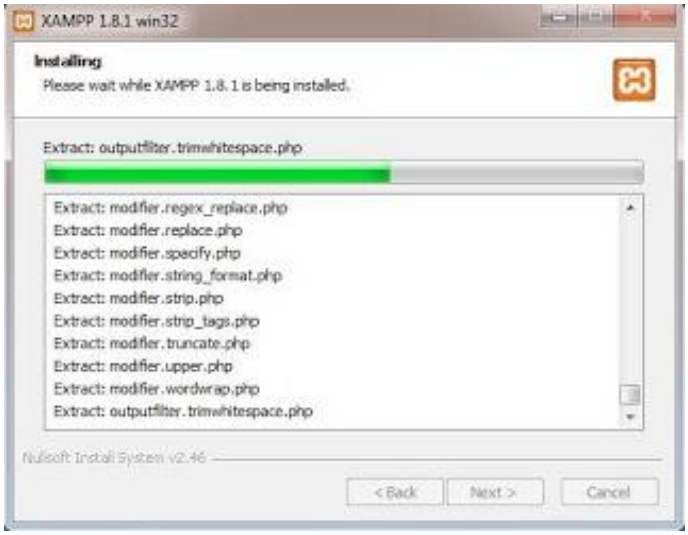
Gambar 4. 46 *Service Section*

5. Berikutnya anda diminta memilih *folder* tempat XAMPP disimpan. Untuk mudahnya pilih saja di c:\, seperti *default*-nya dan klik tombol *Next*.



Gambar 4. 47 Install Location

6. Tunggu beberapa menit sehingga XAMPP *terinstall*.



Gambar 4. 48 Proses Install XAMPP

7. Klik tombol *Finish* setelah XAMPP selesai terinstall.



Gambar 4. 49 Proses Completing XAMPP

8. Anda akan mendapatkan pilihan, apakah akan menjalankan XAMPP *Control Panel* sekarang atau lain kali. Klik tombol *Yes* bila Anda akan menjalankannya.
9. Kalau *Windows Firewall* anda aktif maka anda akan mendapat peringatan karena dianggap berbahaya. Klik tombol *Allow acces* bila Anda tetap ingin melanjutkanya.



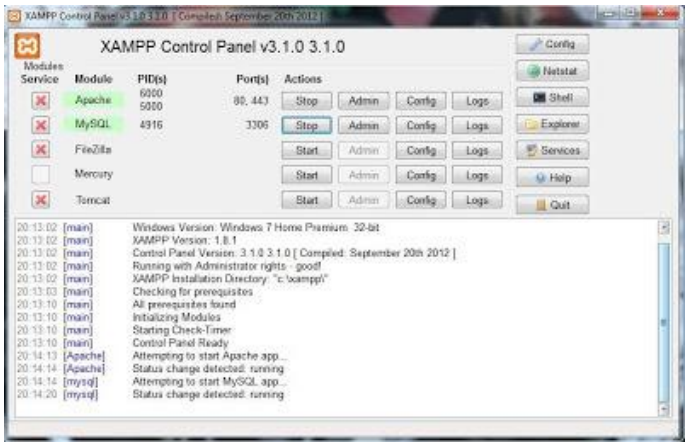
Gambar 4. 50 Peringatan Windows Firewall

10. Setiap kali anda akan mengelola situs *Web* anda di PC lokal, anda harus menjalankan XAMPP. Menjalankan XAMPP berarti memfungsikan PC anda sebagai *server Web*. Untuk menjalankannya, klik ikon XAMPP dan anda dihadapkan pada XAMPP *Control Panel*.



Gambar 4. 51 XAMPP *Control Panel*

11. Klik tombol *Start* masing-masing untuk Apache dan MySql, sehingga muncul tanda *Running* untuk keduanya. PC lokal Anda kini sudah menjadi *server* Web lokal.



Gambar 4. 52 Running Apache dan MySql XAMPP

4.4.7. Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan Bahanbahan Bangunan pada CV ALBET Adapun cara pengoperasiannya dibagi berdasarkan pengguna/*user* menggunakannya adalah sebagai berikut :

A. Seluruh Pengguna/*User*

1. *Form* Menu Login

Sebelum masuk dalam Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan pada CV ALBET kita akan masuk *form login* seperti di bawah ini, hal ini dimaksudkan agar tidak sembarangan petugas bisa memakai program ini.



Gambar 4. 53 Implementasi *Form Login*

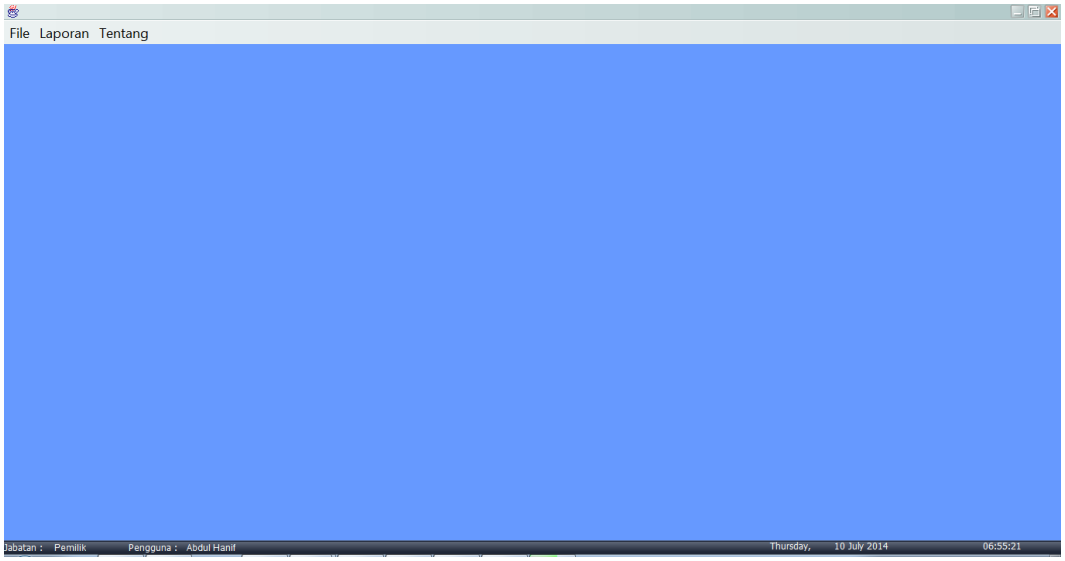
Pada *form* halaman *login* ini kita diminta mengisi *username* dan *password* pada kolom isian *username* dan *password* serta mengklik tombol *LOGIN*. Apabila kolom isian *username* dan *password* diisi dengan valid maka sistem akan memvalidasi *username* dan *password* sudah valid maka pengguna akan masuk ke menu utama, tetapi jika *username* dan *password* yang diisi tidak valid maka akan muncul peringatan kesalahan seperti yang tersebut pada gambar 5.53 dibawah ini.



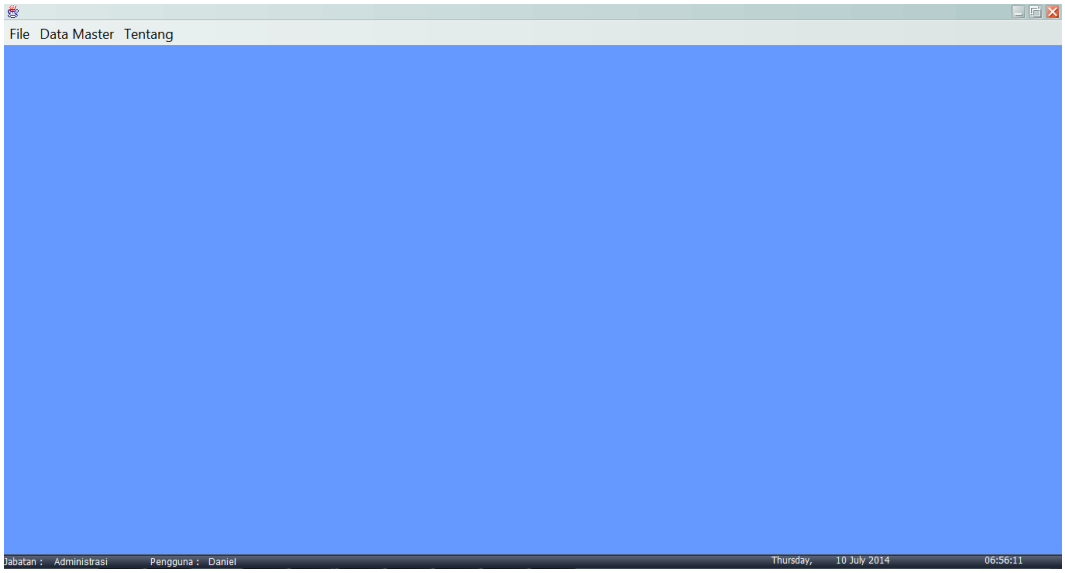
Gambar 4. 54 Kesalahan *Login*

2. *Form* Menu Utama

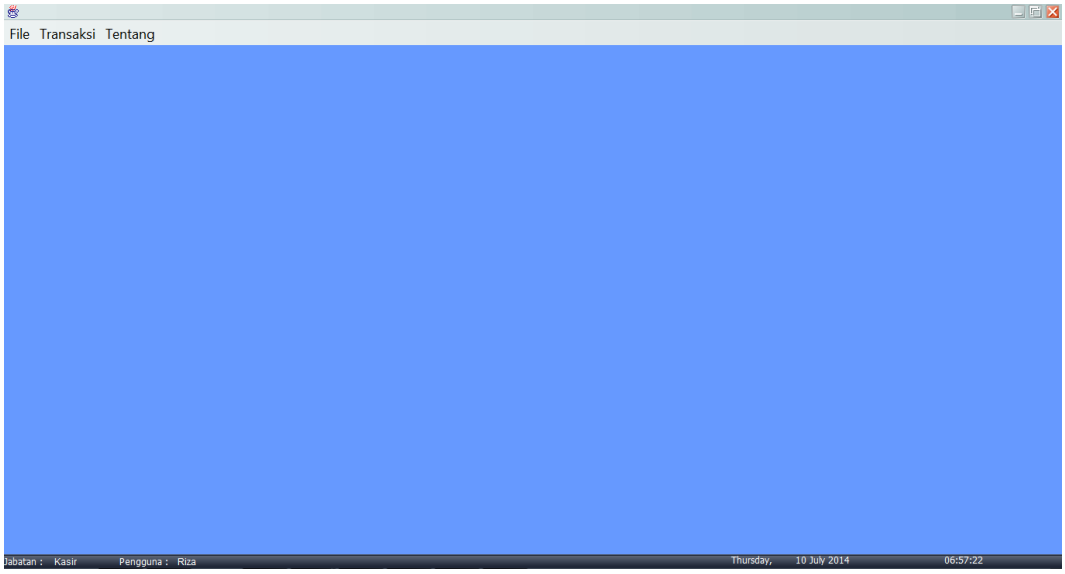
Form Menu Utama sistem in*form*asi pengolahan data pembelian dan penjualan bahan bangunan pada CV ALBET dimana dalam halaman muka ini terdapat deretan menu utama seperti *file*, master, transaksi, laporan dan tentang sesuai dengan *user* atau pengguna yang menggunakan program ini. Pada *form* menu utama pada sub menu *file* terdapat sub menu keluar. Tombol *exit* berfungsi keluar dari proses *login dan program*, sehingga aplikasi ini tidak dapat dipakai dan untuk mengaktifkan kembali sistem aplikasi harus mengisi *login* lagi.



Gambar 4. 55 Implementasi *Form* Menu Utama Pemilik



Gambar 4. 56 Implementasi *Form* Menu Utama Administrasi



Gambar 4. 57 Implementasi *Form* Menu Utama Kasir

B. Bagian Administrasi

1. *Form* Kategori Barang

Berikut ini merupakan *form* kategori barang, *form* ini digunakan untuk mengelola data kategori barang.



Gambar 4. 58 Implementasi *Form* Kategori Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **Kode Kategori** : kode dari kategori, terisi otomatis saat pengguna / *user* ingin menambahkan kategori

b) **Nama Kategori** : nama dari kategori yang diisikan

c) **Keterangan** : penjelasan dari kategori yang diisikan. *Field* ini boleh dikosongkan

Keterangan *Button*:

a) **Tambah** : digunakan untuk menambah jumlah kategori barang atau bahan bangunan yang akan di *input.*

b) **Ubah** : untuk mengubah data kategori bahan bangunan yang telah ada

c) **Hapus** : digunakan untuk menghapus data kategori bahan bangunan

d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

e) **Keluar** : untuk keluar dari *form* kategori barang ke tampilan menu *user* / pengguna

2. *Form* Satuan Barang

Berikut ini merupakan *form* satuan barang, *form* ini digunakan untuk mengelola data satuan barang.



Gambar 4. 59 Implementasi *Form* Satuan Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **Kode Satuan** : kode dari satuan, terisi otomatis saat pengguna / *user* ingin menambahkan satuan

b) **Satuan** : nama dari satuan yang diisikan

Keterangan *Button*:

a) **Tambah** : digunakan untuk menambah jumlah satuan barang atau bahan bangunan yang akan di *input.*

b) **Ubah** : untuk mengubah data satuan bahan bangunan yang telah ada

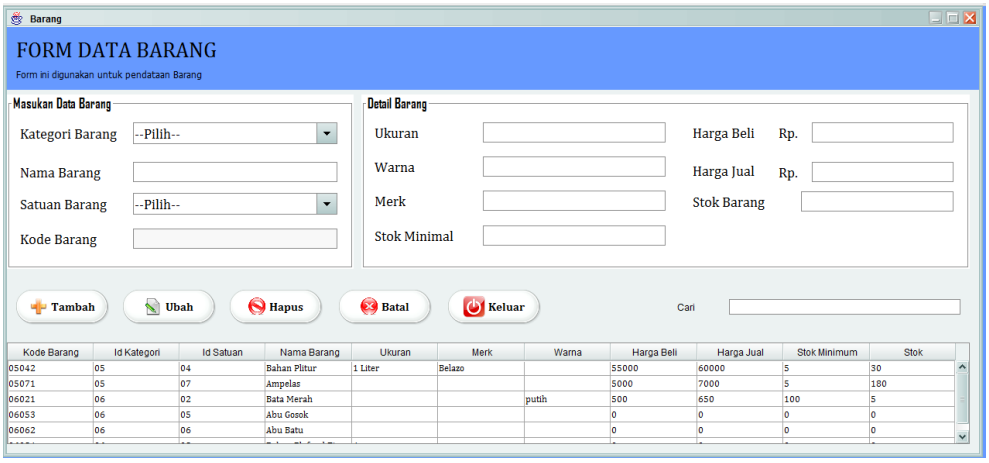
c) **Hapus** : digunakan untuk menghapus data satuan bahan bangunan

d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

e) Keluar : untuk keluar dari *form* satuan barang ke tampilan menu *user* / pengguna

3. *Form* Barang

Berikut ini merupakan *form* barang atau bahan bangunan, *form* ini digunakan untuk mengelola data barang atau bahan bangunan.



Gambar 4. 60 Implementasi *Form* Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

a) Kategori Barang : merupakan kode dari kategori barang dan nama kategori, tersedia otomatis berdasarkan kategori barang yang telah dibuat

b) Nama Barang : merupakan nama dari barang yang diisikan

c) Satuan Barang : merupakan kode dari satuan barang dan nama satuan, tersedia otomatis berdasarkan satuan barang yang telah dibuat

d) Kode Barang : kode dari barang atau bahan bangunan, terisi otomatis saat pengguna / *user* ingin memilih nama satuan. Kode barang dibuat berdasarkan konsep kodifikasi yang akan dijelaskan pada selanjutnya.

e) **Ukuran :** ukuran dari barang yang diisikan. *Field* ini boleh dikosongkan

f) **Warna :** warna dari barang yang diisikan. *Field* ini boleh dikosongkan

g) **Merk** : merk dari barang yang diisikan. *Field* ini boleh dikosongkan

h) **Stok Minimal** : stok minimal dari barang yang harus diisikan. *Field* ini tidak boleh dikosongkan

i) **Harga Beli** : Harga beli dari barang yang harus diisikan sebagai inisialisai harga beli tetapi akan *update* jika terjadi transaksi beli. *Field* ini tidak boleh dikosongkan

j) **Harga Jual** : Harga juali dari barang yang harus diisikan sebagai inisialisai harga beli tetapi akan *update* jika terjadi transaksi jual. *Field* ini tidak boleh dikosongkan

k) **Stok Barang** : stok dari barang yang harus diisikan dengan angka nol (0) sebagai inisialisai stok awal tetapi akan *update* jika terjadi transaksi beli. *Field* ini tidak boleh dikosongkan

Keterangan *Button*:

a) **Tambah** : digunakan untuk menambah jumlah barang atau bahan bangunan yang akan di *input.*

b) **Ubah** : untuk mengubah data barang atau bahan bangunan yang telah ada

c) **Hapus** : digunakan untuk menghapus data barang atau bahan bangunan

d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

e) **Keluar** : untuk keluar dari *form* barang ke tampilan menu *user* / pengguna

Pengkodean atau kodifikasi digunakan untuk tujuan mengklafikasikan data, memasukan data kedalam komputer dan untuk mengambil bermacam-macam in*form*asi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus (misalnya %, /, -, $, #, &, ;, dan sebagainya). Angka merupakan simbol yang banyak digunakan pada sistem pengkodean. Dalam sistem in*form*asi pengolahan data pembelian dan penjualan obat pada CV ALBET terdapat pengkodean yang bertujuan mempermudah dalam memasukan dan pencarian data. Adapun pengkodean tersebut yaitu kode barang atau bahan bangunan seperti dibawah ini :

a. Bahan bangunan terdapat 5 (delapan) digit angka. Untuk lebih jelasnya mengenai pengkodean ini adalah sebagai berikut :

   ***Form*at** : 99-99-9

   Keterangan :

   99 → Kode Kategori

   Kode Kategori dapat diisi dengan :

   01 = Alat Tukang

   02 = Bahan Alat Penggantung

   03 = Bahan Atap

   04 = Bahan Elektrenikal

   05 = Bahan *Finishing*

   06 = Bahan Jadi

   07 = Bahan Kaca

   08 = Bahan Kayu

09 = Bahan Lantai

10 = Bahan Logam

11 = Bahan Paku

12 = Bahan Pengisap Air

13 = Bahan Perpipaan

14 = Bahan Saluran Air

15 = Sanitasi Air

99 → Kode Satuan

Kode Satuan dapat diisi dengan :

01 = Batang

02 = Buah

03 = Galon

04 = Kaleng

05 = Kg

06 = Kubik

07 = Lembar

08 = Meter

09 = Pasang

10 = Zak

9 → No. Urut Barang atau Bahan Bangunan

No. Urut dapat diisi dengan :

1 = No. urut satu

2 = No. urut dua, dan seterusnya

Contoh :

05042

Artinya :

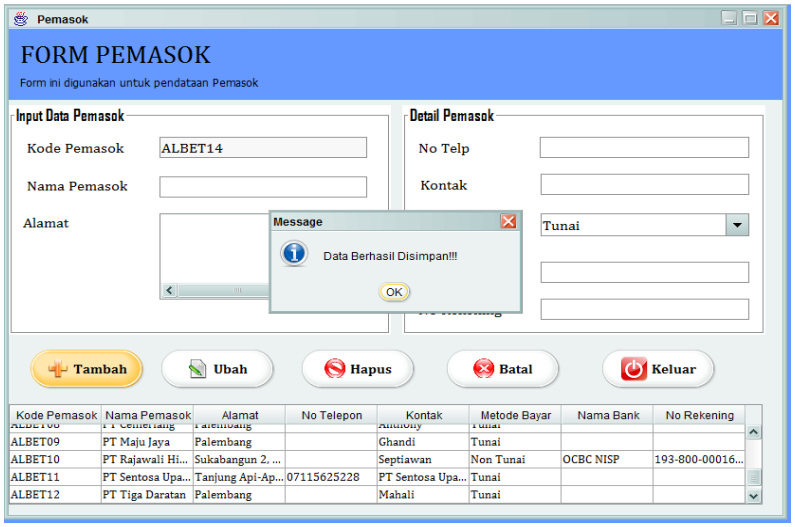
05 → menyatakan kode kategori yang memiliki nama kategori yaitu Bahan *Finishing*

04 → menyatakan kode satuan yang memiliki nama satuan yaitu Kg

2 → No urut dari kode barang berdasarkan jenis kategori

4. *Form* Pemasok

Berikut ini merupakan *form* pemasok atau *supplier*, *form* ini digunakan untuk mengelola data barang atau bahan bangunan.



Gambar 4. 61 Implementasi *Form* Pemasok

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **Kode Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* ingin menambahkan pemasok/*supplier*
b) **Nama Pemasok** : nama dari pemasok/*supplier* yang diisikan
c) **Alamat** : alamat dari pemasok/*supplier* yang diisikan
d) **No. Telp** : no telp dari pemasok/*supplier* yang diisikan
e) **Kontak** : kontak atau pihak dari pemasok/*supplier* yang diisikan
f) **Metode Bayar** : cara pembayaran CV ALBET kepada pemasok/*supplier* yang dipilih
g) **Nama Bank** : nama bank yang ditunjuk oleh pemasok/*supplier* sebagai sarana untuk pembayaran pembelian bahan bangunan. *Field* ini boleh dikosongkan jika metode bayar secara tunai.
h) **No Rekening** : no rekening bank yang ditunjuk oleh pemasok/*supplier* sebagai sarana untuk pembayaran pembelian bahan bangunan. *Field* ini boleh dikosongkan jika metode bayar secara tunai.

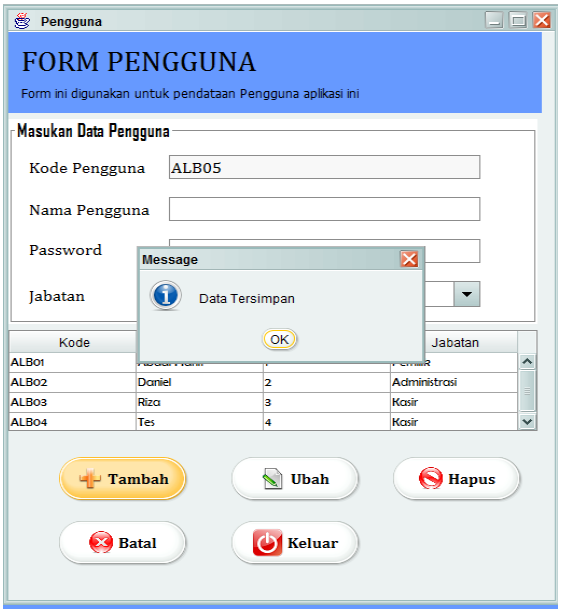
Keterangan *Button*:

a) **Tambah** : digunakan untuk menambah jumlah pemasok/*supplier* yang akan di *input.*
b) **Ubah** : untuk mengubah data pemasok/*supplier* yang telah ada
c) **Hapus** : digunakan untuk menghapus data pemasok/*supplier*
d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

e) **Keluar** : untuk keluar dari *form* pemasok/*supplier* ke tampilan menu *user* / pengguna

5. *Form* Pengguna

Berikut ini merupakan *form* pengguna atau *user*, *form* ini digunakan untuk mengelola data pengguna atau *user*.



Gambar 4. 62 Implementasi *Form* Pengguna

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **Kode Pengguna** : kode dari pengguna/*user* terisi otomatis saat pengguna / *user* ingin menambahkan pengguna/*user*

b) **Satuan** : nama dari pengguna/*user* yang diisikan

c) **Password** : *password* dari pengguna/*user* yang diisikan

d) **Jabatan** : jabatn dari pengguna/*user* yang dipilih

Keterangan *Button*:

a) **Tambah** : digunakan untuk menambah jumlah pengguna/*user* yang akan di *input.*

b) **Ubah** : untuk mengubah data pengguna/*user* yang telah ada

c) **Hapus** : digunakan untuk menghapus data pengguna/*user*

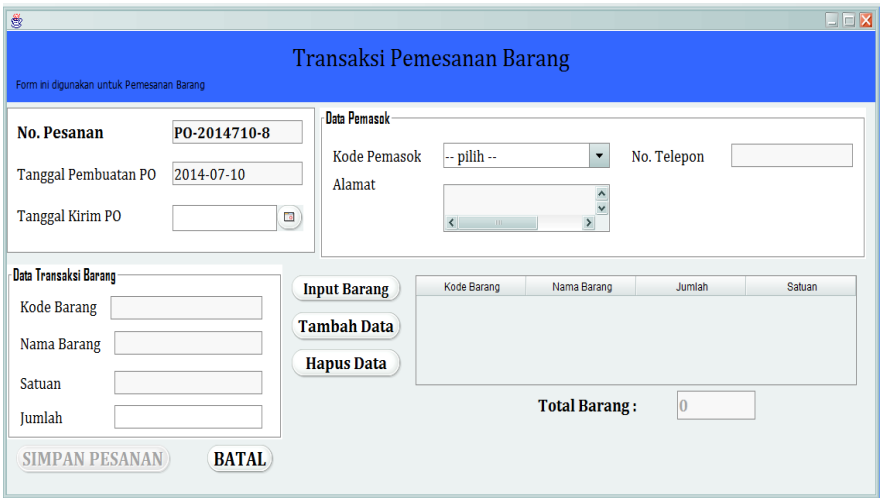
d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

e) **Keluar** : untuk keluar dari *form* pengguna/*user* ke tampilan menu *user* / pengguna

C. Kasir

1. Transaksi Pemesanan Barang

Berikut ini merupakan *form* transaksi pemesanan barang (pembuatan *purchase order*), *form* ini digunakan untuk melakukan transaksi pembuatan *purchase order* (PO) kepada pemasok/*supplier*.



Gambar 4. 63 Implementasi *Form* Transaksi Pemesanan

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **No. Pesanan** : no *purchase order* (PO) yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi pemesanan
b) **Tanggal Pembuatan PO** : tanggal saat *purchase order* (PO) yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi pemesanan
c) **Tanggal Kirim PO**: tanggal yang direncanakan untuk mengirimkan *purchase order* (PO) yang dibuat kepada pemasok/*supplier*, dipilih sesuai keinginan pengguna / *user* saat menambahkan transaksi pemesanan
d) **Kode Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* ingin menambahkan pemasok/*supplier*
a) **Alamat** : alamat dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan pemasok/*supplier*
b) **No. Telp** : no telp dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan pemasok/*supplier*
e) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input* barang
f) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input* barang

g) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input* barang

h) **Jumlah** : jumlah barang atau bahan bangunan yang akan dibeli

i) **Total Barang** : jumlah barang atau bahan bangunan yang akan dibeli

Keterangan *Button*:

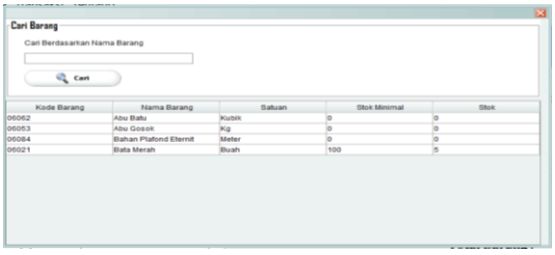
a) **Input Barang** : digunakan untuk menambah barang atau bahan banggunan yang akan di *input* ke dalam *purchase order* (PO) berdasarkan form cari barang seperti gambar 4.64.

b) **Tambah Data** : untuk menambah barang atau bahan banggunan yang telah dipilih ke dalam tabel pemesanan

c) **Hapus Data** : untuk menghapus barang atau bahan banggunan yang telah dipilih dari dalam tabel pemesanan

d) **Simpan Pesanan** : untuk menyimpan transaksi yang telah dilakukan, tombol ini akan aktif setelah ada barang atau bahan bangunan dimasukan ke table pemesanan

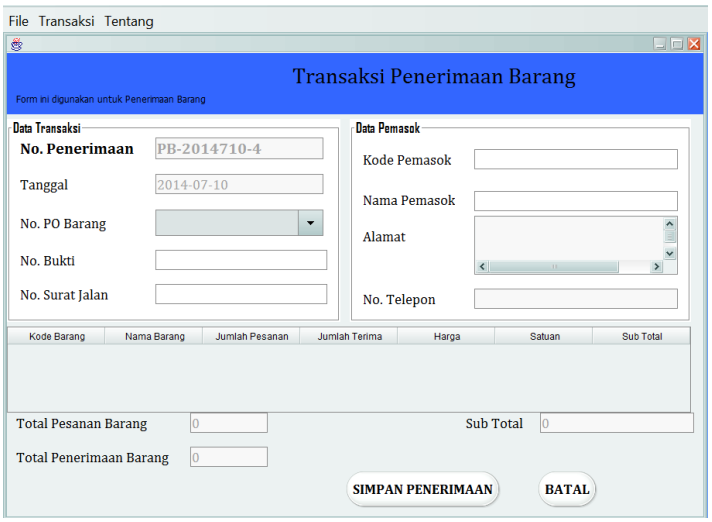
e) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.



Gambar 4. 64 Implementasi *Form* Cari Barang Pembelian

2. Transaksi Penerimaan Barang

Berikut ini merupakan *form* transaksi penerimaan barang, *form* ini digunakan untuk melakukan transaksi penerimaan barang dari pemasok/*supplier*.



Gambar 4. 65 Implementasi *Form* Transaksi Penerimaan Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

c) **No. Penerimaan** : no penerimaan barang yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi penerimaan barang

d) **Tanggal** : tanggal saat transaksi penerimaan barang yang dibuat, terisi otomatis saat pengguna / *user* sesuai tanggal saat itu.

e) **No. PO Barang** : no *purchase order* (PO) yang telah dibuat kepada pemasok/*supplier* saat transaksi pemesanan barang

f) **No**. **Bukti** : no faktur yang terlampir pada faktur pembelian dari pemasok atau *supplier* yang diisikan

g) **No. Surat Jalan** : no surat jalan yang terlampir pada faktur pembelian atau surat jalan dari pemasok atau *supplier* yang diisikan

h) Kode Pemasok : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no po

i) Nama Pemasok : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no po

j) Alamat : alamat dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no po

k) No. Telp : no telp dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no po

l) Sub Total : jumlah total harga seluruh barang yang akan dibeli

Adapun parameter yang terdapat dalam tabel penerimaan barang adalah :

a) Nama Barang : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* memilih no po

b) Satuan Barang : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* memilih no po

c) Kode Barang : kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* memilih no po

d) Jumlah Pesanan : jumlah barang atau bahan bangunan yang dibeli berdasarkan *purchase order*(PO) yeng telah dibuat , yang terisi otomatis saat pengguna / *user* memilih no po

e) Jumlah Terima : jumlah barang atau bahan bangunan yang diterima oleh pihak CV ALBET dari pemasok atau *supplier* yang diisikan

f) **Harga** : harga barang atau bahan bangunan yang diberikan pemasok/*supplier* yang terlampir pada faktur pembelian yang diisikan

g) **Sub Total** : total harga barang dikalikan umlah barang yang diterima, secara otomatis terisi setelah mengisi jumlah terima dan harga.

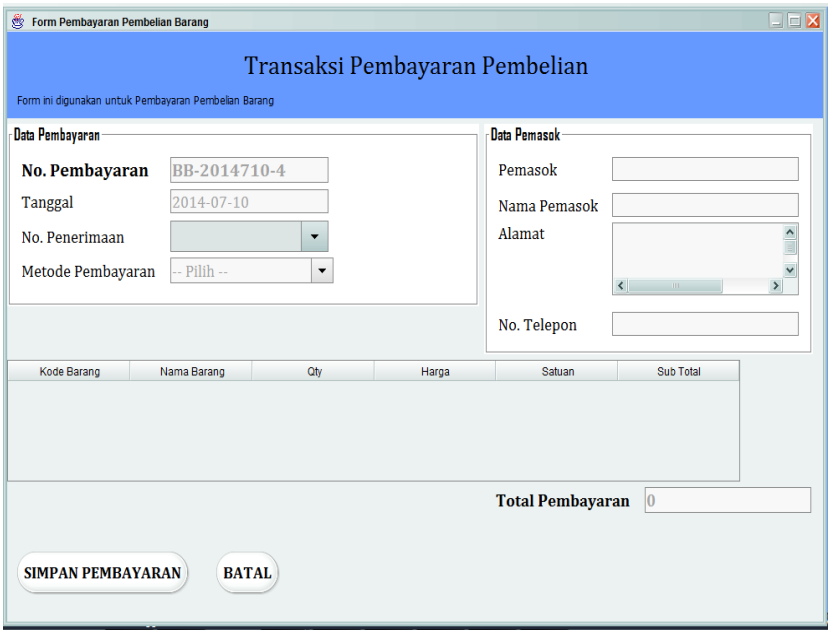
Keterangan *Button*:

a) **Simpan Penerimaan** : untuk menyimpan transaksi yang telah dilakukan

b) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

3. Transaksi Pembayaran Pembelian

Berikut ini merupakan *form* transaksi pembayaran pembelian, *form* ini digunakan untuk melakukan transaksi pembayaran pembelian.



Gambar 4. 66 Implementasi *Form* Transaksi Pembayaran Pembelian

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **No. Pembayaran** : no pembayaran yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi pembayaran pembelian
b) **Tanggal** : tanggal saat transaksi pembayaran pembelian yang dibuat, terisi otomatis saat pengguna / *user* sesuai tanggal saat itu
c) **No. Penerimaan**: no *penerimaan* yang telah dibuat saat transaksi penerimaan barang
d) **Kode Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no penerimaan
e) **Nama Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no penerimaan
f) **Alamat** : alamat dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no penerimaan
g) **No. Telp** : no telp dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no penerimaan
h) **Metode Pembayaran** : cara pembayaran CV ALBET atas pembelian
i) **Total Pembayaran** : jumlah total harga seluruh barang yang telah dibeli

Adapun parameter yang terdapat dalam tabel pembayaran pembelian adalah :

a) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* memilih no penerimaan
b) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* memilih no penerimaan

c) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* memilih no penerimaan

d) **Qty** : jumlah barang atau bahan bangunan yang dibeli oleh pihak CV ALBET dari pemasok atau *supplier*, yang terisi otomatis saat pengguna / *user* memilih no penerimaan

e) **Harga** : harga barang atau bahan bangunan yang diberikan pemasok/*supplier* yang terlampir pada faktur pembelian, yang terisi otomatis saat pengguna / *user* memilih no penerimaan.

f) **Sub Total** : total harga barang dikalikan umlah barang yang diterima, yang terisi otomatis saat pengguna / *user* memilih no penerimaan

Keterangan *Button*:

a) **Simpan Penerimaan** : untuk menyimpan transaksi yang telah dilakukan

b) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

4. Transaksi Retur Pembelian

Berikut ini merupakan *form* transaksi retur pembelian barang, *form* ini digunakan untuk melakukan transaksi retur pembelian kepada pemasok/*supplier* secara langsung pada saat sebelum penerimaan barang dengan syarat yang telah ditentukan oleh pihak pemasok/*supplier* terlebih dahulu agar transaksi dapat diproses.



Gambar 4. 67 Implementasi *Form* Transaksi Retur Pembelian

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **No. Retur** : no retur yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi retur pembelian

b) **Tanggal** : tanggal saat transaksi retur pembelian yang dibuat, terisi otomatis saat pengguna / *user* sesuai tanggal saat itu.

c) **No. PO Barang** : no *purchase order* (PO) yang telah dibuat kepada pemasok/*supplier* saat transaksi pemesanan barang

d) **No. Faktur** : no faktur yang terlampir pada faktur pembelian dari pemasok atau *supplier* yang diisikan

e) **No. Surat Jalan** : no surat jalan yang terlampir pada faktur pembelian atau surat jalan dari pemasok atau *supplier* yang diisikan

f) **Kode Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no po

g) **Nama Pemasok** : kode dari pemasok/*supplier*, terisi otomatis saat pengguna / *user* memilih no po

h) **Alamat** : alamat dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no po

i) **No. Telp** : no telp dari pemasok/*supplier* yang terisi otomatis saat pengguna / *user* menambahkan memilih no po

j) **Sub Total** : jumlah total harga seluruh barang yang akan dibeli

Adapun parameter yang terdapat dalam tabel retur pembelian adalah :

a) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* memilih no po

b) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* memilih no po

c) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* memilih no po

d) **Jumlah Pesanan** : jumlah barang atau bahan bangunan yang dibeli berdasarkan *purchase order*(PO) yeng telah dibuat , yang terisi otomatis saat pengguna / *user* memilih no po

e) **Jumlah Retur** : jumlah barang atau bahan bangunan yang diretur oleh pihak CV ALBET kepada pemasok atau *supplier* yang diisikan

Keterangan *Button*:

a) **Simpan Penerimaan** : untuk menyimpan transaksi yang telah dilakukan

b) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

5. Transaksi Penjualan Barang

Berikut ini merupakan *form* transaksi penjualan barang, *form* ini digunakan untuk melakukan transaksi penjualan barang kepada konsumen secara langsung ataupun melalui pesanan dengan syarat konsumen tersebut harus melakukan pembayaran terlebih dahulu agar transaksi dapat diproses.



Gambar 4. 68 Implementasi *Form* Transaksi Penjualan Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **No. Penjualan** : no penjualan atau *sales order* (SO) yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi penjualan

b) **Tanggal** : tanggal saat penjualan atau *sales order* (SO) yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi penjualan

c) **Konsumen** : nama dari konsumen yang diisikan

d) **Alamat** : alamat konsumen yang diisikan

e) **No. Telp** : no telp dari konsumen yang diisikan

f) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input*

g) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input*

h) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* menambahkan barang dengan menekan tombol *input*

i) **Jumlah** : jumlah barang atau bahan bangunan yang akan dijual kepada konsumen

j) **Stok Barang** : stok yang tersedia dari barang yang telah dipilih

k) **Harga Barang** : harga barang atau bahan bangunan yang akan dijual kepada konsumen, yang diisikan

l) **Grand Total** : total penjualan

m) **Bayar** : total pembayaran yang diberikan oleh konsumen

n) **Kembalian** : sisa pembayaran

Adapun parameter yang terdapat dalam tabel penjualan barang adalah :

a) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* menekan tombol tambah

b) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* menekan tombol tambah

c) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* menekan tombol tambah

d) **Qty** : jumlah barang atau bahan bangunan yang dibeli oleh konsumen, yang terisi otomatis saat pengguna / *user* menekan tombol tambah

e) **Harga** : harga barang atau bahan bangunan yang diberikan pemasok/*supplier* yang terlampir pada faktur pembelian, yang terisi otomatis saat pengguna / *user* menekan tombol tambah

f) **Sub Total** : total harga barang dikalikan umlah barang yang diterima, yang terisi otomatis saat pengguna / *user* menekan tombol tambah

Keterangan *Button*:

a) **Simpan Penjualan** : untuk menyimpan transaksi yang telah dilakukan, tombol ini akan aktif setelah ada barang atau bahan bangunan dimasukan ke table pemesanan

b) **Batal Transaksi** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal

6. Transaksi Keluar Barang

Berikut ini merupakan *form* transaksi pengeluaran barang, *form* ini digunakan untuk melakukan transaksi pegeluaran barang.



Gambar 4. 69 Implementasi *Form* Transaksi Pengeluaran Barang

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **No. Transaksi** : no transaksi atau no surat jalan yang dibuat, terisi otomatis saat pengguna / *user* ingin menambahkan transaksi pengeluaran barang
b) **Tanggal** : tanggal saat transaksi pengeluaran barang yang dibuat, terisi otomatis saat pengguna / *user* sesuai tanggal saat itu.
c) **No. Faktur** : no faktur yang telah dibuat saat transaksi penjualan
d) **Nama Sopir** : nama sopir yang akan mengantarkan barang yang diisikan
e) **No. Kendaraan** : no kendaraan atau no polisi kendaraan yang digunakan sopir untuk mengantarkan barang yang yang diisikan
f) **Nama** : nama dari konsumen, terisi otomatis saat pengguna / *user* memilih no faktur
g) **Alamat** : alamat dari konsumen yang terisi otomatis saat pengguna / *user* menambahkan memilih no faktur
h) **No. Telp** : no telp dari konsumen yang terisi otomatis saat pengguna / *user* menambahkan memilih no faktur

Adapun parameter yang terdapat dalam tabel pengeluaran barang adalah :

a) **Nama Barang** : merupakan nama dari barang yang terisi otomatis saat pengguna / *user* memilih no po
b) **Satuan Barang** : merupakan nama satuan, yang terisi otomatis saat pengguna / *user* memilih no po
c) **Kode Barang :** kode dari barang atau bahan bangunan, yang terisi otomatis saat pengguna / *user* memilih no po

d) **Jumlah Pesan** : jumlah barang atau bahan bangunan yang dibeli konsumen , yang terisi otomatis saat pengguna / *user* memilih no faktur

e) **Jumlah Keluar** : jumlah barang atau bahan bangunan yang dikeluarkan oleh pihak CV ALBET yang diisikan

Keterangan *Button*:

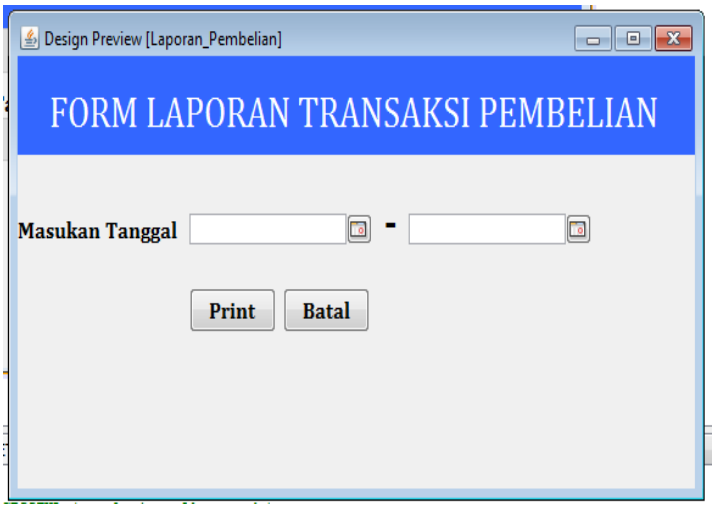
c) **Simpan Penerimaan** : untuk menyimpan transaksi yang telah dilakukan

d) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal.

D. Pemilik

1. *Form* Laporan Transaksi Pembelian

Berikut ini merupakan *form* laporan transaksi pembelian, *form* ini digunakan untuk mencetak laporan transaksi pembelian.



Gambar 4. 70 Implementasi *Form* Laporan Transaksi Pembelian

Adapun parameter yang digunakan dalam *form* ini adalah :

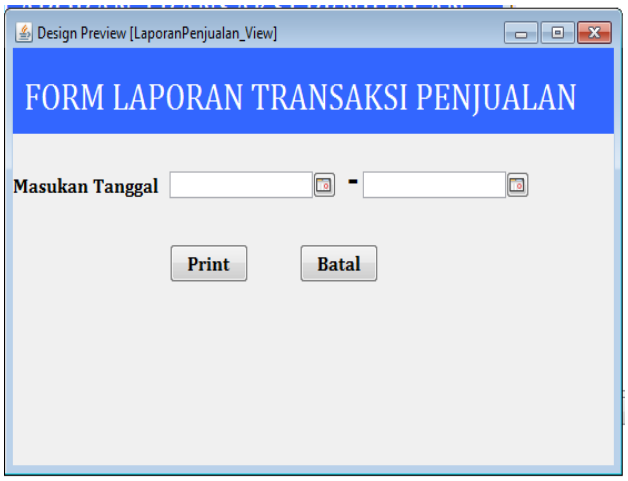
a) **Tanggal 1** : tanggal mulai untuk pencetakan laporan transaksi pembelian.
b) **Tanggal 2** : tanggal akhir untuk pencetakan laporan transaksi pembelian

Keterangan *Button*:

a) **Print** : untuk mencetak laporan transaksi pembelian sesuai tanggal
b) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal

2. *Form* Laporan Transaksi Penjualan

Berikut ini merupakan *form* laporan transaksi penjualan, *form* ini digunakan untuk mencetak laporan transaksi penjualan.



Gambar 4. 71 Implementasi *Form* Laporan Transaksi Pembelian

Adapun parameter yang digunakan dalam *form* ini adalah :

a) **Tanggal 1** : tanggal mulai untuk pencetakan laporan transaksi penjualan
b) **Tanggal 2** : tanggal akhir untuk pencetakan laporan transaksi penjualan

Keterangan *Button*:

a) **Print** : untuk mencetak laporan transaksi penjualan sesuai tanggal
b) **Batal** : digunakan untuk membatalkan semua perintah yang telah diberikan sebelumnya dan mengembalikan tampilan *form* ke kondisi awal

4.5 Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

Pengujian dapat berarti proses untuk memeriksa apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai dengan standar tertentu. Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan.

4.5.1 Rencana Pengujian

Pengujian Rancangan Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan Bahan-Bahan Bangunan Pada CV ALBET di bawah ini menggunakan data uji berupa sebuah data masukan dari bagian administrasi.

Tabel 4. 12 Rencana Pengujian

Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pengujian
Login User	Hak akses setiap username	Sistem	Black Box
Data Kategori	Pengisian data kategori bahan bangunan	Sistem	Black Box
Data Satuan	Pengisian data satuan bahan bangunan	Sistem	Black Box
Data Barang	Pengisian data barang atau bahan bangunan	Sistem	Black Box
Data Pemasok atau Supplier	Pengisian data pemasok atau supplier bahan bangunan	Sistem	Black Box
Data Pengguna	Pengisian data pengguna yang menggunakan aplikasi ini	Sistem	Black Box
Transaksi Pembuatan PO (Purchase Order)	Pengisian data transaksi pembuatan PO (Purchase Order)	Sistem	Black Box
Transaksi Penerimaan Barang	Pengisian data transaksi penerimaan barang atau bahan bangunan	Sistem	Black Box
Transaksi Pembayaran Pembelian Barang	Pengisian data transaksi pembayaran pembelian barang atau bahan bangunan	Sistem	Black Box
Transaksi Pembayaran Penjualan	Pengisian data transaksi penjualan barang atau bahan bangunan	Sistem	Black Box
Laporan Pembelian	Pengisisan tanggal pembelian per periode	Sistem	Black Box
Laporan Penjualan	Pengisisan tanggal pembelian per periode	Sistem	Black Box

4.5.2 Kasus dan Hasil Pengujian

Berikut ini adalah beberapa pengujian yang telah dilakukan, yaitu :

1. Pengujian *Form Login User*

Tabel 4. 13 Pengujian *Form Login User*

Kasus dan Hasil Pengujian			
Masukan	**Keluaran yang Diharapkan**	**Keluaran**	**Kesimpulan**
User : Pemilik Username : ALB01 Password : 1 (Data Valid)	Mempunyai hak akses untuk melihat laporan	Tidak dapat melakukan mengelola master data dan transaksi	[√] Diterima [ ] Ditolak
User : Administrasi Username : ALB02 Password : 2 (Data Valid)	Mempunyai hak akses untuk mengelola semua master data	Tidak dapat melakukan transaksi dan melihat laporan	[√] Diterima [ ] Ditolak
User : Administrasi Username : ALB03 Password : 3 (Data Valid)	Mempunyai hak akses untuk melakukan semua transaksi	Tidak dapat mengelola master data dan melihat laporan	[√] Diterima [ ] Ditolak

2. Pengujian *Form* Kategori Barang

Tabel 4. 14 Pengujian *Form* Kategori Barang

Kasus dan Hasil Pengujian			
Data Masukan	**Data Diharapkan**	**Pengamatan**	**Kesimpulan**
Input kode kategori	Input kode secara otomatis	Kode kategori ter-input secara otomatis dan berurutan	[√] Diterima [ ] Ditolak
Menginput data baru dalam tabel kategori barang	Data disimpan dalam tabel kategori barang	Data akan tersimpan pada tabel kategori barang sesuai dengan urutan kode kategori	[√] Diterima [ ] Ditolak

Kasus dan Hasil Pengujian			
Klik tombol hapus	Data yang di pilih akan terhapus	Data akan terhapus sesuai dengan yang di inginkan	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data dapat di ubah	Data dapat diubah sesuai dengan yang diinginkan	[√] Diterima [ ] Ditolak
Klik tombol batal	*Form* akan kembali pada kondisi awal	*Form* akan kembali ke kondisi awal sesuai dengan yang diharapkan	[√] Diterima [ ] Ditolak
Klik tombol keluar	*Form* akan tertutup atau keluar dari *form*	*Form* tertutup atau keluar dari *form*	[√] Diterima [ ] Ditolak

3. Pengujian *Form* Satuan Barang

Tabel 4. 15 Pengujian *Form* Satuan Barang

Kasus dan Hasil Pengujian (Data Normal)			
Data Masukan	**Data Diharapkan**	**Pengamatan**	**Kesimpulan**
Input kode satuan	Input kode secara otomatis	Kode satuan ter-input secara otomatis dan berurutan	[√] Diterima [ ] Ditolak
Menginput data baru dalam tabel satuan barang	Data disimpan dalam tabel satuan barang	Data akan tersimpan pada tabel satuan barang sesuai dengan urutan kode satuan	[√] Diterima [ ] Ditolak
Klik tombol hapus	Data yang di pilih akan terhapus	Data akan terhapus sesuai dengan yang di inginkan	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data dapat di ubah	Data dapat diubah sesuai dengan yang diinginkan	[√] Diterima [ ] Ditolak
Klik tombol batal	*Form* akan kembali pada kondisi awal	*Form* akan kembali ke kondisi awal sesuai dengan yang diharapkan	[√] Diterima [ ] Ditolak

Kasus dan Hasil Pengujian (Data Normal)			
Klik tombol keluar	*Form* akan tertutup atau keluar dari *form*	*Form* tertutup atau keluar dari *form*	[√] Diterima [ ] Ditolak

4. Pengujian Barang

Tabel 4. 16 Pengujian *Form* Barang

Kasus dan Hasil Pengujian (Data Normal)			
Data Masukan	**Data Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Semua data barang diisi lengkap.	Menampilkan data barang yang diisi.	Data barang tersimpan dalam *database*.	[√] diterima [ ] ditolak

5. Pengujian *Form* Pemasok

Tabel 4. 17 Pengujian *Form* Pemasok

Kasus dan Hasil Pengujian (Data Normal)			
Data Masukan	**Data Diharapkan**	**Pengamatan**	**Kesimpulan**
Input kode pemasok	Input kode secara otomatis	Kode pemasok ter-input secara otomatis dan berurutan	[√] Diterima [ ] Ditolak
Menginput data baru dalam tabel pemasok	Data disimpan dalam tabel satuan barang	Data akan tersimpan pada tabel satuan barang sesuai dengan urutan kode satuan	[√] Diterima [ ] Ditolak
Klik tombol hapus	Data yang di pilih akan terhapus	Data akan terhapus sesuai dengan yang di inginkan	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data dapat di ubah	Data dapat diubah sesuai dengan yang diinginkan	[√] Diterima [ ] Ditolak
Klik tombol batal	*Form* akan kembali pada kondisi awal	*Form* akan kembali ke kondisi awal sesuai dengan yang diharapkan	[√] Diterima [ ] Ditolak
Klik tombol	*Form* akan tertutup	*Form* tertutup atau	[√] Diterima

keluar	atau keluar dari *form*	keluar dari *form*	[ ] Ditolak

6. Pengujian Pengguna

Tabel 4. 18 Pengujian *Form* Pengguna

Kasus dan Hasil Pengujian (Data Normal)			
Data Masukan	**Data Diharapkan**	**Pengamatan**	**Kesimpulan**
Input kode pengguna	Input kode secara otomatis	Kode pengguna ter-input secara otomatis dan berurutan	[√] Diterima [ ] Ditolak
Menginput data baru dalam tabel pemasok	Data disimpan dalam tabel satuan barang	Data akan tersimpan pada tabel satuan barang sesuai dengan urutan kode satuan	[√] Diterima [ ] Ditolak
Klik tombol hapus	Data yang di pilih akan terhapus	Data akan terhapus sesuai dengan yang di inginkan	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data dapat di ubah	Data dapat diubah sesuai dengan yang diinginkan	[√] Diterima [ ] Ditolak
Klik tombol batal	*Form* akan kembali pada kondisi awal	*Form* akan kembali ke kondisi awal sesuai dengan yang diharapkan	[√] Diterima [ ] Ditolak
Klik tombol keluar	*Form* akan tertutup atau keluar dari *form*	*Form* tertutup atau keluar dari *form*	[√] Diterima [ ] Ditolak

7. Pengujian Transaksi Pemesanan Pembelian

Tabel 4. 19 Pengujian *Form* Pemesanan Pembelian

Kasus Dan Hasil Uji Coba			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Memilih *supplier* dan data pesanan barang yang ingin dibeli.	Menampilkan data *supplier* dan pemesanan pembelian yang diisi.	Data pemesanan pembelian tersimpan dalam *database*.	[√] diterima [ ] ditolak

8. Pengujian Transaksi Penerimaan Barang

Tabel 4. 20 Pengujian Transaksi Penerimaan Barang

Kasus Dan Hasil Uji Coba			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Memilih No. PO dan menginput jumlah pesanan barang yang diterima.	Menampilkan data penerimaan barang yang diisi.	Data penerimaan barang tersimpan dalam *database* stok barang.	[√] diterima [ ] ditolak

9. Pengujian Transaksi Pembayaran Pembelian Barang

Tabel 4. 21 : Pengujian Transaksi Pembayaran Pembelian

Kasus Dan Hasil Uji Coba			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Memilih No.PO dan menginput jumlah pembayaran.	Menampilkan data pembayaran pembelian barang yang diisi.	Data pembayaran pembelian barang tersimpan dalam *database*.	[√] diterima [ ] ditolak

10. Pengujian Transaksi Penjualan Langsung

Tabel 4. 22 Pengujian Transaksi Penjualan Langsung

Kasus Dan Hasil Uji Coba (Data Normal)			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Menginput data konsumen dan data barang yang dibeli.	Menampilkan data penjualan barang yang diisi.	Data penjualan barang tersimpan dalam *database*.	[√] diterima [ ] ditolak

11. Pengujian Transaksi Pengeluaran Barang

Tabel 4. 23 Pengujian Transaksi Pengeluaran Barang

Kasus Dan Hasil Uji Coba (Data Normal)			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Klik Tambah Data	Field-field aktif	Field-field aktif	[√] diterima [ ] ditolak
Memilih No.SO dan menginput data barang.	Menampilkan data pengeluaran barang yang diisi.	Data pengeluaran barang tersimpan dalam *database*.	[√] diterima [ ] ditolak

12. Pengujian Laporan Pembelian Barang

Tabel 4. 24 Pengujian Laporan Pembelian Barang

Kasus Dan Hasil Uji Coba (Data Normal)			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Mengisi tanggal periode laporan, laporan pembelian bahan bangunan yang sesuai dengan data yang diinginkan.	Dapat mencetak laporan pembelian bahan bangunan sesuai dengan tanggal periode yang dipilih.	Mencetak laporan pembelian bahan bangunan.	[√] diterima [ ] ditolak

13. Pengujian Laporan Penjualan Barang

Tabel 4. 25 Pengujian Laporan Penjualan Barang

Kasus Dan Hasil Uji Coba (Data Normal)			
Data Masukan	**Hasil Di Harapkan**	**Pengamatan**	**Kesimpulan**
Mengisi tanggal periode laporan, laporan penjualan bahan bangunan yang sesuai dengan data yang diinginkan.	Dapat mencetak laporan penjualan bahan bangunan sesuai dengan tanggal periode yang dipilih.	Mencetak laporan penjualan bahan bangunan.	[√] diterima [ ] ditolak

4.5.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan diatas, maka dapat ditarik kesimpulan bahwa perangkat lunak Sistem In*form*asi Pengolahan Data Pembelian dan Penjualan bahan-bahan bangunan pada CV ALBET tidak menemukan kesalahan sintak dan secara fungsional mengeluarkan hasil sesuai yang diharapkan**.**