

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Permasalahan yang akan di bahas dalam tugas akhir ini adalah membuat sistem yang dapat membantu pihak perusahaan yang memiliki gudang untuk mengatasi permasalahan seperti kehilangan barang, tidak cocoknya data yang sudah ditulis pada buku dll. Pada tahap ini dilakukan analisa terhadap data yang sudah diperoleh menggunakan teknik pengumpulan data, tahap ini merupakan tahap awal dalam pembentukan sistem atau aplikasi untuk mengimplementasikan *personal extreme programming* dan *Adaptive Response Rate Singel Exponential Smoothing* dalam memprediksi jumlah permintaan persediaan yang dibutuhkan untuk periode mendatang.

3.1.1 Identifikasi Masalah

Mengenali permasalahan yang sering terjadi termasuk kehilangan data barang, penumpukan barang pada gudang, keterlambatan dalam membuat laporan, dan tidak seimbangnya pencatatan transaksi keluar dan masuk barang dengan laporan pada saat melakukan stock opname. Pada akhirnya, petugas yang melaksanakan inventarisasi persediaan sering menghadapi kesulitan saat melakukan perhitungan karena rekapitulasi persediaan di gudang yang digunakan sebagai acuan perhitungan tidak dalam kondisi baik karena laporannya masih dalam format kertas. Sebagai hasilnya, petugas harus melengkapi atau memperbarui laporan tersebut sebelum diserahkan kepada admin .

3.1.2 Pemecahan Masalah

Hal yang pertama kali dilakukan pada tahap ini adalah menentukan dan mengungkapkan kebutuhan sistem. Data yang sudah diperoleh akan digunakan sebagai acuan untuk menentukan kebutuhan apa saja yang dibutuhkan oleh sistem. Kebutuhan sistem ada dua yaitu *functional requirements* dan *non-functional requirements*

3.1.2.1 Functional Requirements

Functional requirements merupakan kebutuhan yang memiliki kaitan langsung dengan sistem. Pada kebutuhan fungsional sistem yang dibuat berdasarkan *stakeholder* yang mana akan menjadi sebuah *role* atau hak akses pada sistem informasi inventori gudang. Berikut merupakan tabel kebutuhan fungsional sistem.

Tabel 3 1 Kebutuhan Fungsional Sistem

Stakeholder (Roles [Roles Name])	Kebutuhan Fungsional	Deskripsi
Petugas Gudang (Admin [Warehouse])	Manajemen data penjualan	Melihat data penjualan yang sudah di inputkan oleh staff gudang
	Manajemen Persediaan Barang (Stok Gudang)	Mengelola data master gudang beserta stok persediaan setiap barang.

	Manajemen Data pengguna	Mengelola data pengguna pada staff gudang
Petugas <i>Stock Opname</i> (Staff [Opname])	Audit stok gudang (akhir periode)	Entry data penghitungan stok real pada akhir periode.
	Controlling kondisi fisik Persediaan (stok gudang)	Cek selisih stok real (data audit stok akhir periode) dengan stok gudang. Mendata atau entry (<i>labeling</i> keterangan) pada setiap barang setelah dilakukan audit persediaan atau stok.
	Manajemen Laporan Opname	Mengolah laporan opname (hasil audit stok gudang beserta kondisi fisik)
	Mengakses Laporan (persediaan, mutasi barang [transaksi], opname)	Melihat dan mencetak laporan sesuai dengan jenis laporan yang dipilih

	View data mutasi barang dan peramalan (Stok Gudang)	Melihat data mutasi barang dan data peramalan ARRES (<i>Adaptive Response Rate Single Exponential Smoothing</i>)
--	---	--

3.1.2.2 Non-Functional Requirements

Non-Functional requirements atau kebutuhan non fungsional merupakan sebuah kebutuhan yang secara langsung tidak terkait dengan fitur tertentu pada sistem informasi inventori gudang. Ada dua kebutuhan non fungsional yaitu kebutuhan perangkat lunak dan perangkat keras sebagai berikut :

1. Kebutuhan Perangkat Keras (*Hardware*)

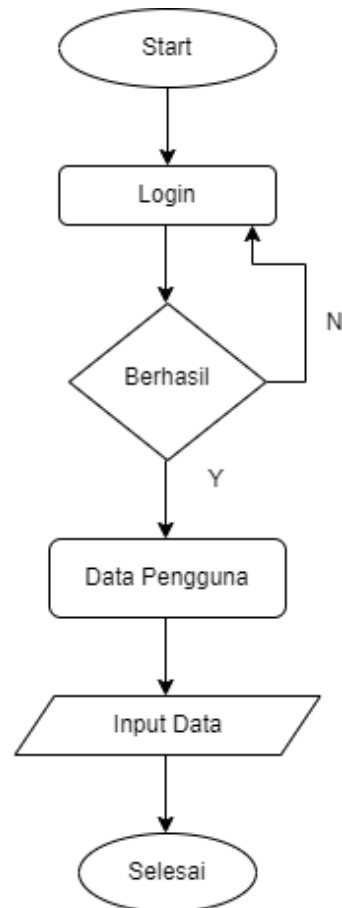
- a. Laptop : Laptop dengan spesifikasi minimum Core i3 atau AMD-A6-3420 dengan ram 2gb

2. Kebutuhan Perangkat Lunak (*Software*)

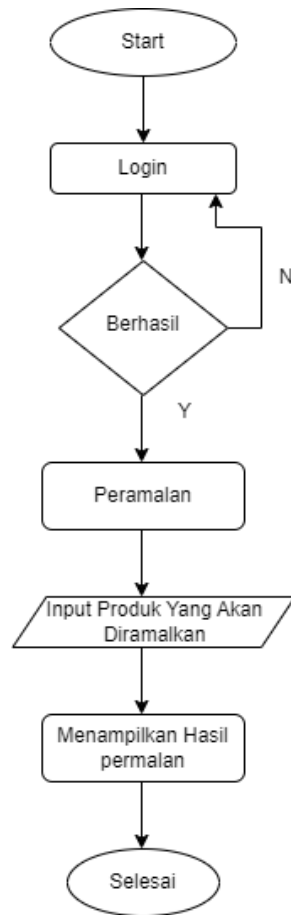
- a. *Windows 10* : Sistem operasi yang digunakan untuk mengerjakan program.
- b. *Framework Laravel 5.8* : Digunakan untuk mengerjakan program dengan *pattern HMVC (Hierarchical Module View Controller)*
- c. *Laragon* : Sebuah *software web server apache* yang didalamnya sudah tersedia *database server MySQL* dan mendukung *script PHP*.

- d. *Web Browser* : Digunakan untuk mengakses aplikasi atau sistem informasi. Dapat menggunakan *web browser* seperti *Google Chrome*, *Mozilla Firefox* atau sejenisnya.
- e. *Microsoft Word* : Digunakan untuk mendokumentasikan laporan hasil penelitian.
- f. *Microsoft Excell* : Digunakan untuk menyimpan data hasil *training*, *testing aplikasi*, dan membuka *report* yang di-*generate* melalui aplikasi.
- g. *Visual Studio Code* : Digunakan untuk *entry* kode program atau dapat menggunakan *text editor* sejenisnya.
- h. *Github* : Sebuah *platform* mendukung *branching*. *Platform* ini akan digunakan untuk membagi pengerjaan aplikasi kedalam *system moduel*.
- i. *Node Js* : Sebuah *module* atau *library* tambahan yang digunakan untuk membantu pengembangan tampilan aplikasi secara dinamis.
- j. *Bootstrap 3* : Merupakan sebuah *framework* yang membantu *development user interface* aplikasi.

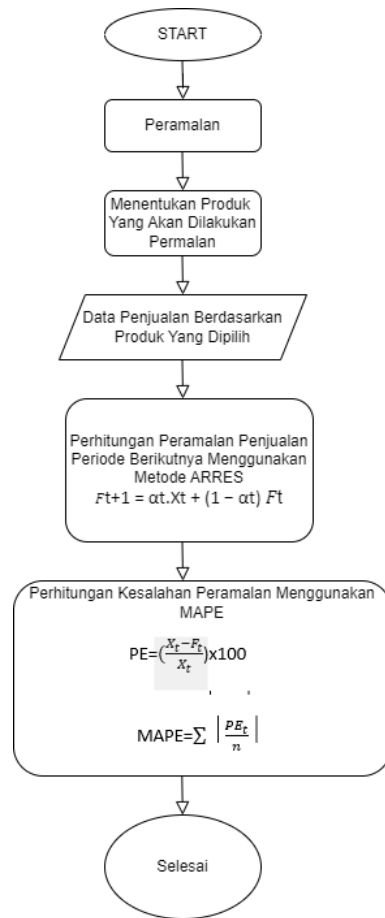
3.1.3 Flowchart



Gambar 3.1 Flowchart Admin

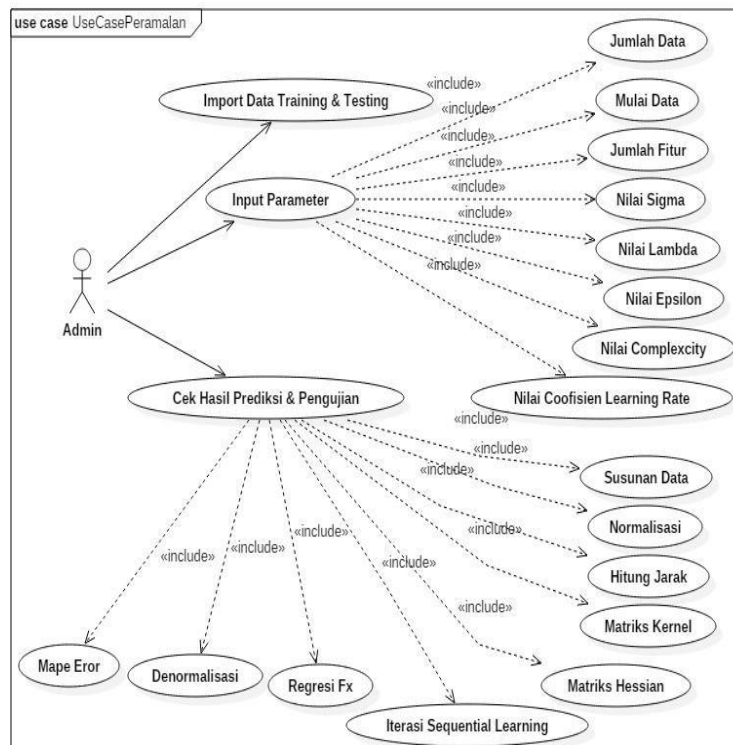
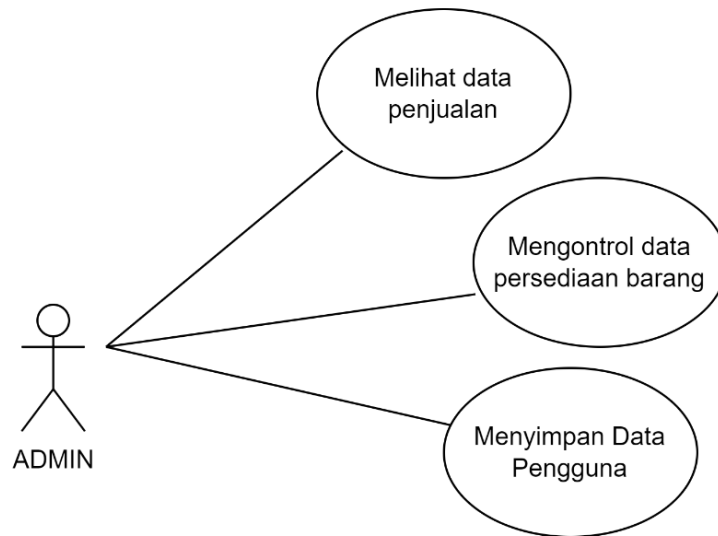


Gambar 3.2 Flowchart Staff



Gambar 3.3 Flowchart Peramalan

3.1.4 Perancangan Use Case Diagram

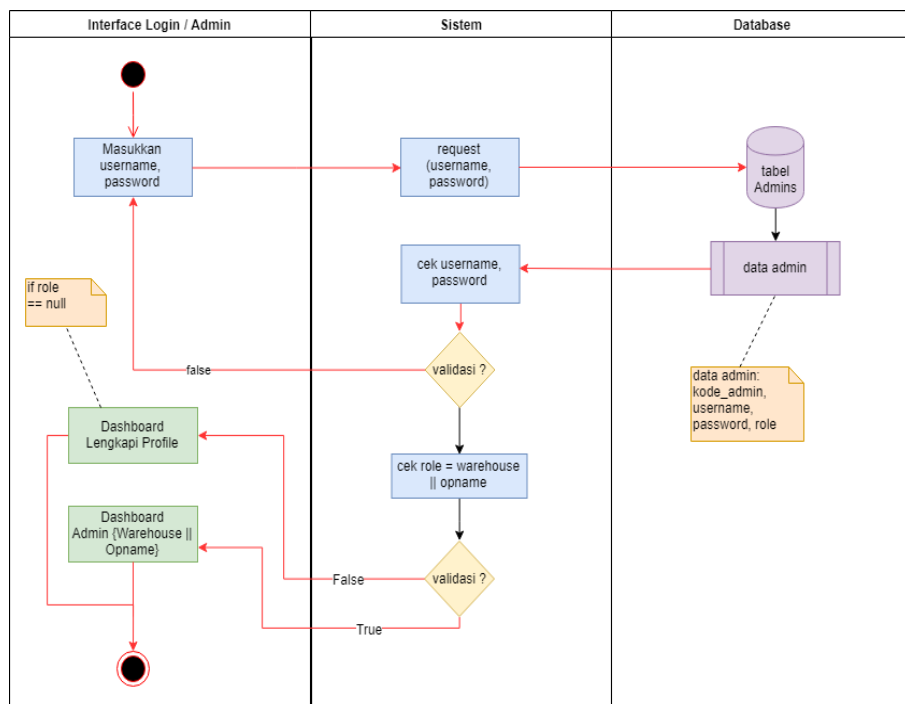


Gambar 3.4 Use Case Diagram Sistem Iinformasi Inventori Gudang

Pada Gambar diatas merupakan pemodelan dari kebutuhan fungsional . Terdapat 2 aktor utama yaitu admin warehouse, petugas *stock opname* (staff). Yang masing masing aktor memiliki kewenangan atau hak akses yaitu untuk petugas gudang dan petugas *stock opname* memiliki hak akses sebagai admin

3.1.5 Perancangan Activity Diagram

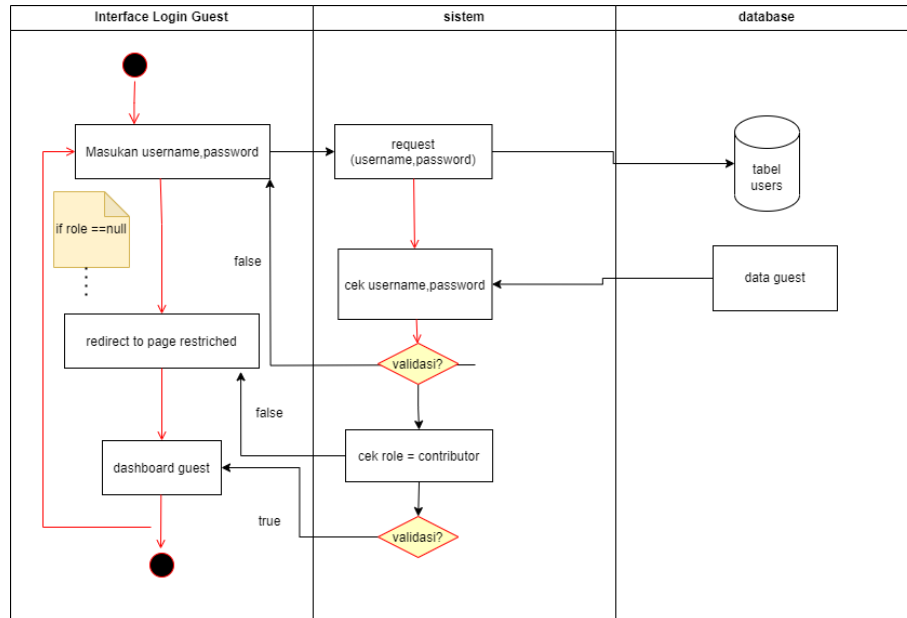
Activity Diagram merupakan suatu gambaran sebuah alur kerja (*workflow*) dari sebuah proses yang mengacu pada pemodelan *use case diagram* yang dimana setiap case akan dimodelkan ulang kedalam bentuk *activity diagram* untuk memperjelas alur yang berjalan pada sistem.



Gambar 3.5 Activity Diagram Login Admin

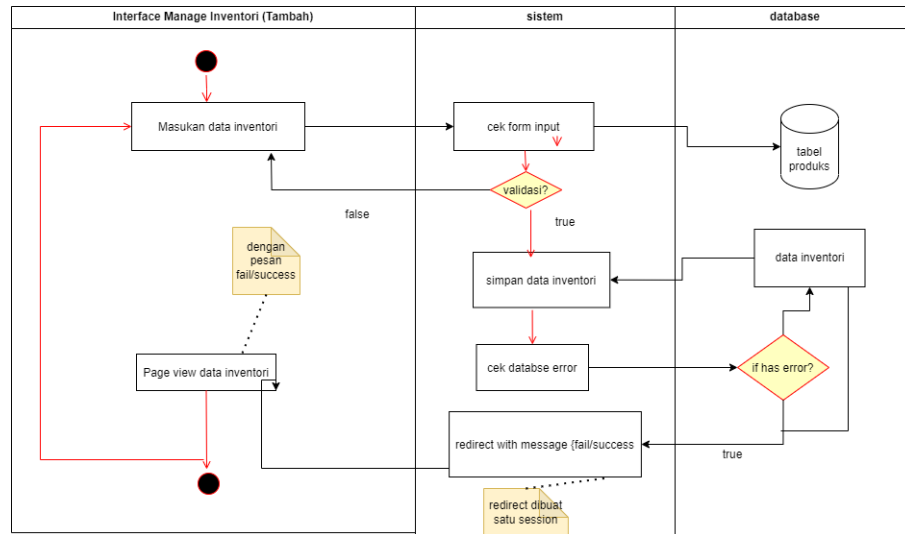
Agar dapat mengakses halaman utama, maka user perlu membuka aplikasi dan melakukan login terlebih dahulu. Setelah itu akan menampilkan halaman awal yaitu form login. User harus memasukkan username dan password jika benar maka

user dapat mengakses menu utama dalam sistem, jika tidak maka user akan melakukan login ulang.



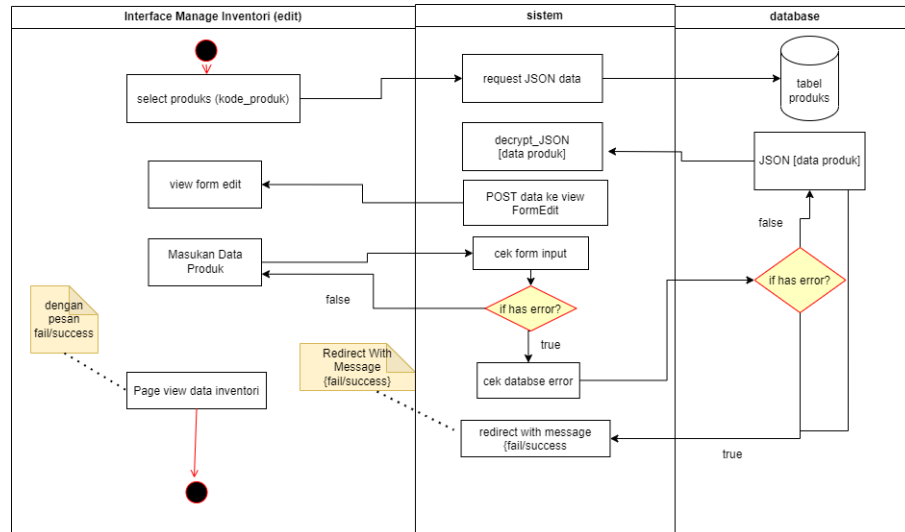
Gambar 3.6 Activity Diagram Login Guest

Gambar diatas merupakan proses login untu pengecekan autentifikasi dan hak akses untuk setiap user. Dimana setiap user atau pengguna aplikasi diwajibkan menginputkan *username* dan *password* untuk proses autentifikasi.



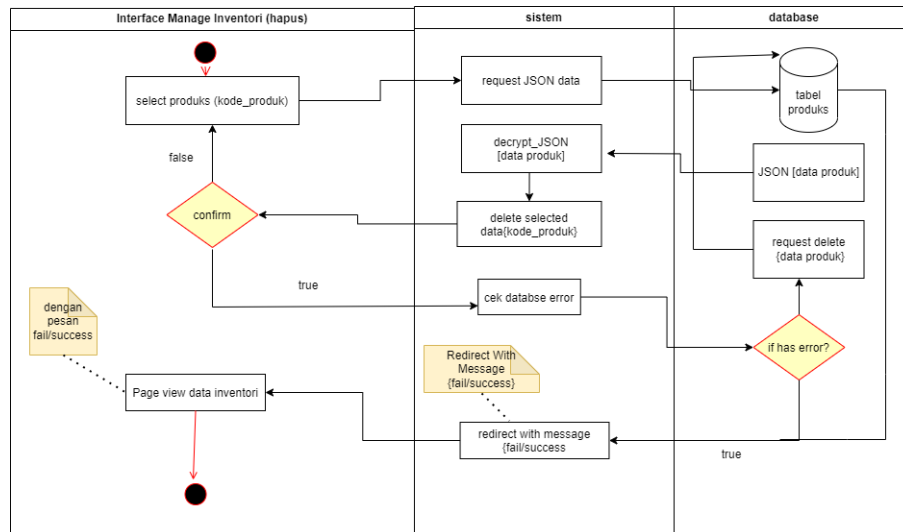
Gambar 3.7 Activity Diagram Tambah Data

Merupakan proses dari manajemen inventori bagian penambahan data. Langkah pertama admin diwajibkan mengisi sebuah *form* yang sudah disediakan sesuai dengan data inventori yang diminta. Kemudian dilakukan pengecekan *input form* guna mengetahui ada kesalahan pada penulisan atau tidak. Apabila ditemukan sebuah kesalahan pada proses *input data* maka akan muncul notif (*error message*). Jika data yang di isi sudah benar atau valid maka dari sisi database akan melakukan pengecekan apakah sudah sesuai dengan *format* dan *type* yang diminta. Dan jika data yang diminta sudah sesuai maka *request* data tersebut akan langsung disimpan dalam *database* kemudian sistem akan *redirect* ke view data inventori disertai *success message*.



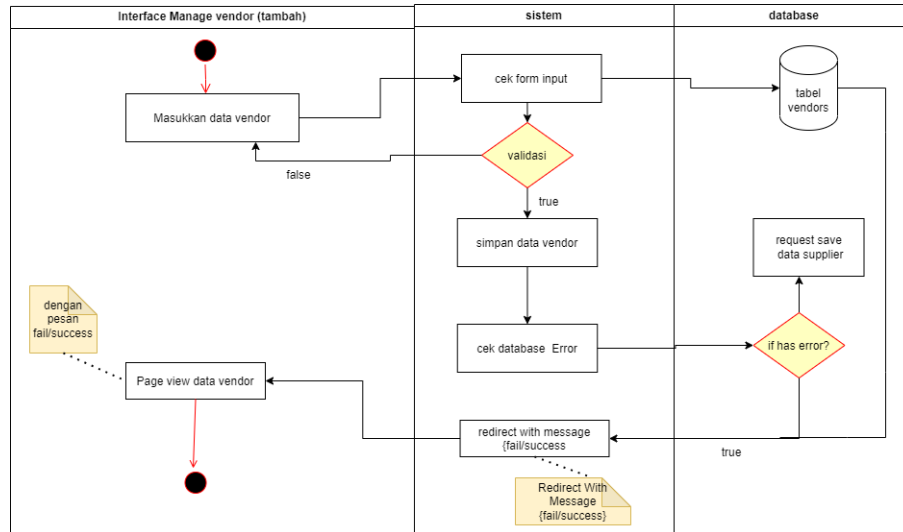
Gambar 3.8 Activity Diagram Manage Inventori Edit Data

Merupakan proses untuk mengedit data pada inventori pertamanya yang harus dilakukan adalah staff memilih data inventori yang akan di edit lalu sistem akan melakukan *request* data tersebut dalam bentuk format *JSON* berdasarkan kode barang yang di inputkan kedalam database. Apabila ada kesalahan penulisan format maka akan keluar notif *error message* pada form edit dan jika data yang dimasukkan sudah valid atau sesuai dengan data yang diminta sesuai dengan request maka data tersebut kan langsung disimpan kedalam database kemudian sistem akan *redirect* kedalam view data inventori disertai tulisan *succes message*.



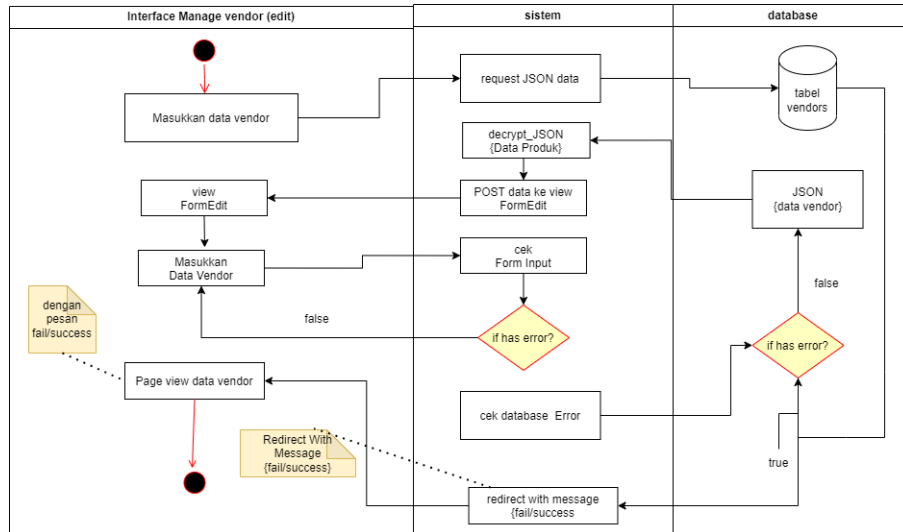
Gambar 3.9 Activity Diagram Proses Dalam Menghapus Data Inventori

Langkah awal yang harus dilakukan adalah admin memilih data inventori yang akan di edit kemudian, sistem akan melakukan *request* data tersebut dalam bentuk format *JSON* berdasarkan *tabel_produk* kepada *database*. Kemudian sistem akan melakukan *decode* data inventori berupa format *JSON* yang sudah didapatkan agar mempermudah *passing parameter*. *Tabel_produk* pada data tersebut akan dijadikan *key* untuk penghapusan inventori oleh sistem, kemudian *database* akan melanjutkannya sebagai *request* untuk penghapusan data pada tabel. Sebelum sistem mengirimkan *request* hapus data, sistem akan mengirimkan pesan berupa konfirmasi apakah data yang akan dihapus sudah sesuai.



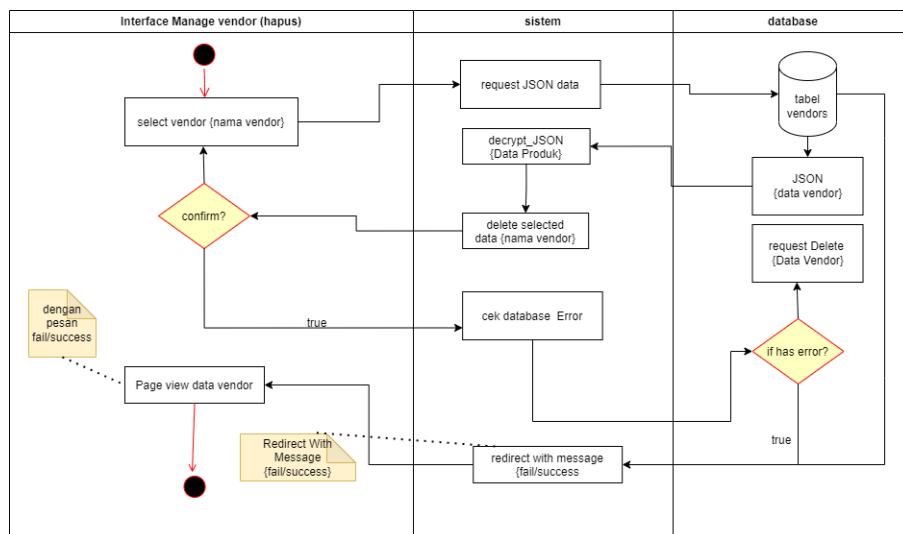
Gambar 3.10 Activity Diagram Manage Vendor Tambah Data

Langkah awal admin diwajibkan mengisi sebuah *form* yang sudah disediakan sesuai dengan data *vendor* yang diminta. Kemudian akan dilakukan pengecekan terhadap form *Input* jika terjadi kesalahan maka akan muncul notif *error message*. Kemudian jika yang dimasukkan sudah sesuai dengan data maka akan muncul tulisan *succes message*.



Gambar 3.11 Activity Diagram Manage Vendor Edit Data

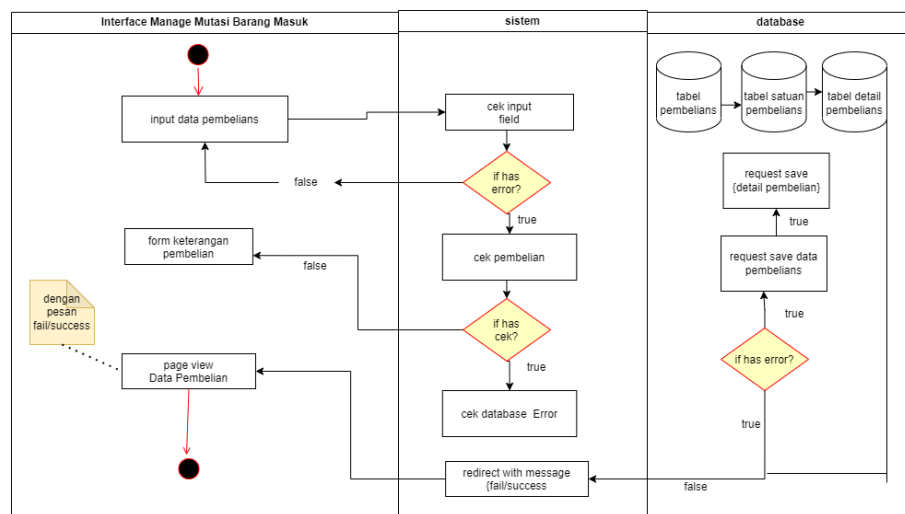
Admin memilih data *vendor* yang akan di edit kemudian sistem akan melakukan *request* data tersebut dalam format *JSON* berdasarkan nama_*vendor* kepada database.



Gambar 3.12 Activity Diagram Manage Vendor Hapus Data

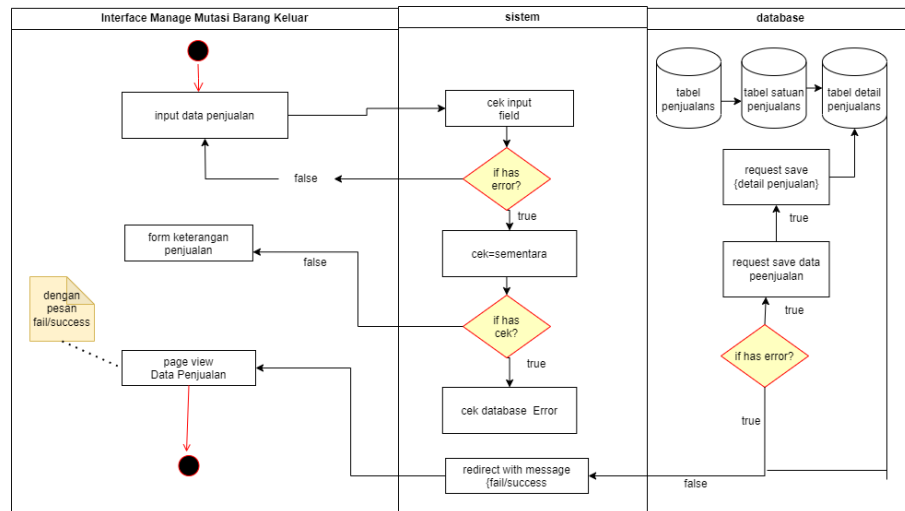
Admin memilih data inventori yang akan diedit kemudian sistem akan melakukan *request* data dalam bentuk forman *JSON* berdasarkan nama_*vendor* yang ada pada

database. Kode_produk pada data tersebut digunakan sebagai *key* untuk menghapus inventori oleh sistem. Setelah *request* data berhasil sistem akan *redirect* ke view data *supplier*.



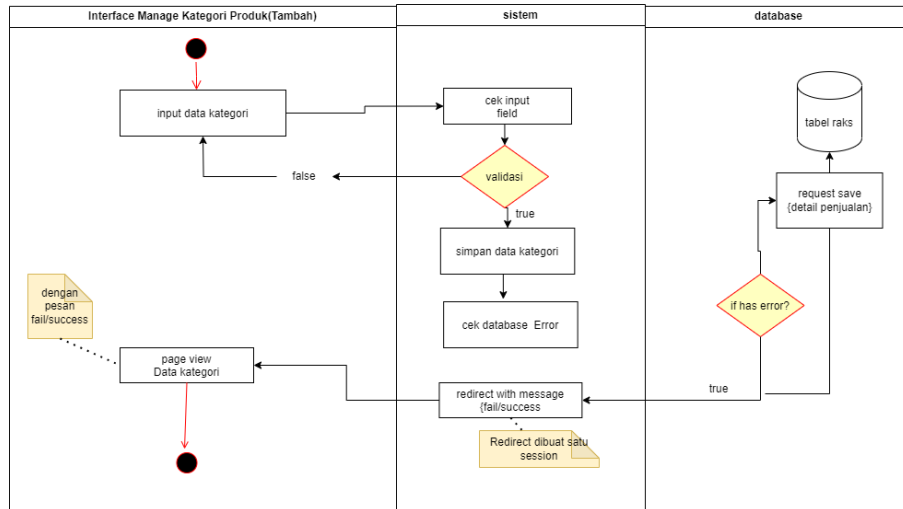
Gambar 3.13 Activity Diagram Mutasi Barang Masuk (Transaksi)

Langkah awal pendataan yaitu dimulai dari input data mutasi barang yang masuk. Kemudian dilakukan pengecekan input untuk mengetahui jika ada kesalahan saat *entry data*.

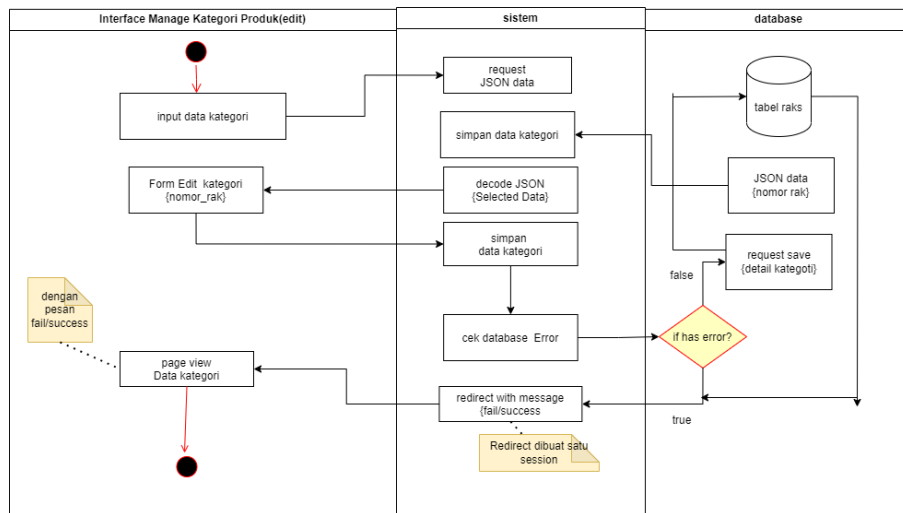


Gambar 3.14 Activity Diagram Mutasi Barang Keluar (Transaksi)

Dimulai dari data input data mutasi barang masuk kemudian dilakukan cek input untuk mengetahui apakah ada kesalahan pada saat entry data. Jika valid sistem akan melakukan pengecekan jenis mutasi barang keluar, jika jenis barang yang keluar berstatus sementara atau disebut *sale* maka sistem akan memproses data kedalam *request* simpan data transaksi dan detail dari transaksi tersebut. Lalu dilakukan pengecekan pada database untuk mengetahui ada error atau tidak.

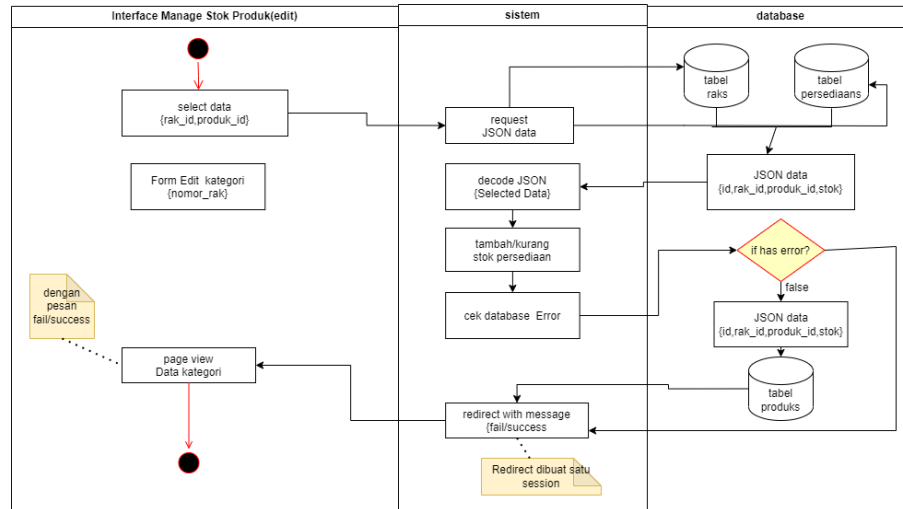


Gambar 3.15 Activity Diagram Manage Kategori Produk Tambah Data
 Proses ini digunakan untuk pengkategorian gudang agar mutasi barang tertata dengan rapi. Langkah pertama yaitu input data gudang kemudian sistem akan melakukan cek pada *form input*. Jika tidak terjadi kesalahan selanjutnya akan dilakukan pengecekan pada database untuk mengetahui ada *error* atau tidak. Kemudian sistem akan mengirimkan *request* simpan data pada database.



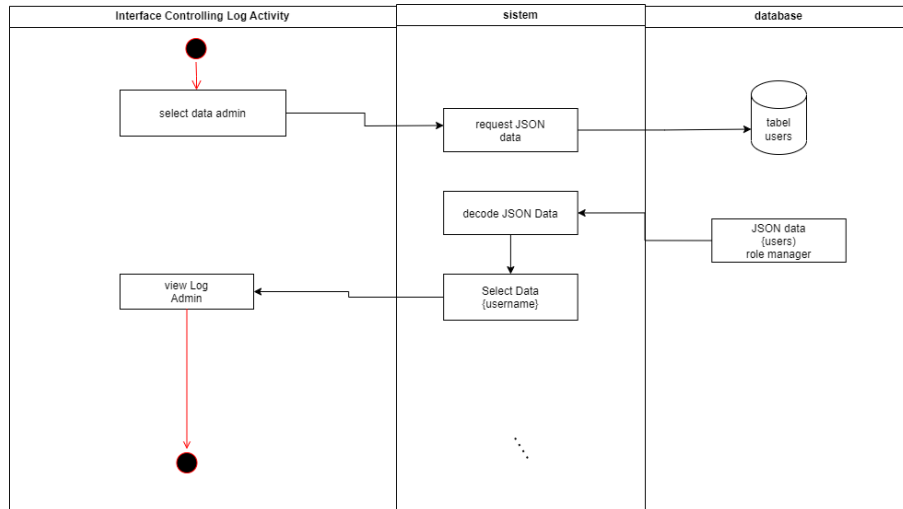
Gambar 3.16 Activity Diagram Manage Kategori Produk Edit Data

Merupakan edit data master gudang. Yang harus dilakukan pertama adalah admin memilih data master gudang yang akan di edit kemudian, sistem akan melakukan sebuah *request* data dalam bentuk format *JSON* berdasarkan nomor_rak yang terdapat pada *database*. Kemudian sistem melakukan *decode* data master gudang berupa format *JSON* yang sudah didapatkan agar mempermudah *passing parameter*. Data yang sudah di *passing* tersebut ditampilkan pada *form* edit, kemudian admin dapat merubah data sesuai dengan *format* yang ada. Sistem melakukan pengecekan *input form* yang didapat apabila terdapat kesalahan penulisan format maka akan muncul *error message* pada *form* edit. Jika data yang dimasukkan sudah *valid* maka akan dilakukan pengecekan tahap berikutnya dari sisi *database* apakah data sudah sesuai dengan *format* dan *type* yang diminta. Jika terdapat kesalahan maka akan *redirect kembali* kedalam *view* data inventori dengan beserta pesan kesalahan *error message*. Dan jika data yang diminta sudah sesuai maka *request* data tersebut akan langsung disimpan kedalam *database* kemudian sistem akan *redirect* ke *view* data inventori disertai dengan *success message*.



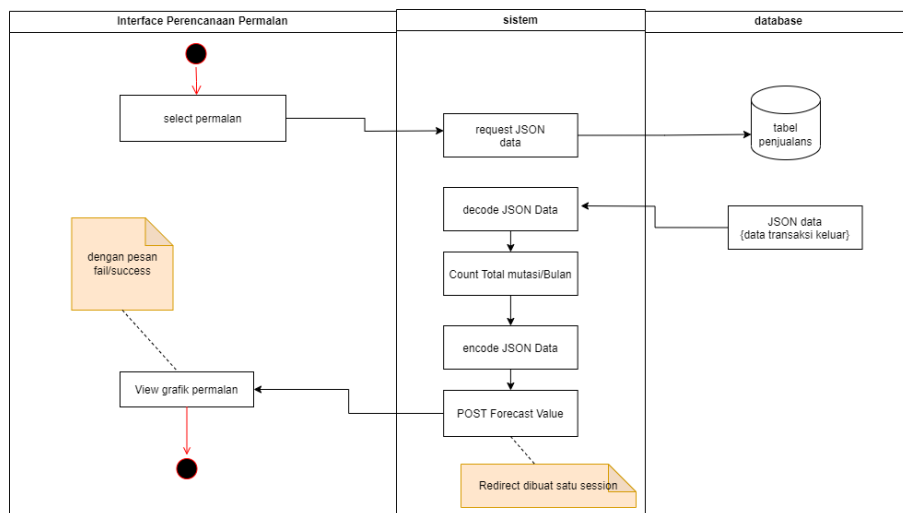
Gambar 3.17 Activity Diagram Manajemen Stock Produk

Proses ini berjalan secara dinamis pada sistem informasi. Artinya, terjadi pertukaran data pada proses transaksi mutasi barang keluar atau masuk yang mempengaruhi jumlah stok gudang. Proses mutasi tersebut akan dikategorikan berdasarkan id_produk pada data persediaan produk di dalam rak. Total stok kategori produk akan diambil berdasarkan hasil dari penghitungan keseluruhan (akumulasi) pada setiap mutasi jumlah stok barang. Total stok gudang tersebut akan disimpan kedalam data master stok gudang.



Gambar 3.18 Activity Diagram Controlling Log Activity

Dilakukan kepala gudang untuk memantau setiap kegiatan atau aktifitas yang dilakukan oleh admin. Langkah awal admin memilih data staff (*key*: kode_admin), kemudian sistem melakukan *request* data *JSON*. Berikutnya, sistem melakukan passing data tersebut kedalam view *Log Admin*.



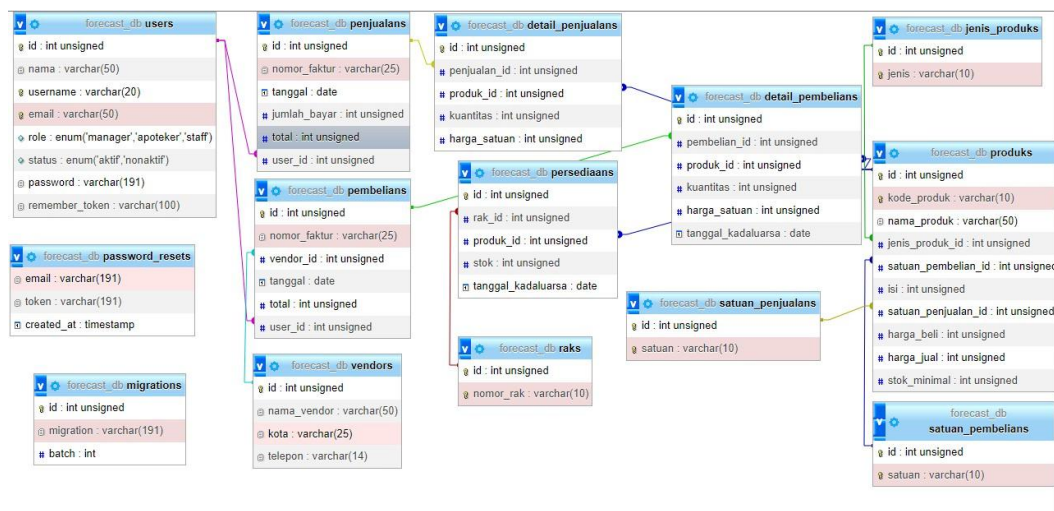
Gambar 3.19 Activity Diagram Perencanaan Pengadaan Stok Gudang (ARRSES)

Activity diagram dari proses input data peramalan dan prediksi, alur dari proses tersebut yaitu, Admin melakukan input parameter yang digunakan yaitu

jumlah data, jumlah iterasi, kemudian diproses di proses prediksi, tetapi sebelum melakukan proses kondisi terlebih dahulu apakah jumlah data lebih besar dari fitur, jika benar maka diproses ke langkah selanjutnya, jika tidak maka kembali pada input parameter tersebut. Setelah diproses kemudian admin dapat melihat hasil prediksi dari peramalan ARRSES.

3.1.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram menjelaskan hubungan antar data dalam basis data yang terdiri dari *object - object* dasar yang mempunyai hubungan atau relasi antar *object- object* tersebut.



Gambar 3 20 ERD

3.1.7 Kamus Tabel

1. Nama Tabel : User

Tabel 3.2 Tabel User

No	Field	Type	Length
1	<u>id</u>	Varchar	15
2	Nama	Varchar	50
3	Username	Varchar	50
4	Email	Varchar	50
5	Role	Varchar	20
6	Password	Varchar (Md5)	50

2. Nama Tabel : Vendors

Tabel 3.3 Vendors

No	Field	Type	Length
1	nama_vendor	Varchar	50
2	kota	Varchar	25
3	telepon	Varchar	14

3. Nama Tabel : Jenis_produk

Tabel 3.4 Tabel Jenis_produk

No	Field	Type	Length
1	jenis	Varchar	10

4. Nama Table : produks

Tabel 3.5 Tabel Produks

No	Field	Type	Length
1	<u>kode_produk</u>	Varchar	10
2	nama_produk	Varchar	50
3	jenis_produk_id	Varchar	50
4	satuan_pembelian_id	Varchar	50
5	harga_beli	Varchar	50
6	harga_jual	Varchar	50
7	stock_minimal	Int	10
8	isi	int	10

5. Nama Tabel : pembelians

Tabel 3.6 Tabel Pemebelians

No	Field	Type	Length
1	<u>id</u>	Int	20
2	nomor_faktur	Varchar	40
3	vendor_id	Int	20
4	tanggal	date	10
5	total	Int	20
6	user_id	int	30

6. Nama Tabel : detail_pembelians

Tabel 3.7 detail_pembelians

No	Field	Type	Length
1	<u>id</u>	varchar	30
2	pembelian_id	Varchar	20

3	produk_id	Varchar	20
4	kuantitas	Int	10
5	harga_satuan	Int	20
6	tanggal_kadaluarsa	date	10

7. penjualans

Tabel 3.8 penjualans

No	Field	Type	Length
1	id	Varchar	30
2	nomor_faktur	Varchar	30
3	tanggal	date	10
4	jumlah_bayar	Int	20
5	user_id	Int	20

8. Nama Tabel : detail_penjualans

Tabel 3.9 detail_penjualans

No	Field	Type	Length
1	id	Int	30
2	penjualan_id	Varchar	30
3	produk_id	Varchar	30
4	kuantitas	Varchar	30
5	harga_satuan	Int	20

9. Nama Tabel : raks

Tabel 3.10 raks

No	Field	Type	Length
1	<u>id</u>	int	30

2	nomor_rak	Varchar	10
---	-----------	---------	----

10. Nama Tabel : persediaans

Tabel 3.11 persediaans

No	Field	Type	Length
1	Id	Int	25
2	rak_id	Varchar	25
3	produk_id	Varchar	25
4	stok	Int	10
5	tanggal_kadaluarsa	date	10

11. Nama Tabel : satuan_penjualans

Tabel 3.12 satuan_penjualans

No	Field	Type	Length
1	id	Int	25
2	satuan	Varchar	10

12. Nama Tabel : satuan_pembelians

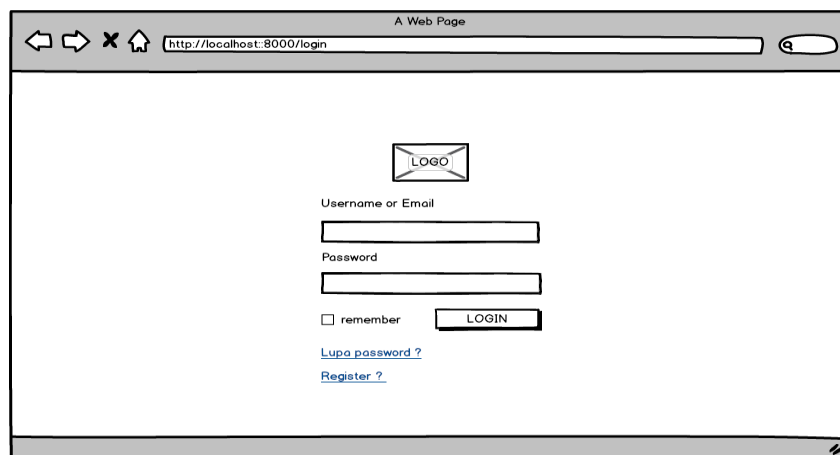
Tabel 3.13 satuan_pembelians

No	Field	Type	Length
1	id	Int	25
2	satuan	Varchar	10

3.1.8 Desain Interface

Tahap ini akan dimulai pembuatan perancangan berbentuk *mock-up* dari sistem atau perangkat lunak yang akan dibangun nanti. Tujuannya dibuat *mock-up* yaitu untuk memvisualisasikan proses bisnis kedalam bentuk realistik. Berikut beberapa desain *interface*.

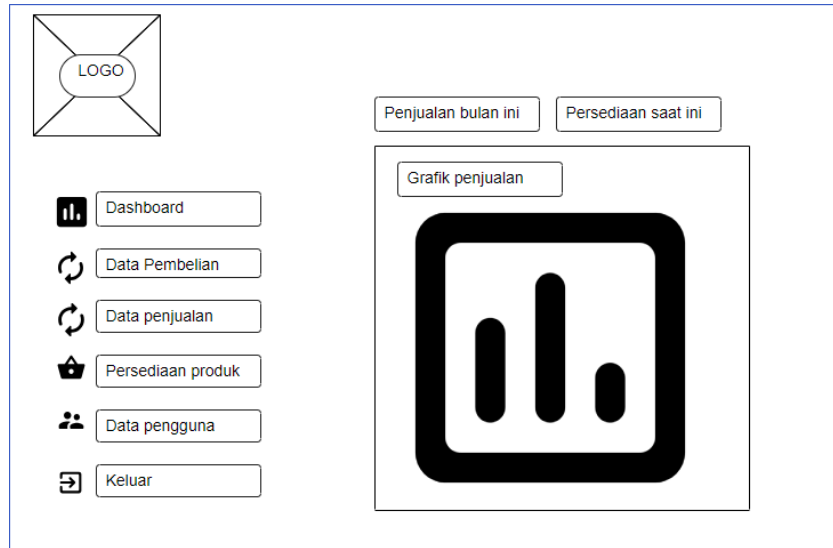
3.1.7.1 Warehouse Admin



The image shows a web browser window with the title "A Web Page". The address bar contains "http://localhost:8000/login". The main content area features a "LOGO" placeholder (represented by an envelope icon), a "Username or Email" input field, a "Password" input field, a "remember" checkbox, a "LOGIN" button, and two links: "Lupa password ?" and "Register ?".

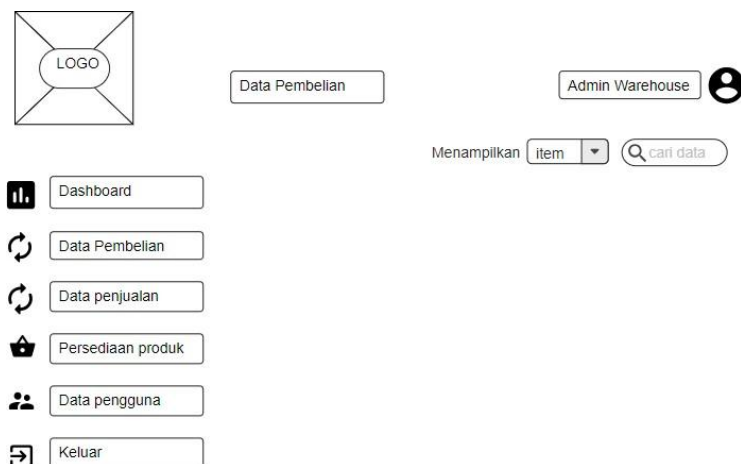
Gambar 3.21 Halaman Login

Pada halaman ini user diwajibkan untuk login dengan mengisi username dan password yang sudah terdaftar dalam sistem. Jika anda belum mempunyai akun anda dapat mengklik tombol register



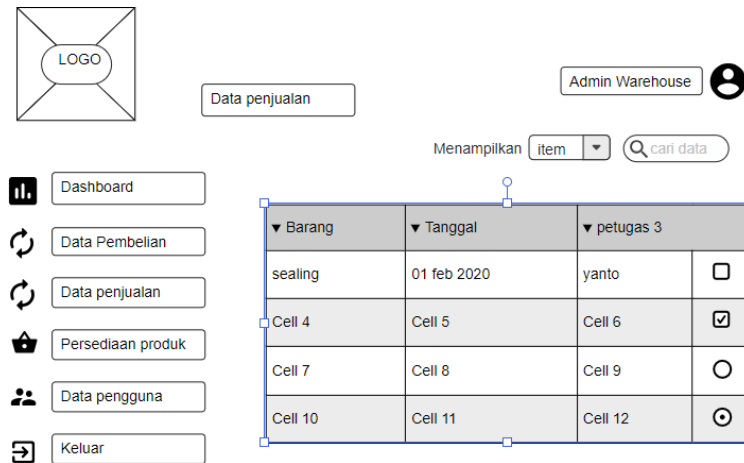
Gambar 3.22 Halaman Dashboard Admin Gudang

Pada halaman ini disajikan beberapa tampilan seperti halaman *dashboard* yang berisi profil singkat admin dan grafik mutasi barang masuk dan keluar. Pada menu ini admin juga dapat melakukan menu apa saja yang diakses antara lain data supplier, mutasi barang, data barang.



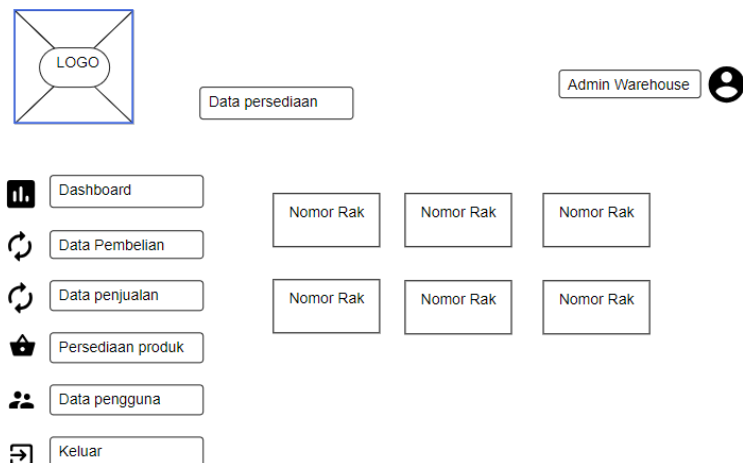
Gambar 3.23 Halaman Data Pembelian

Pada data pembelian digunakan oleh admin gudang . yaitu admin gudang dapat melihat data pembelian yang terjadi di Gudang.



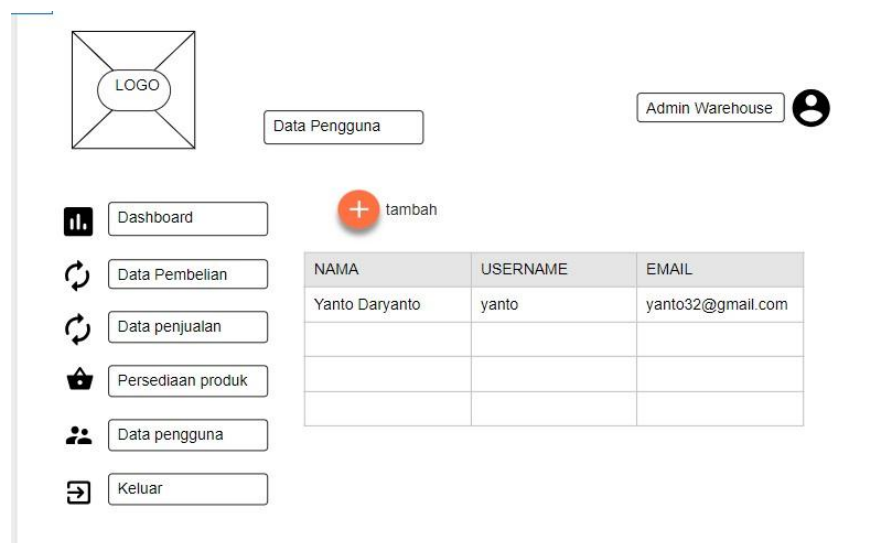
Gambar 3.24 Halaman Data Penjualan

Pada menu ini admin gudang dapat melihat data barang penjualan yang terjadi pada Gudang berdasarkan periode tanggalnya dan admin Gudang juga bisa melihat petugas digudang siapa yang menginputkan data penjualan tersebut



Gambar 3.25 Halaman Data Mutasi Barang

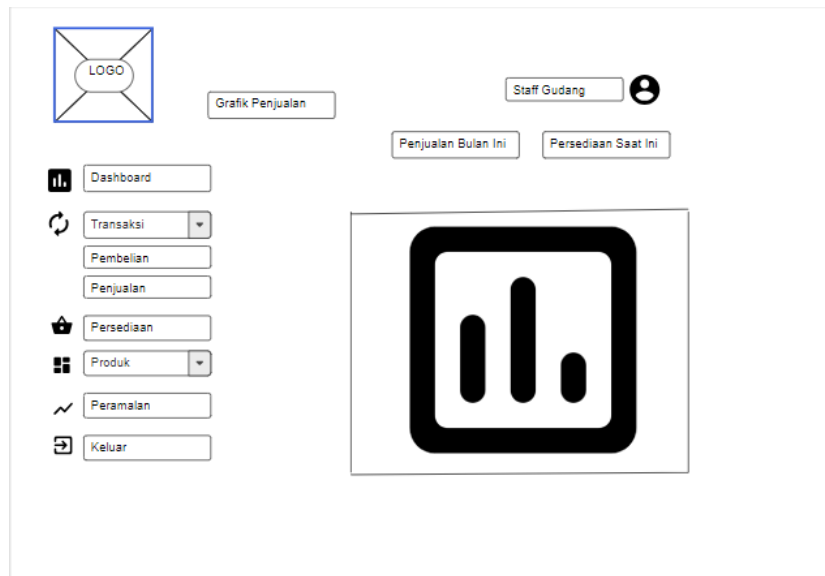
Admin gudang dapat mengontrol barang apa saja yang tersedia pada gudang.



Gambar 3.26 Halaman Data Pengguna

Pada halaman ini admin gudang dapat menambahkan dan menghapus akun *staff* gudang.

3.1.7.2 Opname Staff Warehouse



Gambar 3.27 Halaman Staff Dashboard

Halaman ini akan disajikan beberapa *view* seperti halaman *dashboard* yang berisi beberapa *sections* tentang profil singkat *staff* gudang, menu aplikasi dan beberapa item.

LOGO

Staff Gudang

Data Pembelian

+ Transaksi Baru

Pembelian Barang

Nama Vendor Barang Nomor Faktur Tanggal Pembelian

Produk	Kuantitas	Harga

+ ADD

Gambar 3.28 Halaman Data Pembelian

Pada halaman ini *staff* gudang melakukan input data pembelian barang, pada menu ini *staff* gudang dapat melakukan transaksi pembelian sesuai vendor yang dipilih untuk membeli barang tersebut.

Dashboard

Transaksi

Pembelian

Penjualan

Persediaan

Produk

Peramalan

Keluar

Staff Gudang

Data Penjualan

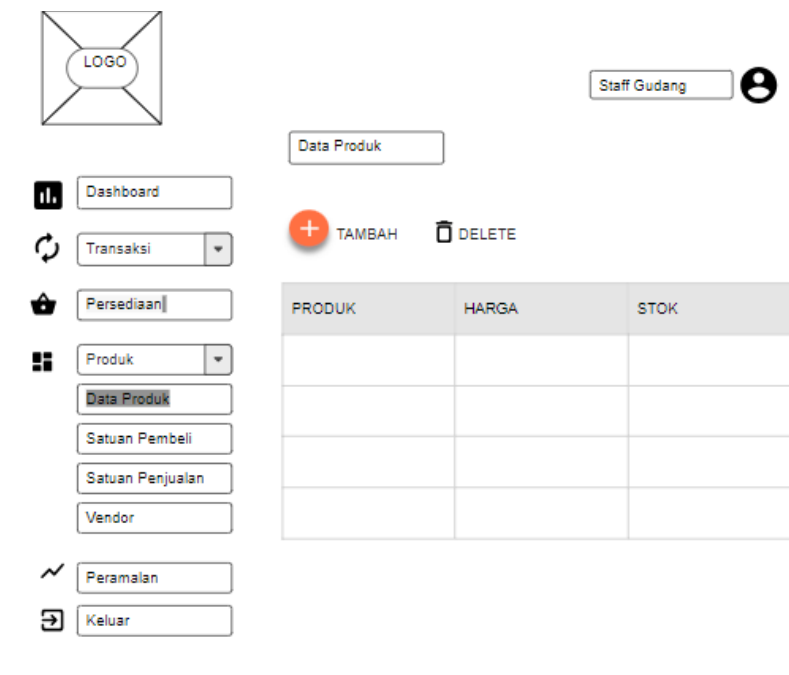
Menampilkan item

cari data

Barang	Tanggal	Petugas

Gambar 3.29 Data Penjualan

Pada halaman ini *staff* gudang menginputkan data barang yang berhasil dijual dari gudang.



Gambar 3.31 Data Produk

Halaman yang menampilkan data produk yang tersedia, *staff* gudang dapat melakukan penambahan barang dan juga bisa menghapus barang.

The screenshot shows a web application interface for managing product types. At the top left is a placeholder for a logo. The top right shows the user's name 'Staff Gudang' and a profile icon. The sidebar menu on the left contains the following items: Dashboard, Transaksi, Persediaan, Produk (with a dropdown arrow), Data Produk, Jenis Produk (highlighted), Satuan Pembeli, Satuan Penjualan, Vendor, Peramalan, and Keluar. The main content area features a 'Data Produk' button, a '+ TAMBAH' button, and a 'DELETED' button. Below these is a table with three columns: JENIS, JUMLAH, and STOK. The table is currently empty.

JENIS	JUMLAH	STOK

Gambar 3.32 Halaman Jenis Produk

Untuk tampilan jenis produk disini *staff* gudang bisa melakukan penambahan kategori barang.

LOGO

Staff Gudang

Satuan Pembelian

+ TAMBAH 🗑️ DELETE

JENIS	JUMLAH	STOK

Dashboard

Transaksi

Persediaan

Produk

Data Produk

Jenis Produk

Satuan Pembelian

Satuan Penjualan

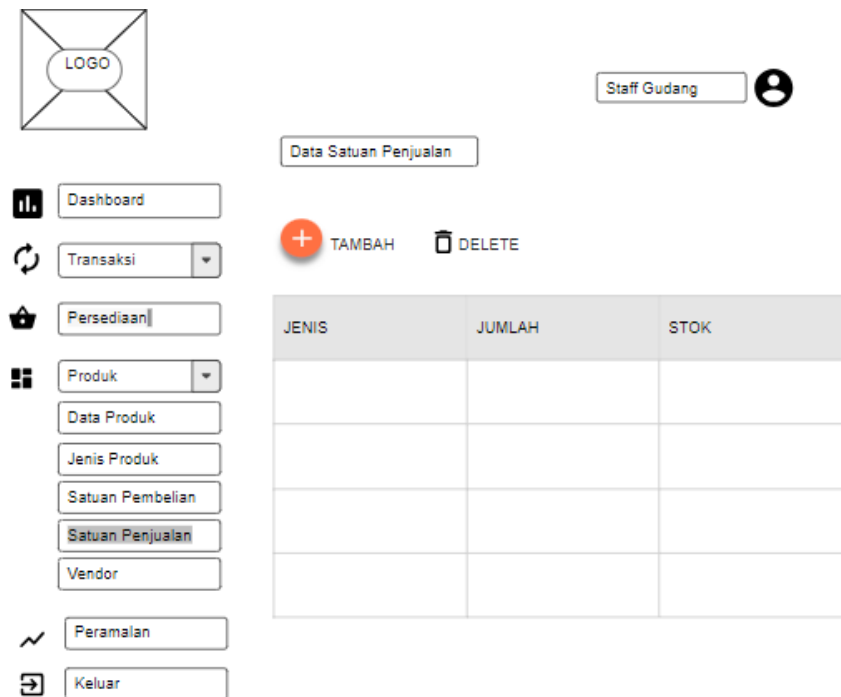
Vendor

Peramalan

Keluar

Gambar 3.33 Halaman Satuan Pembelian

Halaman ini akan disajikan untuk *staff* gudang menambahkan satuan pembelian barang.



LOGO

Staff Gudang

Data Satuan Penjualan

+ TAMBAH DELETE

JENIS	JUMLAH	STOK

Dashboard

Transaksi

Persediaan

Produk

Data Produk

Jenis Produk

Satuan Pembelian

Satuan Penjualan

Vendor

Peramalan

Keluar

Gambar 3.34 Halaman Satuan Penjualan

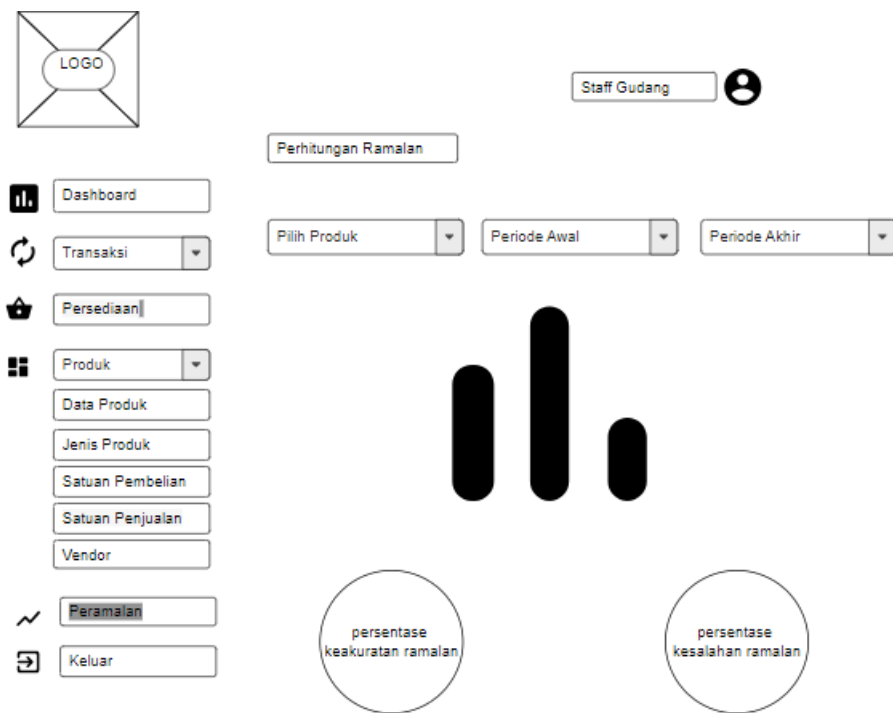
Menu ini berguna untuk menambahkan atau menghapus satuan penjualan yang dilakukan oleh *staff* warehouse.

The screenshot shows a web application interface for Vendor management. On the left is a sidebar with navigation options: Dashboard, Transaksi, Persediaan, Produk, Data Produk, Jenis Produk, Satuan Pembelian, Satuan Penjualan, Vendor, Peramalan, and Keluar. The top header features a logo placeholder and a user profile labeled 'Staff Gudang'. The main content area has a 'Data Vendor' section with '+ TAMBAH' and 'DELETE' buttons. Below this is a table with three columns: Nama Vendor, Asal Kota, and No telepon.

Nama Vendor	Asal Kota	No telepon

Gambar 3.35 Halaman Vendor

Pada bagian ini *staff* gudang bisa menginputkan data vendor



Gambar 3.36 Halaman Peramalan

Halaman yang digunakan untuk manajemen pengadaan barang pada periode mendatang. *Staff* hanya perlu memilih data barang pada *selectbox* pilih barang dan memilih periode yang akan diramalkan.