SKPL-0103

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Raniah Food Application

**(RFA)**

Untuk:

Raniah Food Creation

Dipersiapkan oleh:

Ade Naufal 1301150005

Ariq N. Wibisono 1301160128

Kinegar H. 1301160373

M. Alfarizky S. 1301164188

Rizki Achmad R. 1301160509

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| **Program Studi S1 Teknik Informatika**  **-**  **Fakultas Informatika**  https://lh3.googleusercontent.com/6vpG7AfyJShEr6VhERtbiRizCnn0dq3evygcV7HZcqIGiAPIvvc0Ka8f4ldooLYlm_Z2JmAWnIy_nDfNXPa4eCweJEgOkaEc-RPrRf89KnjX2cuZ4ZqEPDKW6HuVtdgrcIMaIr7bGU5nu96Mkg | **Nomor Dokumen** | | **Halaman** |
| ***SKPL-*0103** | | **47** |
| **Revisi** | ***D*** | ***Tgl: 30/04/2018*** |

# Daftar Perubahan

|  |  |
| --- | --- |
| **Revisi** | **Deskripsi** |
| **A** | Bab 1.2 Konvensi Dokumen  Bab 2.2 Fungsi Produk  Bab 2.3 Kelas dan Karakteristik Pengguna  Bab 2.6 Dokumentasi Pengguna  Bab 2.7 Asumsi dan Dependensi  Bab 3.1 Antarmuka Pengguna  Bab 4 Usecase  Bisnis Proses |
| **B** | Bab 2.2 Fungsi produk  Bab 2.3 Kelas dan Karakteristik Pengguna  Bab 3.3 Antarmuka Perangkat Lunak  Bab 3.4 Antarmuka Komunikasi  Bab 4 Usecase |
| **C** | Bab 4 Model Domain  Bab 5 Usecase  Lamipran B: Flow Map |
| **D** | Bab 2.3  Bab 2.7  Bab 3.1  Proses Bisnis  Aturan Bisnis  Bab 4 Model Domain  Bab 5 Usecase |
| **E** |  |
| **F** |  |
| **G** |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | A | B | C | D | E | F | G |
| TGL | 05/03/2018 | 04/04/2018 | 23/04/2018 | 30/04/2018 |  |  |  |
| Ditulis oleh | Kelompok 5 | Kelompok 5 | Kelompok 5 | Kelompok 5 |  |  |  |
| Diperiksa oleh | Astri Cahyaningtyas | Astri Cahyaningtyas | Astri Cahyaningtyas | Astri Cahyaningtyas |  |  |  |
| Disetujui oleh | Veronikha Efendy | Veronikha Efendy | Veronikha Efendy | Veronikha Efendy |  |  |  |

# Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| **Halaman** | **Revisi** | **Halaman** | **Revisi** |
| 6  7  8  10-30  34  7-8  9-10  12-37  11  12-24  9  10-11  11  14-15  16-43  43-44  44 | A  A  A  A  A  B  B  B  C  C  D  D  D  D  D  D  D |  |  |

# Daftar Isi

[Daftar Perubahan 2](#_Toc512854669)

[Daftar Halaman Perubahan 3](#_Toc512854670)

[Daftar Isi 4](#_Toc512854671)

[Daftar Tabel 6](#_Toc512854672)

[Daftar Gambar 7](#_Toc512854673)

[1. Pendahuluan 8](#_Toc512854674)

[1.1 Tujuan Penulisan Dokumen 8](#_Toc512854675)

[1.2 Konvensi Dokumen 8](#_Toc512854676)

[1.3 Cakupan Produk 8](#_Toc512854677)

[1.4 Referensi 8](#_Toc512854678)

[2. Overall Description 9](#_Toc512854679)

[2.1 Perspektif Produk 9](#_Toc512854680)

[2.2 Fungsi Produk 9](#_Toc512854681)

[2.3 Kelas dan Karakteristik Pengguna 9](#_Toc512854682)

[2.4 Lingkungan Operasi 10](#_Toc512854683)

[2.5 Batasan Perancangan dan Implementasi 10](#_Toc512854684)

[2.6 Dokumentasi Pengguna 10](#_Toc512854685)

[2.7 Asumsi dan Dependensi 10](#_Toc512854686)

[3. Requirements Antarmuka Eksternal 11](#_Toc512854687)

[3.1 Antarmuka Pengguna 11](#_Toc512854688)

[3.2 Antarmuka Perangkat Keras 11](#_Toc512854689)

[3.3 Antarmuka Perangkat Lunak 11](#_Toc512854690)

[3.4 Antarmuka Komunikasi 12](#_Toc512854691)

[4. Model Domain 13](#_Toc512854692)

[5. Fitur Sistem (Use Cases) 15](#_Toc512854693)

[5.1 Use Case Registrasi 15](#_Toc512854694)

[5.2 Use Case Login 17](#_Toc512854695)

[5.3 Use Case Input Data Bahan 19](#_Toc512854696)

[5.4 Use Case Input Data Produksi 21](#_Toc512854697)

[5.5 Use Case View Data Produksi 23](#_Toc512854698)

[5.6 Use Case Input Data Penjualan 24](#_Toc512854699)

[5.7 Use Case Logout 26](#_Toc512854700)

[5.8 Use Case Delete Data Bahan 27](#_Toc512854701)

[5.9 Use Case Delete Data Produksi 28](#_Toc512854702)

[5.10 Use Case Delete Data Penjualan 30](#_Toc512854703)

[5.11 Use Case Update Data Bahan 31](#_Toc512854704)

[5.12 Use Case Update Data Produksi 33](#_Toc512854705)

[5.13 Use Case Update Data Penjualan 35](#_Toc512854706)

[5.14 Use Case View Data Bahan 37](#_Toc512854707)

[5.15 Use Case Kelola Laporan Penjualan 38](#_Toc512854708)

[5.16 Use Case Kelola Laporan Keuangan 39](#_Toc512854709)

[6. Requirements Nonfungsional Lainnya 42](#_Toc512854710)

[6.1 Requirements Performa 42](#_Toc512854711)

[6.2 Requirements Keselamatan 42](#_Toc512854712)

[6.3 Requirements Keamanan 42](#_Toc512854713)

[6.3 Atribut Kualitas Perangkat Lunak 42](#_Toc512854714)

[7. Requirements Lain 42](#_Toc512854715)

[7.1 Requirment Database 42](#_Toc512854716)

[7.2 Requirement Hak Cipta 43](#_Toc512854717)

[Lampiran A: Daftar Kata-Kata Sukar 43](#_Toc512854718)

[Lampiran B: Analysis Models 43](#_Toc512854719)

[Proses Bisnis 43](#_Toc512854720)

[Aturan Bisnis 44](#_Toc512854721)

[Flow Map 45](#_Toc512854722)

[Pemesanan Barang 45](#_Toc512854723)

[Penjualan Barang 45](#_Toc512854724)

[Produksi Barang 47](#_Toc512854725)

# Daftar Tabel

[Tabel 1-Flow Map Pemesanan Barang 45](#_Toc512849648)

[Tabel 2-Flow Map Penjualan Barang 46](#_Toc512849649)

[Tabel 3-Flow Map Produksi Barang 47](#_Toc512849650)

# Daftar Gambar

[Gambar 1-Skema LAN 12](#_Toc512849673)

[Gambar 2-Entity Relationship Diagram 13](#_Toc512849674)

[Gambar 3-Class Diagram 14](#_Toc512849675)

[Gambar 4-Use Case 15](file:///C:\Users\THOSIBA\Downloads\5b.%20Template%20SKPL%20Analisis%20Berorientasi%20Objek%20revisi%204.docx#_Toc512849676)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen SKPL ini dibuat dengan tujuan sebagai bahan pemodelan untuk pembuatan aplikasi RFA yang dapat membantu perusahaan Raniah Food Creation. Diharapkan aplikasi RFA dapat membantu para pegawai perusahaan Raniah Food Creation dalam mengatur dan merapikan data-data yang ada pada perusahaan tersebut dengan lebih efisien.

## Konvensi Dokumen

Requirement yang ada pada aplikasi ini adalah sebagai berikut di mana kata/kalimat yang ditebalkan adalah requirement utama dan yang tidak adalah requirement untuk menunjang requirement utama:

* Aplikasi menggunakan database
* Input data karyawan, produk, bahan, keuangan, pelanggan, dan penjualan
* Edit data karyawan, produk, bahan, keuangan, pelanggan, dan penjualan
* Delete data karyawan, produk, bahan, keuangan, pelanggan, dan penjualan
* Pembagian peran karyawan
* Pembagian akses data berdasarkan peran karyawan
* Menggunakan sorting data
* Menggunakan pencarian data
* Dapat mengkalkulasi pemasukan dan pengeluaran keuangan
* Menggunakan fitur login dan logout

## Cakupan Produk

Dokumen ini mencakup pembuatan sebuah aplikasi yang dapat mencatat, memproses, dan mengatur data pada perusahaan. Aplikasi ini menggunakan sebuah database untuk menyimpan data tersebut. Data tersebut akan diolah supaya dapat mempermudah perhitungan keuangan yang ada pada perusahaan. Aplikasi juga dapat mengurutkan dan mencari data-data sebagai masukan untuk perusahaan itu sendiri.

## Referensi

Referensi yang digunakan dalam pembuatan laporan SKPL ini adalah:

* Template SKPL S1 Teknik Informatika - Universitas Telkom 2017/2018
* Slide mata kuliah APPL S1 Teknik Informatika - Universitas Telkom 2017/2018
* Buku “Software Engineering a Practitioner’s Approach” Edisi 8 karangan Roger S. Pressman Ph.D. dan Bruce R. Maxim Ph.D yang diterbitkan oleh Mc Graw Hill di New York 2014
* Bpk. Kiki - Pemilik Perusahaan Raniah Food Creation Cabang Guntursari Wetan.

# Overall Description

## Perspektif Produk

Program ini merupakan produk yang disengaja ada dan baru untuk kebutuhan perusahaan dalam mengelola data internal pada perusahaan. Program tidak terhubung dengan sistem lainnya dikarenakan belum ada sistem yang dipakai oleh perusahaan tersebut.

## Fungsi Produk

Toko memerlukan pembukuan agar dapat menghitung aliran pemasukan dan pengeluaran. Tiap bidang harus membuat laporan harian yang akan dimasukan ke dalam pembukuan. Namun diperlukan pengelolaan data dari pembukuan yang akan menjadi laporan keuangan. Oleh karena itu, direncanakanlah perangkat lunak yang ditulis di SKPL ini. Program ini berfungsi sebagai alat memantau pengelolaan data internal perusahaan. Selain itu, program ini juga berfungsi sebagai tempat penyimpanan data-data yang penting bagi perusahaan.

Hak akses dibagi berdasarkan status karyawan yaitu admin, koki, dan sales. Status karyawan ditentukan oleh admin dan pengaksesan diperlukan konfirmasi identitas yang dilakukan melalui sistem.

## Kelas dan Karakteristik Pengguna

Terdapat tiga pengguna yaitu admin, koki, dan sales. Admin yang merupakan lulusan minimal S1 dengan pengalaman organisasi bertugas menangani administrasi perusahaan seperti mengatur data keuangan, penjualan, hasil produksi, dan bahan. Admin memiliki hak akses laporan keuangan, penjualan, dan data karyawannya.Koki yang merupakan lulusan minimal SMA atau sederajat dengan pengalaman memasak bertugas untuk mencatat bahan, memasak bahan menjadi produk siap dan mencatatnya lalu diantarkan oleh sales. Koki memiliki hak akses untuk memasukkan, mengubah, dan menghapus data bahan dan data produksi.Sedangkan sales yang merupakan lulusan minimal SMA atau sederajat yang berpengalaman dalam hal penjualan bertugas menangani penjualan produk dan mengirimkan produk dari koki ke pelanggan sesuai pesanan. Sales memiliki hak akses memasukan, mengubah, dan menghapus dan data penjualan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Admin** | **Koki** | **Sales** |
| Mengakses laporan keuangan dan Memasukkan/mengubah/menghapus data karyawan | ✓ |  |  |
| Mengakses data bahan dan data produksi |  | ✓ |  |
| Memasukkan/mengubah/menghapus data bahan dan data produksi |  | ✓ |  |
| Mengakses laporan penjualan | ✓ |  | ✓ |
| Memasukkan/mengubah/menghapus & data penjualan |  |  | ✓ |

## Lingkungan Operasi

Perangkat lunak akan beroprasi di lingkungan toko. Platform yang digunakan adalah komputer *desktop* atau *laptop* dengan sistem operasi minimal Windows 7. Program akan berjalan bersama MySQL untuk kepentingan menyimpan data.

## Batasan Perancangan dan Implementasi

Program hanya sebatas mengolah data. Program tidak dapat menangani penerimaan pesanan langsung dari pelanggan. Program hanya mencakup satu cabang dan tidak dapat terhubung dengan cabang yang lain. Pengguna hanya bisa digunakan oleh karyawan dan admin di area tersebut.

## Dokumentasi Pengguna

* Registrasi (Admin Only)
  + Buka halaman untuk registrasi
  + Isikan data pada tempat yang disediakan
  + Upload foto dengan ukuran 3x4
  + Pilih role/posisi pekerjaan
  + Klik tombol “Submit”
* Pemesanan Bahan
  + Buka Halaman pemesanan
  + Pilih bahan yang ingin dibeli
  + Masukkan jumlah stock yang sudah dibeli
  + Klik tombol “beli”
* Input Produk (Koki Only)
  + Buka halaman produksi.
  + Jika produk sudah ada, cukup isikan nama produk dan stock yang dibuat
  + Jika produk belum ada, isi form produk yang ada pada halaman produksi

## Asumsi dan Dependensi

Asumsi dalam sistem ini yakni sales mendapat data penjualan langsung dari pelanggan. Sales, admin, koki hanya dapat menginputkan, mengubah, melihat dan mendelete data seusai hak akses. Sales yang menerima pesanan dan mengantar hasil produksi dapat dilakukan oleh orang yang sama.

Dependensi dalam sistem ini yakni. koki akan memproduksi barang jika stock produk mau habis dan stock bahan yang diperlukan mencukupi. Sales dapat mengirim barang ke pelanggan apabilia stock produk mencukupi. Karyawan memiliki hak akses di dalam program setelah admin menginputkan datanya pada registrasi.

# Requirements Antarmuka Eksternal

## Antarmuka Pengguna

Pada aplikasi RFA ini akan terdapat 17 halaman. Halaman pertama yang akan dibuka adalah login disini pengguna dapat melakukan login sesuai dengan identitasnya. Lalu pengguna dapat masuk hanya pada halaman yang diberikan hak akses padanya. Admin dapat mengakses halaman input,delete,view registrasi dimana disana ia dapat mengatur data administrasi. Admin juga dapat mengakses halaman view bahan, laporan penjualan dan laporan keuangan. Sales dapat mengakses halaman input,view,delete penjualan dimana ia dapat mengatur data penjualan. Sales juga dapat mengakses halaman view produk. Koki dapat mengakses halaman input, view, delete bahan dan produk. Pada setiap halaman akan tercantum tombol logout.

## Antarmuka Perangkat Keras

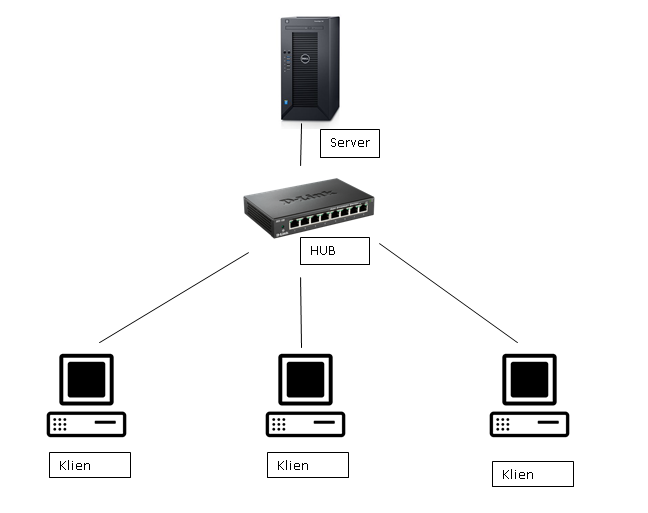
Untuk menjalankan aplikasi RFA dibutuhkan sebuah komputer/laptop yang tersambung dengan perusahaan. Perangkat keras yang membantu penggunaan aplikasi adalah *keyboard* dan *mouse*. *Keyboard* digunakan untuk membantu memasukkan data pada aplikasi. Sedangkan *mouse* digunakan untuk membantu menggunakan aplikasi itu sendiri.

## Antarmuka Perangkat Lunak

Aplikasi RFA hanya dapat dijalankan pada sistem operasi Windows. Ini disebabkan semua komputer yang ada pada perusahaan tersebut menggunakan sistem operasi Windows. Inilah alasan utama mengapa aplikasi ini hanya dapat dijalankan pada sebuah komputer/laptop. Dipilihnya perangkat komputer karena dapat menjalankan server internal yang memerlukan spesifikasi komputer yang tinggi. Hal ini dapat mempermudah pemasangan karena komputer tidak berpindah-pindah

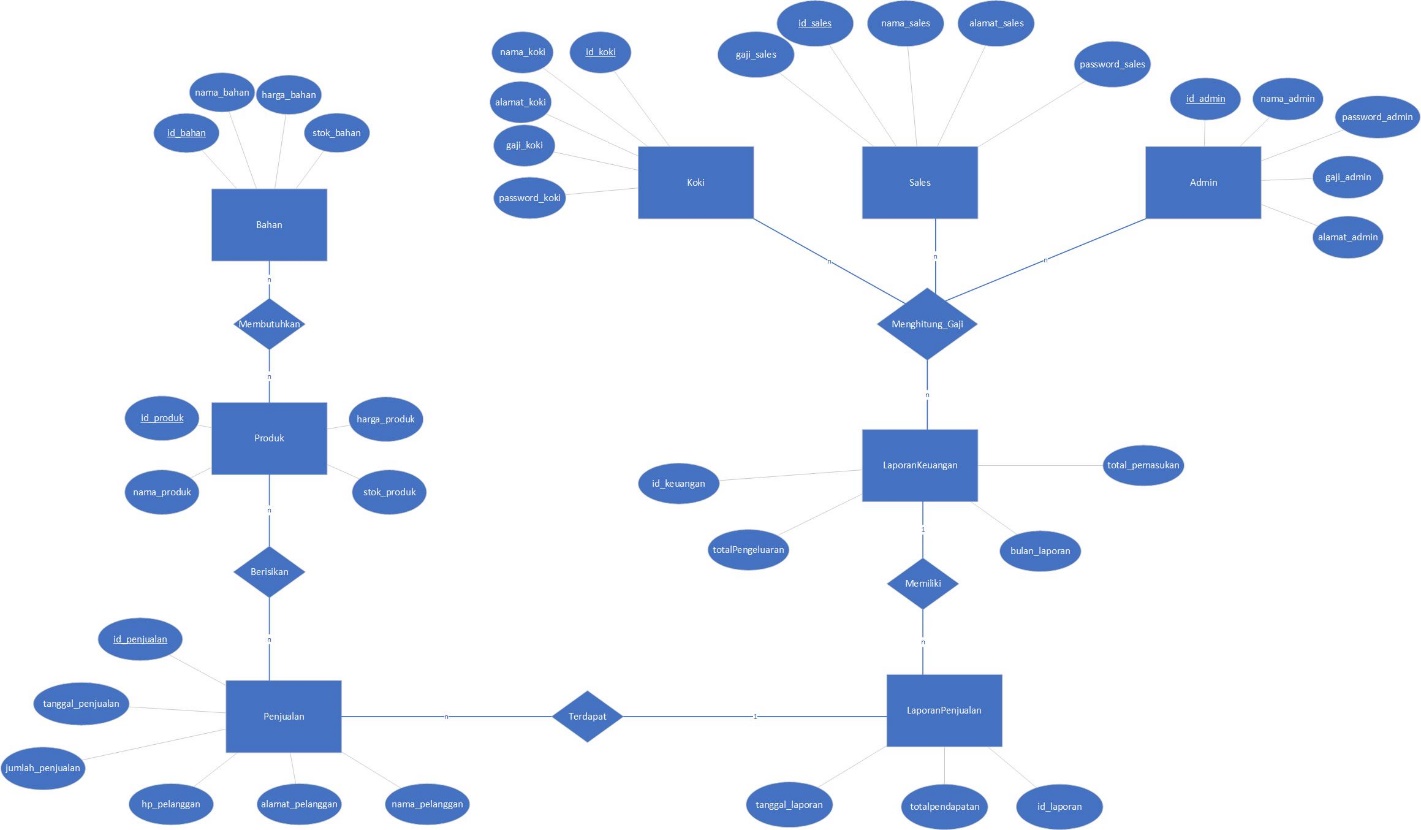
## Antarmuka Komunikasi

Aplikasi ini menggunakan jaringan lan untuk pemindahan data. Ini disebabkan karena pengguna hanya dapat menggunakan aplikasi pada tempat perusahaan tersebut. Sehingga pengguna tidak dapat menggunakan aplikasi apabila diluar jangkauan jaringan lokal perusahaan. Jaringan akan dipusatkan ke komputer tersendiri yang bertindak sebagai server. Koputer lainnya akan bertindak sebagai klien yang diakses oleh karyawan. Jaringan akan terhubung dengan kabel LAN. Berikut contoh skema LAN.

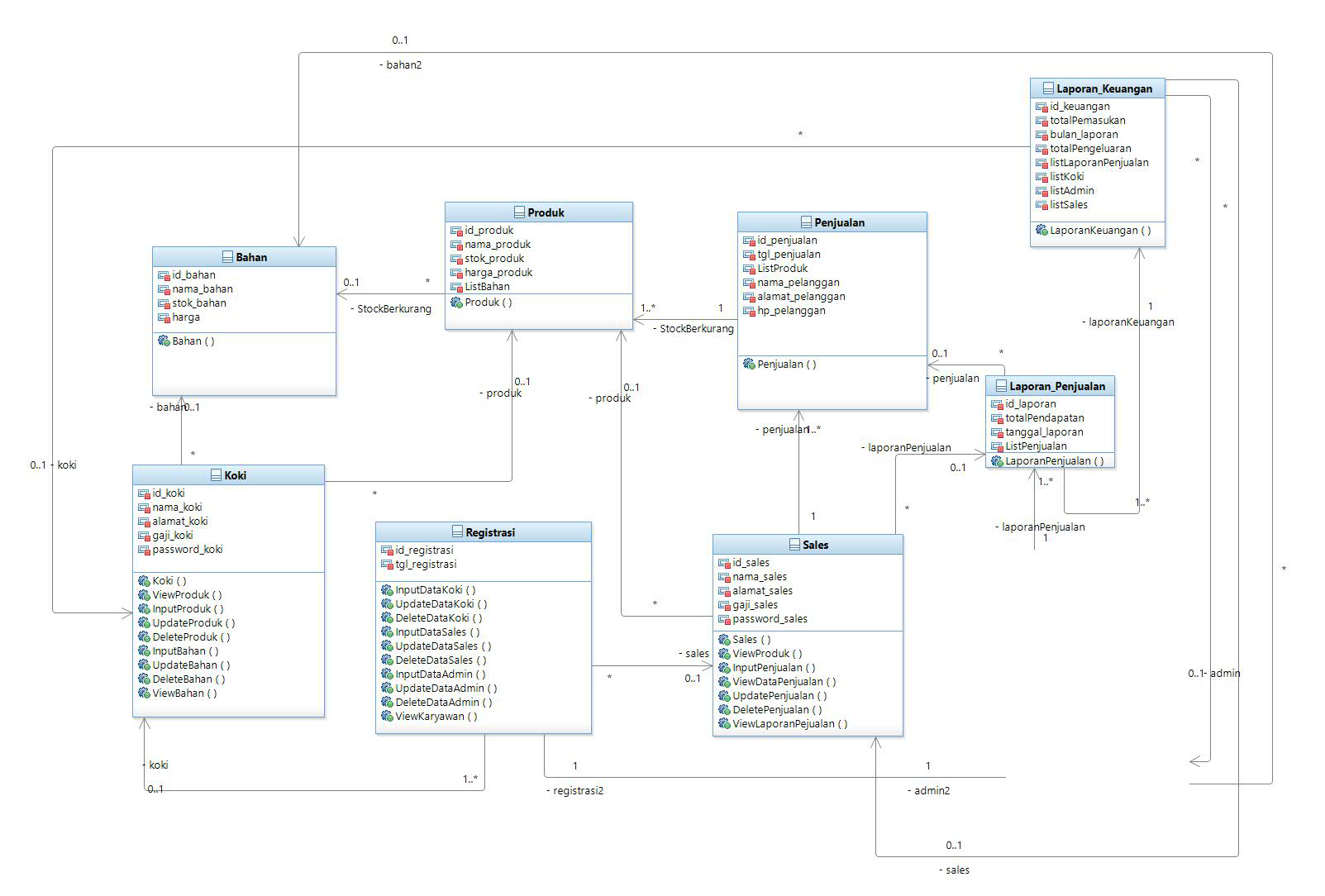


Gambar -Skema LAN

# Model Domain

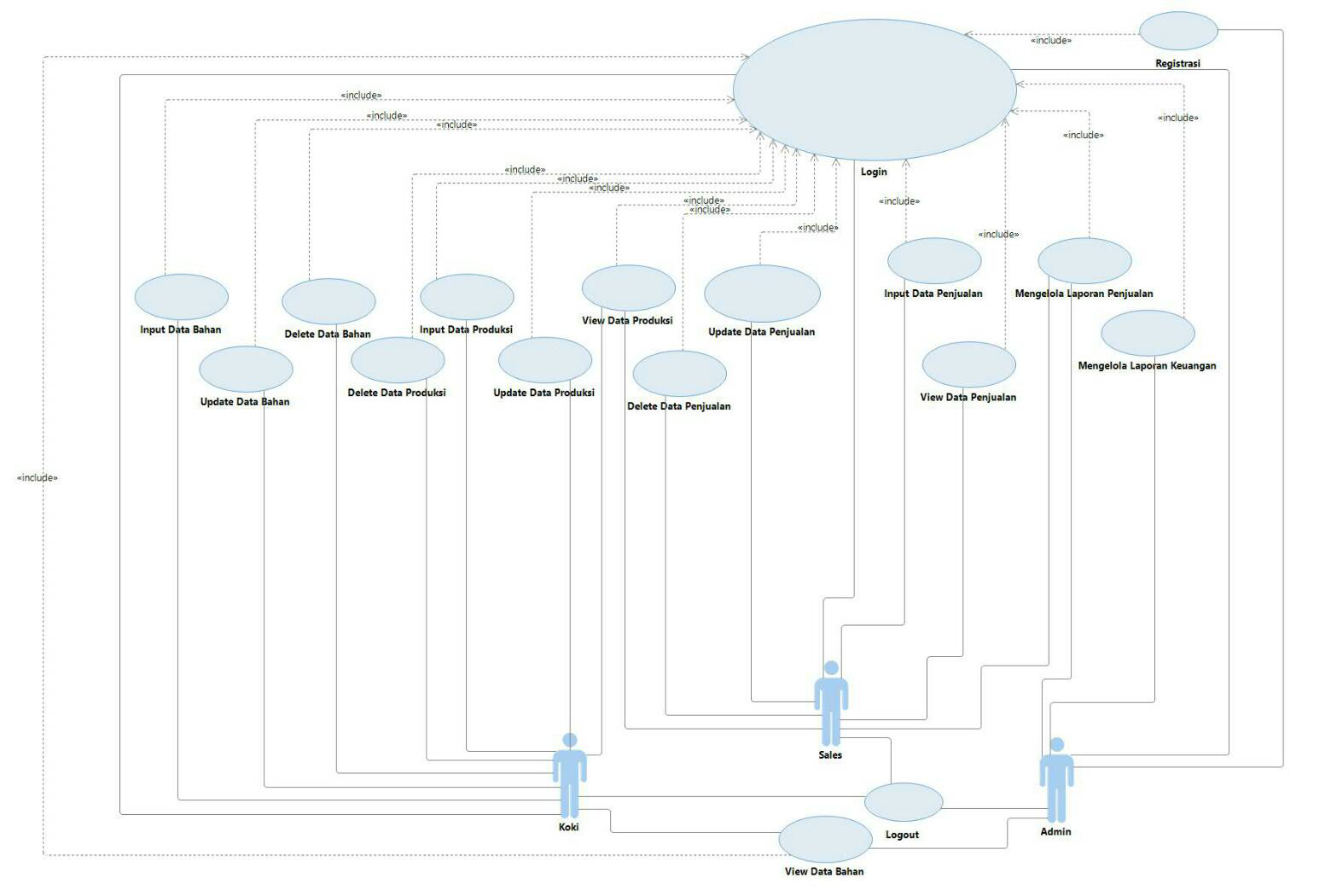


Gambar -Entity Relationship Diagram



Gambar -Class Diagram

# Fitur Sistem (Use Cases)



Gambar -Use Case Diagram

## Use Case Registrasi

|  |  |
| --- | --- |
| Nama Use Case: | Registrasi |
| Tujuan: | Memasukkan data karyawan pada database, sehingga karyawan dapat login dan menggunakan aplikasi. |
| Input: | Data karyawan (sales/koki/admin) |
| Output: | - |
| Skenario Utama: | Admin memasukkan data karyawan. Sistem menerima dan menyimpan data karyawan ke dalam database. |
| Aktor: | Admin |
| Prakondisi: | Admin ingin menginputkan data diri dan admin sudah melakukan login. |
| Langkah-langkah:   |  |  | | --- | --- | | Aktor | Sistem | | 1. Admin sudah melakukan login |  | | 2. Admin memasukkan data sales/koki/admin berupa ID Pegawai, Nama, dan alamat pegawai pada form registrasi serta memilih *role*. |  | | 3. Admin menekan tombol submit saat selesai memasukkan data karyawan pada form. |  | |  | 4. Sistem mengecek pada salah satu atau beberapa field data (seperti nama dan ID pegawai) yang telah diisi dan membandingkan dengan yang ada di database, apakah sudah ada atau belum. Jika sudah ada, maka sistem memberi peringatan berupa pesan kepada admin. | |  | 5. Sistem menerima dan menyimpan data karyawan dan menampilkan kembali data yang telah diinput kepada admin. | | |
| Pascakondisi: | Data\_Karyawan sudah terdaftar di dalam perusahaan |
| Prioritas: | Sedang |
| Jika Dapat Digunakan: | Tiga kali percobaan |
| Frekuensi Digunakan: | Rutin sekali setiap hari |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Pada langkah ke-4, jika nama sudah terdaftar maka akan kembali ke langkah ke 1. |
| Skenario Eksepsional: | Apabila terjadi gangguan pada sebelum langkah 4 selesai, maka sistem tidak akan menyimpan perubahan yang terjadi. |
| Contoh: | Admin memasukkan data karyawan dalam sebuah form. Kemudian sistem mengecek data karyawan dan setelah dicek belum ada. Sistem lalu memasukkan data karyawan pada database dan kemudian menampilkan data karyawan yang sebelumnya dimasukkan dalam preview mode. |
| Isu Terbuka: | 1. Bagaimana kemampuan sekuritas saat registrasi? 2. Apakah sistem submisi form menggunakan enkripsi atau cara lain? |

## Use Case Login

|  |  |
| --- | --- |
| Nama Use Case: | Login |
| Tujuan: | Memberi hak akses kepada user untuk masuk ke dalam sistem dengan mengecek identitas user sesuai dengan yang ada di database. |
| Input: | Id\_admin/sales/koki |
| Output: | - |
| Skenario Utama: | Admin/Sales/Koki memasukkan ID dan password. Sistem mengecek apakah karyawan tersebut ada atau tidak. Jika ada, maka karyawan akan dapat masuk ke dalam aplikasi sebagai user dengan ID karyawan yang dimaksud. |
| Aktor: | Admin/Sales/Koki |
| Prakondisi: | Admin/Sales/Koki belum melakukan login, sehingga tidak mendapat hak akses selain melakukan login |
| Langkah-langkah:   |  |  | | --- | --- | | Aktor | Sistem | | 1. Admin/Sales/Koki memasukkan ID mereka beserta password |  | |  | 2. Sistem mengecek dan memastikan apakah ID karyawan dan password yang dimaksud ada pada sistem | |  | 3. Jika ada, maka karyawan akan diberi akses masuk sebagai user dengan ID karyawan yang terhubung | | |
| Pascakondisi: | Admin/Sales/Koki mendapat hak akses untuk menggunakan aplikasi sesuai role masing-masing. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Tiga kali percobaan |
| Frekuensi Digunakan: | Rutin sekali setiap hari |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Pada langkah ke 2, jika id atau password salah maka akan kembali ke langkah ke 1. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 3, maka sistem akan mengentikan proses login. |
| Contoh: | Sales memasukkan ID karyawan dan password miliknya. Sistem mengecek ID karyawan-nya dan ada. Sistem memberi akses kepada sales. Sales dapat masuk ke dalam aplikasi. |
| Isu Terbuka: | 1. Verifikasi login menggunakan metode sekuritas apa? |

## Use Case Input Data Bahan

|  |  |
| --- | --- |
| Nama Use Case: | Input Data Bahan |
| Tujuan: | Memasukkan Data\_Bahan pada database |
| Input: | Data Bahan Produksi |
| Output: | - |
| Skenario Utama: | Koki memasukkan data bahan pada sistem. Sistem mengecek apakah masing-masing data bahan tersebut sudah ada. Jika ada, maka sistem hanya menambah nilai stok pada data bahan yang sudah ada saja. Jika belum, sistem akan mendaftarkan data bahan baru beserta nilai stoknya. Setelah mencapai inputan data bahan terakhir, sistem akan menambahkan data bahan yang baru dan mengupdate data bahan yang sudah ada dengan nilai stok baru pada database. |
| Aktor: | Koki |
| Prakondisi: | Bahan sudah diterima namun bahan belum diinputkan dan Koki sudah melakukan login |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Koki sudah melakukan login |  | | 2. Koki sudah melihat data bahan dari sistem |  | | 3.Koki memasukan data bahan melalui form tabel yang berisi id\_bahan, nama\_bahan, stock\_bahan, dan harga |  | |  | 4. Sistem mengecek, dimulai dari data bahan pertama, apakah masing-masing data bahan sudah ada. | |  | 5. Jika ada, maka sistem hanya menambah nilai stok pada data bahan yang sudah ada saja. Jika belum, sistem akan mendaftarkan data bahan baru beserta nilai stoknya. | |  | 6. Apabila masih ada data bahan, ulangi langkah 3. Jika sudah tidak ada lagi data bahan yang akan dimasukkan, lanjut ke langkah 5. | |  | 7. Sistem akan menambahkan data bahan yang baru dan mengupdate data bahan yang sudah ada dengan nilai stok baru pada database. | | |
| Pascakondisi: | Data\_Bahan sudah terdaftar ke dalam database. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 5, maka sistem tidak akan menyimpan perubahan yang terjadi |
| Contoh: | Koki memasukkan data bahan pada sebuah form tabel. Sistem mengecek masing-masing data bahan dengan data bahan yang sudah ada pada database. Sistem menambah stok bahan (untuk data bahan yang sudah ada) dan menambah data bahan yang belum ada sebelumnya pada database. |
| Isu Terbuka: | 1. Berapa batasan untuk jumlah total data yang bisa diinput? |

## Use Case Input Data Produksi

|  |  |
| --- | --- |
| Nama Use Case: | Input Data Produksi |
| Tujuan: | Memasukkan data produksi ke dalam database. |
| Input: | Data produksi |
| Output: | - |
| Skenario Utama: | Koki memasukkan data produksi pada sistem. Sistem mengecek apakah masing-masing data produksi tersebut sudah ada. Jika ada, maka sistem hanya menambah nilai stok pada data produksi yang sudah ada saja. Jika belum, sistem akan mendaftarkan data produksi baru beserta nilai stoknya. Setelah mencapai inputan data produksi terakhir, sistem akan menambahkan data produksi yang baru dan mengupdate data produksi yang sudah ada dengan nilai stok baru pada database. |
| Aktor: | Koki |
| Prakondisi: | Koki telah memproduksi makanan dan melakukan login namun belum memasukkan data produksi makanan ke database |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Koki telah melakukan login |  | | 2. Koki memasukkan data produksi pada sebuah form berisi id\_produksi, nama\_produksi , stock\_produksi, harga\_produksi |  | |  | 3. Sistem mengecek, dimulai dari data produksi pertama, apakah masing-masing data produksi sudah ada. | |  | 4. Jika ada, maka sistem hanya menambah nilai stok pada data produksi yang sudah ada saja. Jika belum, sistem akan mendaftarkan data produksi baru beserta nilai stoknya. | |  | 5. Apabila masih ada data produksi, ulangi langkah 3. Jika sudah tidak ada lagi data produksi yang akan dimasukkan, lanjut ke langkah 5. | |  | 6. Sistem akan menambahkan data produksi yang baru dan mengupdate data produksi yang sudah ada dengan nilai stok baru pada database. | | |
| Pascakondisi: | Data\_Produksi sudah terdaftar ke dalam database. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 6, maka sistem tidak akan menyimpan perubahan yang terjadi |
| Contoh: | Koki memasukkan data produksi pada sebuah form. Sistem mengecek masing-masing data produksi dengan data produksi yang sudah ada pada database. Sistem menambah stok makanan (untuk data produksi yang sudah ada) dan menambah data produksi yang belum ada sebelumnya pada database. |
| Isu Terbuka: | 1. Berapa batasan untuk jumlah total data yang bisa diinput? |

## 5.5 Use Case View Data Produksi

|  |  |
| --- | --- |
| Nama Use Case: | View Data Produksi |
| Tujuan: | Melihat Data\_Produksi pada database |
| Input: | - |
| Output: | Data Produksi |
| Skenario Utama: | Koki melihat data\_produksi |
| Aktor: | Sales dan Koki |
| Prakondisi: | Sales ingin melihat Data\_Produksi dan Sales sudah melakukan login |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1.Sales/koki sudah melakukan login |  | | 2. Sales/koki meminta data produksi dari sistem |  | |  | 3.Sistem memanggil Data\_Produksi dari database dengan atribut id\_produksi, nama\_produksi, stock\_produksi, dan harga\_produksi | |  | 4..Sistem menampilkan data produksi kepada sales | | |
| Pascakondisi: | Sales dapat melihat Data\_Produksi. |
| Prioritas: | Sedang |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 4, maka sistem akan menghentikan proses view data produksi |
| Contoh: | Sales/koki meminta sistem untuk menampilkan Data\_Produksi. Sistem memanggil Data\_Produksi dari database. Lalu sistem menampilkan hasilnya ke layar |
| Isu Terbuka: | 1. Apakah terjadi bentrok jika lebih dari satu user menggunakan use case ini? |

## 5.6 Use Case Input Data Penjualan

|  |  |
| --- | --- |
| Nama Use Case: | Input Data Penjualan |
| Tujuan: | Memasukkan Data\_Penjualan pada database |
| Input: | Data\_Penjualan |
| Output: | - |
| Skenario Utama: | Sales memasukkan data\_penjualan pada sistem. Sistem melakukan input data pelanggan. Sistem memasukan data\_penjualan pada database. |
| Aktor: | Sales |
| Prakondisi: | Sales telah menjual makanan dan sudah melakukan login namun belum memasukannya ke database dan telah memasukkan data pelanggan |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Sales telah login |  | | 2. Sales memasukkan Data\_Penjualan melalui form tabel berupa id\_penjualan, tgl\_penjualan, jumlah\_penjualan, ListProduk, nama\_pelanggan, alamat\_pelanggan, hp\_pelanggan. |  | |  | 3. Sistem menerima Data\_Penjualan dari form table. | |  | 4.Sistem memasukkan Data\_Penjualan pada database. | | |
| Alternative Flow Of Events: | - |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunaka: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Pascakondisi: | Data\_Penjualan sudah terdaftar ke database. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 4 selesai, maka sistem tidak akan menyimpan perubahan yang terjadi |
| Contoh: | Sales memasukkan Data\_Penjualan. Sistem melakukan input data penjualan dan data pelanggan. Sistem memasukkan Data\_Penjualan pada database. |
| Isu Terbuka: | 1. Berapa batasan untuk jumlah total data yang bisa diinput? |

## 5.7 Use Case Logout

|  |  |
| --- | --- |
| Nama Use Case: | Logout |
| Tujuan: | Mengeluarkan user dari sistem |
| Input: | - |
| Output: | - |
| Skenario Utama: | Sales/Admin/Koki menekan tombol logout. Sistem mengeluarkan Sales/Admin/Koki dari sistem |
| Aktor: | Sales/Admin/Koki |
| Prakondisi: | Sales/Admin/Koki sudah login ke dalam sistem dan ingin melakukan logout |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Sales/Admin/Koki menekan tombol logout. |  | |  | 1. Sistem mengeluarkan Sales/Admin/Koki dari sistem | | |
| Pascakondisi: | Sales/Admin/Koki sudah keluar dari sistem |
| Prioritas: | Rendah |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 2 selesai, maka sistem menghentikan proses Logout |
| Contoh: | Sales menekan tombol logout.  Sistem mengeluarkan Sales dari sistem |
| Isu Terbuka: | - |

## 5.8 Use Case Delete Data Bahan

|  |  |
| --- | --- |
| Nama Use Case: | Delete Data Bahan. |
| Tujuan: | Menghapus Data\_Bahan dari sistem. |
| Input: | Data\_Bahan |
| Output: | - |
| Skenario Utama: | Koki memilih data bahan dan menekan tombol delete. Sistem menghapus data bahan dari sistem. |
| Aktor: | Koki. |
| Prakondisi: | Koki sudah melakukan login dan ingin menghapus Data\_Bahan dari sistem. |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Koki telah melakukan login |  | | 2. Koki memilih data bahan yang ingin dihapus. |  | | 3. Koki menekan tombol delete. |  | |  | 4. Sistem mengecek ada tidaknya data bahan pada sistem. | |  | 5. Jika ada maka Sistem menghapus data bahan dari sistem. | | |
| Pascakondisi: | Data\_Bahan sudah terhapus dari dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 4 Data Bahan tidak ditemukan maka sistem mengoutputkan “Data Bahan tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 4 selesai, maka sistem menghentikan proses Delete Data Bahan. |
| Contoh: | Admin memilih bahan tepung untuk dihapus dan menekan tombol delete.  Sistem menghapus data tepung dari sistem. |
| Isu Terbuka: | 1. Apakah ada kemungkinan terjadi masalah delete kemudian data masih tersisa? |

## 5.9 Use Case Delete Data Produksi

|  |  |
| --- | --- |
| Nama Use Case: | Delete Data Produksi. |
| Tujuan: | Menghapus Data\_Produksi dari sistem. |
| Input: | Data\_Produksi |
| Output: | - |
| Skenario Utama: | Koki memilih data produksi dan menekan tombol delete. Sistem menghapus data produksi dari sistem. |
| Aktor: | Koki. |
| Prakondisi: | Koki sudah melakukan login dan ingin menghapus Data\_Produksi dari sistem. |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Koki telah melakukan login |  | | 2. Koki memilih Data\_Produksi yang ingin dihapus. |  | | 3.Koki menekan tombol delete. |  | |  | 4. Sistem mengecek ada tidaknya Data\_Produksi pada sistem. | |  | 5. Jika ada maka Sistem menghapus data bahan dari sistem. | | |
| Pascakondisi: | Data\_Produksi sudah terhapus di dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 4 Data Produksi tidak ditemukan maka sistem mengoutputkan “Data Produksi tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 4 selesai, maka sistem menghentikan proses Delete Data Produksi. |
| Contoh: | Koki memilih data brownis untuk dihapus dan menekan tombol delete.  Sistem menghapus data brownis dari sistem. |
| Isu Terbuka: | 1. Apakah ada kemungkinan terjadi masalah delete kemudian data masih tersisa? |

## 5.10 Use Case Delete Data Penjualan

|  |  |
| --- | --- |
| Nama Use Case: | Delete Data Penjualan. |
| Tujuan: | Menghapus Data\_Penjualan dari sistem. |
| Input: | Data\_Penjualan |
| Output: | - |
| Skenario Utama: | Sales memilih data penjualan dan menekan tombol delete. Sistem menghapus data penjualan dari sistem. |
| Aktor: | Sales. |
| Prakondisi: | Sales sudah melakukan login dan ingin menghapus data penjualan dari sistem. |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1.Sales telah melakukan login |  | | 2. Sales memilih Data\_Produksi yang ingin dihapus. |  | | 3.Sales menekan tombol delete. |  | |  | 4. Sistem mengecek ada tidaknya Data\_Produksi pada sistem. | |  | 5. Jika ada maka Sistem menghapus data bahan dari sistem. | | |
| Pascakondisi: | Data\_Penjualan sudah terhapus di dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 4 Data\_Penjualan tidak ditemukan maka sistem mengoutputkan “Data Penjualan tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi gangguan pada sebelum langkah 4 selesai, maka sistem menghentikan proses Delete Data Penjualan. |
| Contoh: | Sales memilih Data\_Penjualan bulan Januari untuk dihapus dan menekan tombol delete.  Sistem menghapus Data\_Penjualan bulan Januari dari sistem. |
| Isu Terbuka: | 1. Apakah ada kemungkinan terjadi masalah delete kemudian data masih tersisa? |

## 5.11 Use Case Update Data Bahan

|  |  |
| --- | --- |
| Nama Use Case: | Update Data Bahan. |
| Tujuan: | Memperbarui Data\_Bahan dari sistem. |
| Input: | Data\_Bahan |
| Output: | - |
| Skenario Utama: | Koki memilih Data\_Bahan, mengisikan data bahan baru dan menekan tombol update. Sistem memperbarui data bahan yang lama dengan yang baru dari sistem. |
| Aktor: | Koki |
| Prakondisi: | Koki sudah melakukan login dan ingin memperbarui Data\_Bahan dari sistem |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Koki memilih Data\_Bahan yang ingin diperbarui. |  | | 1. Koki menekan tombol pilih. |  | |  | 1. Sistem mengecek ada tidaknya Data\_Bahan pada sistem. | | 1. Jika ada maka Koki mengisikan data bahan yang baru. |  | | 1. Koki menekan tombol update. |  | |  | 1. Sistem memperbarui data bahan yang lama dengan Data\_Bahan yang baru dari sistem. | | |
| Pascakondisi: | Data\_Bahan sudah diperbarui di dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 3 Data\_Bahan tidak ditemukan maka sistem mengoutputkan “Data Bahan tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 5 selesai, maka sistem menghentikan proses Update Data Bahan. |
| Contoh: | Admin memilih Data\_Bahan tepung A dan menekan tombol pilih. Kemudian Admin memasukkan Data\_Bahan tepung B dan menekan tombol update.  Sistem memperbarui Data\_Bahan tepung A denganData\_Bahan tepung B dari sistem. |
| Isu Terbuka: | - |

## 5.12 Use Case Update Data Produksi

|  |  |
| --- | --- |
| Nama Use Case: | Update Data Produksi. |
| Tujuan: | Memperbarui Data\_Produksi dari sistem. |
| Input: | Data\_Produksi |
| Output: | - |
| Skenario Utama: | Admin memilih Data\_Produksi, mengisikan Data\_Produksi baru dan menekan tombol update. Sistem memperbarui Data\_Produksi yang lama dengan yang baru dari sistem. |
| Aktor: | Koki |
| Prakondisi: | Koki sudah melakukan login dan ingin memperbarui Data\_Produksi dari sistem |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Admin memilih Data\_Produksi yang ingin diperbarui. |  | | 1. Admin menekan tombol pilih. |  | |  | 1. Sistem mengecek ada tidaknya Data\_Produksi pada sistem. | | 1. Jika ada maka Admin mengisikan Data\_Produksi yang baru. |  | | 1. Admin menekan tombol update. |  | |  | 1. Sistem memperbarui Data\_Produksi yang lama dengan data produksi yang baru dari sistem. | | |
| Pascakondisi: | Data\_Produksi sudah diperbarui di dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 3 Data\_Produksi tidak ditemukan maka sistem mengoutputkan “Data Produksi tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 5 selesai, maka sistem menghentikan proses Update Data Produksi. |
| Contoh: | Admin memilih Data\_Produksi brownis A dan menekan tombol pilih. Kemudian Admin memasukkan Data\_Produksi brownis B dan menekan tombol update.  Sistem memperbarui Data\_Produksi brownis A dengan Data\_Produksi brownis B dari sistem. |
| Isu Terbuka: | - |

## 5.13 Use Case Update Data Penjualan

|  |  |
| --- | --- |
| Nama Use Case: | Update Data Penjualan. |
| Tujuan: | Memperbarui Data\_Penjualan dari sistem. |
| Input: | Data\_Penjualan |
| Output: | - |
| Skenario Utama: | Admin memilih Data\_Penjualan, mengisikan Data\_Penjualan baru dan menekan tombol update. Sistem memperbarui Data\_Penjualan yang lama dengan yang baru dari sistem. |
| Aktor: | Sales |
| Prakondisi: | Sales sudah melakukan login dan ingin memperbarui Data\_Penjualan dari sistem |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Admin memilih Data\_Penjualan yang ingin diperbarui. |  | | 1. Admin menekan tombol pilih. |  | |  | 1. Sistem mengecek ada Data\_Penjualan pada sistem. | | 1. Jika ada maka Admin mengisikan Data\_Penjualan yang baru. |  | | 1. Admin menekan tombol update. |  | |  | 1. Sistem memperbarui Data\_Penjualan yang lama dengan Data\_Penjualan yang baru dari sistem. | | |
| Pascakondisi: | Data\_Penjualan sudah diperbarui di dalam sistem. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | Jika pada tahap 3 Data\_Penjualan tidak ditemukan maka sistem mengoutputkan “Data Penjualan tidak ada” dan kembali ke tahap 1. |
| Skenario Eksepsional 1: | Apabila terjadi gangguan pada sebelum langkah 5 selesai, maka sistem menghentikan proses Update Data Penjualan. |
| Contoh: | Admin memilih Data\_Penjualan pada bulan Januari dan menekan tombol pilih. Kemudian Admin memasukkan Data\_Penjualan pada bulan Januari yang baru dan menekan tombol update.  Sistem memperbarui Data\_Penjualan pada bulan Januari dengan Data\_Penjualan pada bulan Januari yang baru dari sistem. |
| Isu Terbuka: | - |

## 5.14 Use Case View Data Bahan

|  |  |
| --- | --- |
| Nama Use Case: | View Data Bahan |
| Tujuan: | Melihat Data\_Bahan pada database |
| Input: | - |
| Output: | Data\_Bahan |
| Skenario Utama: | Koki/Admin melihat Data\_Bahan |
| Aktor: | Koki/Admin |
| Prakondisi: | Koki/Admin sudah melakukan login dan ingin melihat Data\_Bahan |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Sales meminta Data\_Bahan dari sistem |  | |  | 1. Sistem mengirim Data\_Bahan kepada sales | | |
| Pascakondisi: | Sales dapat melihat Data\_Bahan. |
| Prioritas: | Sedang |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 1, maka sistem akan menghentikan proses view Data\_Bahan |
| Contoh: | Sistem mengirimkan data\_produksi pada sales. |
| Isu Terbuka: | - |

## 5.15 Use Case Kelola Laporan Penjualan

|  |  |
| --- | --- |
| Nama Use Case: | Kelola Laporan Penjualan |
| Tujuan: | Mengelola laporan penjualan berdasarkan Data\_Penjualan pada database |
| Input: | Data\_Penjualan |
| Output: | Laporan\_Penjualan |
| Skenario Utama: | Admin meminta laporan penjualan dari sistem.  Sistem mengambil data\_penjualan dari database. Sistem mengelola dan mengubah Data\_Penjualan menjadi laporan penjualan dan memberikannya pada admin |
| Aktor: | Admin dan Sales |
| Prakondisi: | Admin/Sales sudah melakukan login dan ingin melihat laporan penjualan |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Admin meminta laporan penjualan dari sistem |  | |  | 2. Sistem mengambil Data\_Penjualan dari database | |  | 3. Sistem mengelola dan mengubah Data\_Penjualan menjadi laporan penjualan dan memberikannya pada admin | | |
| Pascakondisi: | Sales dapat melihat laporan penjualan. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi ganguan pada sebelum langkah 1, maka sistem akan menghentikan mengelola laporan penjualan |
| Contoh: | Admin meminta laporan penjualan tanggal 21 dari sistem.  Sistem mengambil data\_penjualan itu dari database. Sistem mengelola dan mengebuah data\_penjualan menjadi laporan penjualan itu dan memberikannya pada admin |
| Isu Terbuka: | 1. Bagaimana solusi untuk mendapatkan laporan penjualan apabila ada lebih dari satu chunk data penjualan? Apakah ada kemungkinan dieliminasi untuk data duplikat? |

## 5.16 Use Case Kelola Laporan Keuangan

|  |  |
| --- | --- |
| Nama Use Case: | Kelola Laporan Keuangan |
| Tujuan: | Mengelola laporan keuangan berdasarkan olahan inputan yang masuk (TotalPemasukan: ListLaporanPenjualan; TotalPengeluaran: ListBahanMasuk, ListKoki, ListAdmin, ListSales) pada database |
| Input: | ListLaporanKeuangan, ListBahanMasuk, ListKoki, ListAdmin, ListSales |
| Output: | Laporan\_Keuangan |
| Skenario Utama: | Admin meminta laporan keuangan dari sistem.  Sistem mengambil ListLaporanKeuangan, ListBahanMasuk, ListKoki, ListAdmin, ListSales dari database. Sistem mengolah TotalPemasukan dari ListLaporanPenjualan dan mengolah TotalPengeluaran dari ListBahanMasuk, ListKoki, ListAdmin, ListSales, sehingga hasilnya akan menjadi laporan keuangan dan memberikannya pada admin. |
| Aktor: | Admin |
| Prakondisi: | Admin sudah melakukan login dan ingin melihat laporan keuangan |
| Langkah-langkah:   |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Admin meminta laporan keuangan dari sistem |  | |  | 2. Sistem mengambil ListLaporanKeuangan, ListBahanMasuk, ListKoki, ListAdmin, ListSales dari database | |  | 3. Sistem mengolah TotalPemasukan dari ListLaporanPenjualan dan mengolah TotalPengeluaran dari ListBahanMasuk, ListKoki, ListAdmin, ListSales | |  | 4. Sistem menuangkan hasilnya menjadi laporan keuangan utuh dan menampilkannya kepada admin. | | |
| Pascakondisi: | Admin dapat melihat laporan keuangan. |
| Prioritas: | Tinggi |
| Jika Dapat Digunakan: | Sekali per percobaan |
| Frekuensi Digunakan: | Rutin |
| Channel untuk Aktor: | Komputer dengan akses LAN. |
| Aktor Kedua: | - |
| Channel untuk Aktor Kedua: | - |
| Alternative Flow Of Events: | - |
| Skenario Eksepsional 1: | Apabila terjadi gangguan pada sebelum langkah 1, maka sistem akan menghentikan mengelola laporan keuangan |
| Contoh: | Admin meminta laporan keuangan pada bulan Mei. Sistem mengambil ListLaporanKeuangan, ListBahanMasuk, ListKoki, ListAdmin, ListSales dari database. Sistem mengolah TotalPemasukan dari ListLaporanPenjualan dan mengolah TotalPengeluaran dari ListBahanMasuk, ListKoki, ListAdmin, ListSales, sehingga hasilnya akan menjadi laporan keuangan dan memberikannya pada admin. |
| Isu Terbuka: | - |

# 6. Requirements Nonfungsional Lainnya

## 6.1 Requirements Performa

Dengan adanya aplikasi diharapkan dapat meningkatkan kinerja perusahaan. Keberadaan aplikasi juga akan mempermudah karyawan perusahaan dalam pekerjaannya. Dan yang paling penting aplikasi akan mengurangi kemungkinan terjadinya kesalahan data yang ada pada perusahaan.

## 6.2 Requirements Keselamatan

Aplikasi ini dapat menangani sebuah kejadian di luar perkiraan seperti kerusakan perangkat yang datang dengan tidak terduga. Ini dapat dilihat pada bab 5, untuk setiap penanganan hal seperti itu. Selain itu admin akan melakukan *backup* data sekitar 1 kali setiap minggu untuk mengurangi kerusakan data yang mungkin terjadi.

## 6.3 Requirements Keamanan

Untuk melindungi data yang aplikasi ini memiliki pembagian peran bagi setiap jenis karyawan yang ada. Setiap jenis karyawan hanya dapat melakukan peran yang dia miliki. Contohnya koki tidak bisa memasukkan data\_penjualan dan sales juga tidak dapat memasukkan data\_makanan.

## 6.3 Atribut Kualitas Perangkat Lunak

Aplikasi ini memiliki harus bersifat sederhana, tepat, fleksibel, aman, dan dapat dijalankan pada sistem operasi Windows. Sederhana agar pengguna aplikasi mudah mempelajari dan menggunakan aplikasi. Tepat karena salah satu fungsi utama aplikasi ini adalah menghitung keuangan perusahaan. Fleksibel karena fungsionalitas aplikasi ini harus dapat diubah dengan mudah mengikuti kondisi perusahaan. Aplikasi ini harus aman agar data perusahaan tidak tersebar kepada perusahaan saingan. Aplikasi juga harus dapat dijalankan pada sistem operasi Windows karena sistem operasi yang digunakan perusahaan adalah Windows.

# 7. Requirements Lain

## 7.1 Requirment Database

Database yang akan digunakan untuk menjalankan aplikasi adalah database MySQL. MySQL ini akan menyambungkan satu komputer dengan komputer lain dengan LAN. Untuk pembuatan database, programmer dapat menggunakan oracle MySQL.

## 7.2 Requirement Hak Cipta

Untuk melindungi aplikasi dari kemungkinan pencurian/pembajakan sebaiknya programmer memasang *copyright* pada aplikasi. Walaupun peraturan hak cipta masih belum diberlakukan dengan kuat di Indonesia, tapi setidaknya dapat mengingatkan pembajak untuk tidak mengambil karya orang lain.

# Lampiran A: Daftar Kata-Kata Sukar

|  |  |
| --- | --- |
| * SKPL (Spesifikasi Kebutuhan Perangkat Lunak): | Laporan yang digunakan sebagai dasar programmer untuk membuat program yang bertujuan untuk menyelesaikan masalah kliennya. |
| * Use Case Diagram: | Diagram yang mendeskripsikan sistem dari pandangan pengguna. |
| * Class Diagram: | Diagram yang menunjukkan kumpulan kelas dan *interface,* termasuk hubungan dan kerjasama antara keduanya. |
| * DFD (Data Flow Diagram): | Diagram yang memodelkan proses dan aliran data pada suatu sistem yang akan dibuat. |
| * Flow Map: | Diagram yang memodelkan proses dan aliran data pada suatu sistem yang sedang berjalan.. |

# Lampiran B: Analysis Models

## Proses Bisnis

Raniah Food Creation yang beralamat di Jalan Guntursari Wetan 17, Bandung, memiliki tiga bagian staf, yaitu: admin, produksi, dan sales. Tugas dari admin adalah membuat data administrasi dan mengatur keuangan. Tugas bagian produksi adalah mengelola bahan yang tersedia dan membuat produk kue dari bahan tersebut. Tugas sales adalah mempromosikan produk dan mengantar produk ke customer. Sales bertugas mendistribusikan ke toko-toko seperti Kartika Sari ataupun rumah pelanggan.

Perusahaan tersebut mendapatkan stok bahan langsung dari pabrik. Lalu stok barang tersebut akan disimpan di gudang penyimpanan dan di data oleh koki. Setelah itu bahan yang tersedia akan di olah menjadi sebuah produk makanan oleh staf koki dan dicatat. Selanjutnya produk yang sudah jadi tersebut akan di berikan kepada sales. Lalu sales akan mengirimkan produk tersebut ke mendistribusikannya ke toko-toko besar,pasar,toko pinggir jalan, dan rumah customer yang telah memesan. Setelah transaksi terjadi maka sales akan mencatat penjualan. Data-data yang ada akan disimpan oleh manajer tiap hari untuk menghitung pendapatan per harinya dan tiap bulan untuk menghitung pendapatan tiap bulan. Proses pendataan dan perhitungan dilakukan secara manual sehingga ada risiko salah perhitungan maupun data yang hilang.

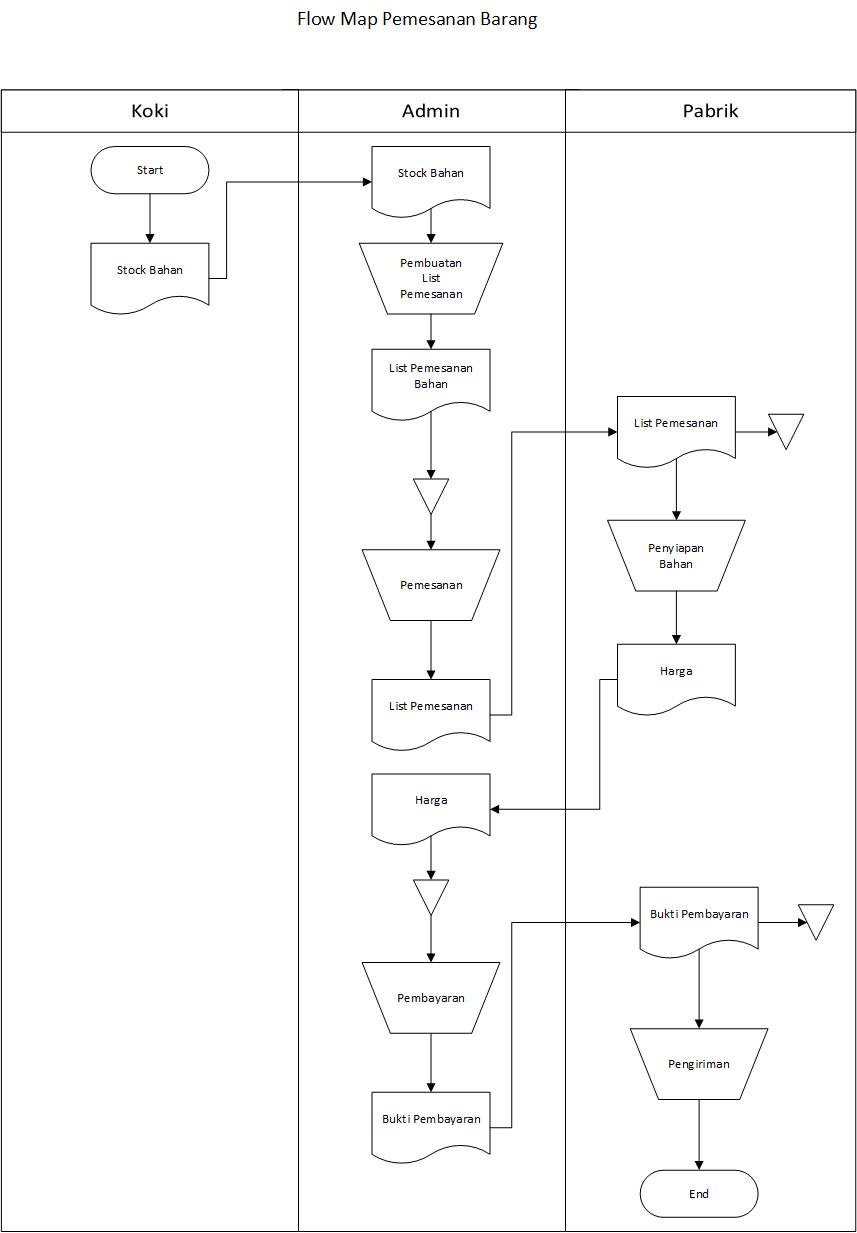
## Aturan Bisnis

* Hanya manajer yang boleh mengatur bagian keuangan.
* Hanya manajer yang boleh memecat/merekrut karyawan.
* Manajer dapat melihat seluruh data karyawannya.
* Manajer menyimpan data perusahaan setelah diserahkan oleh karyawannya tiap harinya untuk dapat melihat pemasukan tiap hari.
* Data tersebut akan kembali diolah tiap bulan untuk melihat pemasukan tiap bulan.
* Sales dapat membatalkan pesanan pelanggan.
* Koki yang mengurus dapur ada yang memasak dan ada yang memonitor data bahan dan data produksi.
* Koki akan memberi tahu manajer untuk memesan bahan apabila stok bahan tinggal sedikit dan manajer akan memesan stok bahan tersebut dari perusahaan.
* Perusahaan akan mengirim bahan setelah manajer mengirim bukti pembayaran bahan.
* Koki akan memproduksi produk apabila stok produk tersebut mau habis dan bahan yang dibutuhkan mencukupi.

## Flow Map

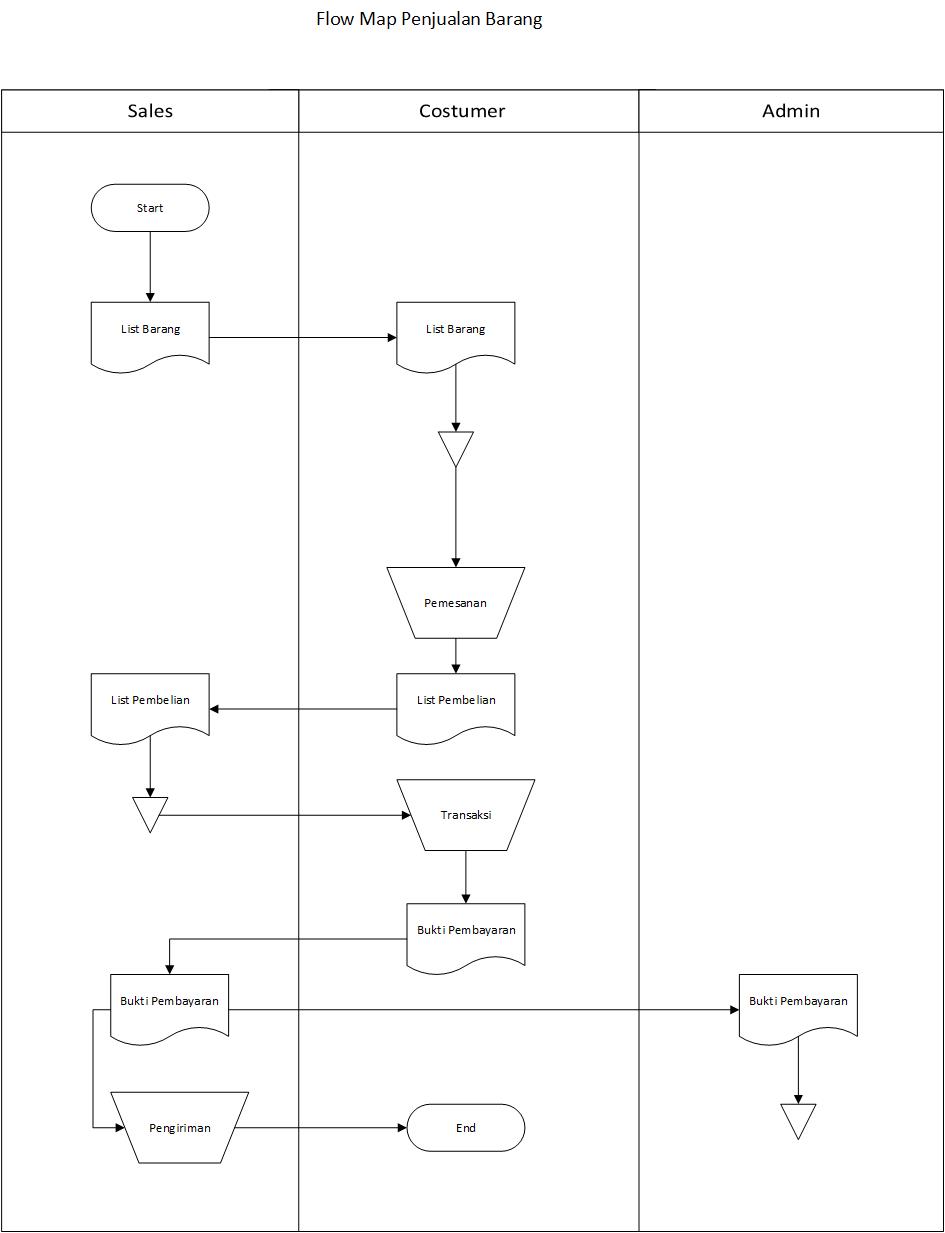
### Pemesanan Barang

Tabel -Flow Map Pemesanan Barang



### Penjualan Barang

Tabel -Flow Map Penjualan Barang



### Produksi Barang

Tabel -Flow Map Produksi Barang

