**BAB IV**

**ANALISA DAN PEMBAHASAN**

1. **Analisa Sistem**

Untuk memperjelas permasalahan yang timbul maka penulis akan mencoba memformulasikan ke dalam bentuk perumusan masalah. Berikut ini akan diuraikan tentang hal-hal yang menjadi penyebab timbulnya masalah yang dapat menimbulkan kebutuhan untuk pembuatan sistem baru :

* + - 1. Penggunaan transaksi data keluar masuk barang, Data pemakaian barang dan Data Supplier masih menggunakan excel.

1. Penyimpanan data secara manual dan tidak terstruktur dengan komputer sehingga menyulitkan pencarian data.
2. Belum adanya pembuatan laporan secara terstruktur dan rinci.
3. **Analisis Kebutuhan Sistem Fungsional**

Kebutuhan fungsional (*functional requirement)* adalah jenis kebutuhan yang berisi proses – proses apa saja yang dapat dibutuhkan oleh system. Kebutuhan fungsional juga berisi informasi – informasi apa saja yang harus ada dan di hasilkan oleh system. Berikut adalah kebutuhan fungsional dari Sistem inventori Gudang:

**Admin**

* + - * 1. Sistem dapat melakukan login.
        2. Sistem bisa menginput data barang masuk dan keluar.
        3. Sistem dapat mengetahui stok barang di Gudang, barang keluar dan masuk.
        4. Sistem bisa mencetak laporan data barang masuk dan keluar.

1. **Analisa Kebutuhan Sistem**
2. **Analisa Kebutuhan Hardware**

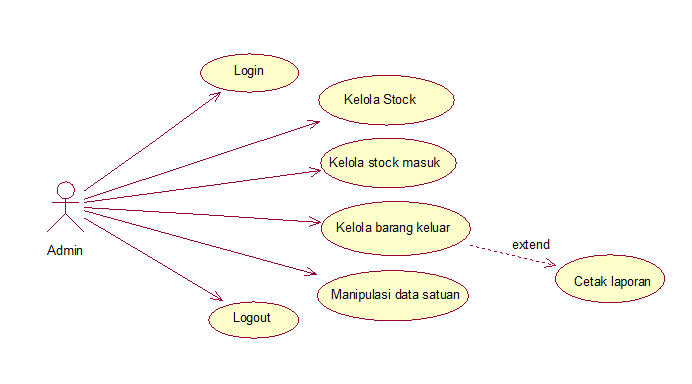
Berikut adalah hardware dengan sepesifikasi minimum yang dapat digunakan penulis untuk pembuatan sistem inventori Gudang di Breadtalk Semarang adalah sebagai berikut :

1. *Processor intel core i3* atau lebih tinggi.
2. Monitor 14” resolusi layar 1024x768 pixels.
3. Hardisk 500 GB atau lebih tinggi.
4. RAM 4GB atau lebih tinggi
5. Keyboard
6. Mouse
7. Printer
8. Modem untuk koneksi internet.
9. **Analisa Kebutuhan Software**

Perangkat lunak yang dibutuhkan dalam pembuatan sistem inventori Gudang di Breadtalk Semarang:

1. Sistem operasi windows 7 atau versi yang lebih tinggi ( windows 10 pro lebih direkomendasikan ).
2. Xampp V3.2.1.
3. Web browser (Mozila firefox, Crome, Opera, dll).
4. Notepad++ atau sublime text 3.
5. Framework CodeIgniter 3.1.6.
6. **Perancangan Sistem UML (*****Unified Modeling Language).***
   1. ***Use Case Diagram***

*Use case* diagrammenggambarkan fungsionalitas sistem secara global. Berikut ini adalah *use case* yang berjalan di bagian



Gambar 4.1*Use Case Diagram*

**Sekenario Use Case melakukan Login**

Nama *Use Case* : Melakukan Login

Aktor : Admin

Deskripsi : Proses login admin merupakan proses

Masuk agar diizinkan mengakses dengan

memasukkan username dan password

yang telah divalidasi

*Pre – Condition* : Admin belum login ke dalam sistem

*Post – Condition* : Admin berhasil masuk ke dalam sistem

Dan dapat mengaksesn

Tabel 4.1 Skenario Melakukan Login

|  |  |
| --- | --- |
| **Aksi aktor** | **Reaksi Sistem** |
| **Alur untuk melakukan login** | |
| 1. Admin masuk sistem dan memilih menu login |  |

Tabel 4.1 Skenario Melakukan Login (Lanjutan)

|  |  |
| --- | --- |
| **Aksi aktor** | **Reaksi Sistem** |
|  | Sistem menampilkan halaman login |
| Admin mengisi username dan password |  |
|  | Sistem akan melakukan validasi data username dan password |

1. **Skenario Use Case Mengelola Stock**

Nama *use case* : Mengelola Stock

Aktor : Admin

Deskripsi : Kegiatan untuk menambah atau menginput stock baru.

*Pre-condition* : Admin harus berada di menu utama dan

sudah *login*.

*Post*-*condition* : Data informasi input baru tersimpan

Tabel 4.3 Skenario Mengelola Stock

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk menambah data barang baru** | |
| 1. Klik menu data master kemudian pilih stock barang |  |
|  | 1. Manampilkan data stock barang |
| **Alur untuk tambah stock barang** | |
| 1. Klik tombol tambah stock |  |
|  | 1. Menampilkan form tambah stock |
| 1. Isikan data stock barang baru |  |

Tabel 4.3 Skenario Mengelola Stock (Lanjutan)

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Klik tombol “Submit” untuk mnyimpan input stock barang baru |  |
|  | 1. Sistem memvalidasi |
|  | 1. Jika data barang yang dimasukkan valid, maka data akan disimpan kedatabase |
|  | 1. Data stock baru berhasil di input |
| **Alur untuk update stock barang** | |
| 1. Memilih icon “Update” |  |
|  | 1. Menampilkan form update |
| 1. Isikan data yang akan di update |  |
| 1. Menekan tombol submit |  |
|  | 1. Memvalidasi data terupdate |
|  | 1. Data stock barang berhasil diupdate |
| **Alur untuk untuk hapus stock** | |
| 1. Memilih icon delete |  |
|  | 1. Tampil alert yakin ingin menghapus data |
| 1. Memilih tombol Ok |  |
|  | 1. Data berhasil terhapus |

1. **Skenario *Use Case* Kelola Stock Barang Masuk**

Nama *use case* : Kelola stock barang masuk

Aktor : Admin

Deskripsi : Proses ini adalah kegiatan untuk update stok keluar, hapus stok dan mengeluarkan stok

*Pre-condition* : Admin sudah harus berada di menu transaksi, stock gudang dan sudah *login.*

*Post*-*condition* : Data stok berhasil di keluarkan.

Tabel 4.5 Skenario kelola stock barang masuk

|  |  |  |
| --- | --- | --- |
| **Aksi Aktor** | | **Reaksi Sistem** |
| **Alur untuk update barang masuk** | | |
| 1. Memilih menu transaksi, stock gudang | |  |
|  | | 1. Menampilkan tabel barang masuk |
| 1. Memilih icon “Update” | |  |
|  | | 1. Menampilkan form update |
| 1. Menekan tombol submit | |  |
|  | | 1. Memvalidasi data terupdate |
|  | | 1. Data barang berhasil diupdate |
| **Alur untuk delete barang masuk** | | |
| 1. Memilih menu transaksi, stock gudang | |  |
|  | | 1. Menampilkan tabel barang masuk |
| 1. Memilih icon hapus | |  |
|  | | 1. Tampil alert yakin ingin menghapus data |
| 1. Memilih tombol Ok | |  |
|  | | 1. Data berhasil terhapus |
| **Alur untuk keluarkan barang** | | |
| 1. Memilih menu transaksi, stock gudang |  | |
|  | 1. Menampilkan tabel barang masuk | |
| 1. Memilih icon keluarkan barang |  | |
|  | 1. Tampil form keluarkan barang | |

Tabel 4.5 Skenario list data barang masuk (Lanjutan)

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk keluarkan barang** | |
| 1. Isikan tanggal barang keluar |  |
|  | 1. Barang berhasil di keluarkan |

1. **Skenario *Use Case* Kelola** B**arang Keluar**

Nama *Use* *Case* : Kelola barang keluar

*Actor* : Admin

Deskripsi : Proses admin melakukan *invoice*

*Pre – Condition* : Admin sudah login ke dalam aplikasi

*Post – Condition* : Data stok invoice berhasil tercetak

Tabel 4.6 Skenario usecase Kelola barang keluar

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk invoice barang** | |
| 1. Memilih menu transaksi, tabel barang keluar |  |
|  | 1. Menampilkan data yang sudah di keluarkan. |
| 1. Pilih icon invoice |  |
|  | 1. Memproses printah invoice |
|  | 1. Tampil data barang yang akan di cetak. |

Tabel 4.6 Skenario usecase Kelola barang keluar(Lanjutan)

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk invoice semua barang** | |
| 1. Memilih menu transaksi, tabel barang keluar |  |
|  | 1. Menampilkan data yang sudah di keluarkan |
| 1. Pilih tombol semua invoice |  |
|  | 1. Memproses printah semua invoice |
|  | 1. Menampilkan data semua invoice |
|  | 1. Semua invoice berhasil dicetak |
| **Alur untuk hapus barang yang sudah di keluarkan** | |
| * + - 1. Memilih menu transaksi, tabel barang keluar |  |
|  | * + - 1. Menampilkan data yang sudah di keluarkan |
| * + - 1. Memilih icon hapus |  |
|  | * + - 1. Tampil alert yakin ingin menghapus data |
|  | * + - 1. Data barang keluar dihapus |

1. **Skenario *Use Case* Manipulasi Data Satuan**

Nama *Use* *Case* : Kelola data satuan

*Actor* : Admin

Deskripsi : Proses ini merupakan proses untuk

Menentukan jenis satuan suatu barang.

*Pre – Condition* : Admin sudah login ke dalam aplikasi

*Post – Condition* : Admin berhasil keluar dari aplikasi

Tabel 4.7 Skenario usecase manipulasi data satuan

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk manipulasi tambah data satuan** | |
| 1. Memilih menu transaksi tabel satuan |  |
|  | 1. Menampilkan tabel satuan |
| 1. Pilih tombol tambah satuan |  |
|  | 1. Tampil form tambah satuan |
| 1. Isikan satuan |  |
|  | 1. Klik tombol submit |
|  | 1. Data satuan tersimpan |
| **Alur untuk manipulasi update data satuan** | |
| 1. Memilih menu transaksi tabel satuan |  |
|  | 1. Menampilkan tabel satuan |
| 1. Memilih icon update |  |
|  | 1. Menampilkan form update satuan |
| 1. Isikan kode satuan dan nama satuan |  |
| 1. Memilih tombol update |  |
|  | 1. Data bethasil di update |
| **Alur untuk untuk hapus data satuan** | |
| 1. Memilih menu transaksi tabel satuan |  |
|  | 1. Menampilkan tabel satuan |
| 1. Memilih icon hapus |  |
|  | 1. Tampil alert yakin ingin menghapus data satuan |
| 1. Menekan tombol ok |  |
|  | 1. Data satuan berhasil dihapus |

1. **Skenario *Use Case* melakukan *logout***

Nama *Use* *Case* : Melakukan *Logout*

*Actor* : Admin

Deskripsi : Proses *logout* merupakan proses untuk

keluar dari aplikasi

*Pre – Condition* : Admin sudah login ke dalam aplikasi

*Post – Condition* : Admin berhasil keluar dari aplikasi

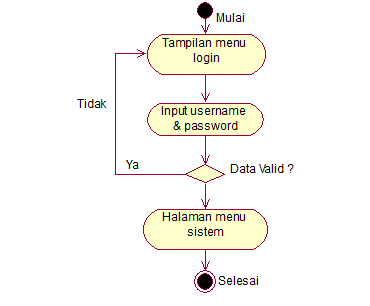
Tabel 4.8 Skenario Melakukan *Logout*

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Alur untuk keluar dari system** | |
| 1. Admin memilih menu logout |  |
|  | 1. Sistem merespon dan memverifikasi |
|  | 1. Sistem melakukan logout |

* + 1. ***Activity Diagram***

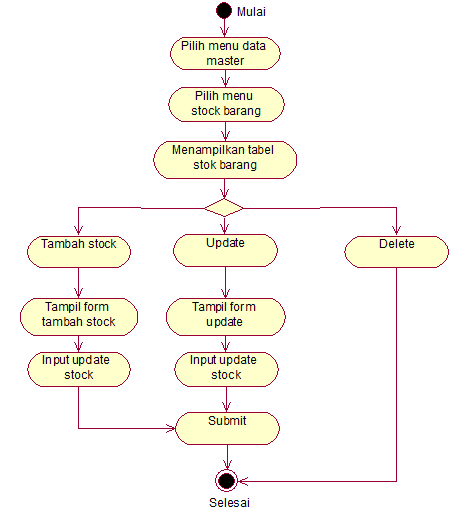
*Activity diagram* menggambarkan rangkaian aliran dari aktivitas yang dapat digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas yang lain seperti *use case* atau interaksi. Aktivitas merupakan bentuk khusus dari keadaan dimana aktivitas menggambarkan kegiatan yang dilakukan. Berikut ini adalah *activity diagram* yang berjalan pada Sistem inventori Gudang di breadtalk Semarang.

1. *Activity Diagram* melakukan Login

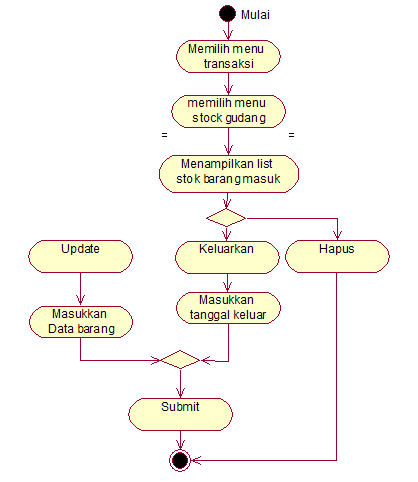
****

Gambar 4.2 *Activity Diagram* melakukan Login

1. *Activity Diagram* Kelola Stock

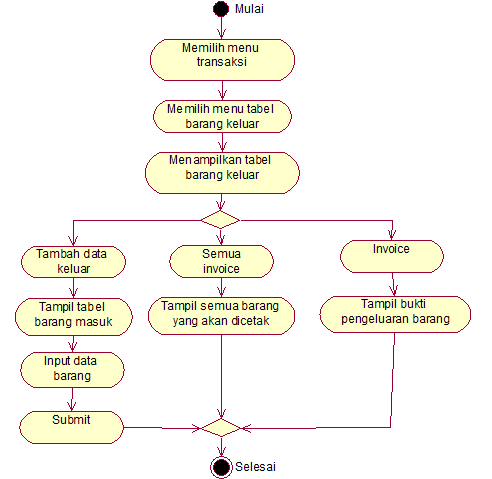
**

Gambar 4.3 *Activity Diagram* Kelola Stock

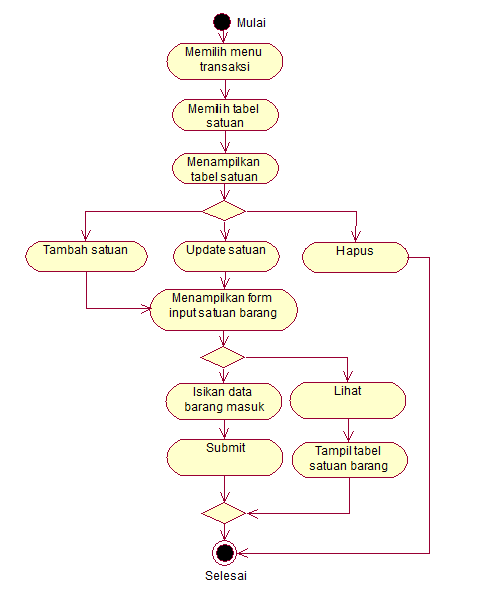
1. *Activity Diagram* Kelola Stock Barang Masuk

Gambar 4.4 *Activity Diagram* Kelola Stock Barang Masuk

1. *Activity Diagram* Kelola Barang Keluar

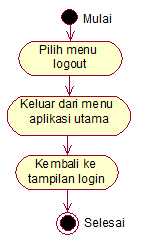


Gambar 4.5 *Activity Diagram* Kelola Barang Keluar

1. *Activity Diagram* Manipulasi Data Satuan

Gambar 4.6 *Activity Diagram* Manipulasi Data Satuan

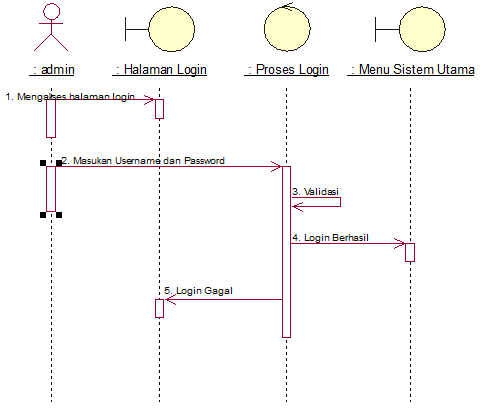
1. *Activity Diagram* Logout



Gambar 4.7 *Activity Diagram* Logout

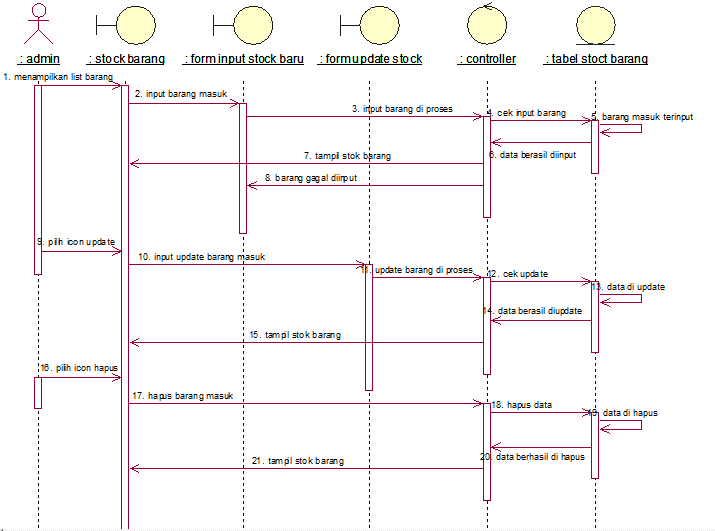
* + 1. ***Sequence Diagram***

*Sequence diagram* berisi gambaran dari urutan kejadian suatu kegiatan. Menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

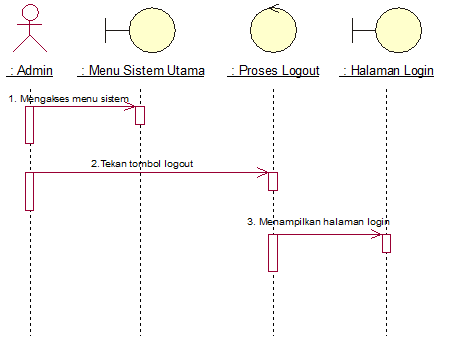
1. *******Sequence Diagram* Login

Gambar 4.8 *Sequence Diagram* Login

1. *Sequence Diagram* Mengelola Stock Barang

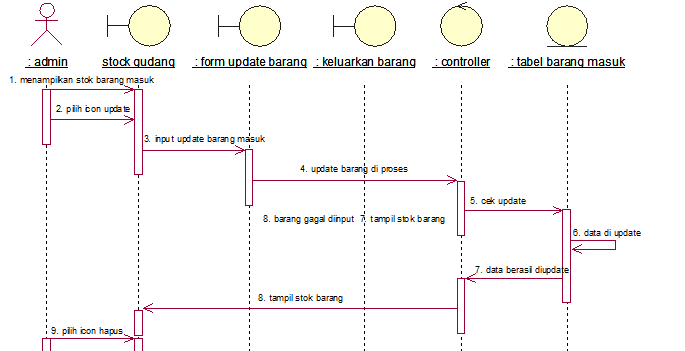
**

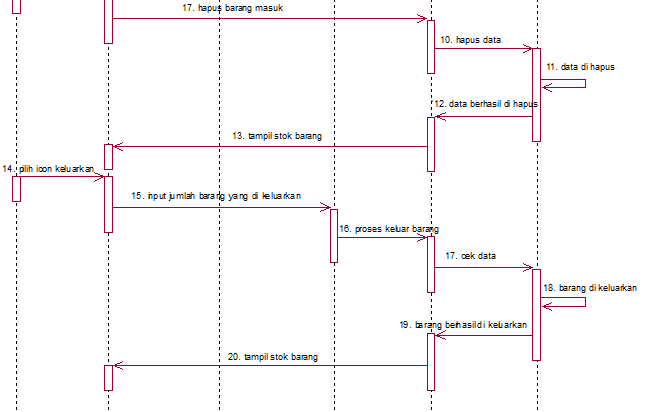
Gambar 4.9 *Sequence Diagram* Mengelola tambah barang masuk

1. *Sequence Diagram* Mengelola Logout

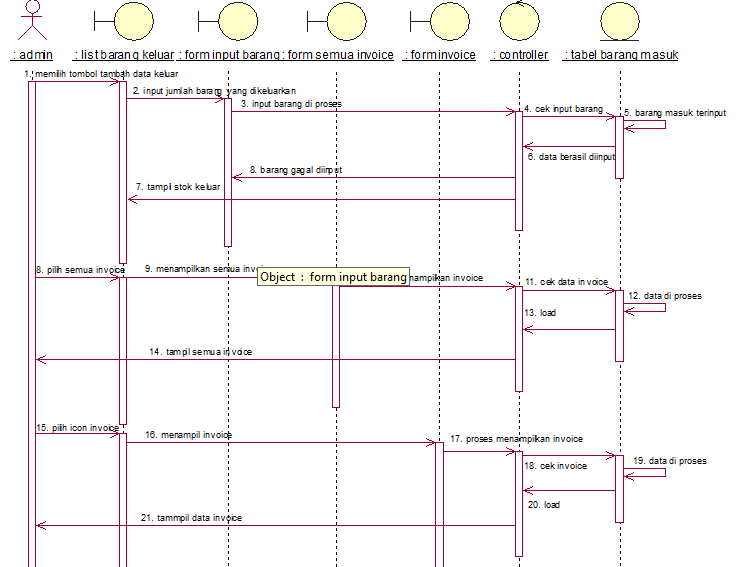
Gambar 4.10 *Sequence Diagram* Mengelola Logout

1. *Sequence Diagram* Kelola Stock Barang Masuk

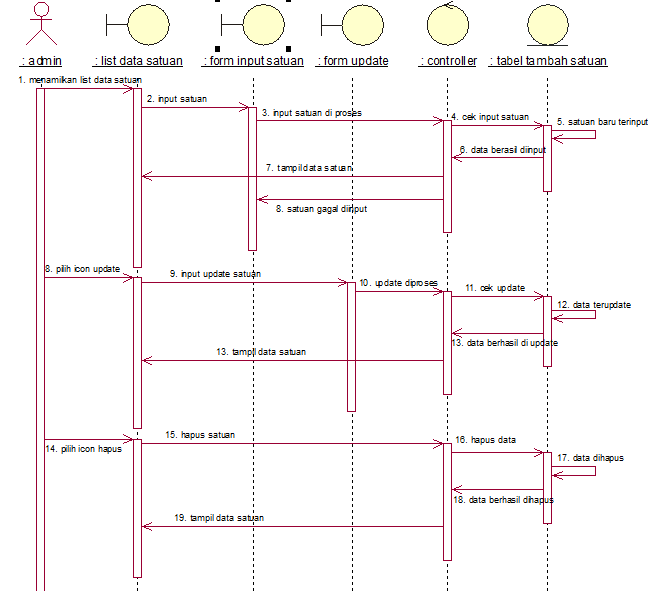




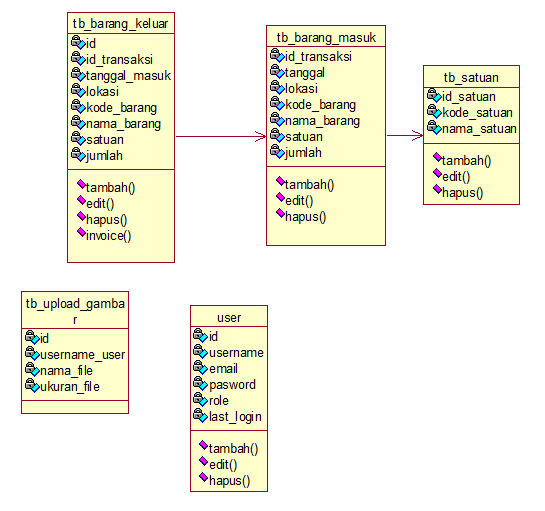
Gambar 4.11 *Sequence Diagram* Kelola Stock Barang Masuk

1. *Sequence Diagram* Kelola Barang Keluar

Gambar 4.12 *Sequence Diagram* Kelola Barang Keluar

1. *Sequence Diagram* Manipulasi Data Satuan

Gambar 4.13 *Sequence Diagram* Manipulasi Data Satuan

* + 1. ***Class Diagram***

Gambar 4.14 *Class Diagram*

* 1. **Perancangan Database**

Fungsi dari suatu perancangan *database* adalah untuk menampung beberapa tabel yang digunakan sebagai sumber pengolahan data. Perancangan *database* akan memberikan gambaran mengenai pembuatan tabel didalam *database* beserta atribut – atributnya. Berikut adalah nama – nama tabel yang digunakan beserta field – field yang terdapat pada masing – masing tabel.

* + 1. **Perancangan Tabel Barang Masuk**

Tabel ini berfungsi untuk menyimpan data barang masuk

Tabel 4.9 Perancangan Tabel Barang Masuk

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id\_transaksi | Int(50) | *Primary Key* |
| tanggal | Varchar(20) |  |
| lokasi | Varchar(100) |  |
| Kode\_barang | Varchar(100) |  |
| nama\_barang | Varchar(100) |  |
| satuan | Varchar(50) |  |
| jumlah | |  |  | | --- | --- | | Varchar(10) |  | |  |

* + 1. **Perancangan Tabel Barang Keluar**

Tabel ini berfungsi untuk menyimpan data barang keluar

Tabel 4.10 Perancangan Tabel Barang Keluar

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | int(10) | *Primary Key* |
| id\_transaksi | Varchar(50) | Foreign key |
| tanggal\_masuk | Varchar(20) |  |
| tangal\_keluar | Varchar(20) |  |
| lokasi | Varchar(100) | |  |  | | --- | --- | |  |  | |
| kode\_barang | varchar(100) |  |
| nama\_barang | Varchar(100) |  |
| satuan | Varchar(50) |  |
| jumlah | Varchar(10) |  |

* + 1. **Perancangan Tabel Satuan**

Tabel ini berfungsi untuk menyimpan data satuan barang.

Tabel 4.11 Perancangan Tabel Satuan

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| Id\_satuan | Int(11) | *Primary Key*   |  |  | | --- | --- | | (*auto\_increment*) | [Change](http://localhost/phpmyadmin/tbl_structure.php?db=ket_surat&table=penduduk&token=c74aa92b6907c66973e02b8535d1bc83&goto=tbl_structure.php&back=tbl_structure.php&field=id&change_column=1) | |
| kode\_satuan | Varchar(100) |  |
| nama\_satuan | Varchar(100) |  |

* + 1. **Perancangan Tabel Upload Gambar**

Tabel ini berfungsi untuk menyimpan informasi data gambar yang berhubungan dengan admin.

Tabel 4.12 Perancangan Tabel Upload Gambar

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | int(11) | *Primary Key*  (*auto\_increment*) |
| username\_user | Varchar(100) |  |
| nama\_file | Varchar(220) |  |
| ukuran\_file | Int(8) |  |

* + 1. **Perancangan Tabel User**

Tabel ini berfungsi untuk menyimpan informasi data user

Tabel 4.13 Perancangan Tabel User

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| id | Int(12) | *Primary Key*  (*auto\_increment*) |
| username | Varchar(200) |  |
| email | Varchar(100) |  |
| password | Varchar(200) |  |

Tabel 4.13 Perancangan Tabel User (Lanjutan)

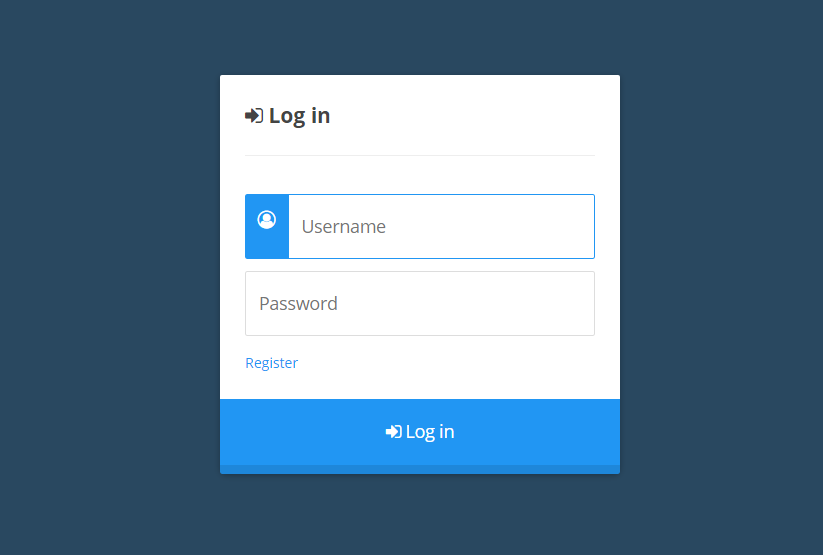
|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Keterangan** |
| role | Tinyint(4) |  |
| last\_login | Varchar(20) |  |

* 1. **Tampilan Antarmuka Aplikasi**

Tanpilan antarmuka aplikasi ini didesign sesederhana mungkin dan sesimple mungkin agar user bisa menyesuaikan dengan sistem dengan cepat, berikut adalah tampilan sistem inventori gudang sebagai berikut:

1. Halaman Login

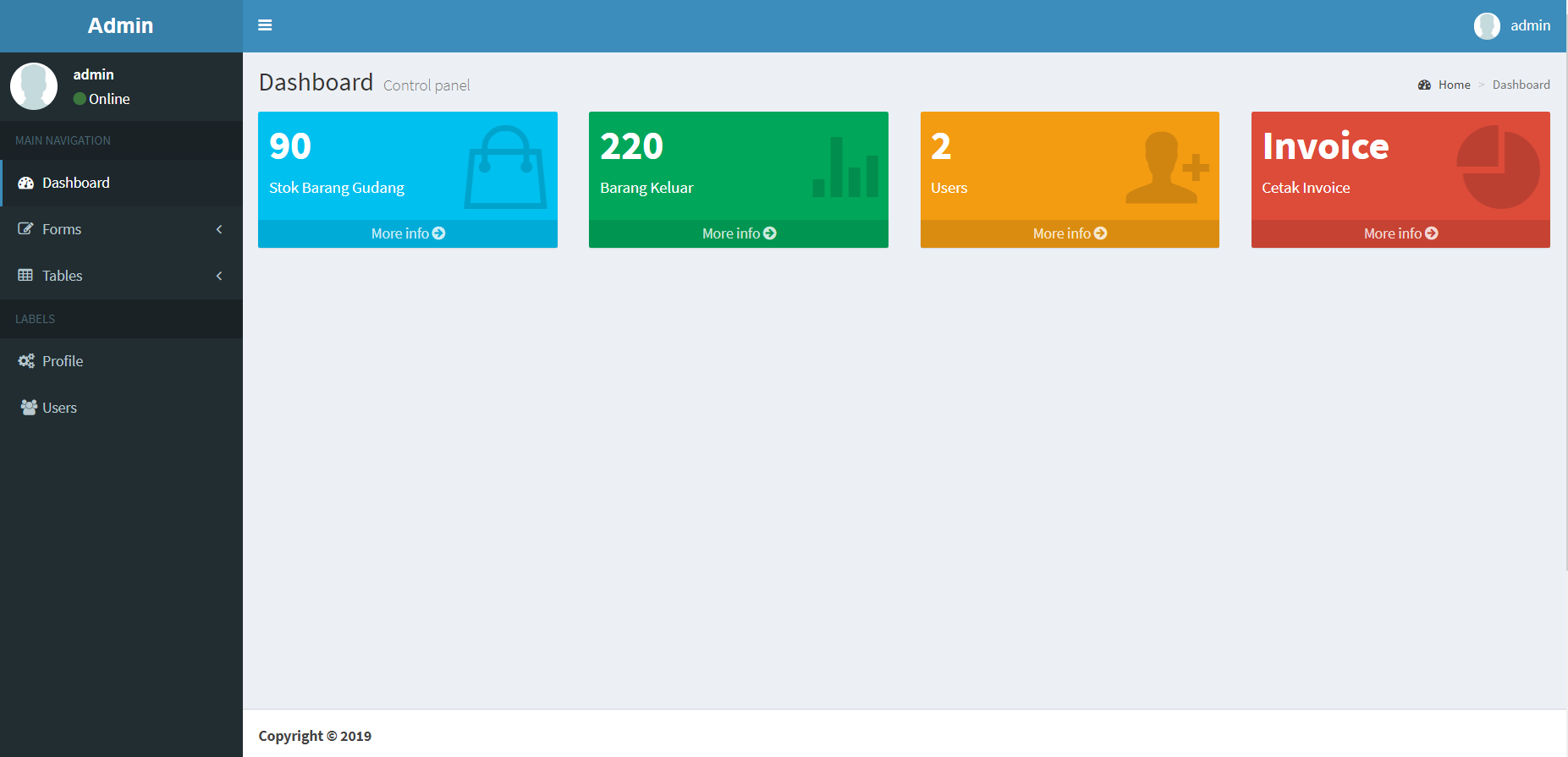
Halaman ini akan muncul saad admin mengakses alamat sistem contoh <http://localhost/KPinventory/>. Pada halaman ini admin diwajibkan memasukan username dan passwoard agar bisa masuk ke halaman dashboard.



Gambar 4.20 Tampilan Halaman Login

1. Halaman dashboard

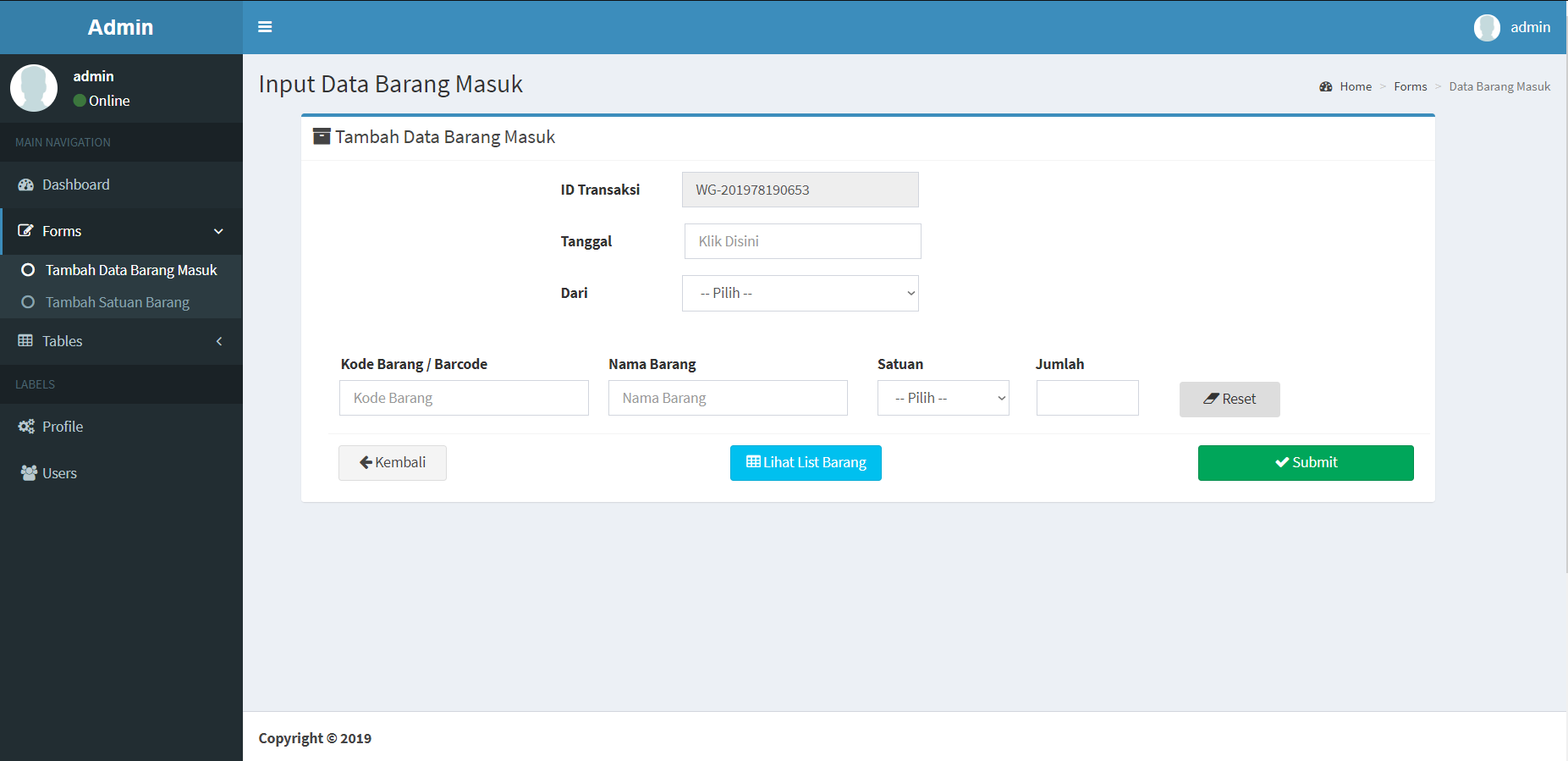
Halaman dashboard berisi informasi barang masuk, barang keluar, jumlah user dan i*nvoice.*



Gambar 4.21 tampilan Halaman Dashboard

1. Halaman Input Data Barang Masuk

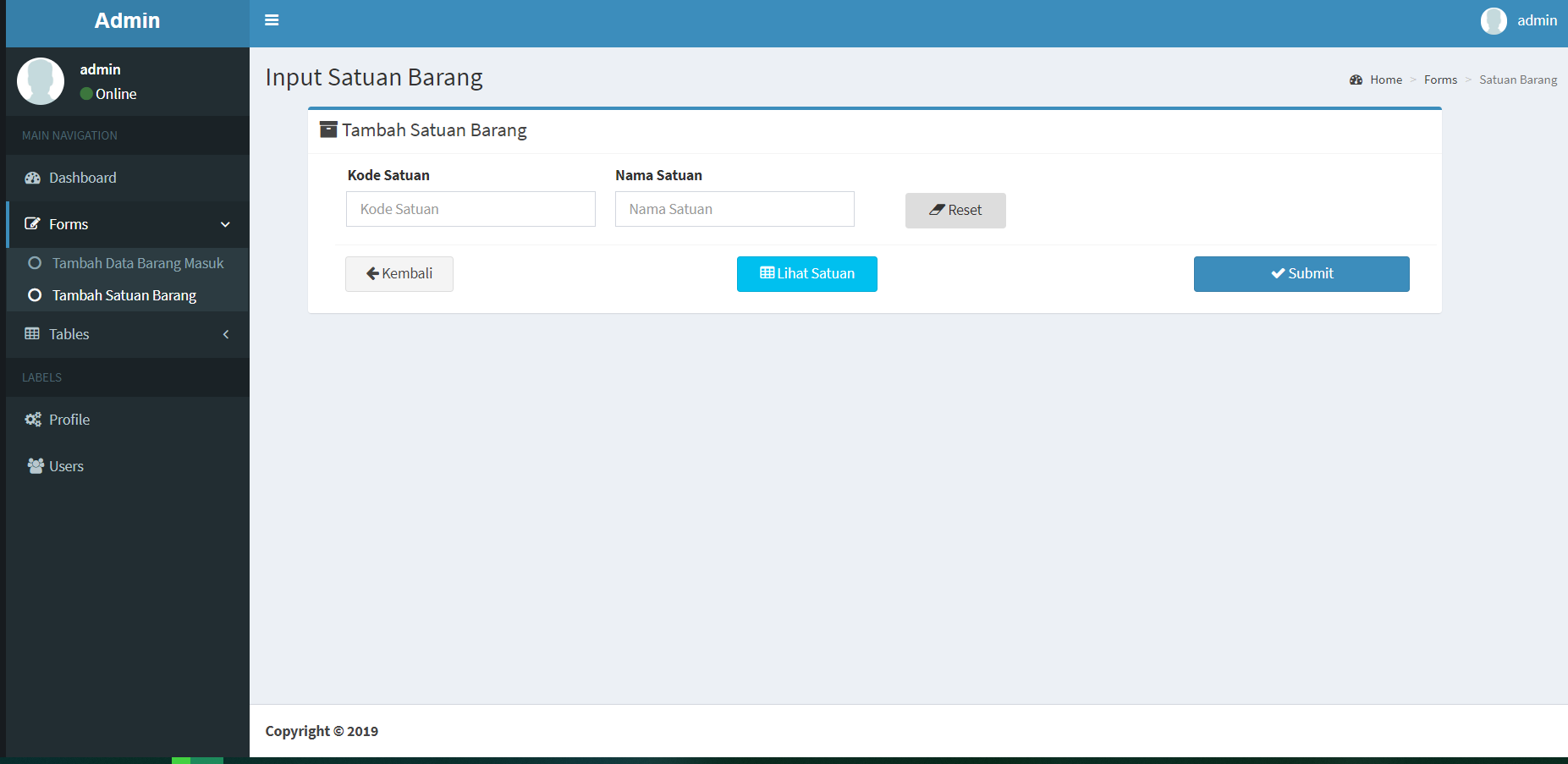
Halaman ini digunakan untuk memasukan barang baru yang masuk ke gudang.



Gambar 4.22 tampilan Halaman Input Barang Masuk

1. Tambah Satuan Barang

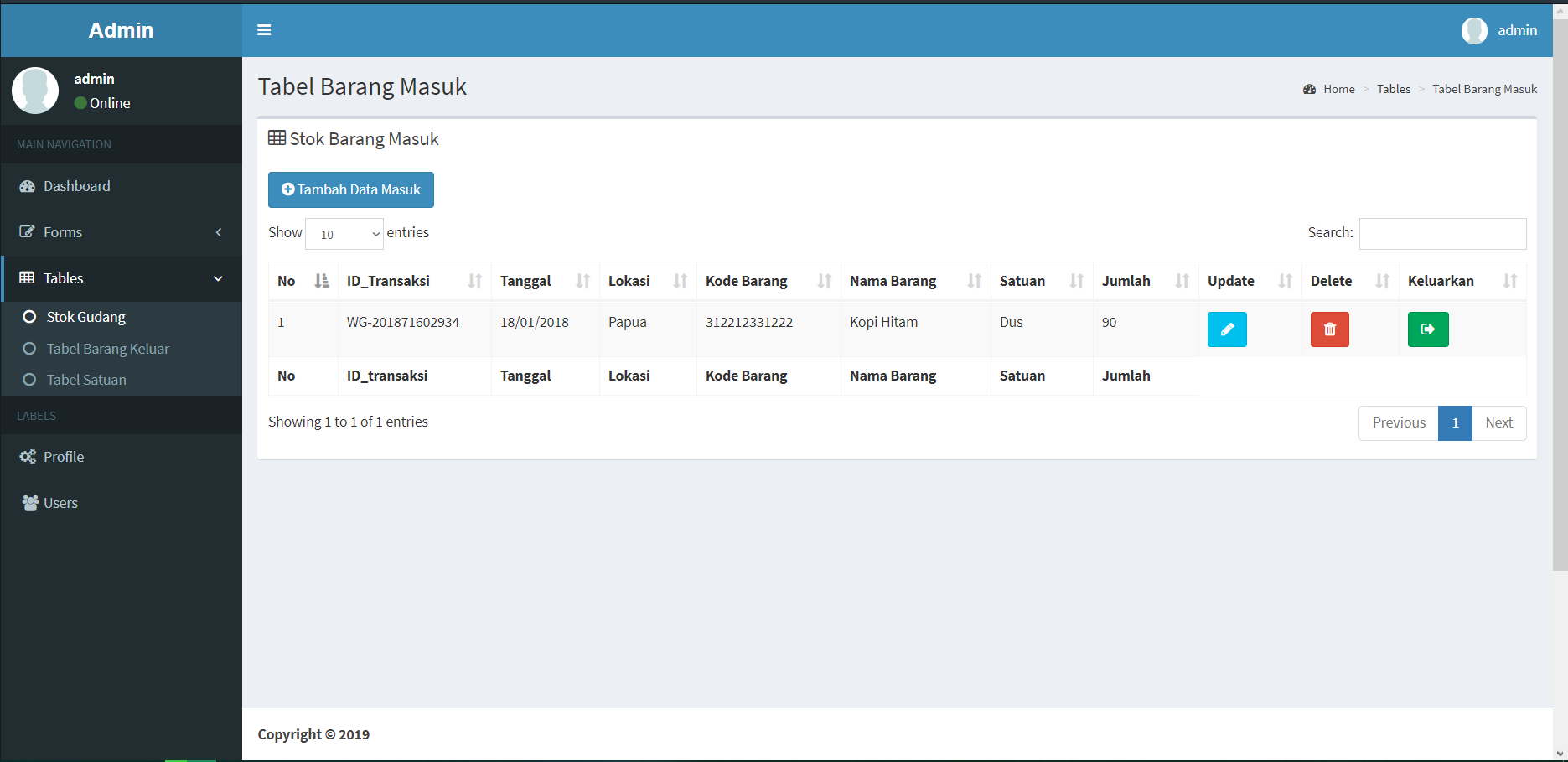
Halaman tambah satuan digunakan untuk menentukan satuan suatu barang seperti dus, pcs, pack.

****

Gambar 4.23 tampilan Halaman Tambah Satuan

1. List Barang Masuk

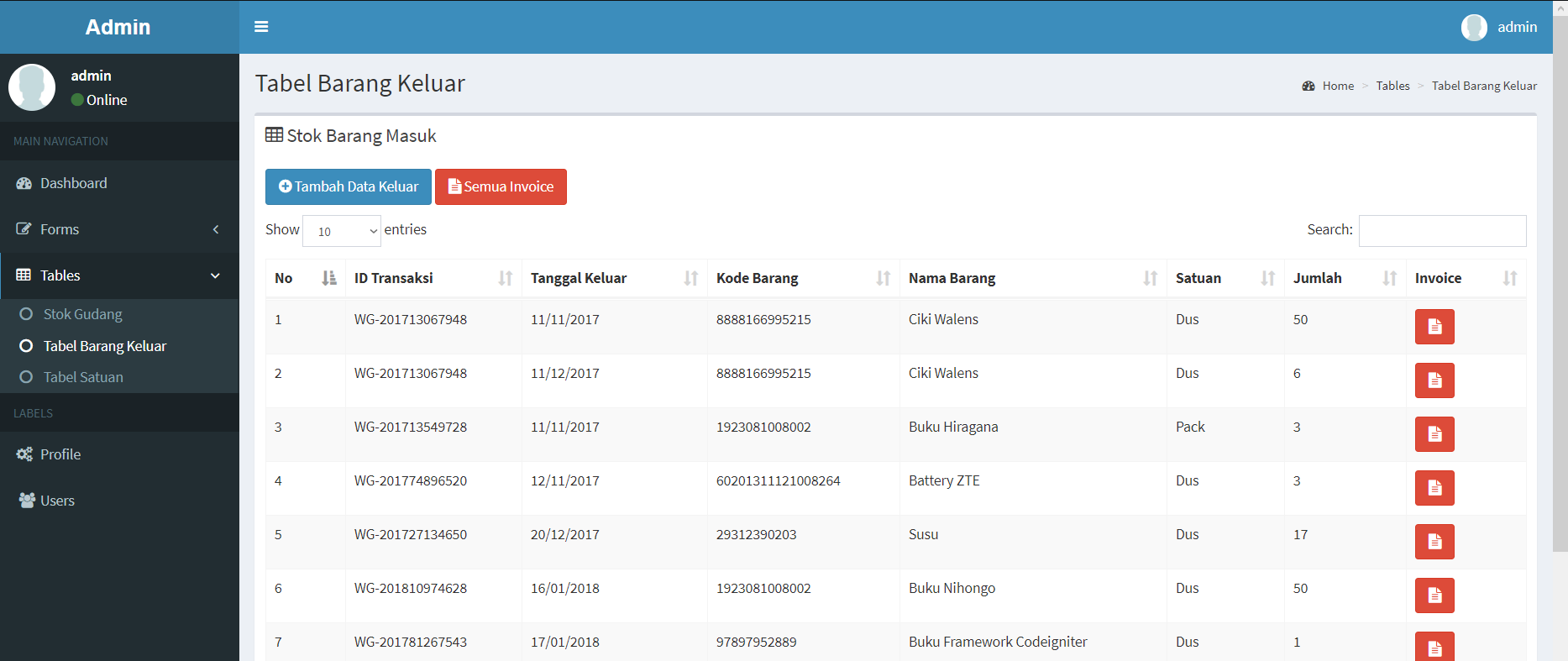
Halaman ini digunakan untuk mengetahui stock barang dan mengeluarkan barang.

****

Gambar 4.24 Tampilan halaman list barang masuk atau stok barang

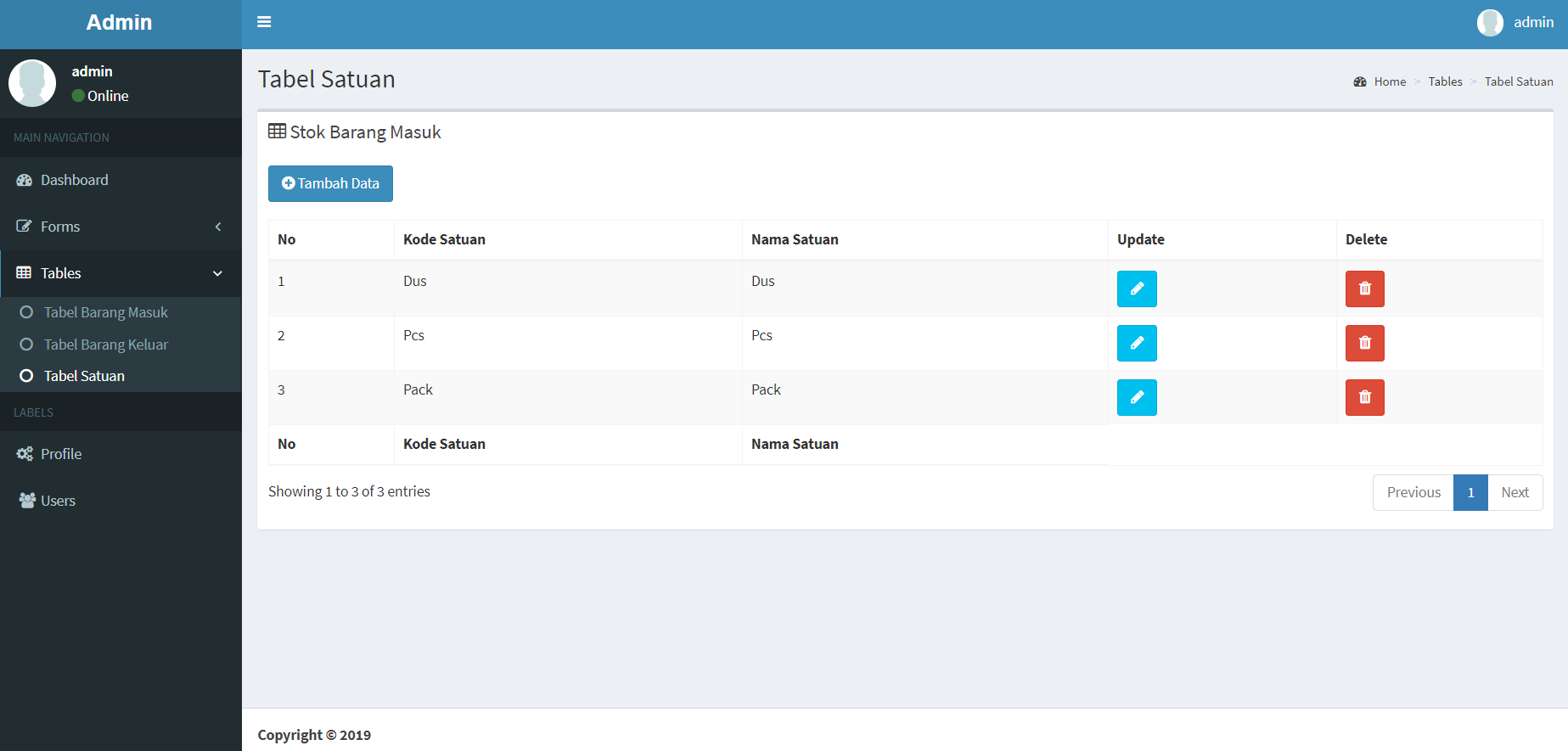
1. List Barang Keluar

Halaman ini digunakan untuk cetak detail barang yang keluar invoice .



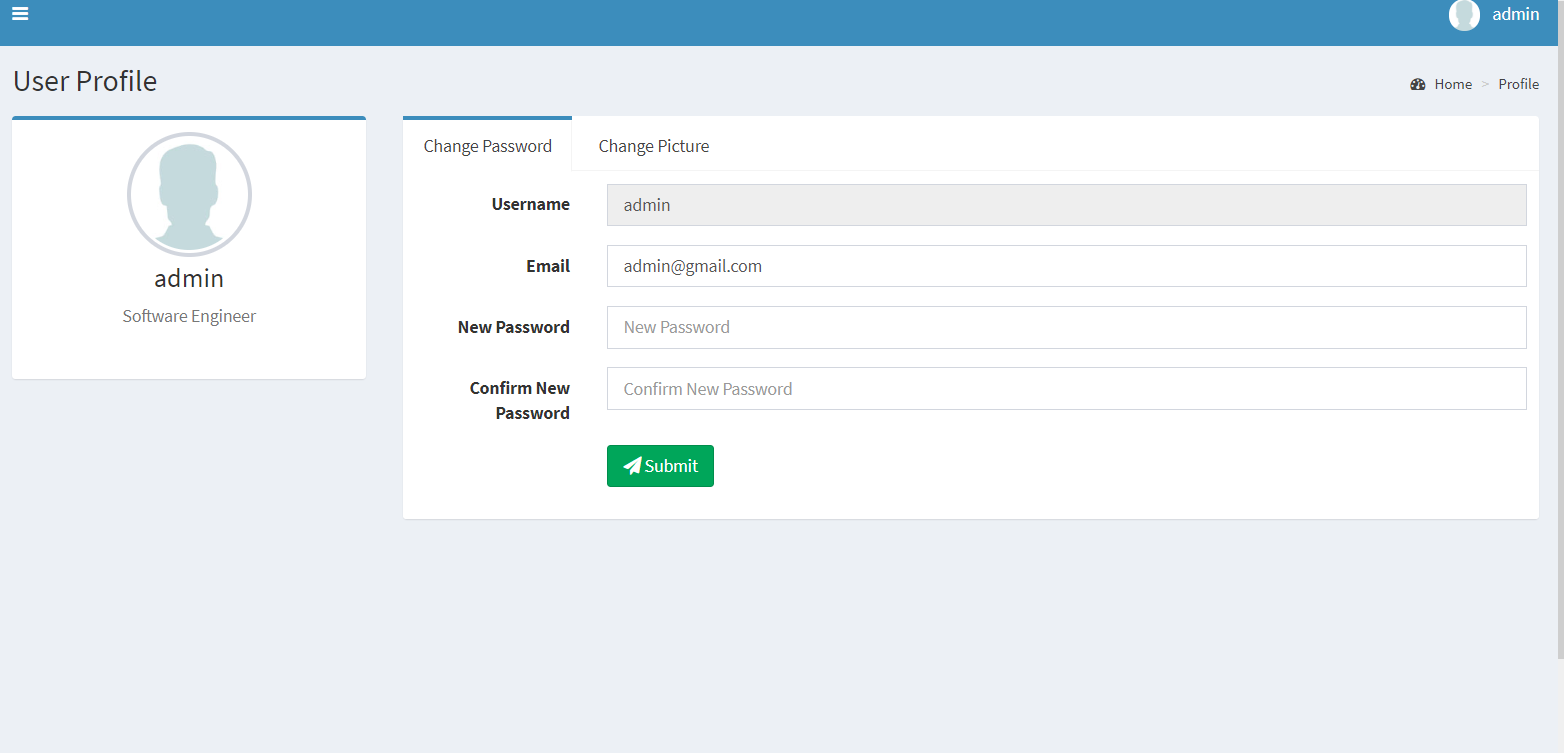
Gambar 4.25 Tampilan Halaman Data Barang Keluar atau *Invoice*

1. List Satuan

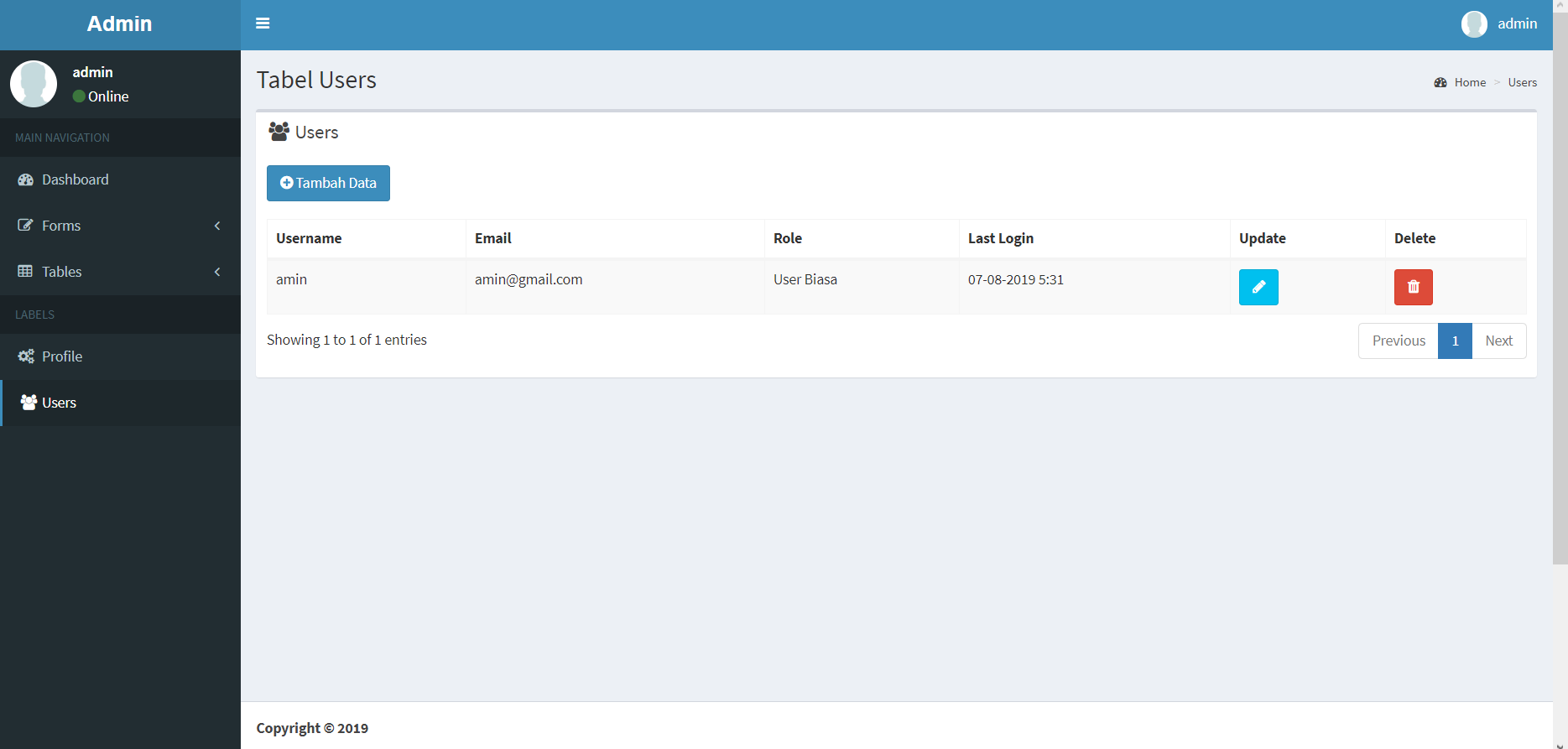


Gambar 4.26 Tampilan Halaman List Satuan

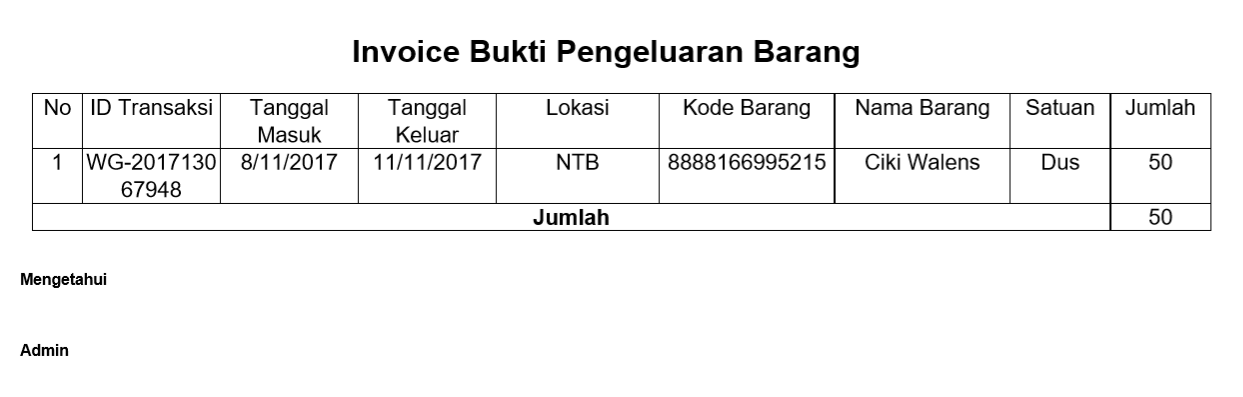
1. Profil

Halaman ini berfungsi untuk mengetahui admin yang login.

Gambar 4.27 Tampilan Halaman Profil Admin

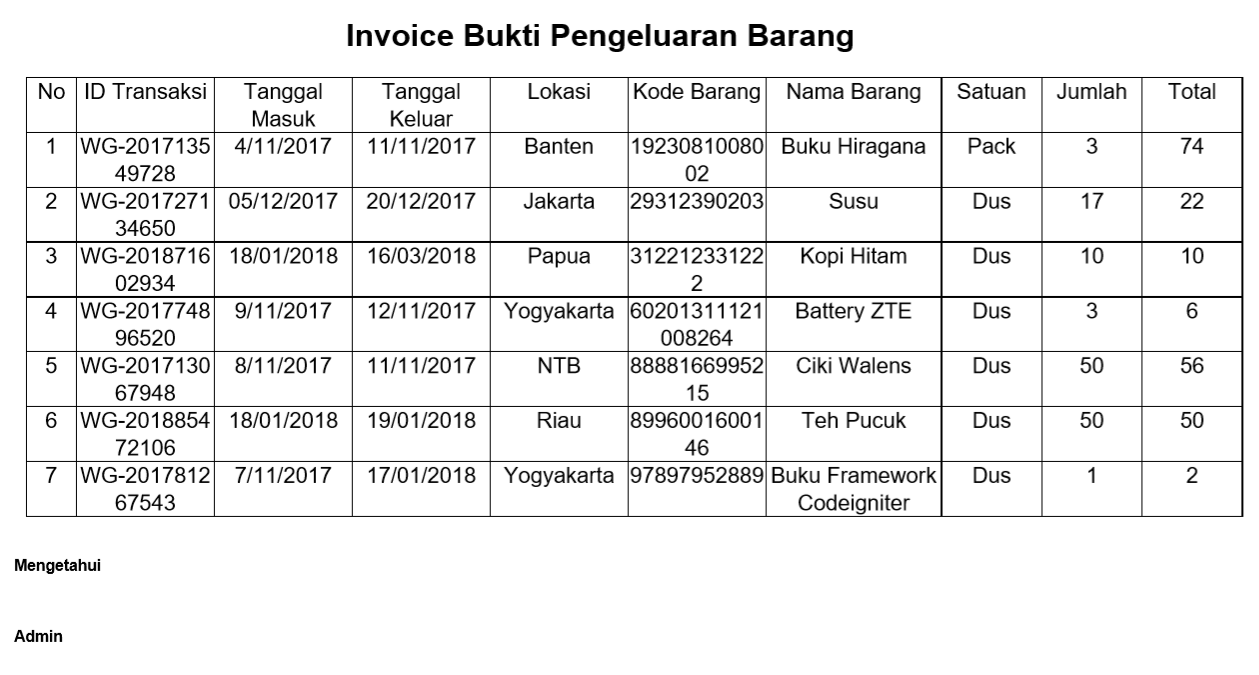
1. ****User

Gambar 4.28 Tampilan Halaman Data admin

1. Hasil cetak invoice

Gambar 4.29 Tampilan Cetak *Invoice*

1. Hasil Cetak Semua Invoice



Gambar 4.30 Tampilan Cetak All *Invoice*