# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Kehidupan sebuah perusahaan sangat bergantung pada logistik dan pergudangan. Mengingat perannya yang sangat penting, kontrol yang tepat pada logistik akan memberikan peningkatan produktivitas dan pelayanan terhadap pergudangan yang akan mempengaruhi pada performa perusahaan secara keseluruhan.

Untuk menganalisa kinerja proses logistik, maka perlu dilaksanakan *stock opname*. *Stock opname* merupakan prosedur wajib dalam sistem logistik dan pergudangan untuk menganalisa apakah telah berjalan dengan efektif dan efisien kinerja proses logistik tersebut. Namun pada prakteknya, proses *stock opname* ini masih terbilang manual dengan mencatat satu persatu barang yang terdapat digudang. Hal tersebut tentunya akan memakan waktu dan diperlukan ketelitian yang tinggi agar data yang diambil tidak salah.

Pesatnya teknologi dan informasi saat ini didukung dengan berkembangnya sistem operasi pada *smartphone* karena beragam fitur dapat ditampilkan untuk memenuhi kebutuhan dan daya tarik tersendiri bagi penggunanya. Jenis-jenis sistem operasi *smartphone* diantaranya adalah *Android, Windows Mobile, Blackberry, Symbian, Iphone* dan sebagainya.

Sistem operasi *Android* merupakan salah satu sistem operasi yang belakangan ini berkembang dengan pesat. Terdapat keunggulan dari sistem operasi ini antara lain dapat diubah atau di modifikasi sesuai dengan keinginan penggunanya dan banyaknya aplikasi komputer yang sudah tersedia. Selain itu, *Android* mudah digunakan dan bisa didapatkan dengan harga terjangkau.

Saat ini, teknologi komunikasi dan informasi terdapat suatu perkembangan yang signifikan dalam penggunaan kode dua dimensi misalnya *pdf147, Datamatrix, Maxicode, Bar Code* dan *QR (Quick Response) Code* (Ariadi, 2011).

*QR Code* adalah simbol *matriks* yang terdiri dari sebuah untaian kotak persegi yang disusun dalam sebuah pola persegi yang lebih besar. Kotak persegi ini kemudian disebut sebagai modul. Luasnya pola persegi ini akan menentukan versi dari *QR Code*.

*QR Code* sangat banyak digunakan dalam kehidupan sehari-hari. *QR Code* cukup membantu para konsumen untuk mendapatkan informasi atas produk yang mereka konsumsi. Seperti halnya *barcode*, di balik kode *matriks* itu tersimpan data, yakni informasi kontak, alamat *e*-*mail*, URL, nomor telepon, nama perusahaan, *geoposition,* jadwal acara, dan *teks* (Hadi, 2009). *QR Code* mengandung informasi yang hanya bisa dibaca dengan sebuah *decoder (*pembaca *QR Code*) yang berupa perangkat *mobile* berbasis *Android*.

Dengan memanfaatkan aplikasi pihak ketiga sebagai *encoder* (pembuat *QR Code*) seperti yang disediakan pada portal web gratis [*https://www.the-qrcodegenerator.com/*,](https://www.the-qrcode-generator.com/) perangkat *mobile* yang berbasis *Android*, dan fitur yang dimiliki *web browser* seperti *Chrome* dan *Mozilla* *Firefox* maka dibuatlah gambar *QR Code*.

Penulis memanfaatkan gambar *QR Code* tersebut dalam membuat sistem informasi logistik dan proses *stock opname.* Sistem informasi logistik ini nantinya akan membantu dalam pencarian barang. Sedangkan proses *stock* *opname* akan dipermudah dengan menggunakan *QR Code.*

Setiap barang-barang di gudang nantinya akan diberikan gambar *QR Code* yang sudah dibuat dan dibaca oleh perangkat *mobile* berbasis *Android* sebagai *decoder* (pembaca *QR Code*). Informasi detail dari barang yang telah dibaca akan keluar yang diambil dari *database* *server* dan dipanggil ke perangkat *mobile* berbasis *Android*. Setelah memasukkan dan menyimpan jumlah barang yang baru, data *stock opname* dapat dilihat di halaman admin yang berbasis web.

Sehingga kedepannya dalam proses *stock opname* bisa lebih cepat lagi dan tidak manual. Mencermati hal tersebut maka penulis ingin membuat sistem informasi logistic *stock opname* menggunakan *QR Code* berbasis *Android*.

Aplikasi tersebut berguna bagi CV Cahaya Sakti Malang yang merupakan salah satu badan usaha yang bergerak dalam bidang *supplier sparepart* mobil. Karena dalam prakteknya, badan usaha tersebut masih menerapkan sistem informasi logistik dan proses *stock opname* yang manual. Dengan begitu, maka CV Cahaya Sakti Malang bisa lebih optimal dalam pemrosesan logistik dan *stock opname*.

## 1.2 Rumusan Masalah

Adapun rumusan masalah yang akan dibahas dalam skripsi ini adalah:

1. Bagaimana membuat sistem informasi logistik secara *mobile*?
2. Bagaimana mengimplementasikan *QR Code decoder* ke dalam perangkat *mobile* berbasis *Android*?
3. Bagaimana membuat proses *stock opname* yang terintegrasi dengan *QR Code* dan dapat memanggil *data* dari *server* untuk ditampilkan di perangkat tersebut?
4. Bagaimana menampilkan dan mengolah semua data barang dan hasil *stock opname* melalui admin berbasis web?

## 1.3 Batasan Masalah

Adapun batasan masalah dalam skripsi ini adalah:

1. Aplikasi ini diimplementasikan di CV Cahaya Sakti Malang.
2. Aplikasi yang dibuat menggunakan perangkat *Androiod Mobile minimal* versi 4.0.3 *Ice Cream Sandwich*
3. *Software* pendukung perangkat ini adalah *Eclipse*.
4. *QR Code* diterjemahkan menjadi inputan berupa gabungan huruf, gabungan angka ataupun huruf dan angka (alfanumerik).
5. Dalam pembuatannya, aplikasi pihak ketiga digunakan sebagai *encoder* (pembuat *QR Code*) yaitu yang terdapat pada [*https://www.the-qrcodegenerator.com/*](https://www.the-qrcode-generator.com/)

## 1.4 Tujuan Penelitian

Penelitian ini dilaksanakan dengan tujuan:

1. Untuk merancang sebuah perangkat lunak berbasis *Android* yang dapat membantu informasi logistik.
2. Untuk merancang sebuah perangkat lunak berbasis *Android* yang dapat mengidentifikasi *QR Code* yang mengandung informasi kode barang.
3. Untuk memudahkan dan mempercepat proses *stock opname* yang terintegrasi dengan *QR Code* dan dapat memanggil *data* dari *server* untuk ditampilkan di perangkat tersebut.
4. Untuk merancang sebuah perangkat lunak berbasis web yang dapat mengelola data barang dan hasil *stock opname.*

## 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Memberikan wawasan tersendiri bagi penulis tentang penerapan proses logistik dan proses *stock opname* yang dipadukan dengan teknologi *QR Code* secara *mobile.*
2. Memudahkan dalam mengatur logistik dan *stock opname*, baik dalam memberikan informasi maupun mengolahnya dengan hanya men-scan gambar *QR Code* pada barang tersebut.
3. Aplikasi ini dapat membantu *user* dalam mengoptimalkan kinerjanya dengan mempercepat proses *stock opname* tanpa mengurangi tingkat ketelitiannya.
4. Aplikasi ini dapat membantu admin dalam mengolah data barang dan *stock opname*.

# BAB II LANDASAN TEORI

## 2.1 Studi Penelitian-penelitian Terdahulu

*QR Code* adalah sebuah metode penyimpanan data yang sederhana. Data yang dapat disimpan juga bermacam-macam. Selain itu kebutuhan untuk menghasilkan dan membaca *QR Code* tidak banyak, cukup dengan menggunakan kamera ponsel yang sudah banyak dimiliki di era modern ini. *QR Code* memiliki keunggulan kapasitas besar, mudah dibaca dari berbagai arah, ukuran kecil, tahan terhadap kotor dan rusak.

Menurut penelitian yang dilakukan oleh Antonius Hendry Setyawa (2012) dari jurnal yang berjudul Perancangan Aplikasi Sistem Presensi Mahasiswa Menggunakan *QR Code* Pada Sistem Operasi Android membuktikan bahwa *QR Code* dapat dimanfaatkan sebagai alat identifikasi mahasiswa pada sistem presensi dan data mahasiswa dapat disimpan dalam bentuk gambar *QR Code.*

Atas dasar penelitian terdahulu tersebut saat ini *QR Code* dimanfaatkan untuk proses *stock opname,* dan gambar *QR Code* nya dimanfaatkan untuk menampung data nomor part.

## 2.2 Studi Pustaka

### 2.2.1 Pengertian Sistem Informasi Logistik

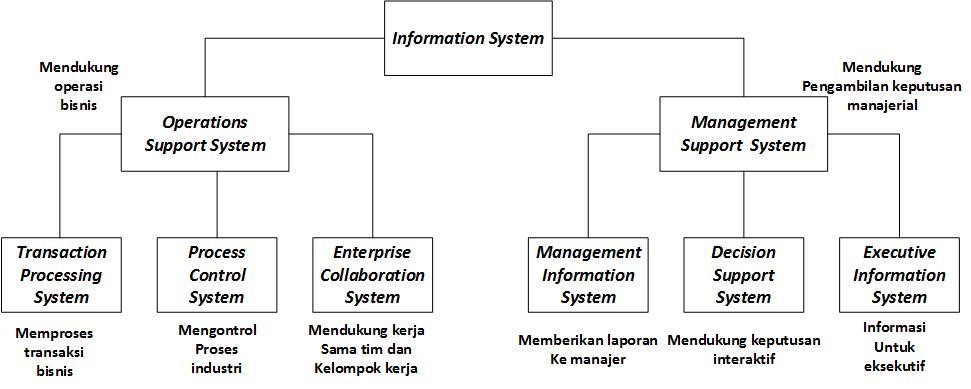
Menurut Al Fatta (2007), untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Akhirnya Sistem Informasi dapat didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada progres.

Sedangkan menurut Marimin (2006), sistem informasi adalah suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Sistem informasi dapat juga didefinisikan sebagai suatu sistem yang menerima sumber data sebagai *input* dan mengolahnya menjadi produk informasi sebagai *output*. Sistem informasi merupakan sistem yang terdiri dari beberapa subsistem atau komponen *hardware, software* dan *brainware*, data dan prosedur untuk menjalankan *input, proses, output,* penyimpanan, dan pengontrolan yang mengubah sumber data menjadi informasi. Atau dapat juga didefinisikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Data merupakan bahan mentah (*raw material*) untuk suatu informasi. Perbedaan informasi dan data sangat relative, tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya. Informasi merupakan data yang telah diolah menjadi suatu bentuk yang penting bagi penerima dan mempunyai nilai nyata bagi pengambilan keputusan-keputusan saat ini atau waktu yang akan datang. Informasi memberikan sesuatu yang berguna jika sesuai dengan kebutuhan *end user,* mempunyai ketelitian dalam pengolahan data, tidak kadaluwarsa (*up to date*) dan dapat dipergunakan secara efektif.

Secara konseptual, menurut O’Brien (2004) *dalam* Marimin (2006), mengklasifikasikan sistem informasi tersebut menjadi dua kelompok besar. Kelompok pertama yaitu *operations support system* yang terdiri dari *transaction processing system, process control system,* dan *enterprise collaboration system.* Kelompok kedua yaitu *management support system* yang terdiri dari *management information system, decision support system,* dan *executive information system.*



**Gambar 2.1** Model Sistem Informasi

Sistem Informasi yang profesional adalah sistem informasi yang mampu menyajikan informasi secara tepat, cepat, aman, dan akurat. Hal ini dapat terwujud apabila SIM tersebut ditunjang oleh sumber daya manusia yang profesional. Beberapa variabel yang digunakan untuk mendefinisikan logika kerangka kerja SI sebagai karakteristik terdiri dari beberapa faktor, adalah:

* Akurat dan teruji
* Lengkap dan detail
* Aktual dan mutakhir (mengikuti teknologi modern)
* Frekuensi penggunaanya sering
* Sumber informasi jelas
* Relevan dan tepat
* Dapat dinilai
* Dapat diterima.

Sistem Informasi juga memegang peranan yang sangat penting dalam organisasi, di antaranya:

1. Mendukung operasional organisasi sehingga organisasi mampu beroperasi secara efisien.
2. Mendukung pengambilan keputusan oleh manajerial organisasi di mana sistem informasi menyajikan informasi yang akurat dan tepat waktu serta tersaji dalam bentuk sesuai dengan yang diinginkan.
3. Mendukung strategi bisnis organisasi dan implementasi strategi sehingga organisasi mampu bertahan (dalam persaingan) dan bahkan lebih maju (Marimin, 2006).

Menurut Bateman (2008), Logistik adalah pemindahan sumber daya ke dalam organisasi (masuk) dan produk-produk dari organisasi kepada para pelanggan (keluar).

Logistik merupakan seni dan ilmu yang mengatur mengatur atau mengontrol arus barang, energi, informasi, dan sumber daya lainnya, seperti produk, jasa, dan manusia, dari sumber produksi ke pasar dengan tujuan mengoptimalkan penggunaan modal Manufaktur dan marketing akan sulit dilakukan tanpa dukungan logistik. Logistik juga mencakup integrasi informasi, transportasi, inventori, pergudangan, reverse logistics dan pemaketan.

Berdasarkan pengertian di atas, maka misi logistik adalah "mendapatkan barang yang tepat, pada waktu yang tepat, dengan jumlah yang tepat, kondisi yang tepat, dengan biaya yang terjangkau, dengan tetap memberikan kontribusi profit bagi penyedia jasa logistik"

Logistik selalu berkutat dalam menemukan keseimbangan untuk 2 hal yang amatlah sulit untuk disinergikan, yaitu menekan biaya serendah-rendahnya tetapi tetap menjaga tingkat kualitas jasa dan kepuasan konsumen.Dalam dunia bisnis yang selalu berubah, manajemen logistik yang baik merupakan sebuah keharusan. Menurut Siagian (2005), Sistem Informasi Logistik dapat dijelaskan dalam beberapa bagian fungsi dan operasi internalnya. Tujuan utamanya adalah berupaya mengumpulkan, memperkuat, dan memanfaatkan data perusahaan sebagai dasar pengambulan keputusan tentang strategi yang akan digunakan serta memfasilitasi transaksi bisnis.

Sedangkan Sistem Informasi Manajemen Logistik merupakan sistem pengelolaan persediaan barang habis pakai yang dapat memberikan kemudahan dalam pengelola data dan informasi logistik, menyediakan data logistik yang akurat untuk keperluan perencanaan kebutuhan barang, analisa kebutuhan dan distribusi, Membantu kelancaran pekerjaan di bidang logistik, terutama dalam *stock opname* yang memegang peranan yang penting.

### 2.2.2 Quick Response Code

Menurut Wahyono (2010), *barcode* atau kode baris digambarkan dalam bentuk baris hitam tebal dan tipis yang disusun berderet sejajar horizontal. Satu unit barcode sebenarnya terdiri dari salah satu warna hitam atau putih. Sebuah unit yang berwarna hitam ditunjukkan dengan sebuah bar, sedangkan yang berwarna putih ditunjukkan dengan sebuah spasi.

*Barcode* adalah suatu kumpulan *data* optik yang dapat dibaca oleh mesin. Manusia tidak bisa membacanya karena *barcode* ini berbentuk gambar lebar garis dan spasi garis pararel atau simbologi linear 1D (1 dimensi). Selain 1D (1 dimensi), ada juga bentuk persegi, titik, heksagon dan bentuk geometri lainnya di dalam gambar yang disebut kode *matriks* atau simbologi 2D (2 dimensi) atau *QR Code*. Kode QR (*Quick Response*) atau biasa dikenal dengan istilah *QR Code* adalah bentuk evolusi *barcode* dari satu dimensi menjadi dua dimensi. Kode QR adalah suatu jenis kode *matriks* atau *barcode* dua dimensi dengan fungsionalitas utama yaitu dapat dengan mudah dibaca dan untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan *barcode*, yang hanya menyimpan informasi secara *horizontal*, kode QR mampu menyimpan informasi secara *horizontal* dan *vertikal*, oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak daripada *barcode*.

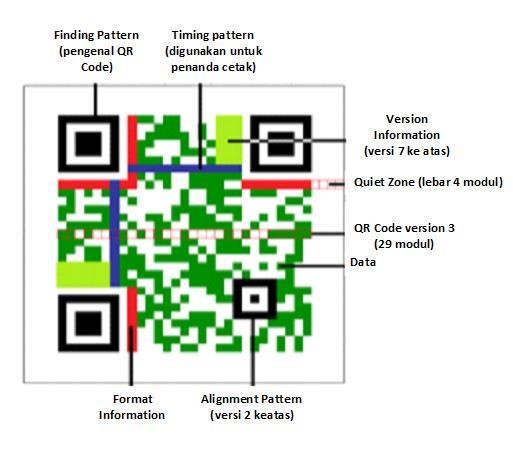
*QR Code* adalah simbol *matriks* dua dimensi yang terdiri dari sebuah untaian kotak persegi yang disusun dalam sebuah pola persegi yang lebih besar. Kotak persegi ini kemudian disebut sebagai modul. Luasnya pola persegi ini akan menentukan versi dari *QR Code*. Gambar 2.2 berikut adalah salah satu contoh bentuk *QR Code*.



## Gambar 2.2 Contoh *QR Code*

### 2.2.2.1 Struktur *QR Code*

*QR Code* memiliki bagian-bagian yang akan dipaparkan pada gambar 2.3 di bawah ini.



**Gambar 2.3** Struktur *QR Code* (Ariadi, 2011)

Gambar di atas menyajikan struktur dari sebuah *QR Code*. Istilah – istilah yang berkenaan dengan *QR Code* adalah :

* *Finding Pattern*: Pola untuk mendeteksi posisi dari *QR Code*.
* *Timing pattern*: Pola yang digunakan untuk identifikasi koordinat pusat dari *QR Code*. Dibuat dalam bentuk modul hitam putih bergantian.
* *Version Information*: Versi dari sebuah *QR Code*. Versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (177 x 177 modul).
* *Quiet Zone*: Daerah kosong dibagian terluar *QR Code* yang mempermudah mengenali pengenal QR oleh sensor CCD.
* *QR Code* version: Versi *QR Code*. Pada contoh gambar versi yang digunakan adalah versi 3 (29 x 29 modul).
* *Data*: Daerah tempat *data* tersimpan (dikodekan).
* *Alignment Pattern*: Pola yang digunakan untuk memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
* *Format information*: Informasi tentang error correction *level* dan mask pattern.

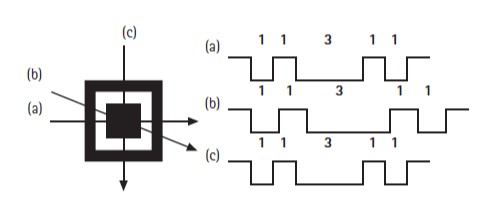
*QR Code* dikembangkan sebenarnya adalah untuk mengambil kelebihan dari *Pdf147*, kepa*data*n *data* yang tinggi dari dari *Data*matrix, dan kecepatan membaca dari Maxicode. Simbol dua dimensi umumnya berisi lebih banyak *data* jika dibandingkan dengan simbol *linier*, kurang lebih 100 kali lebih banyak (Soon, 2008).

### 2.2.2.2 Karakteristik *QR Code*

Karakteristik dari kode dua dimensi *QR Code* yang membedakannya dengan *barcode* adalah dapat menampung jumlah data yang besar. Secara teori sebanyak 7089 karakter numerik maksimum data dapat tersimpan di dalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol linier) dan pembacaan kode dengan cepat. *QR Code* juga memiliki kelebihan lain baik dalam hal unjuk kerja dan fungsi.

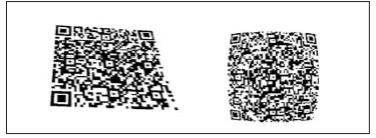
1. Pembacaan Data dari Segala Arah (3600)

Pembacaan kode matriks dengan menggunakan sensor kamera CCD (*Charge Coupled Device*) dimana data akan memindai baris per baris dari citra yang ditangkap dan kemudian disimpan dalam memori. Dengan menggunakan suatu perangkat lunak tertentu, detail citra akan dianalisa, *finding pattern* akan dikenali dan posisi simbol dideteksi. Setelah itu proses pembacaan kode akan diproses. Sedangkan pada simbol linier ataupun kode dua dimensi dimensi lain akan memakan lebih lama waktu untuk mendeteksi letak atau sudut ataupun besar dari simbol tersebut. *QR Code* memiliki *finding pattern* untuk memberitahukan letak simbol matriks dua dimensi *QR Code* yang disusun pada ketiga sudutnya. Hal inilah yang membuat *QR Code* dapat dibaca dari segala arah atau 360 derajat. Rasio antara modul hitam dan modul putih pada finding pattern-nya selalu 1:1:3:1:1 Dengan rasio ini, *finding pattern* dapat mendeteksi keberadaan citra yang ditangkap sensor. Sebagai tambahan, dengan adanya ketiga *finding pattern* maka pengkodean akan lebih cepat dua puluh kali dibandingkan kode matriks lain. Gambar 2.3 berikut ini menggambarkan *finding pattern QR Code*.



**Gambar 2.4** Salah satu *Finding Pattern QR Code* (Soon, 2008)

1. Ketahanan terhadap Penyimpangan simbol Simbol matriks 2 dimensi akan rentan terhadap penyimpangan bentuk ketika ditempatkan pada permukaan yang tidak rata (bergelombang) sehingga sensor pembaca menjadi miring karena sudut antara sensor CCD dan simbol matriks 2 dimensi ini telah berubah. Untuk memperbaiki penyimpangan ini, *QR Code* memiliki perata pola (*alignment pattern*) yang menyusun dengan jarak yang teratur dalam suatu daerah. *Alignment pattern* akan memperhitungkan titik pusat dengan daerah terluar dari simbol matriks, sehingga dengan cara ini penyimpangan linier maupun non-linier masih dapat terbaca. Gambar 2.4 berikut ini adalah contoh penyimpangan yang terjadi pada *QR Code*.



**Gambar 2.5** Jenis Penyimpangan pada *QR Code*

1. Fungsi Pemulihan Data (Ketahanan terhadap kotor maupun kerusakan) *QR Code* mempunyai empat tingkatan koreksi *error* (7%, 15%, 25% dan 30%). Dalam mengendalikan kerusakan yang diakibatkan kotor ataupun rusak, *QR Code* memanfaatkan algoritma Reed-Solomon yang tahan terhadap kerusakan tingkat tinggi. Jadi, ketika *QR Code* akan digunakan dalam lingkungan yang rawan kerusakan akibat lingkungan, disarankan menggunakan koreksi *error* 30%. Gambar 2.5 berikut adalah contoh gambar *QR Code* yang kotor atau rusak.



**Gambar 2.6** Kerusakan pada *QR Code*

1. Kemampuan enkode karakter Kanji dan Kana Jepang

*QR Code* berkembang pesat di negara Jepang. Hal ini yang menyebabkan perkembangan *QR Code* untuk dapat menerima input data berupa karakter yang non-alfabetis. Ketika pembuatan *QR Code* dengan inputan berupa huruf Jepang, maka data tersebut akan diubah kedalam bentuk biner 16 bit (2 byte) untuk karakter tunggal sedangkan untuk gabungan karakter akan di-enkode dalam biner 13 bit. Hal ini memberikan keuntungan lain dimana proses enkode huruf Jepang akan meningkatkan efisien 20 % lebih banyak dari kode simbol dua dimensi lain. Dimana dengan volume data yang sama akan dapat dibuat pada area pencetakan yang lebih kecil.

1. Fungsi *Linking* pada Simbol

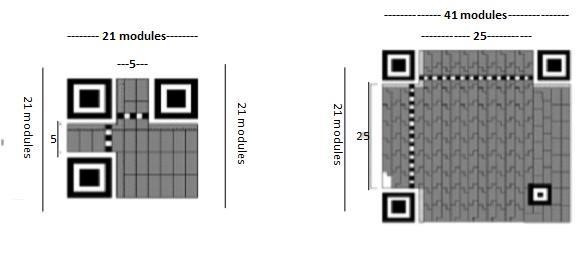
*QR Code* memiliki kemampuan dapat dipecah menjadi beberapa bagian dengan maksimum pembagian adalah 16 bagian (Soon, 2008). Dengan fungsi *linking* ini maka *QR Code* dicetak pada daerah yang tidak terlalu luas untuk sebuah *QR Code* tunggal.

1. Proses *Masking*

Proses *Masking* pada *QR Code* berperan sangat penting dalam hal penyusunan modul hitam dan modul putih agar memiliki jumlah yang seimbang. Untuk memungkinkan hal ini digunakan operasi XOR yang diaplikasikan diantara area data dan daerah *mask pattern*. Ada sebanyak delapan *mask pattern* dalam *QR Code* yang kesemuanya itu dalam bentuk biner tiga bit.

### 2.2.2.3 Versi *QR Code* dan Ukurannya

Ada 40 jenis ukuran dari *QR Code* yang dinyatakan dengan versi 1, versi 2, hingga versi 40. Versi 1 berukuran 21 x 21 modul, versi 2 berukuran 25 x 25 modul dan seterusnya dimana apabila versi meningkat satu maka jumlah modul akan meningkat sebanyak 4 modul x 4 modul.



**2.2.3**

**Gambar 2.7** *QR Code* versi 1 (kiri) dan versi 2 (kanan) (Ariadi, 2011)

### Pengertian *Stock Opname*

*Stock opname* adalah proses menghitung jumlah barang yang ada di gudang dan mencocokkannya dengan catatan pembukuan persediaan, Jika ternyata jumlah di gudang tidak sama dengan catatan, maka perlu dilakukan koreksi atas jumlah barang persediaan (Himayati, 2008). *Stock opname* merupakan istilah lain dari penghitungan fisik persediaan. Tujuan diadakannya *stock opname* adalah untuk mengetahui kebenaran catatan dalam pembukuan, yang mana merupakan salah satu fungsi sistem pengendalian intern (SPI). Dengan diadakannya *stock opname* maka akan diketahui apakah catatan dalam pembukuan stock persediaan benar atau tidak. Jika ternyata ada selisih antara *stock opname* dengan catatan pada pembukuan, kemungkinan ada transaksi yang belum tercatat, atau bahkan ada kecurangan yang berkaitan dengan persediaan.

*Stock opname* biasanya diadakan setiap akhir tahun, tetapi kalau perusahaan dengan SPI yang lebih rapi, *stock opname* biasanya dilakukan tiap triwulan atau caturwulan. *Stock opname* bukan hanya untuk persediaan perusahaan. *Stock opname* juga semestinya dilakukan untuk kas, aktiva, piutang, hutang. Tetapi, perusahaan biasanya hanya melakukan *stock opname* untuk persediaan barang dan kas. Untuk perusahaan manufaktur, *stock opname* persediaan barang dilakukan untuk persediaan bahan baku, bahan penolong, barang setengah jadi, dan barang jadi.

Untuk pengendalian Intern, *stock opname* dilakukan oleh petugas yang bukan merupakan petugas pencatat persediaan, dalam perusahaan biasanya ada petugas audit tersendiri.

Jika setelah dilakukan *stock opname* terjadi selisih maka perlakuan selisih ini biasanya sesuai kebijakan perusahaan. Jika selisih kurang, kekurangannya ini dibebankan perusahaan maka bagian pembukuan membuat jurnal penyesuaian. Tapi jika kebijakan perusahaan mengharuskan petugas persediaan yang harus mengganti kekurangan persediaan maka tidak perlu ada jurnal penyesuaian, kecuali harga untuk penggantian tidak sama dengan harga pokok persediaan.

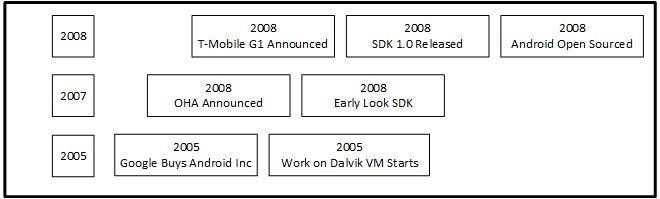
Jika ternyata terjadi selisih lebih pada persediaan barang atau kas maka yang harus dilakukan adalah mengecek kembali catatan bila kemungkinan ada transaksi yang belum dicatat. Untuk selisih kurang pun juga harus dilakukan hal yang sama. Jika ternyata setelah di cek ulang tidak ada transaksi yang terlewatkan maka atas selisih lebih dilakukan jurnal penyesuaian.

#### 2.2.4 *Android*

*Android*, sebagai sebuah sistem, adalah sistem operasi berbasis *Java* yang berjalan pada kernel 2.6 *Linux*. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan mudah menyesuaikan ke *platform* baru. *Android* merupakan satu kumpulan lengkap perangkat lunak yang dapat berupa sistem operasi, middleware, dan aplikasi kunci perangkat *mobile*. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari boot loader, device driver, dan fungsi-fungsi pustaka, hingga perangkat lunak API (Application Programming Interface), termasuk aplikasi SDK (*Software* Development Kit). Jadi, sebenarnya *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai konfigurasi perangkat keras. Walaupun kelas utama perangkat yang didukung oleh *Android* adalah telepon *mobile*, tetapi sekarang ini juga digunakan pada *electronic book readers, netbooks, tablet, dan set-top boxes* (STB) (Collins, 2012).

##### 2.2.4.1 Sejarah *Android*

Telepon *mobile* menggunakan sebuah sistem operasi sejenis *Symbian OS*, Windows *Mobile*, *Mobile* *Linux*, *iPhone OS*, dan banyak sistem operasi lainnya. Standar-standar pendukung dan penerbitan API akan sangat meluas, pengembangan aplikasi *mobile* dengan biaya rendah, tetapi tidak ada sistem operasi yang memegang peranan utama dalam penggunaan *Android*. Kemudian *Google* memasuki bagian ini dengan *Platform* *Android*nya, membuka peluang harapan, harga terjangkau, pengkodean yang bersifat open source, dan sebuah kerangka kerja pengembangan. *Google* memulai perusahaan *Android* Inc. pada tahun 2005, memulai pengembangan *Platform* *Android* (lihat gambar). Pemain-pemain dalam *Android* Inc. termasuklah didalamnya Andy Rubin, Rich Miner, Nick Sears, dan Chris White.



**Gambar 2.8** *Android Timeline*

Pada akhir tahun 2007, sebuah kelompok industri muncul bersamaan dengan

*Platform* *Android* untuk membentuk *Open Handset Alliance* (*http://www.openhandsetalliance.com*). Beberapa diantaranya adalah Sprint Nextel, T-*Mobile*, *Motorola, Samsung, Sony Ericson, Toshiba, Vodafone, Google, Intel, Texas Instrument*.

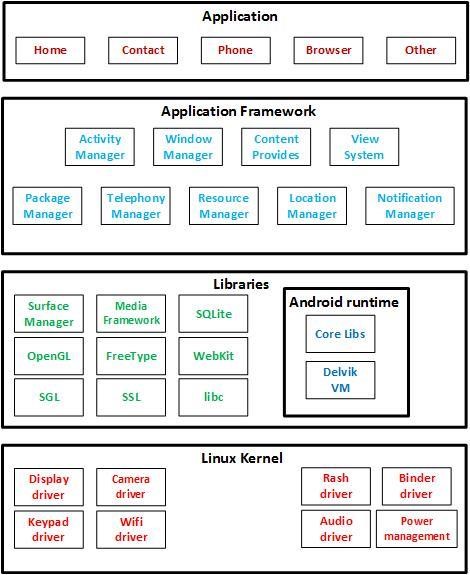
Bagian dari tujuan aliansi adalah mendapatkan paham baru dengan sangat cepat dan merespon kebutuhan konsumen dengan lebih baik, dan *Platform* *Android* sebagai yang utama. *Android* didesain untuk melayani kebutuhan-kebutuhan operator-operator *mobile*, perusahaan penyedia perangkat, dan pengembang aplikasi. Anggota-anggota yang melakukannya merilis hal-hal intelektual yang spesifik melalui *open source Lisensi Apache*, Versi 2.0 (Hashimi & Komatineni, 2009).

##### 2.2.4.2 Platform Android

*Android* dibangun di atas *Linux*. *Linux* adalah sistem operasi yang baik dan bersifat open source. Banyak alasan mengapa memilih *Linux* sebagai dasar dari susunan *Android*. Beberapa hal yang penting adalah portabilitas, keamanan, dan fitur-fiturnya.

*Linux* adalah sebuah *platform* *portable* yang mudah dikompilasi pada bermacam-macam arsitektur perangkat keras. Yang dibawa *Linux* pada *Android* adalah sebuah *level* abstraksi perangkat. Dengan adanya *Android* pada *Linux*, kita tidak perlu terlalu khawatir mengenai fitur-fitur perangkat keras yang menggunakan ini. Kebanyakan bagian-bagian *Linux* pada *level* rendah sudah ditulis dengan pengkodean C yang *portable*, yang mengizinkan bagian ketiga meletakkan *Android* terhadap perangkat jenis apa saja.

*Linux* merupakan sistem operasi dengan keamanan yang tinggi. *Android* mengandalkannya karena keamanan yang dimiliki *Linux*. Seluruh aplikasi *Android* berjalan sesuai dengan proses-proses *Linux* dengan izin-izin yang diatur oleh sistem *Linux* itu sendiri. *Linux* hadir dengan banyak fitur-fitur yang bermanfaat. *Android* mengambil banyak keuntungannya, seperti dukungan manajemen memori, manajemen power, dan jaringan (Gargenta, 2011).



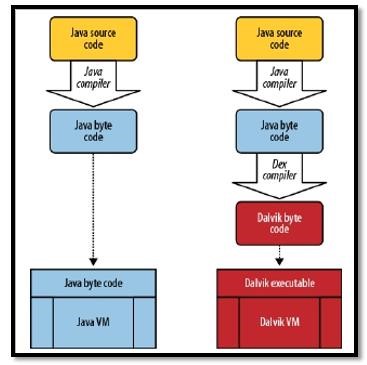
## Gambar 2 9 Tumpukan *Android*

**2.2.4.3**

### *Android* dan *Java*

Pada *Java*, kita menulis *source file* *Java*, mengkompilasinya ke dalam *byte code* *Java* menggunakan kompiler *Java*, kemudian menjalankan *byte code* *Java* pada *Java* VM. Berbeda pada *Android*, ketika *source file* *Java* sudah dikompilasi ke dalam *byte code*, kita harus mengkompilasinya sekali lagi menggunakan *Dalvik* kompiler menjadi *byte code* Dalvik. *Byte code* *Dalvik* inilah yang akan dieksekusi pada *Dalvik VM*. Gambar berikut mengilustrasikan perbandingan antara standar

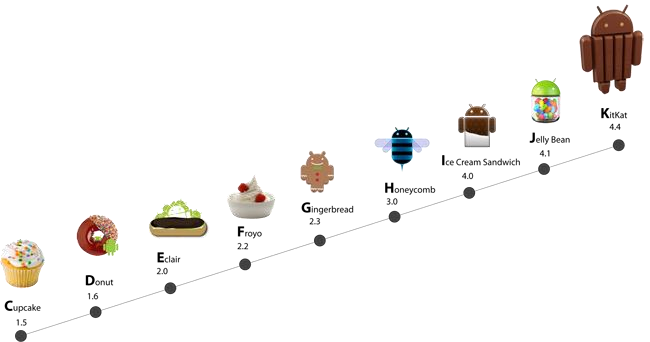
*Java* (disebelah kiri) dengan *Android* yang menggunakan *Dalvik* (disebelah kanan).



## Gambar 2.10 *Android* vs *Java*

### 2.2.4.4 Versi-Versi *Android*

Seperti perangkat lunak lainnya, *Android* terus diperbaharui, yang direfleksikan ke dalam urutan versi-versinya. Berikut ini adalah gambar yang menunjukkan perkembangan versi-versi pada *Android*.



## Gambar 2.11 Versi *Android*

Angka versi *Android* menceritakan tentang besar dan kecil *platform* yang dirilis. Yang paling penting adalah *level* API dari setiap versi. Angka versi berubah setiap waktu. Terkadang disebabkan oleh perubahan API, dan bisa juga karena kecilnya perbaikan bug terhadap kemajuan kinerja. Sebagai informasi, di *Android* *Cupcake* *level* API nya baru 3, sedangkan pada *Android* 4.4 *Kitkat* *level* API sudah mencapai angka 19. Hal itu menunjukan pesatnya perkembangan *Android* hingga saat ini.

### 2.2.4.5 Eclipse IDE (Integrated Development Environment)

*Integrated Development Environment* (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

*Eclipse* adalah sebuah IDE untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi *Eclipse* adalah *Microsoft* *Windows*, *Linux*, *Solaris*, AIX, HP-UX dan Mac OS X.
2. *Multi-language*: *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis pemrograman lainnya, seperti C/C++, *Cobol*, *Phyton*, *Perl*, PHP, dan lain sebagainya*.*
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* dapat digunakan juga untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya (Prasetyo, 2012).

### 2.2.4.6 Android SDK (Sofrware Development Kit)

*Android* SDK adalah *tools* API (*Application* *Programming* *Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* *Android* menggunakan bahasa pemrogrman *Java*. *Android* merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh Google. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan dari *Handphone*/*Smartphone*.

### 2.2.4.7 ADT (Android Development Tools)

ADT (*Android* *Development* *Tools*) adalah *plugin* untuk *Eclipse* yang didesain untuk pengembangan aplikasi *Android*. ADT memungkinkan *Eclipse* untuk digunakan dalam membuat aplikasi *Android* baru, membuat *User* *Interface*, menambahkan komponen berdasarkan *framework* API *Android*, *debug* aplikasi, dan pemaketan aplikasi *Android* (Prasetyo, 2012).

### 2.2.4.8 XML (Extensible Markup Language)

*Extensible* *Markup* *Language* (XML) adalah bahasa markup serba guna yang direkomendasikan oleh W3C (*World* *Wide* *Web* *Consortium*) untuk mendeskripsikan berbagai macam data. XML menggunakan markup tags seperti halnya HTML (*Hypertext* *Markup* *Language*) namun penggunaanya tidak terbatas pada tampilan web saja.

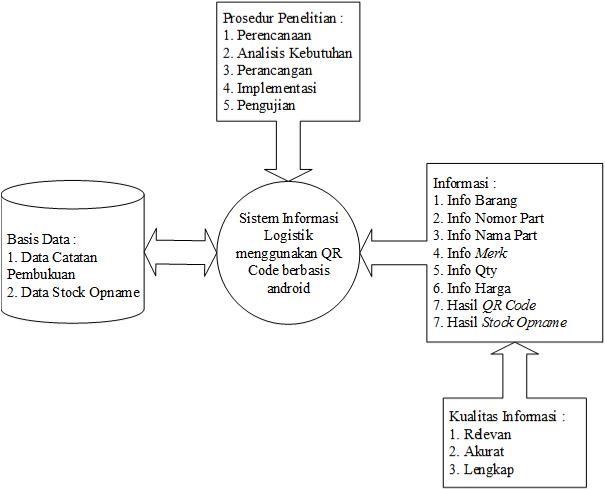
*Layout* mendefenisikan struktur visual pada *user* *interface* (UI), seperi UI dalam *activity* atau *app* *widget*. Pengembang aplikasi *Android* dapat menggunakan XML dalam mendeklarasikan elemen-elemen UI yang dibangun. *Android* menyediakan pembendaharaan kata pada XML.

Kegunaan menggunakan XML dalam pendeklarasian elemen-elemen UI adalah memungkinkan pengembang aplikasi *Android* agar lebih mudah dalam memisahkan tampilan suatu aplikasi dengan code yang mengontrol perilaku tampilan aplikasi tersebut. Pendeskripsian UI yang merupakan bagian yang terpisah dari kode aplikasinya, dapat dimodifikasi dan diadaptasi tanpa harus memodifikasi *source* *code* aplikasi tersebut, apalagi harus melakukan *recompile*.

# BAB III KERANGKA KONSEP PENELITIAN

## 3.1 Kerangka Konsep

Kerangka konseptual dibangun agar penelitian yang dilakukan dapat berjalan terarah sesuai dengan rumusan masalah dan tujuan yang akan dicapai. Berikut ini adalah kerangka konsep yang akan dilakukan:



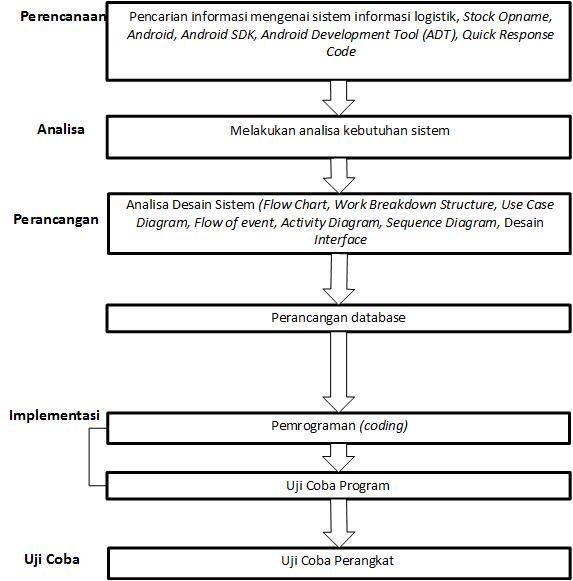
**Gambar 3.1** Kerangka Konsep

Sistem informasi logistik *stock opname* menggunakan *quick response code* berbasis *android* ini dibangun dengan metode SDLC (*Software Development Life Cycle*) dengan model *waterfall.* Metode *waterfall* mempunyai tahapan diantaranya adalah perencanaan, analisis kebutuhan, perancangan, implementasi, dan pengujian*.* Data-data yang dibutuhkan adalah data catatan pembukan (database logistik) dan data stock opname. Keluaran atau *output* yang diharapkan berupa informasi tentang barang logistik, baik nomor dan nama part, info merk (kategori), qty, harga, hingga hasil *QR Code* dan hasil stock opname.

Kualitas informasi yang dikeluarkan harus relevan, akurat dan lengkap berkaitan dengan *record* yang ada.

## 3.2 Tahapan Penelitian

Dalam tahap penelitian ini penulis akan menjelaskan tentang metode SDLC (*Software Development Life Cycle*) dengan model *waterfall.*



**Gambar 3.2** Diagram Tahapan Penelitian

1. Perencanaan

Pada proses perencanaan, penulis mengumpulkan segala informasi terkait sistem yang akan dibangun, seperti Sistem Informasi Logistik, *Stock Opname, Android, Android SDK, Android Development Tool (ADT), QR Code.*

1. Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan *data* dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah konsep sistem *Android* yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai *data* yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrogram.

1. Perancangan Sistem

Desain dan perancangan sistem merupakan pendefinisian dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi. Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, meliputi *Flow Chart, Work Breakdown Structure, Use Case Diagram, Flow of event, Activity Diagram, Sequence Diagram, Desain Interface,* Basis Data.

1. Implementasi

Implementasi sistem dilakukan berdasarkan perancangan aplikasi. Implementasi perangkat lunak mengimplementasikan analisa kebutuhan perangkat sebagai penunjang untuk membangun sistem, meliputi kebutuhan *software* berupa bahasa pemrograman *Java,* PHP dan basis data MySQL serta *tools* pendukung lainnya sehingga dapat mengimplementasikan kebutuhankebutuhan tersebut. Proses perancangan yang dapat dihasilkan adalah sebagai berikut :

* 1. Implementasi *QR Code reader* pada *android* dengan menggunakan bahasa pemrograman *Java.*
  2. Implementasi admin berupa halaman atau tampilan *web* dengan mengimplemeasikan bahasa pemrograman PHP.
  3. Data logistik dimasukkan ke dalam basis data dengan mengimplementasikan basis data MySQL, sehingga data yang tersimpan dalam basis data tersebut akan diolah oleh sistem untuk kebutuhan proses *stock opname*.

1. Pengujian Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

Pengujian sistem dilakukan dengan menggunakan metode *BlackBox Testing*. *BlackBox Testing* memberikan hasil tes kesesuaian aplikasi dengan fungsi atau kemampuan sistem secara fungsional. *BlackBox Testing* yaitu pengujian yang difokuskan terhadap fungsionalitas perangkat lunak tanpa melihat struktur internal program. Sehingga dimungkinkan pembuat program atau petugas penguji mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

*BlackBox Testing* berusaha menemukan kesalahan yang termasuk kategori dibawah ini :

* 1. Fungsi-fungsi yang hilang atau tidak benar.
  2. Kesalahan pada *design interface*.
  3. Kesalahan pada *performance.*
  4. Kesalahan pada inisialisasi dan terminasi.

# BAB IV ANALISA DESIGN SISTEM

## 4.1 Analisis Kebutuhan Sistem

Pada analisis kebutuhan sistem terdiri atas analisis kebutuhan fungsional sistem, kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*), *Flow Chart, Work Breakdown Structure, Use Case Diagram, Flow of event, Activity Diagram, Sequence Diagram, Desain Interface,* Basis Data.

### 4.1.1 Analisis Kebutuhan Fungsional

Berikut adalah beberapa kebutuhan fungsional yang harus dipenuhi oleh sistem, dalam hal ini adalah sistem informasi logistik *stock opname* menggunakan *Quick Response Code* :

1. Menampilkan *QR Code decoder* (*scanner).*
2. Menampilkan gambar *QR Code* yang akan dibaca.
3. Menampilkan hasil gambar *QR Code* yang telah terbaca.
4. Menampilkan hasil detail barang setelah *QR Code* terbaca.
5. Menampilkan hasil pencarian berdasarkan kategori.
6. Menampilkan hasil pencarian berdasarkan kata kunci.
7. Menampilkan hasil pencarian berdasarkan *scan QR Cdoe.*

### 4.1.2 Kebutuhan Perangkat Keras

1. Komputer / Laptop
2. Mouse
3. *Mini USB 2.0 Cable*
4. *Smartphone* Sistem Operasi *Android* versi 4.0 keatas.

### 4.1.3 Kebutuhan Perangkat Lunak

1. Sistem Operasi Windows 8.1
2. Eclipse *Standard* 4.3.1
3. *Sublime Text* 3.0
4. XAMPP sebagai web server
5. MySQL sebagai DBMS untuk database
6. *Browser Chrome*
7. *Android* SDK
8. *Java Development Kit (*JDK)
9. *Android Development Tool (*ADT)
10. *Genymotion*

## 4.2 Desain Alur Sistem

Secara sederhana deskripsi umum sistem dapat dilihat melalui *flowchart* yang terdiri dari sistem barang masuk, sistem barang keluar, dan sistem *stock opname*.

1. Alur Sistem Barang Masuk



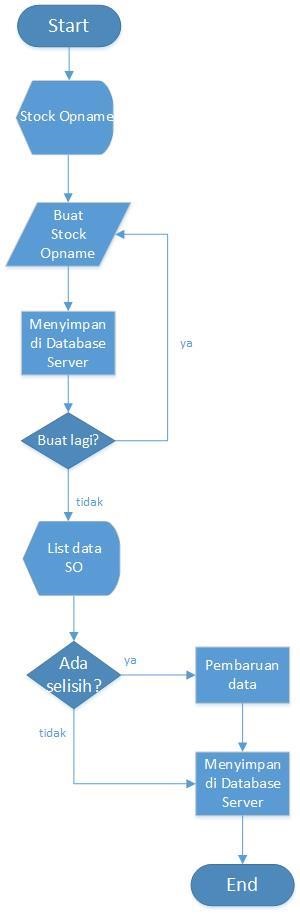
**Gambar 4.1** *Flowchart* sistem barang masuk

1. Alur Sistem Barang Keluar



**Gambar 4.2** *Flowchart* Sistem Barang Keluar

1. Alur Sistem *Stock Opname*



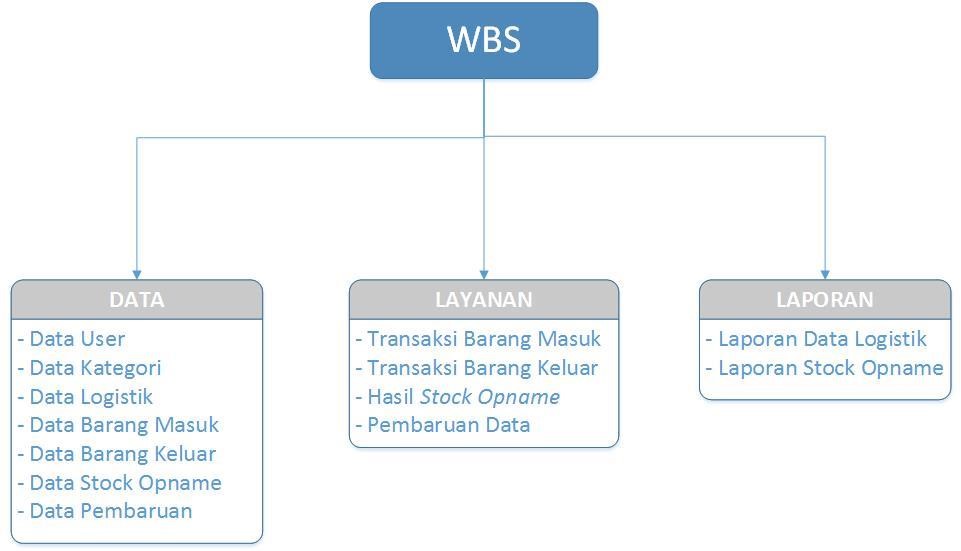
**Gambar 4.3** *Flowchart* Sistem *Stock Opname*

## 4.3 Perancangan Sistem

Perancangan sistem merupakan pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi. Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, meliputi *Flow Chart, Work Breakdown Structure, Use Case Diagram, Flow of event, Activity Diagram, Sequence Diagram, Desain Interface,* dan Basis Data.

### 4.3.1 WBS (*Work Breakdown Structure*)

WBS adalah perangkat yang membantu untuk menentukan dan mengelompokkan elemen-elemen pekerjaan (tugas) dan membantu mengetahui Ruang Lingkup pekerjaan. WBS digunakan untuk melakukan *breakdown* atau memecahkan tiap proses pekerjaan menjadi lebih detail. Hal ini dimaksudkan agar proses perencanaan proyek memiliki tingkat yang lebih baik.



**Gambar 4.4** *Work Breakdown Structure*

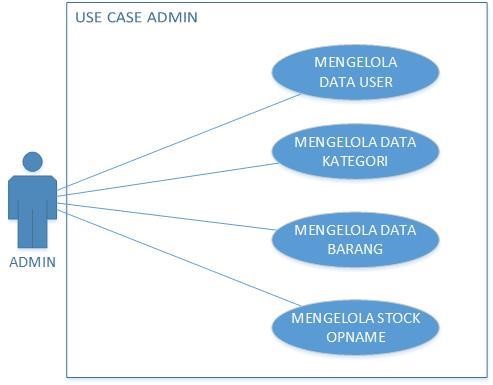
WBS (*Work Breakdown Structure*) terdiri dari 3 (tiga) bagian yaitu Data,

Layanan, Laporan. Data merupakan komponen utama yang digunakan untuk membangun sebuah sistem, meliputi data user, data kategori, data logistik, data barang masuk, data barang keluar, data *stock opname* dan data pembaruan. Layanan dalam sistem ini adalah transaksi barang masuk, transaksi barang keluar, hasil *stock opname*, dan pembaruan data. Untuk Laporan, sistem ini memberikan laporan berupa hasil laporan data logistik, dan laporan stock opname.

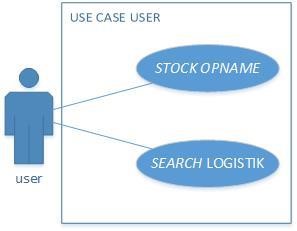
### 4.3.2 *Use Case*

*Use Case* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Bisa dikatakan usecase merupakan dokumen yang digunakan untuk memahami fungsional sub sistem secara skenario. *Use Case* digunakan untuk membentuk tingkah-laku benda/ things dalam sebuah model serta di realisasikan oleh sebuah collaboration. Umumnya *Use Case* digambarkan dengan sebuah elips dengan garis yang solid, biasanya mengandung nama. *Use Case* menggambarkan proses system (kebutuhan sistem dari sudut pandang *user*).

Berikut *usecase diagram* untuk rancangan awal pembuatan sistem :



**Gambar 4.5** *Use Case* *Diagram* Admin



**Gambar 4.6** *Use Case* *Diagram User*

Berikut adalah definisi aktor pada sistem informasi logistik *stock opname* menggunakan *Quick Response Code*:

**Tabel 4.1** Definisi Aktor

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1. | Pengguna (*User*) | Merupakan orang yang memiliki hak akses untuk menjalankan sistem informasi logistik *stock opname* menggunakan *Quick Response Code* berbasis *android* |
| 2. | Admin | Administrator memiliki wewenang mengelola data.  yaitu menambah, merubah, dan menghapus data barang. |

*Use Case Diagram* akan dijelaskan satu persatu menggunakan *Flow of Event.* *Flow of Event* bertujuan untuk mendokumentasikan alur logika dalam usecase yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan.

## a. *Use Case Diagram* Mengelola *Data User*



**Gambar 4.7** *Use Case Diagram* Mengelola *Data User*

**Tabel 4.2** *Flow of Event* Mengelola *Data User*

|  |  |  |
| --- | --- | --- |
| **Nama *Use Case*** | | Mengelola data *user* |
| **Aktor** | | Admin |
| **Pre-condition / Kondisi Awal** | | Admin telah melakukan *login* |
|  | **SKENARIO UTAMA** | |
|  | **Aksi Aktor** | **Reaksi Sistem** |
| 1. | Memilih menu *user* | 2. Menampilkan *list* data *user* |
| 3. | Memilih tombol Tambah/ *Edit* / Hapus / Lihat user | 4. Menampilkan *form* |
| 5. | Admin melakukan penambahan / perubahan / menghapus / melihat data |  |
| 6. | Admin menekan tombol simpan | 7. Sistem menyimpan data di *database*. |
| ***Post-condition* / Kondisi Akhir** | | Data telah tersimpan, dan data *user* ditampilkan. |

## b. *Use Case Diagram* Mengelola Data Kategori Barang

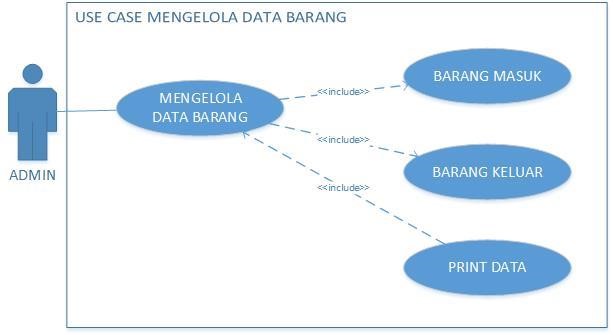


**Gambar 4.8** *Use Case Diagram* Mengelola Data Kategori Barang

**Tabel 4.3** *Flow of Event* Mengelola *Data* Kategori Barang

|  |  |  |
| --- | --- | --- |
|  | **Nama *Use Case*** | Mengelola data kategori |
|  | **Aktor** | Admin |
|  | ***Pre-condition* / Kondisi Awal** | Admin telah melakukan *login* |
|  | **SKENARIO UTAMA** | |
|  | **Aksi Aktor** | **Reaksi Sistem** |
| 1. | Memilih menu kategori | 2. Menampilkan *list* data kategori barang |
| 3. | Memilih tombol Tambah/ *Edit* / Hapus / Lihat kategori | 4. Menampilkan *form* |
| 5. | Admin melakukan penambahan / perubahan / menghapus / melihat data |  |
| 6. | Admin menekan tombol simpan | 7. Sistem menyimpan data di *database* |
| ***Post-condition* / Kondisi Akhir** | | Data telah tersimpan, dan data kategori barang ditampilkan |

### c. *Use Case Diagram* Mengelola Data Barang

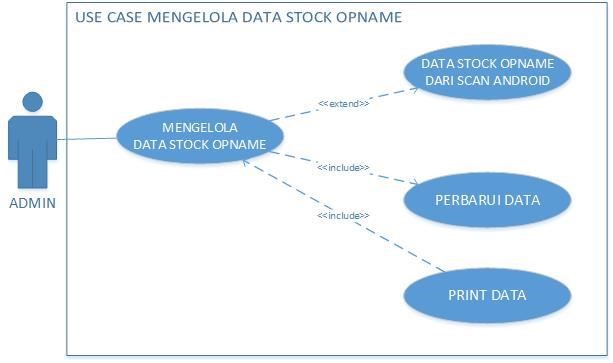


**Gambar 4.9** *Use Case Diagram* Mengelola Data Barang

**Tabel 4.4** *Flow of Event* Mengelola Data Barang

|  |  |  |
| --- | --- | --- |
|  | **Nama *Use Case*** | Mengelola data barang |
|  | **Aktor** | Admin |
|  | ***Pre-condition* / Kondisi Awal** | Admin telah melakukan *login* |
|  | **SKENARIO UTAMA** | |
|  | **Aksi Aktor** | **Reaksi Sistem** |
| 1. | Memilih menu *database* | 2. Menampilkan *list* data barang |
| 3. | Memilih tombol Tambah/ *Edit* / Hapus data barang | 4. Menampilkan *form* |
| 5. | Admin melakukan penambahan / perubahan / menghapus data |  |
| 6. | Admin menekan tombol simpan | 7. Sistem menyimpan data di *database* |
| 8. | Memilih tombol barang masuk / barang keluar | 9. Menampilkan *form* |
| 10. Admin melakukan penambahan / pengurangan stok qty | |  |
| 11. Admin menekan tombol simpan | | 12. Sistem menyimpan data di database dan *list* barang masuk / *list* barang keluar |
| 13. Admin memilih tombol *list* barang masuk / barang keluar untuk melihat catatan transaksi | | 14. Sistem menampilkan l*ist* barang masuk / barang keluar |
| 15. Admin menekan tombol *print* data | | 16. Mengambil data barang untuk siap di *print* |
| ***Post-condition* / Kondisi Akhir** | | Data telah tersimpan, qty berubah sesuai barang masuk / barang keluar, dokumen hasil *print* data barang |

### d. *Use Case Diagram* Mengelola *Stock Opname*

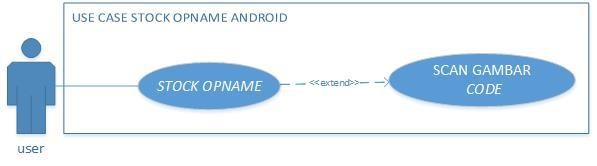


**Gambar 4.10** *Use Case Diagram* Mengelola *Stock Opname*

**Tabel 4.5** *Flow of Event* Mengelola *Stock Opname*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nama *Use Case*** | Mengelola *Stock Opname* | |
|  | **Aktor** | Admin | |
|  | ***Pre-condition* / Kondisi Awal** | Admin telah melakukan *login,* proses *stock opname* menggunakan *android* sudah dilakukan | |
|  | **SKENARIO UTAMA** | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1. | Memilih menu *stock opname* | 2. | Menampilkan *list* data *stock opname* |
| 3. | Memilih tombol Perbarui (jika terdapat selisih)/Hapus data *stock opname* | 4. | Menampilkan *form* |
| 5. | Admin menekan tombol simpan atau hapus | 6. | Sistem menyimpan data di *database* dan *list* data yang telah di perbarui |
| 7. | Admin memilih tombol *list* data yang telah diperbarui | 8. | Sistem menampilkan l*ist* data yang telah di perbarui |
| 9. | Admin menekan tombol *print* data | 10. Mengambil data *stock opname* untuk siap di *print* | |
| ***Post-condition* / Kondisi Akhir** | | Data telah tersimpan, dokumen hasil *print* data *stock opname* | |

### e. *Use Case Diagram* *Stock Opname Android*



**Gambar 4.11** *Use Case Diagram* *Stock Opname Android*

**Tabel 4.6** *Flow of Event* *Stock Opname Android*

|  |  |  |
| --- | --- | --- |
|  | **Nama *Use Case*** | *Stock Opname Android* |
|  | **Aktor** | *User* |
|  | ***Pre-condition* / Kondisi Awal** | Kamera dan koneksi internet harus aktif |
|  | **SKENARIO UTAMA** | |
|  | **Aksi Aktor** | **Reaksi Sistem** |
| 1. | Memilih menu *stock opname* | 2. Menampilkan halaman awal *scan stock opname* |
| 3. | Menekan tombol scan | 4. Memasuki scanner untuk membaca gambar  *QR Code* dan menampilkan hasil detail barang |
| 5. | Mengisi Qty yang baru beserta keterangan barang |  |
| 6. | Admin menekan tombol simpan | 7. Sistem menyimpan data di dalam *database* |
| ***Post-condition* / Kondisi Akhir** | | Data telah tersimpan, dan kembali pada halaman awal *scan stock opname* |

### f. *Use Case Diagram* *Search* Logistik *Android*



**Gambar 4.12** *Use Case Diagram* *Search* Logistik *Android*

**Tabel 4.7** *Flow of Event* *Search* Logistik *Android*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nama *Use Case*** | *Stock Opname Android* | |
|  | **Aktor** | *User* | |
|  | ***Pre-condition* / Kondisi Awal** | Kamera dan koneksi internet harus aktif | |
|  | **SKENARIO UTAMA** | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1. | Memilih menu *search* menggunakan *scan* | 2. | Menampilkan halaman awal *search scan* |
| 3. | Menekan tombol scan | 4. | Memasuki scanner untuk membaca gambar *QR Code* dan menampilkan hasil *search* detail barang |
| 5. | Memilih menu *search* menggunakan kata kunci | 6. | Sistem menampilan list data logistik |
| 7. | Memilih kategori barang atau inputkan *keyword* untuk melakukan pencarian | 8. | Menampilkan hasil pencarian data logistik berdasarkan pilihan kategori atau hasil inputan keyword |
| ***Post-condition* / Kondisi Akhir** | | Menampilkan detail barang yang di cari | |

#### 4.3.3 *Activity Diagram*

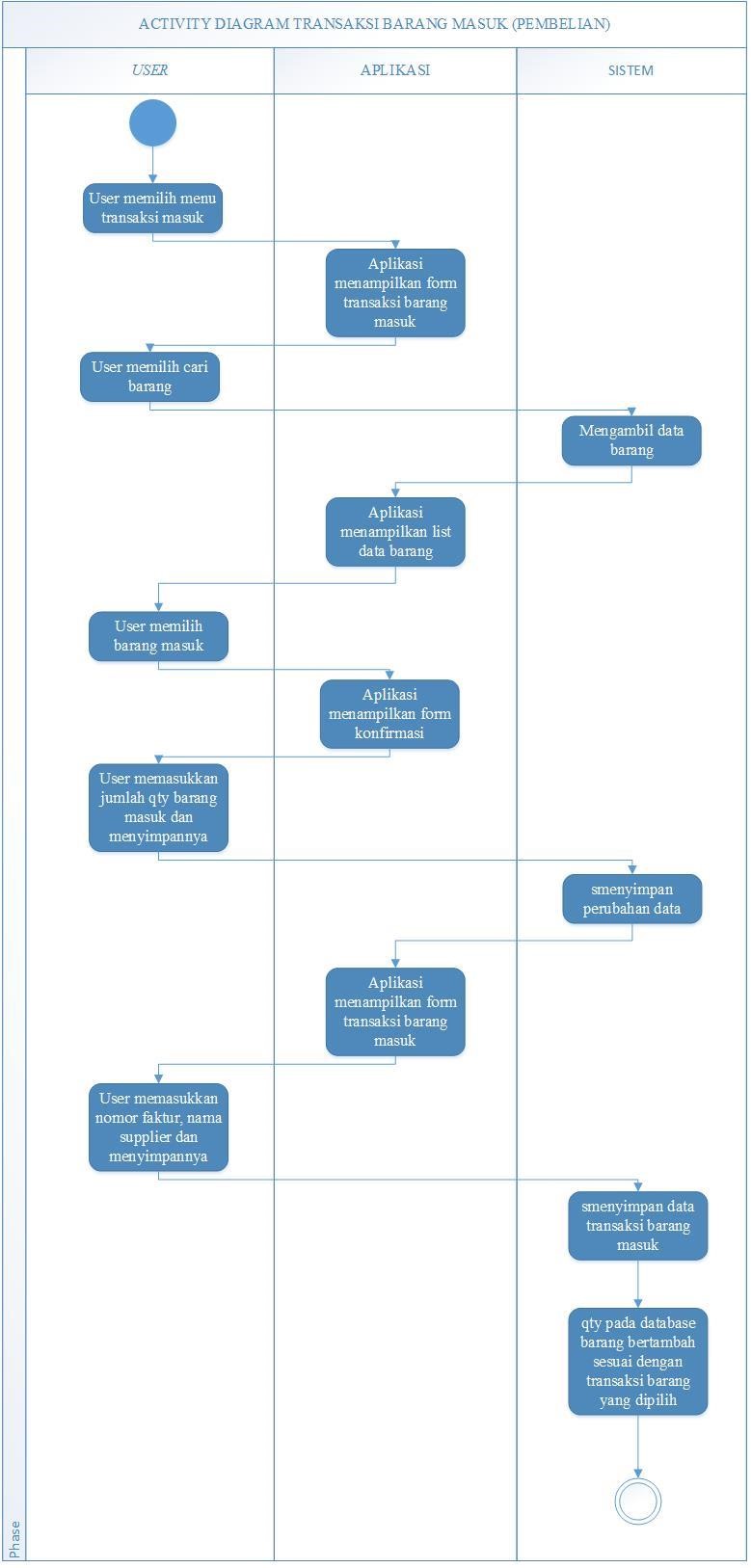
*Activity diagram* menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.

*Activity diagram* sangat bermanfaat untuk membantu memahami proses secara keseluruhan.

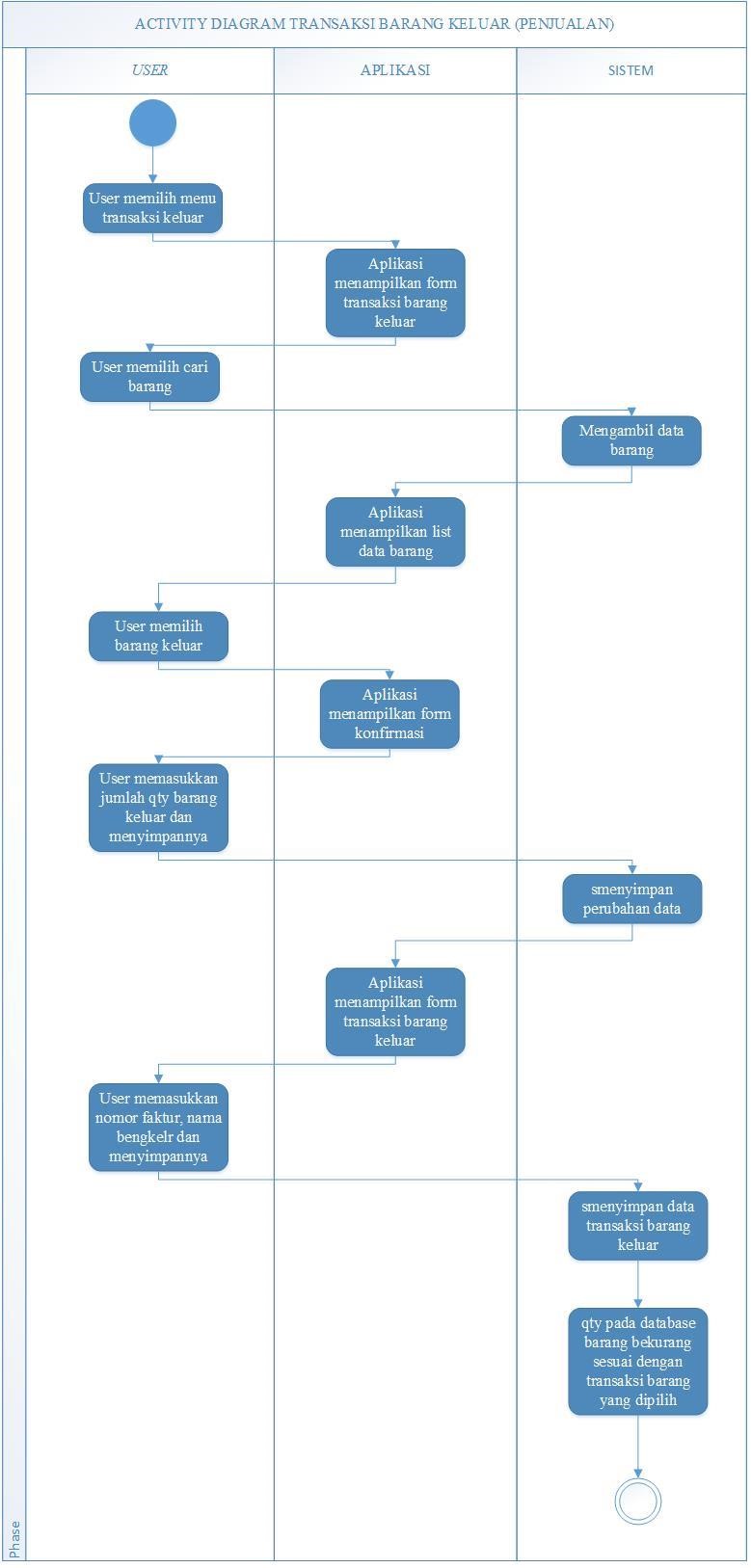
Berikut *activity diagram* untuk transaksi barang masuk, transaksi barang keluar, dan *stock opname.*

## a. Transaksi Barang Masuk



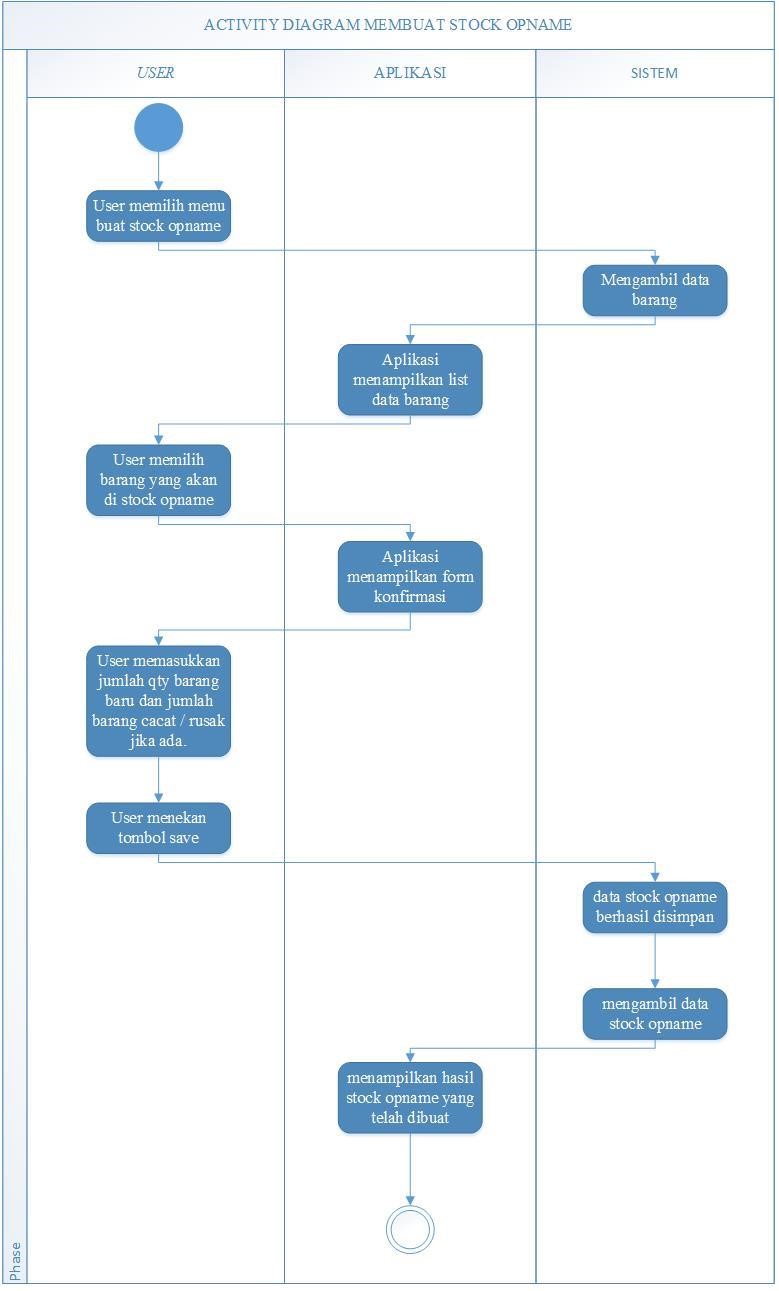
**Gambar 4.13** *Activity Diagram* Transaksi Barang Masuk

## b. Transaksi Barang Keluar



**Gambar 4.14** *Activity Diagram* Transaksi Barang Keluar

### C. *Stock Opname*



**Gambar 4.15** *Activity Diagram* *Stock Opname*

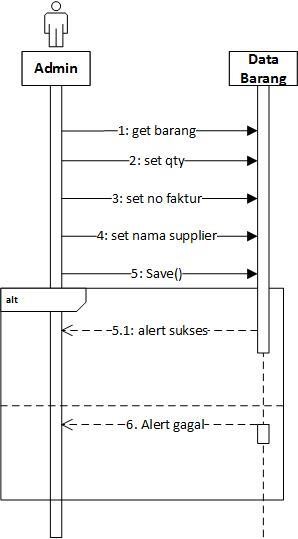
#### 4.3.4 *Sequence Diagram*

*Sequence diagram* (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

*Sequence Diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output tertentu. *Sequence Diagram* diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Berikut *sequence diagram* untuk transaksi masuk, transaksi keluar dan proses stock opname :

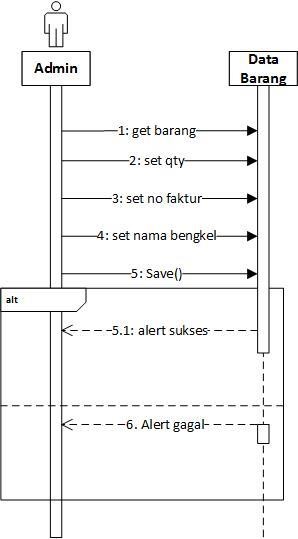
## a. *Sequence Diagram* Transaksi Barang Masuk



**Gambar 4.16** *Sequence Diagram* Transaksi Barang Masuk

Saat admin menekan tombol transaksi masuk untuk pembelian, sistem akan merespon dengan memberikan form untuk tambah transaki masuk. Lalu admin mencari barang dan mengisi jumlah qty dari pembelian barang yang nantinya akan berpengaruh pada penambahan jumlah qty pada database barang. Admin mengisi form berupa nomor faktur dan nama supplier. Setelah admin menekan tombol simpan, sistem akan memberikan respon ketika data berhasil atau gagal disimpan.

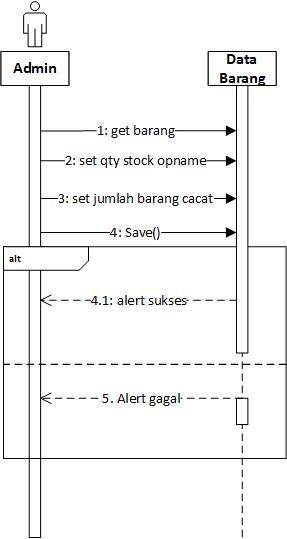
## b. *Sequence Diagram* Transaksi Barang Keluar



**Gambar 4.17** Sequence Diagram Transaksi Barang Keluar

Serupa dengan transaksi barang masuk, saat admin menekan tombol transaksi keluar untuk penjualan, sistem akan merespon dengan memberikan form untuk tambah transaki keluar. Lalu admin mencari barang dan mengisi jumlah qty dari penjualan barang yang nantinya akan berpengaruh pada pengurangan jumlah qty pada database barang. Admin mengisi form berupa nomor faktur dan nama bengkel. Setelah admin menekan tombol simpan, sistem akan memberikan respon ketika data berhasil atau gagal disimpan.

### c. *Sequence Diagram* *Stock Opname*



**Gambar 4.18** *Sequence Diagram* *Stock Opname*

Saat admin menekan tombol untuk membuat *stock opname*, sistem akan merespon dengan memberikan form daftar barang untuk di proses *stock opname*.

Lalu admin mengisi jumlah qty yang baru saat *stock opname* dan mengisi apakah ada barang cacat / rusak. Setelah admin menekan tombol simpan, sistem akan memberikan respon ketika data berhasil atau gagal disimpan.

#### 4.3.5 Desain Interface

Berikut merupakan desain *interface* (antar muka) pada Sistem Informasi Logistik Stock Opname.



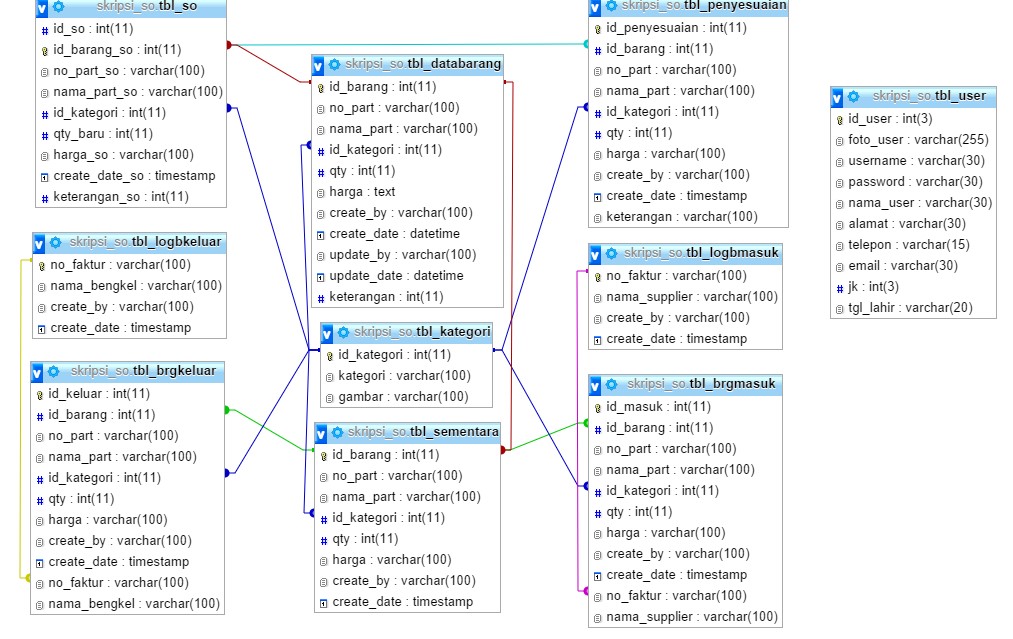
## Gambar 4.19 Desain *Interface*

Menu 1 merupakan tab halaman *search*. Dimana *content* nya berisikan 2 tombol, yaitu tombol cari menggunakan *QR Code* yang berfungsi untuk mencari dan menampilkan detail barang dengan *scan* gambar *QR Code* yang berisi informasi nomor part, dan cari menggunakan kata kunci yang berfungsi mencari barang dengan memilih kategori barang dan memasukkan kata kunci untuk mendapatkan barang yang ingin di cari.

Menu 2 merupakan tab halaman *scanner.* Dimana *content* nya berfungsi untuk melakukan proses *stock opname* yang data nya langsung terkirim ke halaman web untuk admin.

### 4.3.5 Basis Data

Basis data merupakan media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Perancangan basis data dalam membangun sistem ini terdiri dari 10 tabel, yaitu tbl\_user, tbl\_kategori, tbl\_so, tbl\_databarang, tbl\_brgmasuk, tbl\_brgkeluar, tbl\_penyesuaian, tbl\_logbmasuk, tbl\_logbkeluar, dan tbl\_sementara.



**Gambar 4.20** Desain Relasi Pada Basis Data

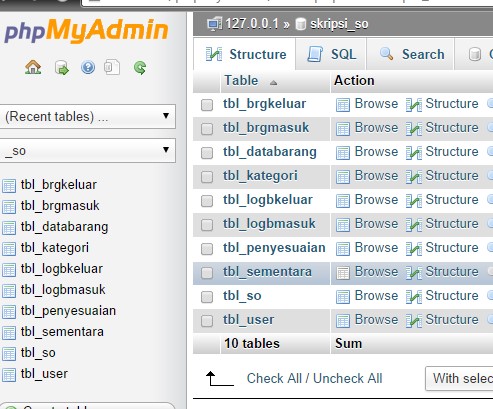
# BAB V IMPLEMENTASI

Setelah melakukan tahapan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi, yaitu realisasi perancangan menjadi nyata. Bagian pada bab ini meliputi implementasi basis data dan implementasi sistem.

Implementasi basis data menggunakan MySQL dan implementasi sistem yang terdiri dari *web* admin dan aplikasi *android*. Halaman admin dibangun menggunakan bahasa pemrograman PHP sebagai bahasa pemrograman berbasis web. Sedangkan aplikasi *android* dibangun dengan bahasa pemrograman *Java.*

## 5.1 Implementasi Basis Data

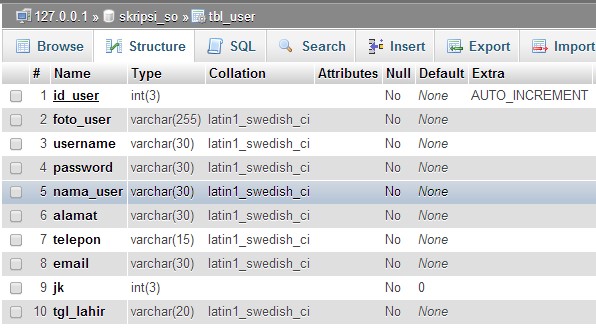
Implementasi basis data merupakan hasil implementasi dari perancangan basis data sesuai perancangan yang telah dilakukan sebelumnya pada bab sebelumnya. Basis data yang dibuat diberi nama diagnosis. Gambar 5.1 merupakan hasil implementasi basis data yang dibangun menggunakan MySQL.



**Gambar 5.1** Basis data

Basis data ini memiliki 10 tabel, yaitu tbl\_user, tbl\_kategori, tbl\_so, tbl\_databarang, tbl\_brgmasuk, tbl\_brgkeluar, tbl\_penyesuaian, tbl\_logbmasuk, tbl\_logbkeluar, dan tbl\_sementara. Implementasi tabel-tabel tersebut adalah sebagai berikut :

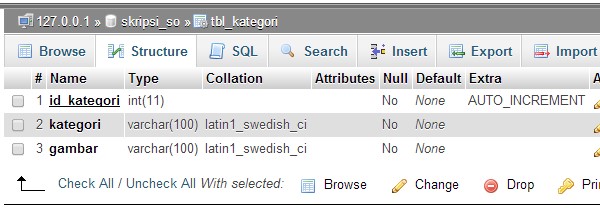
1. Tabel *User*



**Gambar 5.2** Implementasi Tabel *User*

Gambar 5.2 adalah implementasi dari rancangan tabel *user* yang berfungsi menampung data pengguna sistem, dalam hal ini adalah admin. Tabel ini memiliki *field* antara lain : id\_user (*primary key*), foto\_user, *username, password¸* nama\_user, alamat, telepon, email, jk, dan tgl\_lahir.

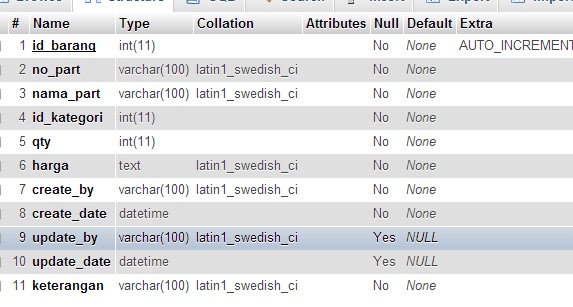
1. Tabel Kategori



**Gambar 5.3** Implementasi Tabel Kategori

Gambar 5.3 adalah implementasi dari rancangan tabel kategoriyang berfungsi menampung data kategori merk barang. Tabel ini memiliki *field* antara lain : id\_kategori (*primary key*), kategori*,* dan gambar.

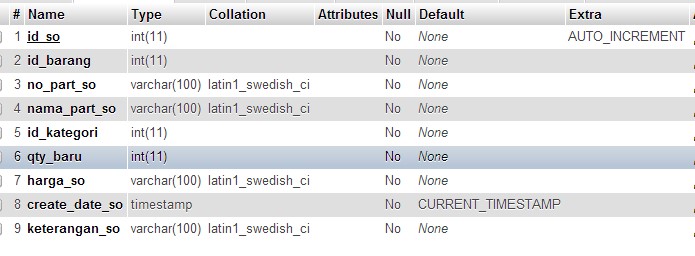
1. Tabel Databarang



**Gambar 5.4** Implementasi Tabel Databarang

Gambar 5.4 adalah implementasi dari rancangan tabel databarang yang berfungsi menampung data barang-barang yang ada. Tabel ini memiliki *field* antara lain : id\_barang (*primary key*), no\_part, nama\_part, id\_kategori, kategori, qty, harga, *create\_by, create\_date, update\_by, update\_date,* dan keterangan.

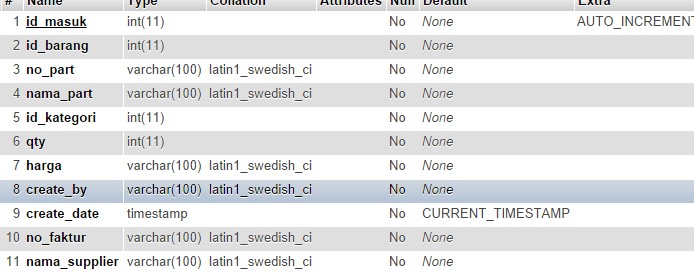
1. Tabel SO



**Gambar 5.5** Implementasi Tabel SO

Gambar 5.5 adalah implementasi dari rancangan tabel soyang berfungsi menampung data hasil *stock opname*. Tabel ini memiliki *field* antara lain : id\_so(*primary key*), id\_barang, no\_part\_so, nama\_part\_so, id\_kategori, qty\_baru, harga\_so, dam *create date*\_sodan keterangan\_so*.*

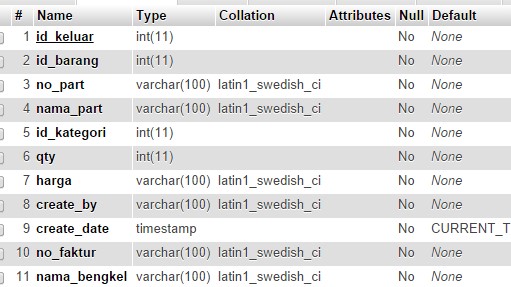
1. Tabel Barang Masuk



**Gambar 5.6** Implementasi Tabel Barang Masuk

Gambar 5.6 adalah implementasi dari rancangan tabel barang masuk yang berfungsi menampung data transaksi barang masuk. Tabel ini memiliki *field* antara lain : id\_masuk (*primary key*), id\_barang, no\_part, nama\_part, id\_kategori, qty, harga, *create\_by, create\_date,* no\_faktur, dan nama\_supplier

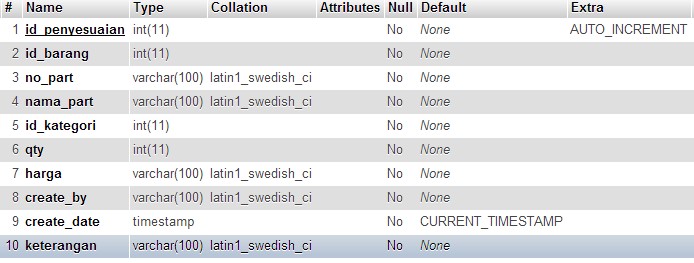
1. Tabel Barang Keluar



**Gambar 5.7** Implementasi Tabel Barang Keluar

Gambar 5.7 adalah implementasi dari rancangan tabel barang keluar yang berfungsi menampung data transaksi barang keluar. Tabel ini memiliki *field* antara lain : id\_keluar (*primary key*), id\_barang, no\_part, nama\_part, id\_kategori, qty, harga, *create\_by, create\_date,* nomor faktur, dan nama\_bengkel.

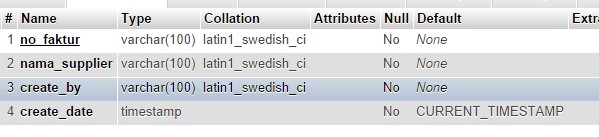
1. Tabel Penyesuaian



**Gambar 5.8** Implementasi Tabel Penyesuaian

Gambar 5.8 adalah implementasi dari rancangan tabel penyesuaian yang berfungsi menampung data stock opname yang telah diperbarui. Data tersebut perlu di perbarui karena terdapat selisih qty *stock opname* dan qty data barang. Setelah data di perbarui maka otomatis data pada tbl\_databarang juga akan menyesuaikan. Tabel ini memiliki *field* antara lain : id\_penyesuaian (*primary key*), id\_barang, no\_part, nama\_part, id\_kategori, qty, harga, *create\_by, create\_date* dan keterangan.

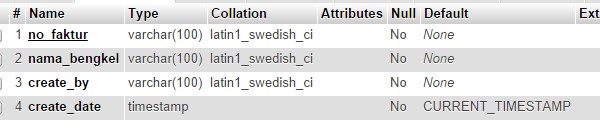
1. Tabel Log Barang Masuk



**Gambar 5.9** Implementasi Tabel Log Barang Masuk

Gambar 5.9 adalah implementasi dari rancangan tabel log barang masuk yang berfungsi menampung data barang masuk berdasarkan nomor faktur dan nama *supplier*. Tabel ini memiliki *field* antara lain : no\_faktur (*primary key*), nama\_supplier, *create\_by,* dan *create\_date.*

1. Tabel Log Barang Keluar



**Gambar 5.10** Implementasi Tabel Log Barang Keluar

Gambar 5.10 adalah implementasi dari rancangan tabel log barang keluar yang berfungsi menampung data barang masuk berdasarkan nomor faktur dan nama bengkel*.* Tabel ini memiliki *field* antara lain : no\_faktur (*primary key*), nama\_bengkel, *create\_by,* dan *create\_date.*

1. Tabel Sementara



**Gambar 5.11** Implementasi Tabel Sementara

Gambar 5.11 adalah implementasi dari rancangan tabel sementara. Berfungsi sebagai *temporary table*, dan membantu untuk menampung sementara data barang masuk maupun barang keluar. Tabel ini memiliki *field* antara lain id\_barang (*primary key*), no\_part, *nama\_part,* id\_kategori, qty, harga, *create\_by,* dan *create\_date.*

## 5.2 Implementasi Sistem

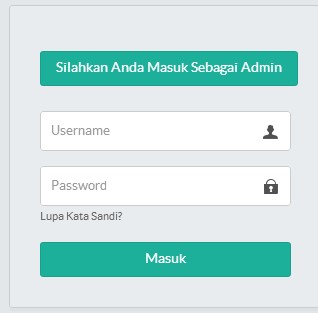
Implementasi Sistem merupakan implementasi dari rancangan antarmuka atau *interface*. Seperti yang telah dijelaskan sebelumnya, implementasi sistem ini terdiri dari web admin dan aplikasi android.

### 5.2.1 Web Admin

Menggunakan bahasa pemrograman PHP sebagai bahasa pemrograman berbasis web dan ditujukan untuk admin dalam mengolah data lebih lanjut.

#### 5.2.1.1 Halaman Login

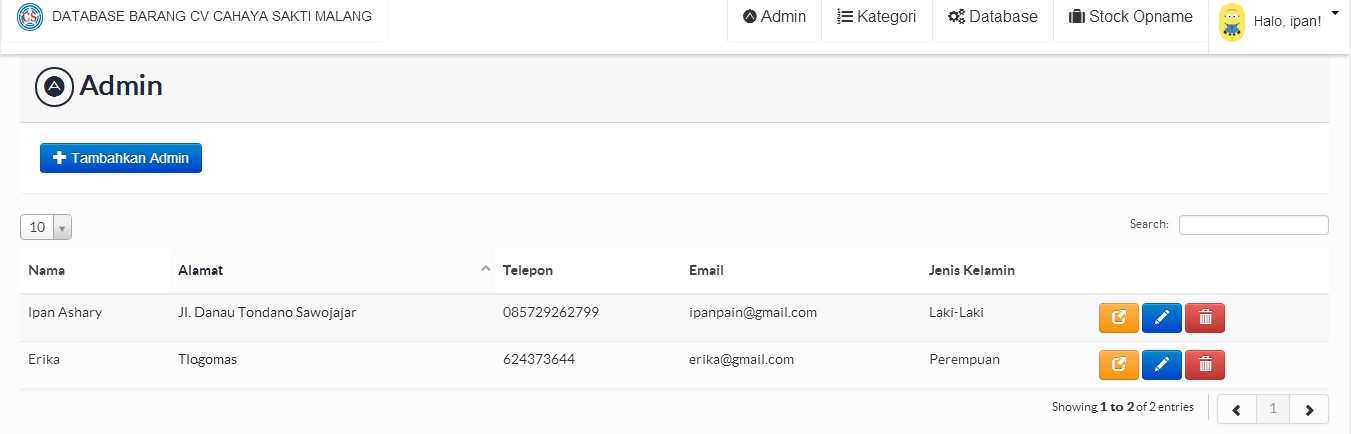
Implementasi rancangan antarmuka halaman login digambarkan pada gambar 5.6. Setiap menggunakan aplikasi admin harus melakukan proses login terlebih dahulu.



**Gambar 5.12** Halaman Login

#### 5.2.1.2 Halaman Admin

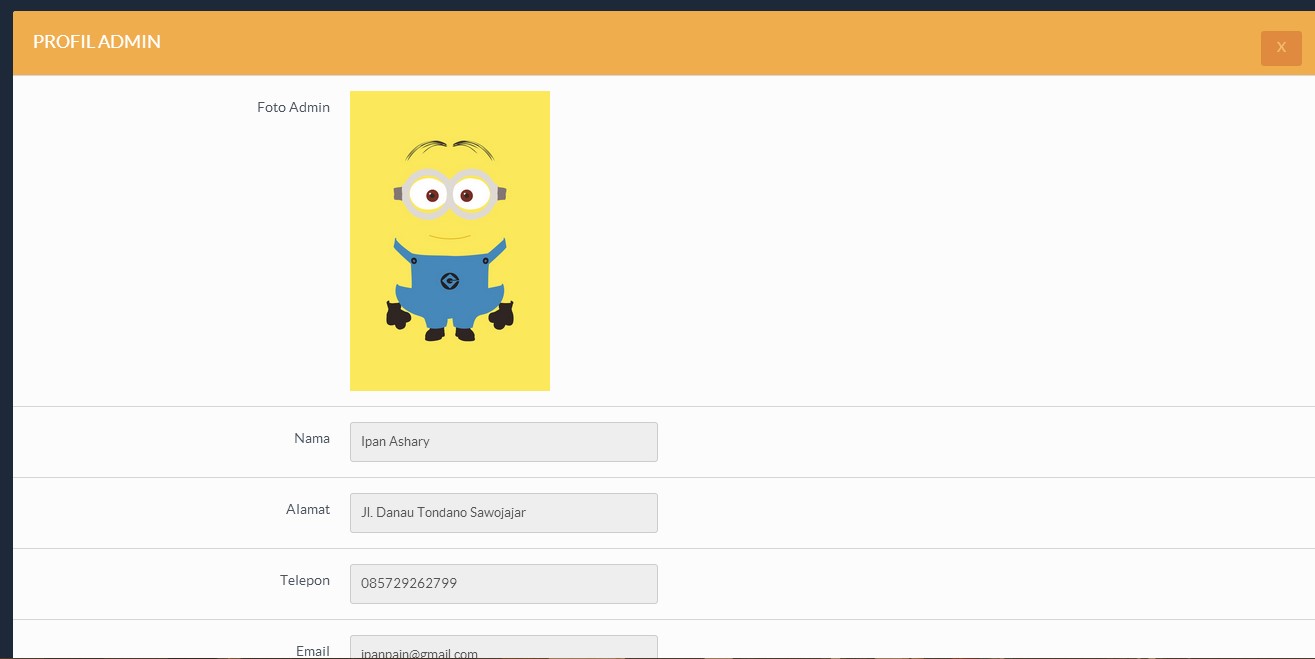
Merupakan halaman admin, berisi informasi daftar admin yang terdaftar dalam aplikasi ini. Tombol berwana kuning untuk melihat data, biru untuk merubah data dan merah untuk menghapus data.



**Gambar 5.13** Halaman Admin

#### 5.2.1.3 Halaman Profil Admin

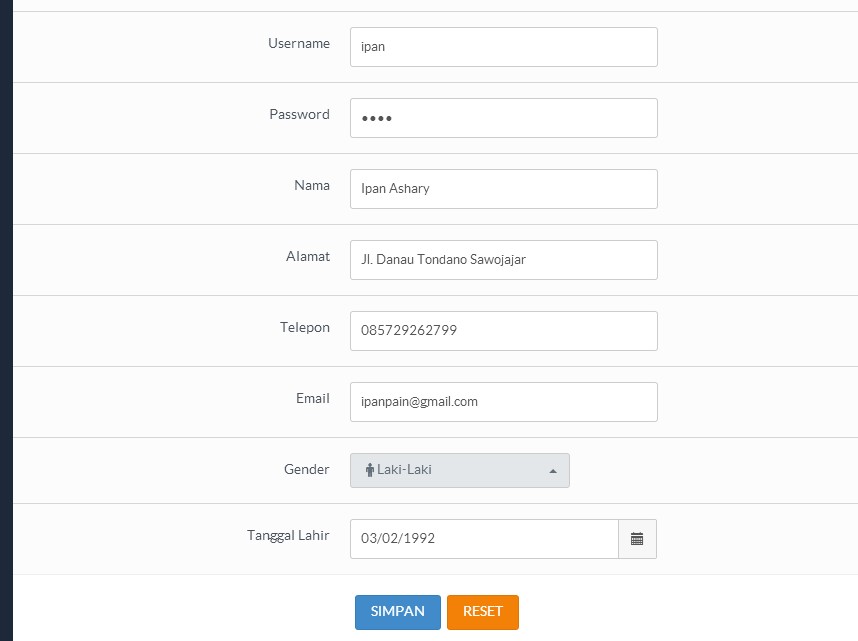
Tampilan detail informasi tentang data admin.



**Gambar 5.14** Gambar Halaman Profil Admin

#### 5.2.1.4 Halaman Edit Admin

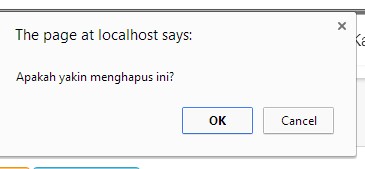
Berisi tentang edit informasi data admin.



**Gambar 5.15** Halaman Edit Admin

#### 5.2.1.5 Halaman Hapus Data

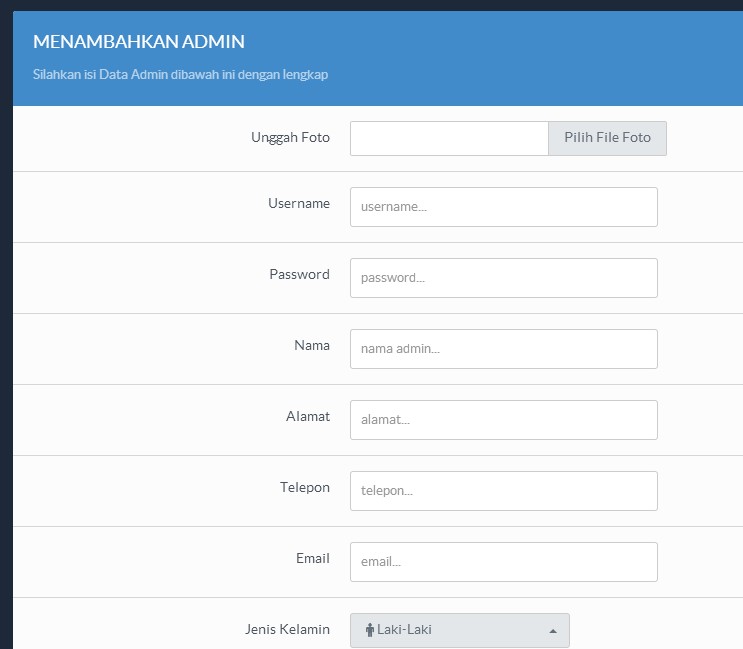
Halaman ini merupakan halaman konfirmasi ketika admin ingin menghapus salah satu data. Baik itu data admin, data kategori, data barang, maupun data stock opname.



**Gambar 5.16** Halaman Hapus Data

#### 5.2.1.6 Halaman Insert Admin

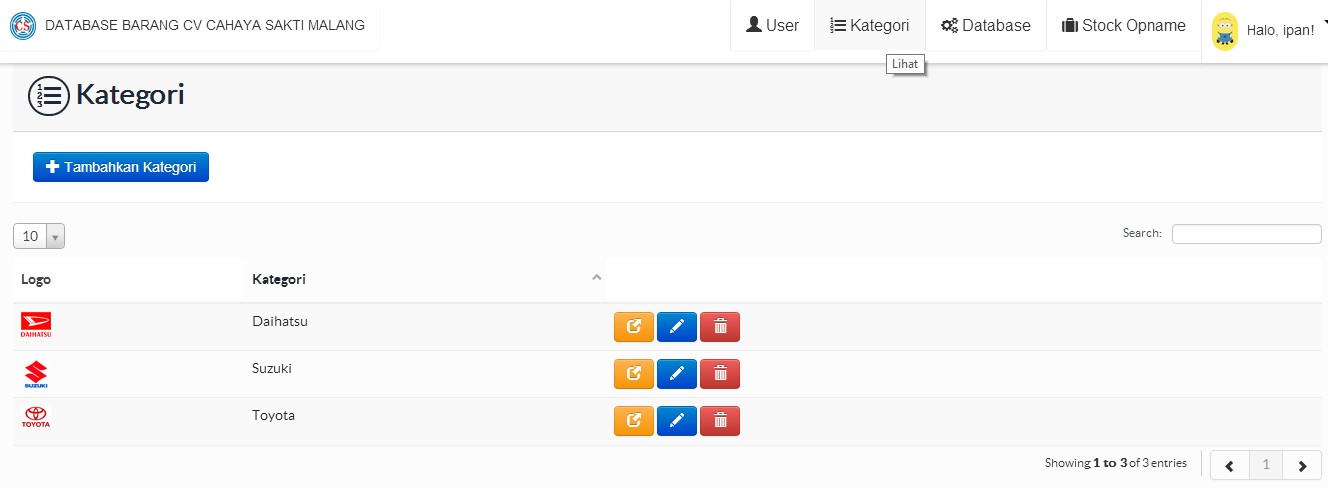
Halaman insert admin ini digunakan untuk menambahkan data admin yang baru.



**Gambar 5.17** Halaman Insert Admin

#### 5.2.1.7 Halaman Kategori

Halaman kategori merupakan list kategori barang (merk part) yang ada.



**Gambar 5.18** Halaman Kategori

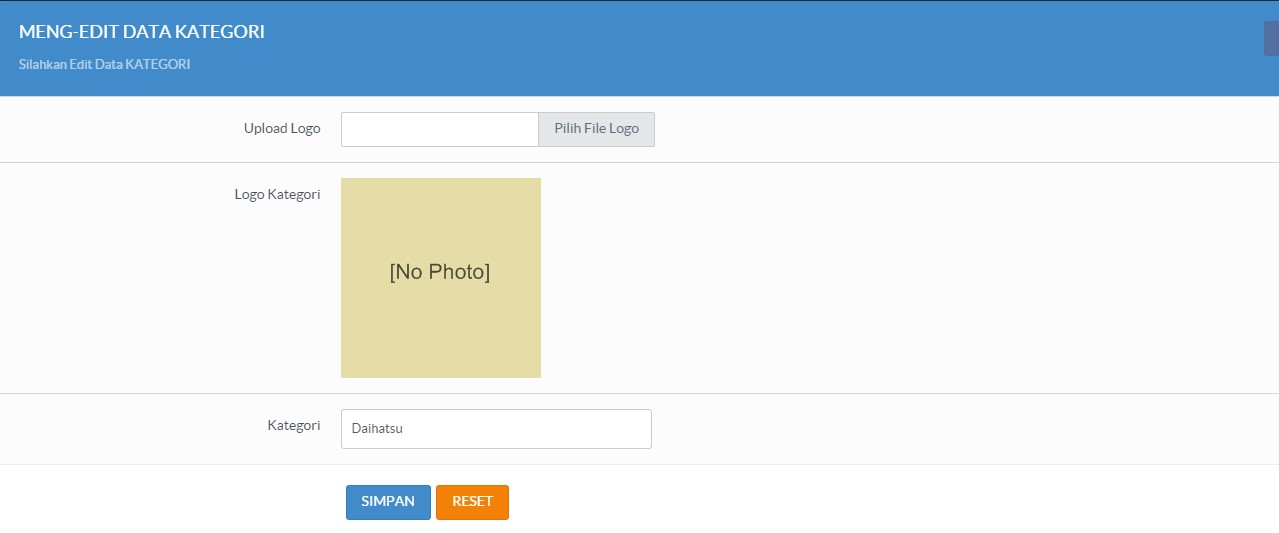
#### 5.2.1.8 Halaman Profil Kategori



## Gambar 5.19 Halaman Profil Kategori

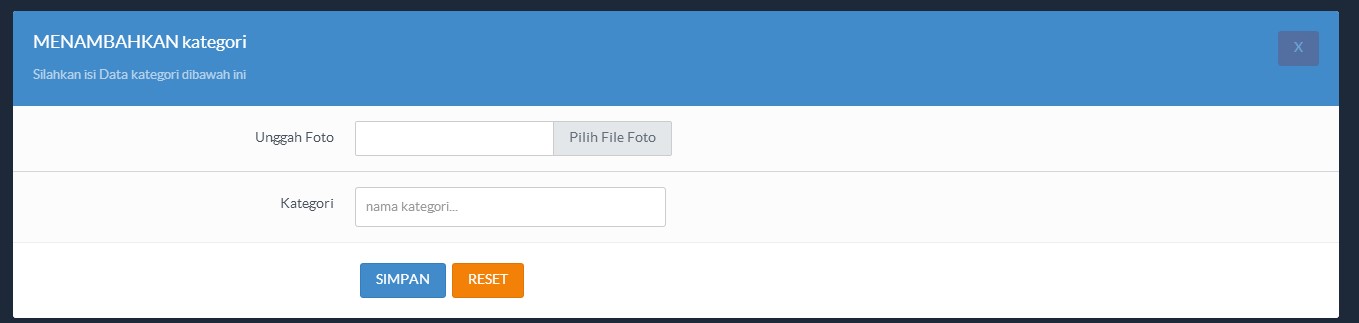
### 5.2.1.9 Halaman Edit Data Kategori

Merupakan halaman untuk mengubah detail kategori yang sudah ada. Terdapat fitur upload logo untuk mengambil gambar logo merk yang diinginkan.



**Gambar 5.20** Halaman Edit Data Kategori

### 5.2.1.10 Halaman Insert Kategori

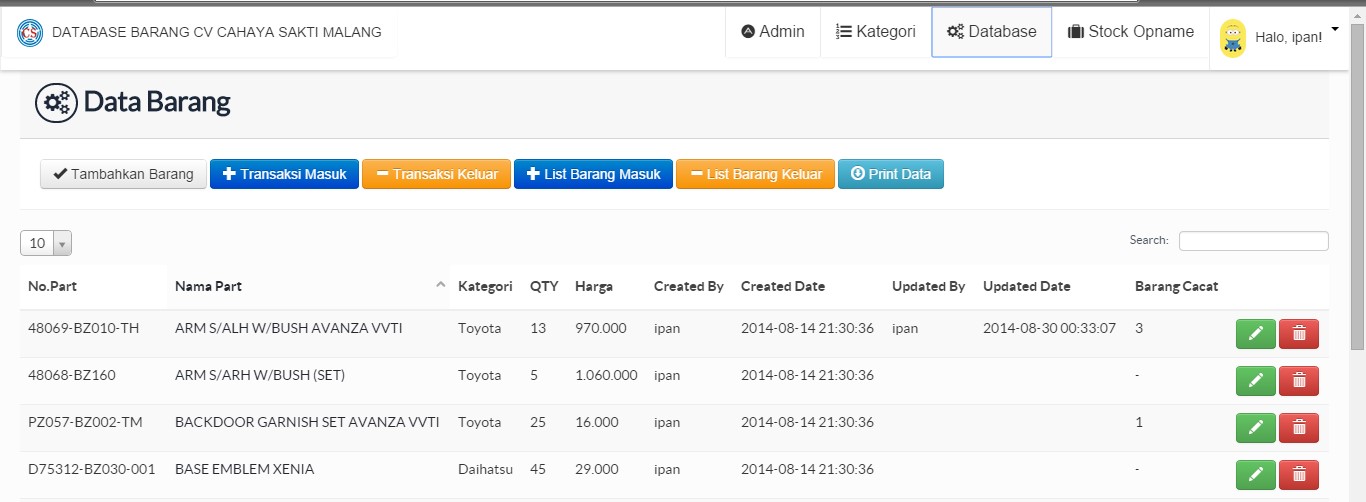


**Gambar 5.21** Halaman Insert Kategori

### 5.2.1.11 Halaman Database Barang

Merupakan halaman yang berisikan list database barang. Terdapat berbagai macam fitur yang sangat berguna. Seperti search untuk mencari barang, table view dapat mengatur jumlah list barang yang ingin ditampilkan dalam satu halaman (10, 25, 50 hingga 100 list barang), serta fitur *sorting* otomatis ketika mengklik judul kolom tabel. Data dapat diurutkan dari yang jumlahnya paling besar ataupun sebaliknya, hingga diurutkan sesuai abjad.

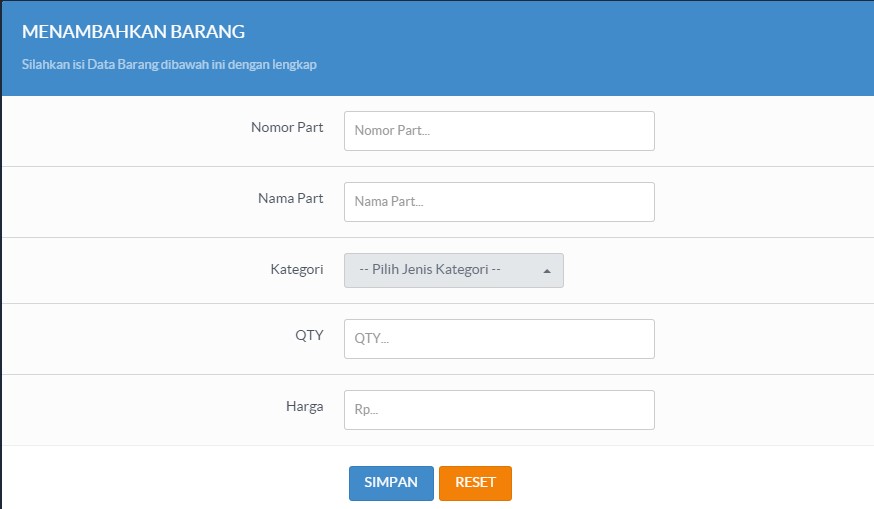
Terdapat tombol tambahkan data untuk melakukan tambah barang, tombol transaksi masuk untuk memulai melakukan proses transaksi masuk barang (pembelian) dari *supplier*, transaksi keluar untuk memulai proses transaksi keluar barang (penjualan) kepada bengkel, serta list barang masuk dan list barang keluar untuk mengetahui daftar barang dari hasil transaksi masuk maupun keluar. Fitur *print* pun disediakan untuk mencetaknya langsung atau menyimpan dokumen tersebut dalam bentuk file berformat .pdf. Tombol berwarna hijau untuk rubah data, dan merah untuk hapus data.



**Gambar 5.22** Halaman Database Barang

### 5.2.1.12 Halaman Tambahkan Barang

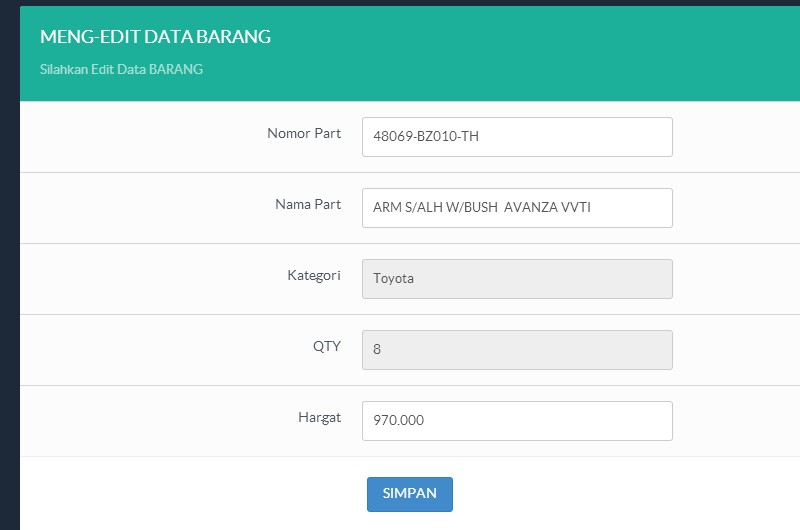
Halaman ini berfungsi untuk menambahkan barang baru ke dalam database barang.



**Gambar 5.23** Halaman Tambah Barang

### 5.2.1.13 Halaman Edit Data Barang

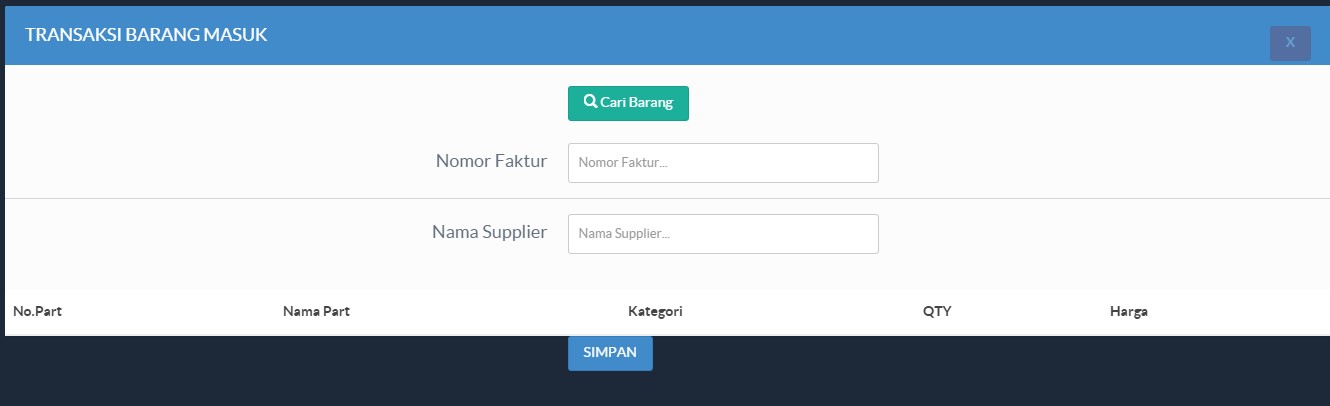
Halaman ini akan tampil ketika user menekan tombol berwarna hijau. Edit data barang hanya bisa dilakukan pada nomor part, nama part dan harga.



**Gambar 5.24** Halaman Edit Data Barang

### 5.2.1.14 Halaman Transaksi Barang Masuk

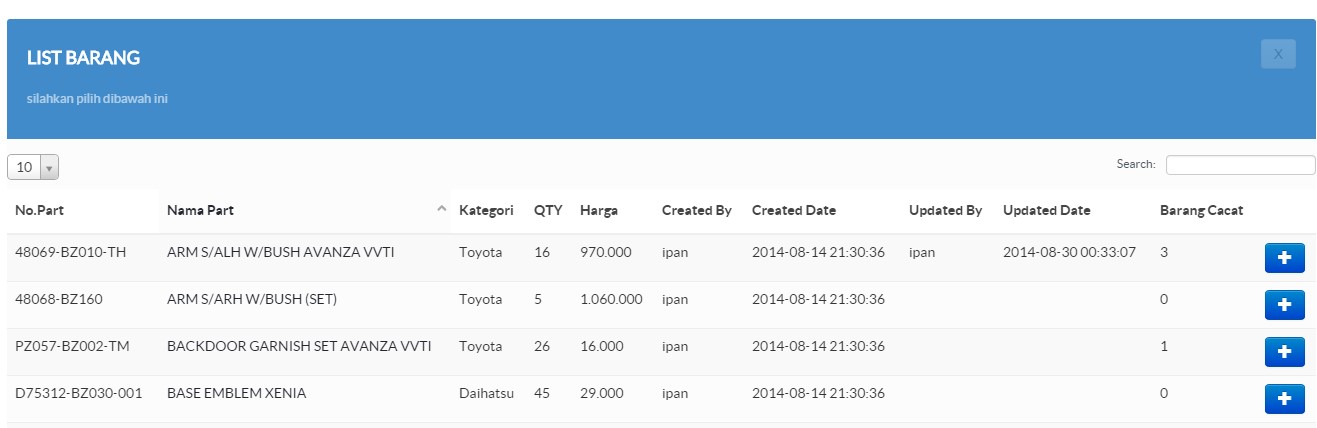
Halaman ini akan tampil ketika user menekan tombol berwarna biru pada halaman database barang yaitu transaksi masuk.



**Gambar 5.25** Halaman Transaksi Barang Masuk

Pada gambar 5.25 terlihat bahwa data barang yang akan melalui proses transaksi barang masuk masih kosong. Maka dari itu terdapat tombol cari barang untuk mencari barang mana saja yang akan di proses.

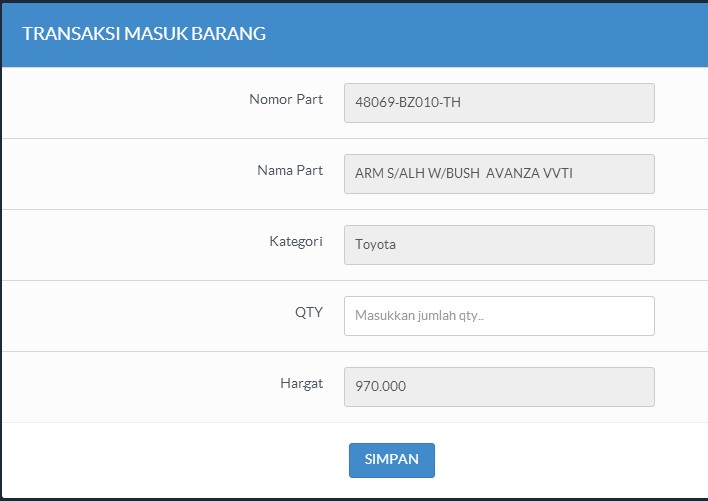
### 5.2.1.15 Halaman Tambah Data Barang Transaksi Masuk



**Gambar 5.26** Halaman Tambah Data Barang Transaksi Masuk

Setelah memilih tombol cari barang pada gambar 5.25, akan muncul halaman pada gambar 5.26 di atas. Terdapat kolom search untuk mudahkan dalam pencarian barang, lalu pilih tombol plus untuk menambahkan daftar transaksi barang masuk.

### 5.2.1.16 Halaman Konfirmasi Transaksi Barang Masuk

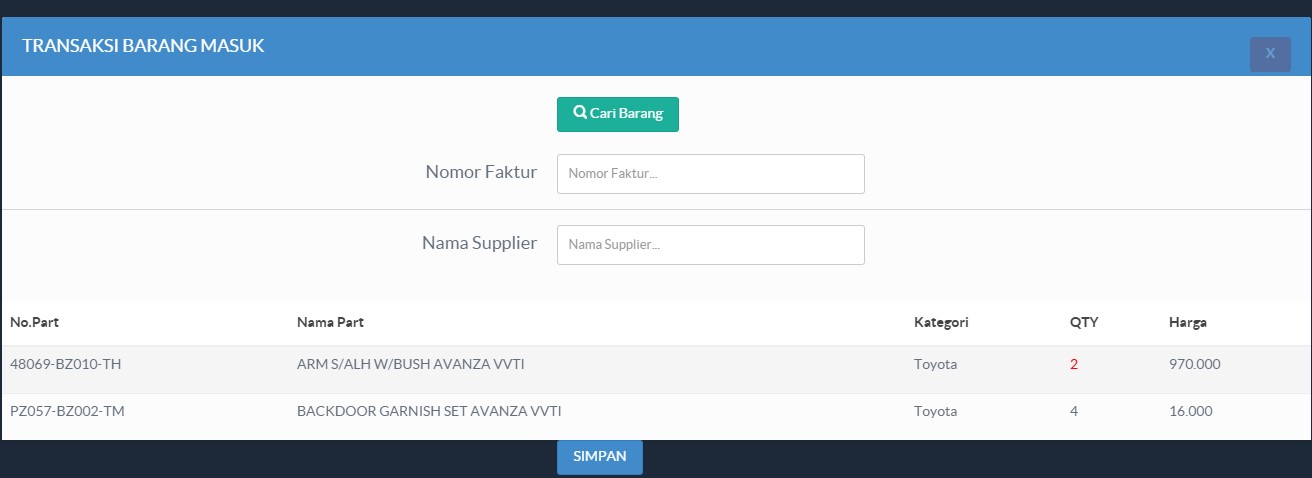


**Gambar 5.27** Halaman Konfirmasi Transaksi Barang Masuk

Tombol plus untuk menambahkan daftar transaksi barang masuk pada gambar 5.26 yang telah dipilih, akan menampilkan halaman konfirmasi barang seperti pada gambar 5.27. Masukkan jumlah qty sesuai dengan jumlah pembelian barang tersebut. Lalu pilih simpan. Setelah itu sistem akan kembali pada proses gambar

5.26 untuk memlilih barang lainnya jika dalam 1 transaksi barang masuk (pembelian) terdapat lebih dari 1 barang. Setelah semua barang sudah di simpan, tutup halaman pada gambar 5.26 tersebut yang berada pada pojok kanan atas. Maka sistem akan menampilkan daftar data transaksi barang masuk yang telah dibuat.

Masukkan nomor faktur dan nama supplier lalu pilih tombol simpan.



**Gambar 5.28** Halaman Transaksi Barang Masuk yang Telah Dibuat

### 5.2.1.17 Halaman List Barang Masuk

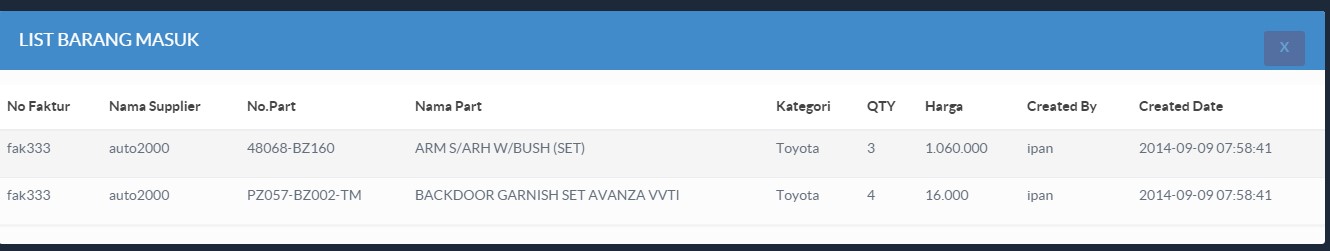
Halaman ini merupakan catatan list transaksi barang masuk yang pernah dibuat.



**Gambar 5.29** Halaman List Barang Masuk

### 5.2.1.18 Halaman Detail List Barang Masuk

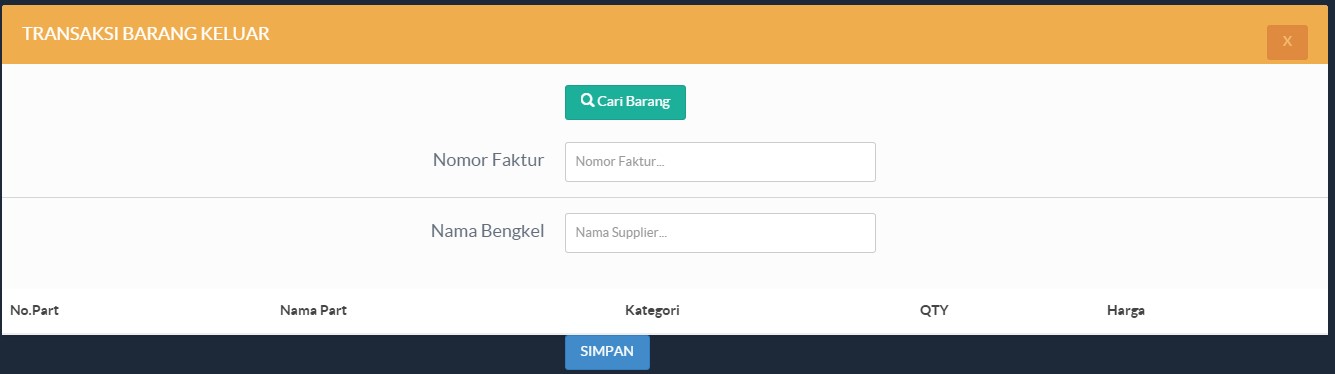
List tersebut bisa dilihat lebih detail mengenai barang apa saja yang pernah masuk melalui tombol lihat yang berwarna kuning pada gambar 5.29, dan akan muncul halaman dibawah ini.



**Gambar 5.30** HalamanDetail List Barang Masuk

### 5.2.1.19 Halaman Transaksi Barang Keluar

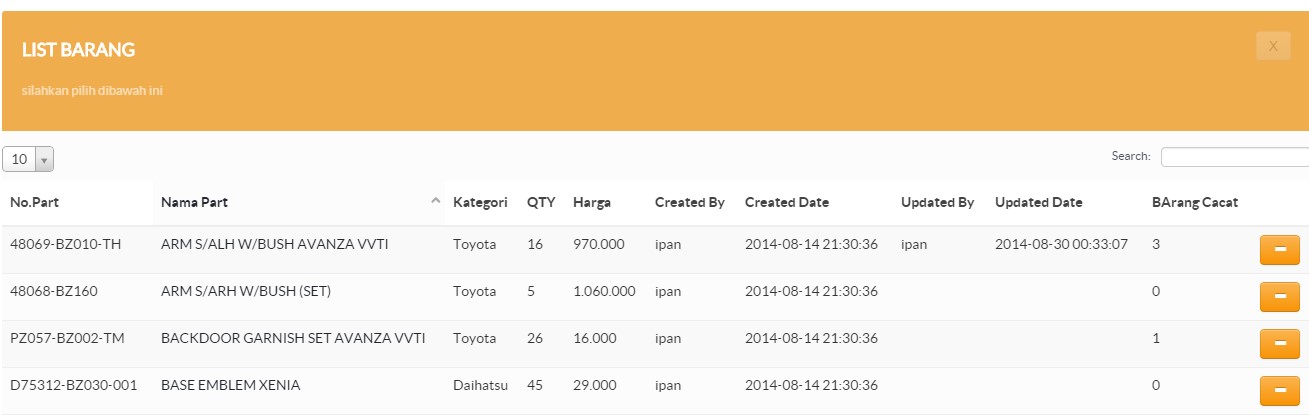
Halaman ini akan tampil ketika user menekan tombol berwarna kuning pada halaman database barang yaitu transaksi keluar.



**Gambar 5.31** Halaman Transaksi Barang Masuk

Pada gambar 5.31 terlihat bahwa data barang yang akan melalui proses transaksi barang keluar masih kosong. Maka dari itu terdapat tombol cari barang untuk mencari barang mana saja yang akan di proses.

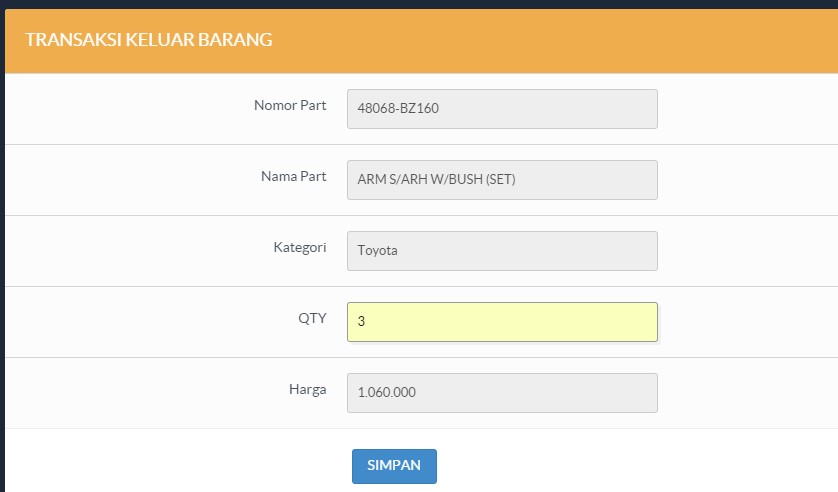
### 5.2.1.20 Halaman Tambah Data Barang Transaksi Keluar



**Gambar 5.32** Halaman Tambah Data Barang Transaksi Masuk

Setelah memilih tombol cari barang pada gambar 5.31, akan muncul halaman pada gambar 5.32 di atas. Terdapat kolom search untuk mudahkan dalam pencarian barang, lalu pilih tombol minus untuk menambahkan daftar transaksi barang keluar.

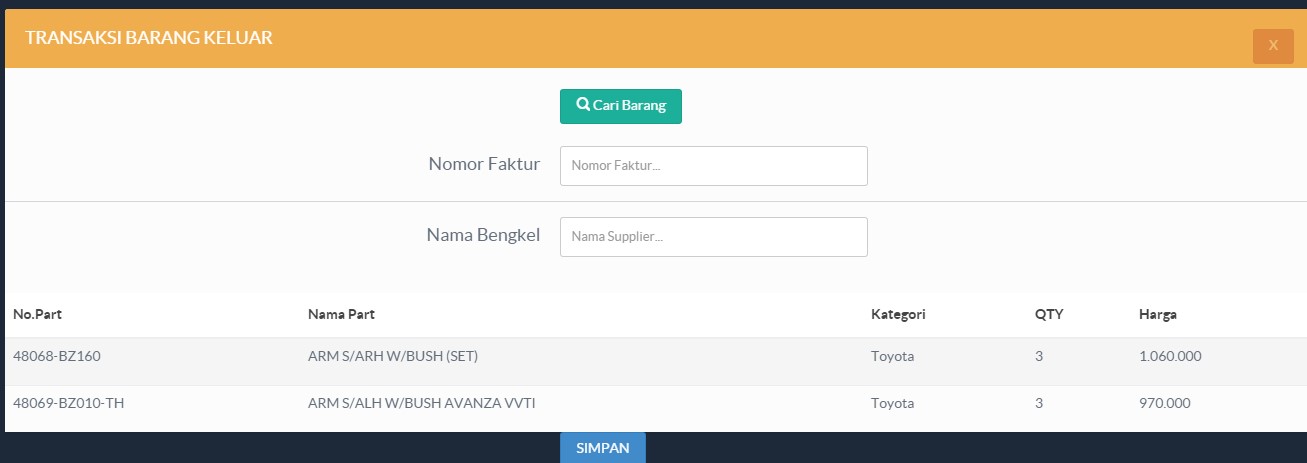
### 5.2.1.21 Halaman Konfirmasi Transaksi Barang Keluar



**Gambar 5.33** Halaman Konfirmasi Transaksi Barang Keluar

Tombol minus untuk menambahkan daftar transaksi barang keluar pada gambar 5.32 yang telah dipilih, akan menampilkan halaman konfirmasi barang seperti pada gambar 5.33. Masukkan jumlah qty sesuai dengan jumlah penjualan barang tersebut. Lalu pilih simpan. Setelah itu sistem akan kembali pada proses gambar 5.32 untuk memlilih barang lainnya jika dalam 1 transaksi barang keluar (penjualan) terdapat lebih dari 1 barang. Setelah semua barang sudah di simpan, tutup halaman pada gambar 5.32 tersebut yang berada pada pojok kanan atas. Maka sistem akan menampilkan daftar data transaksi barang masuk yang telah dibuat.

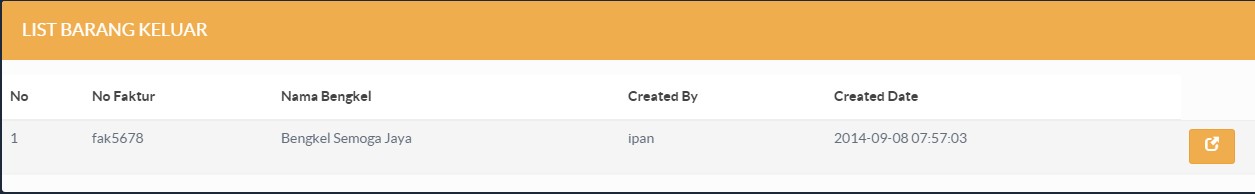
Masukkan nomor faktur dan nama supplier lalu pilih tombol simpan.



**Gambar 5.34** Halaman Transaksi Barang Masuk yang Telah Dibuat

### 5.2.1.22 Halaman List Barang Keluar

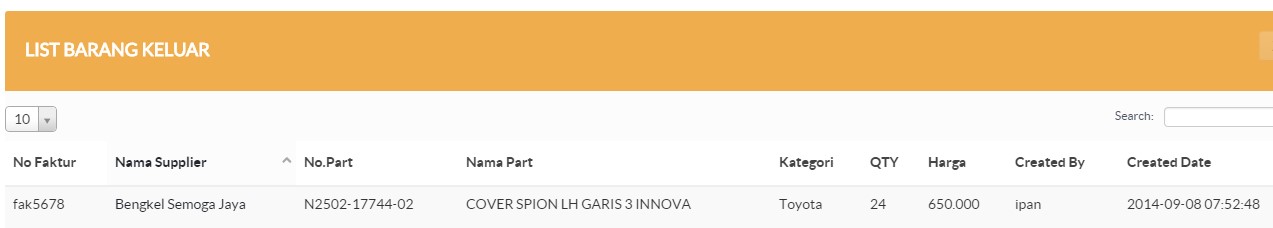
Halaman ini merupakan catatan list transaksi barang keluar yang pernah dibuat.



**Gambar 5.35** Halaman List Barang Keluar

### 5.2.1.23 Halaman Detail List Barang Keluar

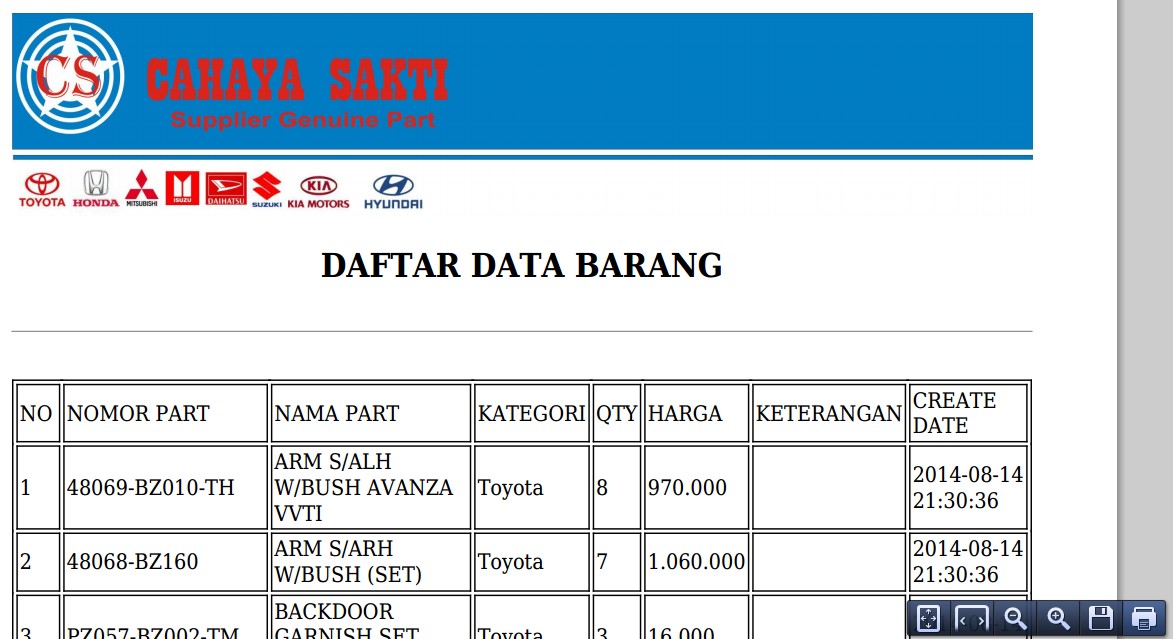
List tersebut bisa dilihat lebih detail mengenai barang apa saja yang pernah keluar melalui tombol lihat yang berwarna kuning pada gambar 5.35, dan akan muncul halaman dibawah ini.



**Gambar 5.36** HalamanDetail List Barang Keluar

### 5.2.1.24 Halaman Laporan Data Barang

Tampilan laporan data barang yang siap di *print* maupun di simpan dalam bentuk file berformat .pdf.



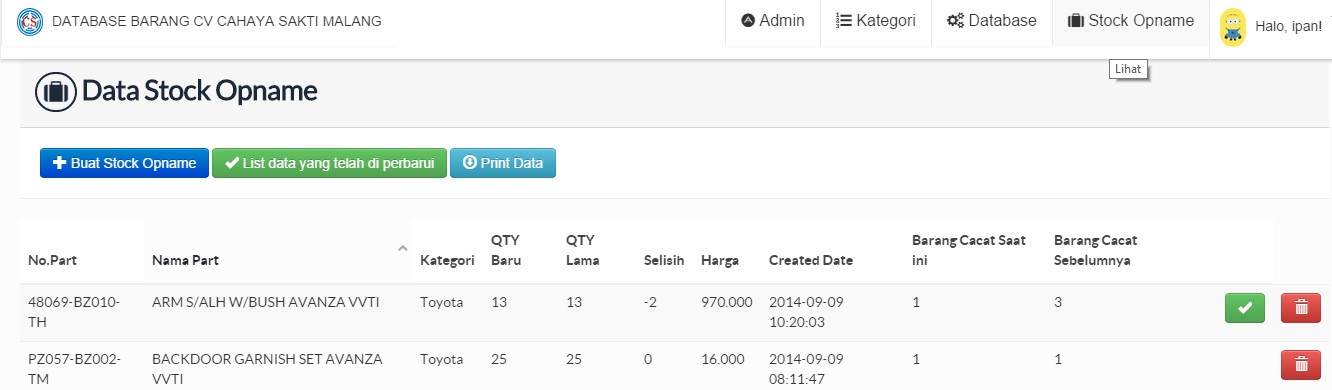
**Gambar 5.37** Laporan Data Barang

### 5.2.1.25 Halaman Data *Stock Opname*

Gambar 5.38 berikut merupakan halaman data *stock opname*. Data ini diperoleh dengan 2 cara. Cara pertama yaitu diperoleh setelah melakukan proses *stock opname* menggunakan aplikasi *android* yang telah disediakan. Hasil *scan QR Code* barang menjadi data *stock opname* ini. Sedangkan cara ke kedua, kita bisa buat *stock opname* langsung dari halaman tersebut dengan cara memilih tombol buat *stock opname*.

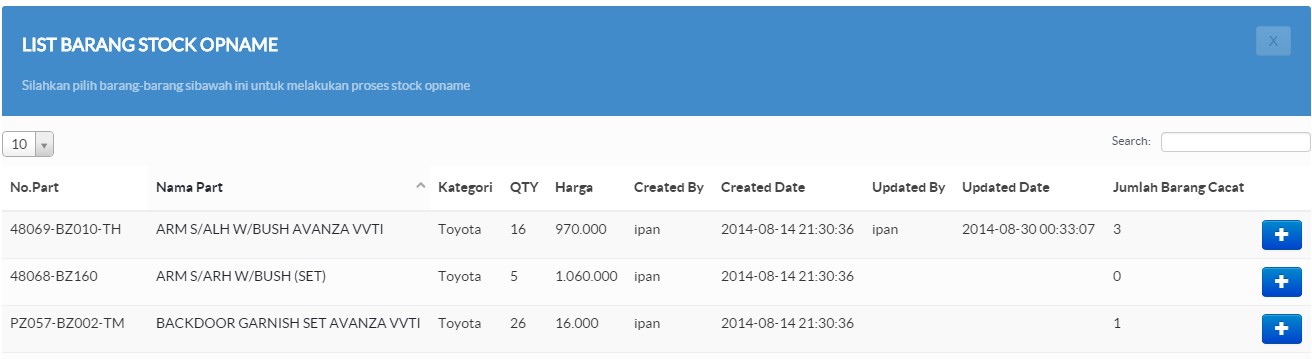
Data *stock opname* ini bisa di expo*rt* ke dalam *file* berformat .pdf maupun langsung di *print*. Tombol hijau berlogo centang berfungsi untuk memperbarui data. Tombol ini akan muncul hanya jika terdapat selisih dari qty baru dan qty lama. Jika selisih = 0 maka tombol ini tidak muncul. Sedangkan tombol merah untuk menghapus data *stock opname.*

Tombol list data yang telah diperbarui dapat menampilkan daftar barang yang telah melalui proses pembaharuan *stock opname.*



**Gambar 5.38** Halaman Data Stock Opname

### 5.2.1.26 Halaman Buat *Stock Opname*



**Gambar 5.39** Halaman Buat *Stock Opname*

Setelah memilih tombol buat stock opname pada gambar 5.38, akan muncul daftar barang seperti pada gambar 5.39. Pilih barang tersebut dengan cara menekan tombol plus berwarna biru.

### 5.2.1.27 Halaman Konfirmasi Proses *Stock Opname*

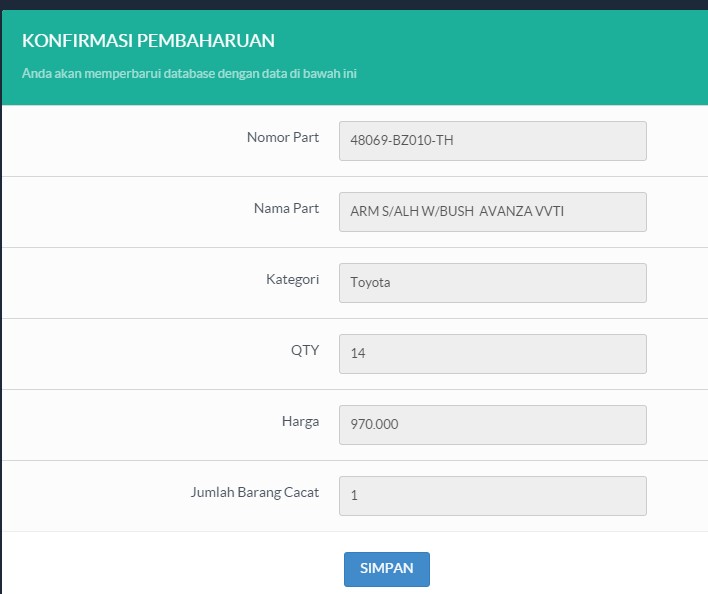
Detail barang yang dipilih akan ditampilkan, masukkan qty baru *stock opname* dan jumlah barang yang cacat jika ada, lalu simpan. Data ini akan masuk kedalam data stock opname seperti pada gambar 5.38 diatas.



**Gambar 5.40** Halaman Konfirmasi Proses Stock Opname

### 5.2.1.28 Halaman Perbarui Data

Halaman ini merupakan halaman konfirmasi pembaharuan yang muncul saat memilih tombol hijau berlogo centang. Konfirmasi ini di perlukan karena akan mengubah qty dan keterangan jumlah barang cacat dari data barang.



**Gambar 5.41** Halaman Perbarui Data

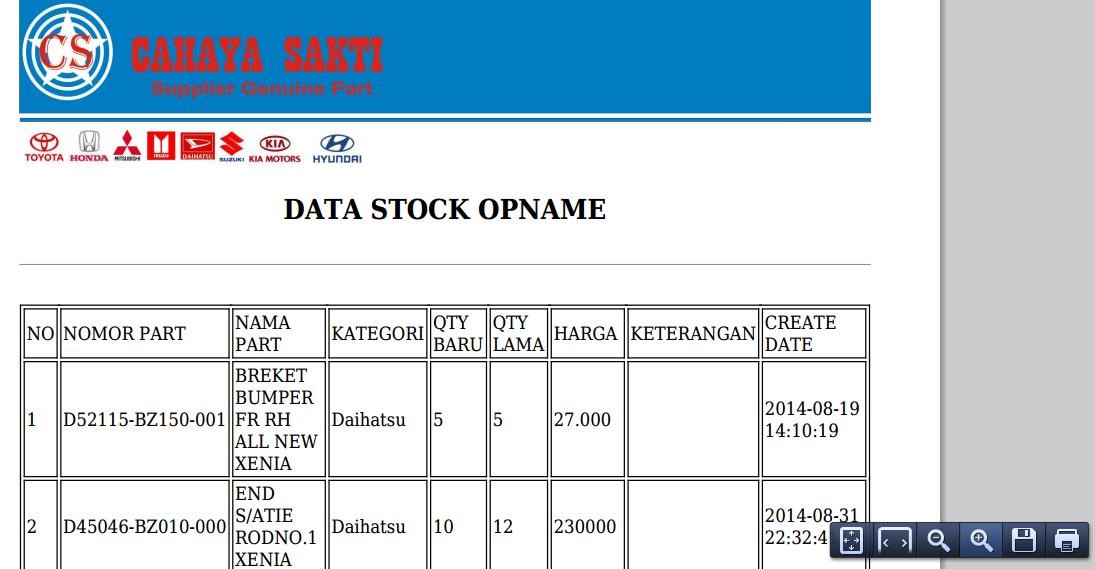
### 5.2.1.29 Halaman List Data Pembaharuan

Halaman list data pembaharuan merupakan daftar data *stock opname* yang telah diperbarui karena terjadi selisih.



**Gambar 5.42** Halaman List Data Pembaharuan

### 5.2.1.30 Halaman Laporan Data Stock Opname



**Gambar 5.43** Laporan Data Stock Opname

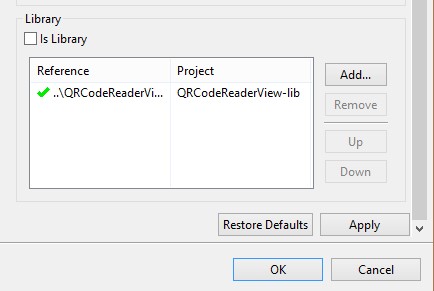
### 5.2.2 Aplikasi *Android*

Aplikasi android ini diperuntukkan bagi *user* untuk melakukan proses *stock opname* dengan menggunakan *QR Code*, serta mempermudah pencarian barang dengan fitur mencari melalui gambar *QR Code* atau melalui kata kunci tertentu.

Sebelumnya akan dijelaskan secara singkat bagaimana proses *QR Code* pada aplikasi ini bekerja.

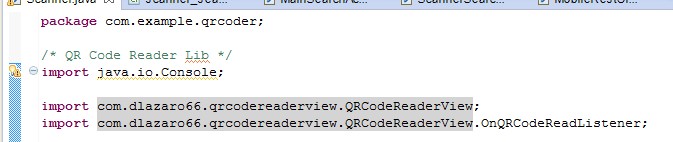
Aplikasi ini menggunakan *library* yang bernama QRCodeReaderView-lib.

Library ini berfungsi hanya untuk membaca gambar *QR Code.*

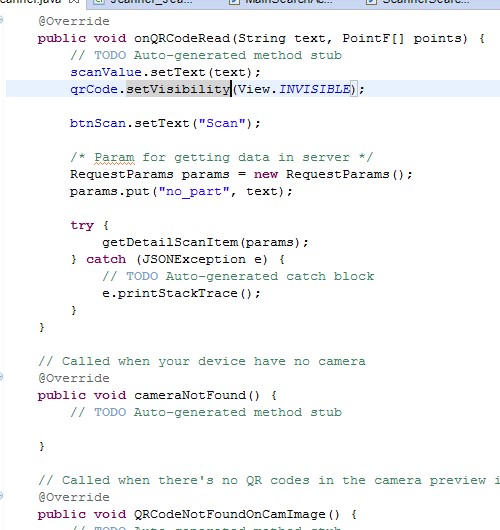


**Gambar 5.44** Project Terhubung dengan *Library*

Setelah terhubung, *library* tersebut di panggil dengan perintah dibawah ini:

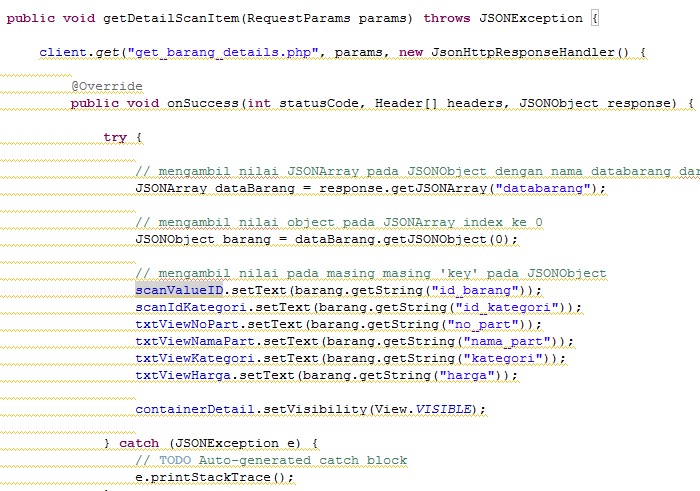


**Gambar 5.45** Import Library ke Dalam Project Lalu akan di baca dengan menggunakan perintah di bawah ini:



**Gambar 5.46** Script Membaca Gambar *QR Code*

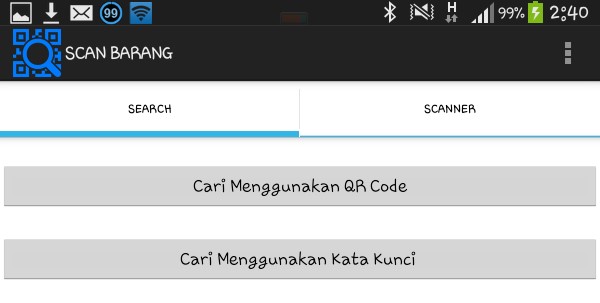
Setelah nomor part berhasil dibaca, detail barang berdasarkan nomor part tersebut akan muncul dengan perintah di bawah ini:



**Gambar 5.47** Script Menampilkan Detail Barang

#### 5.2.2.1 Halaman Utama Pencarian

Tampilan *android* ini terdiri dari 2 bagian, yaitu *search* dan *scanner.* Pada halaman *search* akan disediakan 2 pilihan pencarian, yaitu cari menggunakan *QR Code* dan cari menggunakan kata kunci. Hal ini ditujukan untuk lebih mempermudah *user* dalam pencarian barang yang diinginkan.

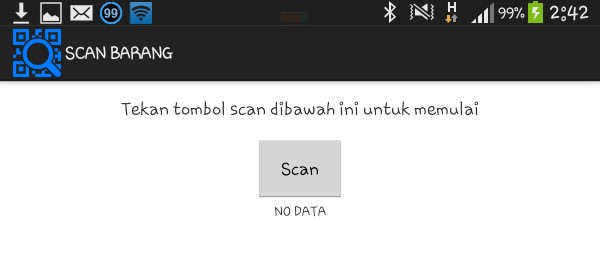


**Gambar 5.48** Halaman Pencarian Android

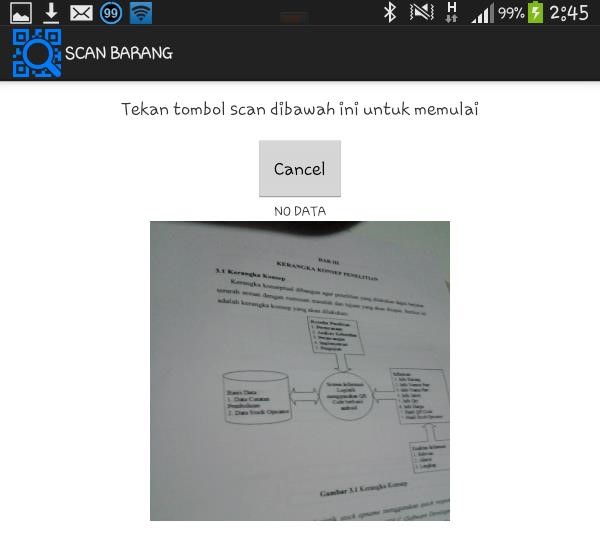
#### 5.2.2.2 Halaman Cari Menggunakan *QR Code*

Halaman ini terdiri dari 3 tampilan. Gambar 5.49 menampilkan tampilan awal sebelum pencarian, gambar 5.50 menampilkan bagaimana ketika fitur kamera aktif, dan gambar 5.51 menampilkan proses pencarian barang telah selesai.

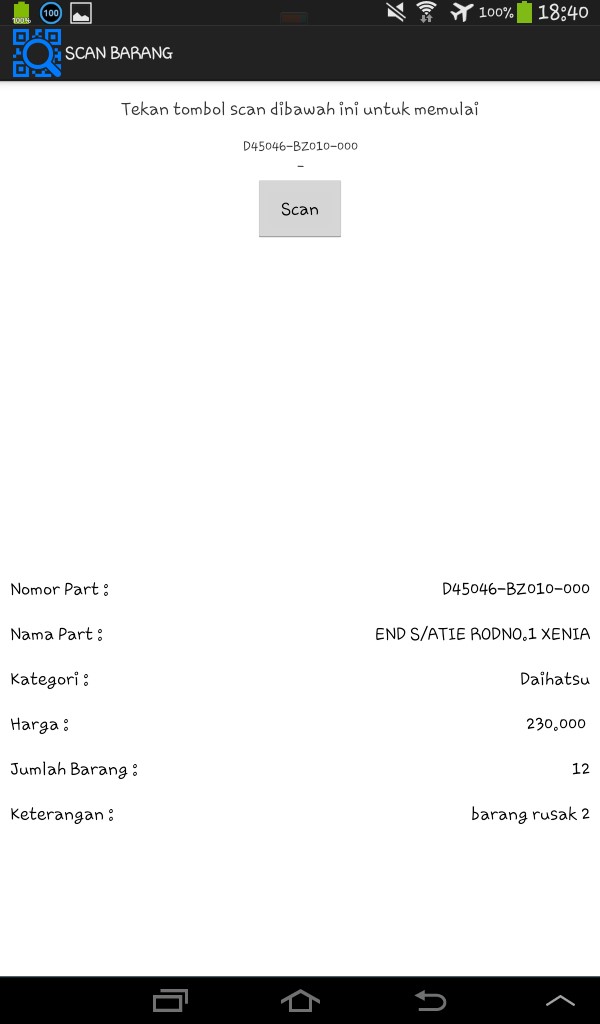
Fitur pencarian ini akan langsung mendapatkan detail barang yang dicari dengan cara scan gambar *QR Code* yang berisikan nomor part.



**Gambar 5.49** Tampilan Awal Pencarian Menggunakan *QR Code*



**Gambar 5.50** Kamera Aktif Saat Menekan Tombol Scan

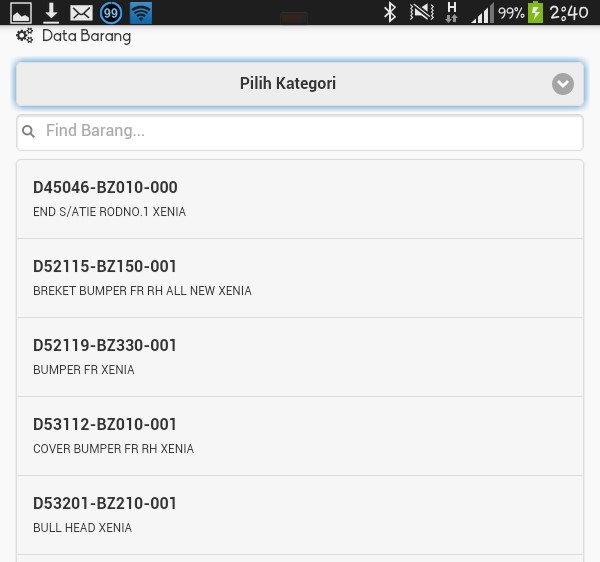


**Gambar 5.51** Hasil Search Barang

Fitur kamera yang aktif pada gambar 5.50 akan otomatis menutup sendiri ketika berhasil scan nomor part yang terdapat dalam gambar *QR Code*. Hasilnya, detail barang yaitu nomor part, nama part, kategori, harga, jumlah barang, dan keterangan dapat diketahui.

#### 5.2.2.3 Halaman Cari Menggunakan Kata Kunci

Halaman ini menampilkan seluruh data barang yang ada di *database server*. Pada gambar 5.52 terdapat pilihan menu pilih kategori untuk mempermudah dalam menyeleksi pencarian barang dan masukkan kata kunci pada kolom yang telah disediakan.



**Gambar 5.52** Tampilan Pencarian Barang Menggunakan Kata Kunci

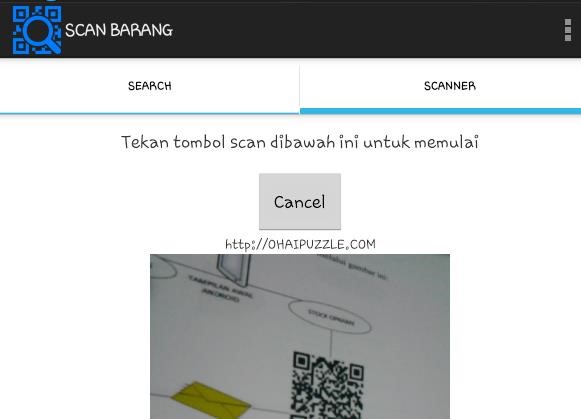
Gambar 5.53 dibawah ini adalah tampilan detail barang setelah mendapatkan barang yang sudah di cari. Detail barang tidak otomatis keluar seperti pada pencarian menggunakan *QR Code*. *User* harus memilih terlebih dahulu pada data barang yang telah berhasil dicari sebelumnya.



**Gambar 5.53** Tampilan Detail Barang

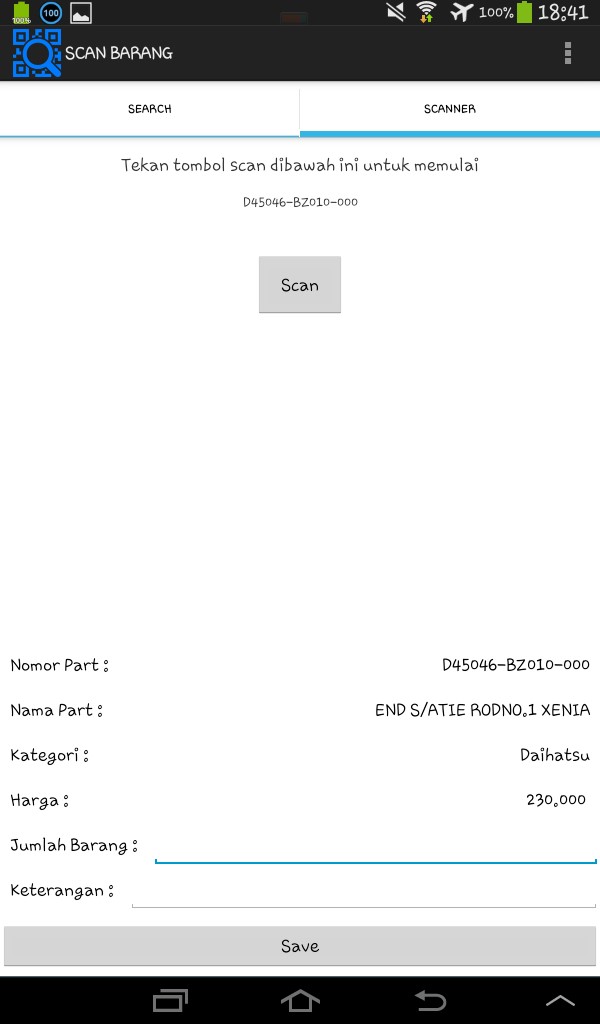
#### 5.2.2.4 Halaman Scanner

Halaman ini hampir sama dengan halaman pencarian menggunakan *QR Code,* yang membedakan adalah adanya proses lebih lanjut yaitu *stock opname.*



**Gambar 5.54** Tampilan Sebelum Melakukan Scan Barang

Setelah nomor part berhasil didapatkan dari scan gambar *QR Code*, detail barang akan muncul kecuali jumlah barang (qty) dan keterangan. Jumlah barang dan keterangan harus kita inputkan sendiri untuk membuat *data record* baru yaitu *stock opname*. Proses ini akan terus terulang (scan barang) hingga semua barang di gudang berhasil di scan.



**Gambar 5.55** Halaman Proses Stock Opname

**BAB VI**

# PENGUJIAN DAN PEMBAHASAN

## 6.1 Pengujian

Pengujian perlu dilakukan setelah implementasi database, sistem, aplikasi, dan konten. Apakah aplikasi ini berjalan sesuai dengan ketentuan dan keinginan atau tidak maka harus dilakukan pengujian secara fungsional. Pengujian pada aplikasi ini meliputi dua jenis pengujian, yaitu pengujian fungsional dan pengujian *QR Code*.

Pengujian sistem dilakukan dengan menggunakan metode *BlackBox Testing*. *BlackBox Testing* memberikan hasil tes kesesuaian aplikasi dengan fungsi atau kemampuan sistem secara fungsional untuk mengetahui apakah fitur-fitur dari aplikasi ini berjalan dengan semestinya atau tidak. Dengan mencoba segala kemungkinan yang terjadi, maka kita bisa mengetahui fitur mana sajakah yang perlu diperbaiki.

Pengujian yang kedua adalah pengujian *QR Code*. Pengujian ini bertujuan untuk mengetahui sejauh mana perangkat *android* dapat membaca gambar *QR Code*.

### 6.1.1 Skenario Uji Coba

Untuk mendapatkan hasil pengujian yang relevan, sistem diujicobakan dengan beberapa skenario uji coba. Skenario uji coba yang dilakukan adalah dengan melakukan proses *stock opname* melalui dua sistem yang berbeda, yaitu *stock opname* menggunakan *android* dan *stock opname* menggunakan web.

1. Skenario I (*Stock Opname* menggunakan *Android*)

Pada skenario uji coba I dengan menggunakan perangkat mobile berbasis *android,* user melakukan proses *stock opname* dengan *scan* gambar *QR Code.* Gambar tersebut terdiri dari nomor part untuk memanggil detail barang dan mengisikan form yang tersedia untuk membuat *stock opname*.

Operasi proses *stock opname* melalui *android* berhasil dijalankan pada

skenario ini :

* 1. Mengaktifkan fitur kamera pada *android.*
  2. Mengambil data dari *database server.*
  3. Menampilkan detail data barang dan menambahkan data baru kedalam tabel *stock opname* pada *database*
  4. Menampilkan informasi proses yang sedang berlangsung pada server.

1. Skenario II ((Stock Opname menggunakan Web)

Pada skenario uji coba II dengan menggunakan komputer *client* yang sudah terinstall sistem aplikasi web ini*,* user melakukan proses *stock opname* dengan memilih tombol buat *stock opname.* Halaman baru yang memuat daftar data barang akan muncul, dan user harus mencari lalu memilih barang tersebut untuk memulai proses stock opname. Setelah dipilih akan muncul halaman konfirmasi dan isi form yang tersedia untuk membuat *stock opname* Operasi proses *stock opname* melalui web berhasil dijalankan pada skenario

ini :

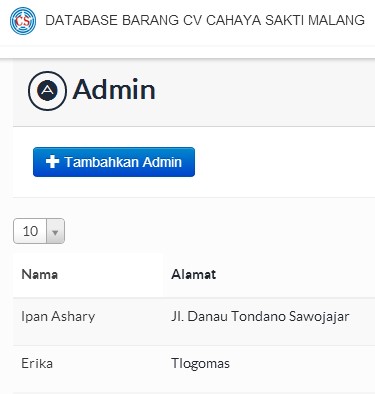
* 1. Mengambil data dari *database server.*
  2. Menampilkan daftar data barang yang tersedia pada tbl\_databarang
  3. Menampilkan detail data barang dan menambahkan data baru kedalam tabel *stock opname* pada *database.*
  4. Menampilkan informasi proses yang sedang berlangsung pada server.

Hasil skenario uji coba proses *stock opname* baik menggunakan *android* dan web pada skenario I dan skenario II dapat berjalan dengan baik. Uji coba skenario I dapat menampilkan detail data barang tanpa harus mencari barang tersebut. Berbeda dengan skenario II yang harus mencari dan memilih barang yang akan di *stock opname* untuk menampilkan detail barang. Hal tersebut membuat proses *stock opname* menggunakan *android* lebih efektif dan efisien.

### 6.1.2 Pengujian Fungsional (*BlackBox Testing)*

Pengujian fungsional merupakan tahapan dimana data-data yang ada dimasukkan untuk dilakukan uji coba bersamaan komponen sistem lain untuk mengetahui bahwa komponen sistem sudah berfungsi sesuai yang diharapkan dan sehingga berjalan secara fungsional. Pengujian Fungsional dilakukan dengan cara menjalankan aplikasi secara langsung dan runtut agar dapat diketahui mana fitur yang sudah berfungsi dengan baik, dan juga fitur mana yang tidak berjalan sesuai dengan fungsinya.

a. Pengujian Fungsional Data Admin

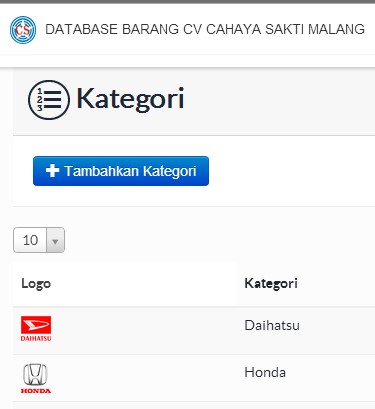


**Gambar 6.1** Pengujian Fungsional Data Admin

**Tabel 6.1** Pengujian Fungsional Data Admin

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| Mengisi *username* dan *password* saat *login* | *Username* dan *password* valid *Login* berhasil | Sesuai |
| Mengisi *username* dan *password* asal / kosong  saat *login* | *Username* dan *password* tidak valid/kosong.  *Login* gagal dan muncul pesan *error* | Sesuai |
| Mengisi data diri admin dengan data yang lengkap | Keluar peringatan sukses dan data ditampilkan dalam tabel | Sesuai |
| Mengisi data diri admin dengan data yang tidak  lengkap | Muncul peringatan gagal | Sesuai |
| Mengubah data diri admin dengan data yang lengkap | Muncul peringatan sukses dan data tersimpan | Sesuai |
| Mengubah data diri admin dengan data yang tidak  lengkap | Muncul peringatan gagal | Sesuai |
| Menghapus data diri admin | Muncul konfirmasi | Sesuai |
| Melihat data diri admin | Muncul data diri admin sesuai dengan data yang terpilih | Sesuai |

Pengujian Fungsional Kategori Barang

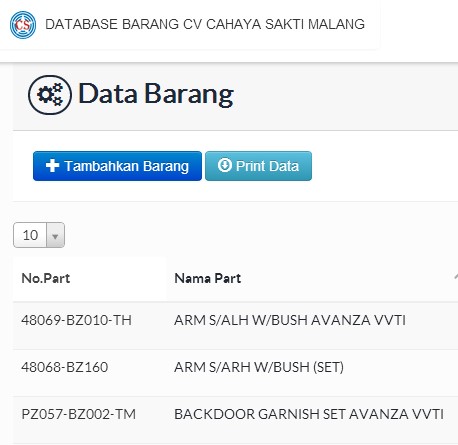


**Gambar 6.2** Pengujian Fungsional Kategori Barang

**Tabel 6.2** Pengujian Fungsional Kategori Barang

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| Mengisi data kategori barang dengan data yang  lengkap | Keluar peringatan sukses dan data ditampilkan dalam tabel | Sesuai |
| Mengisi data kategori barang dengan data yang tidak lengkap | Muncul peringatan gagal | Sesuai |
| Mengubah data kategori barang dengan data yang tidak lengkap | Muncul peringatan gagal | Sesuai |
| Menghapus data kategori barang | Muncul konfirmasi | Sesuai |
| Melihat data diri admin | Muncul data diri admin sesuai dengan data yang  terpilih | Sesuai |

Pengujian Fungsional Database Barang

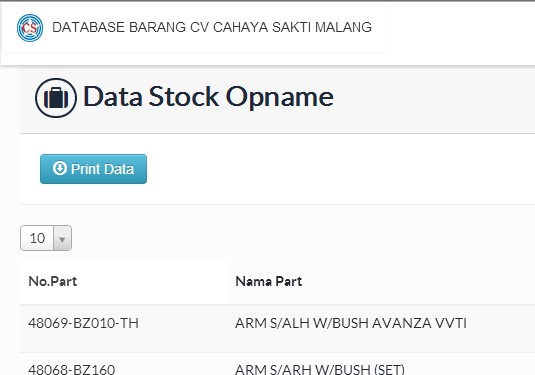


**Gambar 6.3** Pengujian Fungsional Database Barang

**Tabel 6.3** Pengujian Fungsional Database Barang

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| Mengisi data barang dengan data yang lengkap | Keluar peringatan sukses dan data ditampilkan dalam tabel | Sesuai |
| Mengisi data barang dengan data yang tidak  lengkap | Muncul peringatan gagal | Sesuai |
| Mengubah data barang dengan data yang tidak  lengkap | Muncul peringatan gagal | Sesuai |
| Menghapus data barang | Muncul konfirmasi | Sesuai |
| Melihat data barang | Muncul data barang sesuai dengan data yang terpilih | Sesuai |
| Download Laporan (Eksport) | Laporan data barang siap di cetak dan di simpan | Sesuai |

Pengujian Fungsional Data Stock Opname

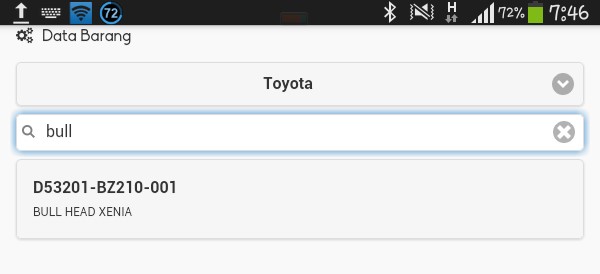


**Gambar 6.4** Pengujian Fungsional Data Stock Opname

## Tabel 6.4 Pengujian Fungsional Data Stock Opname

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| Eksport data | Keluar data *stock opname* siap cetak | Sesuai |

1. Pengujian Fungsional Search Android

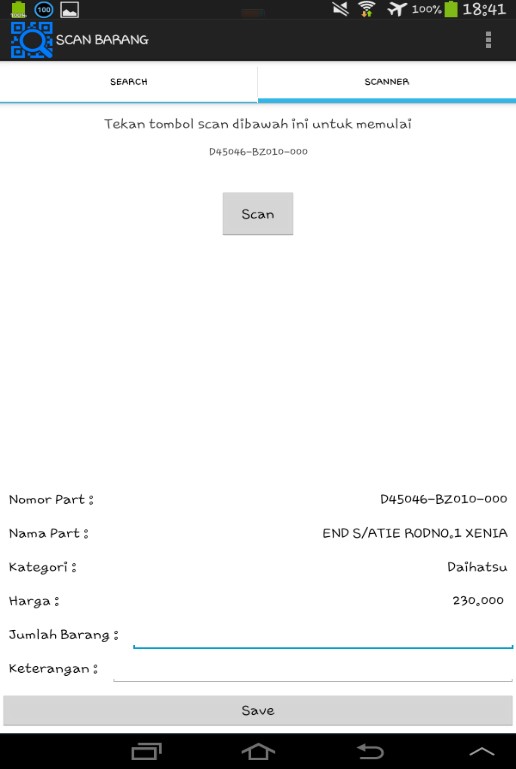


**Gambar 6.5** Pengujian Fungsional Search Android

**Tabel 6.5** Pengujian Fungsional Search Android

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| Pencarian dengan *scan* gambar *QR Code* dengan  nomor part sesuai dengan  *database* | Keluar nomor part beserta detail barang | Sesuai |
| Pencarian dengan *scan* gambar *QR Code* dengan  nomor part tidak sesuai dengan *database* | Keluar nomor part tetapi tidak dengan detail barangnya | Sesuai |
| Pencarian dengan *scan* gambar *QR Code* dengan  nomor part sesuai dengan *database* tetapi tidak terhubung dengan internet | Keluar nomor part tetapi tidak dengan detail barangnya | Sesuai |
| Pencarian dengan *scan* gambar *QR Code* tetapi kamera tidak aktif | Tidak bisa scan | Sesuai |
| Pencarian dengan memasukkan kata kunci | Hasil pencarian ditemukan | Sesuai |
| Pencarian dengan tidak memasukkan kata kunci | Muncul list kategori dan list data barang | Sesuai |
| Pencarian dengan memasukkan kata kunci tanpa terhubung internet | List kategori dan list data barang tidak muncul | Sesuai |

1. Pengujian Fungsional Scanner Android



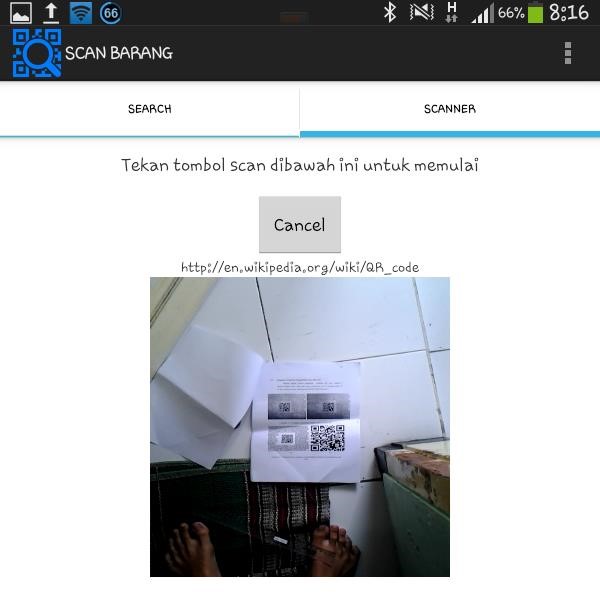
**Gambar 6.6** Pengujian Fungsional Scanner Android

**Tabel 6.6** Pengujian Fungsional Scanner Android

|  |  |  |
| --- | --- | --- |
| **Proses** | **Hasil** | **Keterangan** |
| *Scan* gambar *QR Code* dengan nomor part sesuai dengan *database* | Keluar nomor part, detail barang, tombol save, dan masukkan qty | Sesuai |
| *Scan* gambar *QR Code* dengan nomor part tidak sesuai dengan *database* | Hanya keluar nomor part | Sesuai |
| Pencarian dengan *scan* gambar *QR Code* dengan  nomor part sesuai dengan *database* tetapi tidak terhubung dengan internet | Keluar nomor part tetapi tidak dengan detail barang dan tombol save | Sesuai |
| Pencarian dengan *scan* gambar *QR Code* tetapi kamera tidak aktif | Tidak bisa scan | Sesuai |

### 6.1.3 Pengujian *QR Code*

1. Pengujian ukuran, jarak dan tingkat kecahayaan.



(a) (b)

**Gambar 6.7** (a)Pengujian *QR Code* 6x6 cm; (b) Pengujian *QR Code* 3x3 cm **Tabel 6.7** Pengujian ukuran, jarak dan tingkat kecahayaan

|  |  |  |
| --- | --- | --- |
| **Kondisi *QR Code*** | ***QR Code* 3x3 cm** | ***QR Code* 6x6 cm** |
| Normal (jarak ± 10 cm) | Diterima | Diterima |
| Jarak menengah (± 35 cm) | Diterima | Diterima |
| Jarak menengah (± 35 cm) Cahaya minim | Ditolak | Diterima |
| Jarak maksimal (± 70 cm) | Ditolak | Diterima |
| Jarak maksimal (± 70 cm) Cahaya minim | Ditolak | Ditolak |

1. Pengujian tingkat kerusakan atau terkena kotoran

(c)



(

a

)

(

b

)

**Gambar 6 8** (a)Pengujian Kerusakan Kecil; (b) Pengujian Kerusakan Besar; (c) Pengujian Arah terbalik

**Tabel 6.8** Pengujian Tingkat Kerusakan atau Terkena Kotoran

|  |  |
| --- | --- |
| **Deskripsi** | **Keterangan** |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian kecil pada bagian pojok bawah | Diterima |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian kecil secara vertikal | Diterima |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian kecil secara horizontal | Diterima |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian kecil pada bagian pojok atas | Diterima |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian | Ditolak |
| Menguji pembacaan dengan *QR Code* yang kotor sebagian besar | Ditolak |
| Menguji pembacaan dengan *QR Code* yang kotor di 1 bagian *finder pattern* | Ditolak |
| Menguji pembacaan dengan *QR Code* yang kotor di 2 bagian *finder pattern* | Ditolak |
| Menguji pembacaan dengan *QR Code* yang rusak sebagian kecil | Diterima |
| Menguji pembacaan dengan *QR Code* yang rusak sebagian | Diterima |
| Menguji pembacaan dengan *QR Code* yang rusak sebagian besar | Ditolak |
| Menguji pembacaan dari *QR Code* dengan arah terbalik | Diterima |

### 6.2 Pembahasan

#### 6.2.1 Analisa Pengujian *QR Code*

Perlu diperhatikan bahwa perangkat lunak atau aplikasi yang dibangun ini tidak mempengaruhi kinerja dari sistem pada s*martphone* yang digunakan dan tidak membebani RAM. Aplikasi ini membutuhkan kamera untuk melakukan pembacaan *QR Code,* karena itu s*martphone* yang digunakan untuk memasang aplikasi ini harus terdapat kamera. Kamera yang digunakan untuk melakukan pembacaan sebaiknya memiliki resolusi minimal 3,2 megapiksel karena kualitas gambar akan mempengaruhi kinerja pembacaan yang dilakukan oleh aplikasi s*canner QR Code.* Berdasarkan pengujian yang telah dilakukan pada sistem *stock opname,* kebutuhan fungsional dari seluruh aplikasi sudah terpenuhi. Berdasarkan pengujian *QR Code* yang dilakukan diatas, aplikasi *scanner* *reader* yang ada pada sistem ini dibangun cukup handal dan dapat membaca *QR Code* dengan berbagai kondisi. Kondisi-kondisi yang telah diuji untuk membuktikan kehandalan aplikasi *scanner* *reader* adalah kondisi *QR Code* rusak, kotor, tidak dalam posisi yang benar, jarak, ukuran, hingga kepada pencahayaan.

Namun, toleransi terhadap kerusakan dan kotor pada gambar *QR Code* yang dibaca terbatas. Pada pengeksekusiannya tidak selalu didapat hasil yang optimal. Berikut adalah contoh kegagalan yang terjadi pada saat pengujian *QR Code* dengan kondisi normal jarak ± 10 cm.



**Gambar 6.9** Kegagalan pendekteksian *QR Coede* dalam kondisi normal

Dari gambar 6.9 diatas dapat diketahui bahwa faktor yang menyebabkan kegagalan dalam proses pendeteksian *QR Code* adalah karena bentuk *QR Code* yang tidak simetris dan objek *QR Code* yang rusak.

Dari beberapa analisis faktor kegagalan dalam pengujian diatas, maka didapatkan beberapa factor kegagalan tersebut, antara lain :

1. Pencahayaan yang terlalu gelap ataupun terlalu terang sehingga membuat *QR Code* tidak terlihat sepenuhnya.
2. Kondisi *QR Code* yang kurang baik, seperti warnanya sudah pudar atau kertas tempat *QR Code* berada dalam kondisi yang jelek.
3. Terdapat berbagai macam benda yang dapat menutupi *QR Code* lebih dari 30%, sehingga mengakibatkan data pada *QR Code* tidak dapat terbaca.
4. Semakin jauh pengambilan jarak akan sangat mempengaruhi dalam pembacaan *QR Code*

#### 6.2.2 Analisa Pengujian *User*

Pada tahap ini dilakukan pengujian aplikasi sistem informasi logistik *stock opname* menggunakan *QR Code* pada CV Cahaya Sakti Malang yang digunakan untuk mengetahui apakah aplikasi sudah dibangun sudah bekerja sesuai dengan tujuan. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan sebenarnya. Penulis melakukan penilaian terhadap *user* dengan menggunakan media wawancara.

Pengujian aplikasi dilakukan dengan wawancara, selama proses uji coba aplikasi yang dilakukan oleh *user* *secara* langsung juga disertakan wawancara dengan menanyakan pendapat *user* dengan aplikasi tersebut.

Wawancara dilakukan dengan menggunakan teknik *purposive sampling* dengan mengambil sample dari berbagai profesi kerja, serta menentukan usia dari responden, yang disebarkan kepada 6 responden yang merupakan user dari aplikasi tersebut. Berikut beberapa pertanyaan yang diberikan kepada responden dapat dilihat dalam tabel berikut.

**Tabel 6.9** Tabel Pertanyaan Wawancara

|  |  |
| --- | --- |
| No | Pertanyaan |
| 1 | Apakah sebelumnya anda pernah mengetahui tentang teknologi *Quick Response Code*? |
| 2 | Bagaimana dengan tampilan dari aplikasi sistem informasi logistik *stock opname* untuk android ini dan tampilan web untuk admin? |
| 3 | Apakah aplikasi dan web nya dapat digunakan dengan cukup mudah atau cukup sulit? |
| 4 | Apakah aplikasi ini dapat membantu dalam praktek lapangan ketika melakukan *stock opname?* |
| 5 | Bagaimana perusahaan ini menyikapi jika ada barang yang selisih (tidak sesuai atau hilang)? |

Berikut data yang didapat berdasarkan hasil wawancara: a. 3 responden berumur 19-31 tahun

1. 3 responden berumur 32-44 tahun
2. 4 responden adalah laki-laki
3. 2 responden adalah wanita

Berikut ini hasil pernyataan masing-masing nilai jawaban yang diujikan kepada 6 responden.

1. Sdri Erika Dian (Admin Stock)

Dalam hasil wawancara yang dilakukan dengan sdri Erika, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa belum mengetahui tentang *QR Code,* dan tampilan aplikasinya kurang menarik. Namun saat diuji coba beliau terkesan karena sangat membantu dan lebih memudahkan. Beliau bercerita sebelumnya proses *stock opname* di perusahaan ini masih manual, setiap 6 bulan sekali. Satu persatu barang harus dihitung dan saat sudah selesai catatanya harus dipindah ke dalam file excel secara manual juga. Berhubung tugas saya merekap stock, maka dengan aplikasi ini dapat banyak membantu. Sekali scan lalu memasukkan qty, jumlah barang sudah langsung masuk ke web dan otomatis terekap. Untuk barang selisih beliau mengatakan akan dicari lebih lanjut lagi datanya karena pekerjaan beliau hanya berupa data. Biasanya dicari dari faktur *invoice* yang hilang atau barangnya di jual kemana.

1. Bapak Andri Zainul (Logistik)

Dalam hasil wawancara yang dilakukan dengan Bapak Andri, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa belum mengetahui tentang *QR Code,* sedangkan tampilannya menurut beliau cukup bagus, walaupun sederhana. Beliau mengatakan bahwa pada saat *stock opname,* bagian logistic berperan untuk menghitung barang satu persatu dan mencari nomor part nya untuk dimasukkan ke rekapan stock. Dengan adanya aplikasi ini, tak perlu sulit sulit mencari nomor part nya karena sudah disediakan pencarian melalui scan tersebut. Dengan scan *QR Code* nya sudah muncul nomor part dan keterangan detail dari part tersebut. Sementara mengenai barang yang hilang, harus di cek lagi dan dibenarkan data yang ada di rekap *stock.* Karena kemungkinan bisa saja ada nota barang masuk dan keluar yang terselip.

1. Bapak Ichrom Masrudin (Logistik)

Dalam hasil wawancara yang dilakukan dengan Bapak Ichrom, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa belum pernah mengetahui tentang *QR Code.* Tanggapan beliau biasa saja baik tampilan android maupun web nya. Menurut beliau aplikasi ini sebenarnya bagus, tetapi karena harus menggunakan aplikasi *android* jadi agak kesulitan. Kita harus punya *handphone* *android,* sedangkan disini tidak semua punya *android.* Beliau juga tidak begitu mengerti menggunakan aplikasi aplikasi di *handphone* *android.* Lalu mengenai kemungkinan selisih sebenarnya sudah di perkecil dengan diterapkannya peraturan untuk menerima dan mengeluarkan barang yang harus melalui pengecekan pihak logistic dan admin *stock.* Tetapi kesalahan bisa terjadi, dan solusinya adalah *stock opname* berkala untuk mengetahui jumlah pasti barang yang ada saat ini.

1. Sdr Achmad Nuri (Sales)

Dalam hasil wawancara yang dilakukan dengan Sdr Achmad, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa pernah dengar tentang *QR Code* dan tidak peduli tentang tampilannya. Menurut beliau yang penting fungsi dari aplikasi ini. Beliau terbiasa bekerja dilapangan untuk menarik pelanggan. Terkadang ketika pelanggan sudah didapat, pelanggan tersebut suka menanyakan hal-hal seperti sisa berapa ini part nya, berapa harganya. Beliau sendiri mengaku sering tidak ingat karena banyak nya jenis part / barang yang ada. Dengan aplikasi ini, fitur pencarian nya sangat berguna. Baik itu langsung scan, maupun melalui kata kunci. Mengenai barang selisih menurt beliau pasti ada di kesalahan kita sebagai manusia, dalam hal ini lupa dan lalai. Maka dari itu biasanya langsung diusut hingga tuntas, dan langsung di rubah data sebenarnya saat ini yang benar berapa.

1. Bapak Moch Ali (Kepala Sales)

Dalam hasil wawancara yang dilakukan dengan Bapak Ali, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa mengetahui tentang teknologi *QR Code* yang biasanya di pakai dalam produk makanan, minuman hingga koran. Sedangkan tampilannya menurut beliau cukup sederhana, tidak ribet, dan tidak terlalu banyak fungsi. Hal tersebut yang menurut beliau jadi mudah untuk digunakan. Aplikasi *stock opname* sudah sangat membantu, yang lebih saya lihat itu web rekapan stocknya. Sebagai sales, beliau harus selalu tahu jumlah barang di stock. Dengan adanya web ini bisa dilihat dengan mudah jumlah barang yang ada di stock. Mengenai barang yang selisih, konsep perusahaan ini adalah kekeluargaan. Kebanyakan orang yang kerja disini sudah saling kenal lama atau berteman. Jadi tidak mungkin ada barang yang hilang karena di curi. Kemungkinan besar pasti karena ada kesalahan dari data atau kesalah orang (*Human Error*) baik itu sales, bagian logistik, atau admin *stock* yang lalai.

1. Sdri Qisthina (Kepala Admin)

Dalam hasil wawancara yang dilakukan dengan Sdri Qisthina, dari beberapa pertanyaan yang diajukan beliau menyatakan bahwa mengetahui tentang teknologi *QR Code* yang biasa diterapkan di *social media*. Mengenai tampilan menurut beliau terlalu biasa, mungkin bisa dikembangkan dan dibuat lebih menarik lagi. Saat digunakan untuk *stock opname* sangat membant, cukup efektif dan menghemat waktu. Apalagi untuk rekapan data barang setelah *stock opname* yang akan diserahkan kepada *owner* dapat segera dilaporkan saat itu juga dengan sekali print. Fitur *sorting* tiap kolom di web sangat membantu, juga fitur searching nya sekaligus memudahkan untuk mencari part yang kurang dan perlu di order lagi. Sedangkan mengenai barang yang hilang atau selisih, beliau mengatakan tidak ada ganti rugi. Karena pasti adanya *human error.* Jadi harus langsung di betulkan rekapannya yang benar jumlahnya berapa. Tetapi tetap harus dicari kesalahannya ada di data atau di barang, dan kelalaian ada di pihak admin, sales, atau bagian logistic. Karena kedepannya untuk memperbaiki alur keluar masuk barang agar tidak terjadi kesalahan lagi. Sehingga setiap kesalahan yang terjadi akan membuat perbaikan agar tidak terulang kembali.

**Tabel 6.10** Tabel Kesimpulan Pengujian User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Responden | Sistem informasi logistik *stock opname* menggunakan *QR Code* | | | | |
| STM | KM | CM | M | SM |
| A |  |  |  |  |  |
| B |  |  |  |  |  |
| C |  |  |  |  |  |
| D |  |  |  |  |  |
| E |  |  |  |  |  |
| F |  |  |  |  |  |

Keterangan:

* + STM = Sangat Tidak Menarik
  + KM = Kurang Menarik
  + CM = Cukup Menarik
  + M = Menarik
  + SM = Sangat Menarik

Dari hasil wawancara kepada 6 responden dapat disimpulkan bahwa aplikasi *stock opname* yang telah dibuat mengungkapkan tingkat ketertarikan terhadap aplikasi ini.

**BAB VII**

# PENUTUP

## 7.1 Kesimpulan

Berdasarkan implementasi, pengujian, dan analisis yang telah dilakukan, ada beberapa kesimpulan yang dapat diambil. Kesimpulan-kesimpulan tersebut adalah sebagai berikut :

1. Sistem informasi logistik dapat diterapkan secara mobile dengan menampilkan daftar data barang, detail data barang, dan fitur pencarian baik itu melalui *QR Code* atau kata kunci.
2. *QR Code* dapat dimanfaatkan sebagai alat identifikasi dalam sebuah sistem informasi logistik *stock opname* menggunakan perangkat mobile berbasis *android*, karena *QR Code* dapat menyimpan informasi yang cukup, dalam hal ini adalah untuk nomor part.
3. *Smartphone Android* dapat dimanfaatkan dan diintegrasikan untuk membaca *QR Code* dengan menggunakan sistem informasi logistik *stock opname* tanpa membebani kinerja dari smartphone tersebut. Meskipun mengambil *data* dari *server*.
4. Semua data barang dan hasil *stock opname* dapat ditampilkan dan di olah melalui halaman web untuk admin.
5. Sistem informasi logistik *stock opname* yang telah dibangun terbukti cukup handal dalam melakukan pembacaan *QR Code* baik itu *QR Code* yang rusak, kotor, ataupun terbalik posisinya.

## 7.2 Saran

Penelitian ini masih dapat dikembangkan dan disempurnakan, berikut merupakan beberapa hal yang disarankan untuk pengembangan aplikasi ini, yaitu:

1. Penggunaan *QR Code* bisa lebih dimanfaatkan dalam hal barang masuk maupun barang keluar yang terjadi menggunakan android.
2. Fitur cetak laporan berdasarkan tanggal tertentu atau rentang waktu tertentu yang dapat di inputkan user atau admin sesuai ke hendaknya.