

# Rancang Bangun Aplikasi Pendataan Persediaan Barang untuk Proses *Stock Opname* Menggunakan *Barcode* Berbasis Android pada Perusahaan Manufaktur

Nurul Huda  
Program Studi Ilmu Komputer  
Universitas Negeri Jakarta  
Jakarta, Indonesia  
nurulhuda0379@gmail.com

Ir. Fariani Hermin Indiyah, M.T.  
Program Studi Ilmu Komputer  
Universitas Negeri Jakarta  
Jakarta, Indonesia  
farianihermin828@gmail.com

Ratna Widyati, S.Si., M.Kom  
Program Studi Ilmu Komputer  
Universitas Negeri Jakarta  
Jakarta, Indonesia  
ratna.widyati@unj.ac.id

**Abstraksi**—*Stock opname* merupakan kegiatan membandingkan jumlah persediaan barang yang ada di gudang atau toko dengan yang ada di catatan pembukuan. Kegiatan ini merupakan salah satu kegiatan yang penting untuk dilakukan oleh sebuah perusahaan agar perusahaan tersebut mendapatkan data mengenai jumlah persediaan barang di gudang, sehingga dapat memastikan kebenaran dari data dalam pembukuan perusahaan tersebut. Penelitian ini dilakukan pada sebuah perusahaan cat di Tangerang. Perusahaan tersebut membutuhkan sistem baru untuk menggantikan sistem yang sebelumnya digunakan dalam melakukan pendataan persediaan barang pada proses *stock opname*. Adapun tujuan yang akan dicapai pada penelitian ini adalah membuat aplikasi yang dapat menggantikan sistem yang sebelumnya digunakan dalam melakukan proses pendataan persediaan barang di gudang oleh perusahaan terkait, sehingga dapat membantu perusahaan tersebut dalam menjalankan proses *stock opname*. Aplikasi tersebut dibangun dengan berbasis Android dan diharapkan dapat melakukan pemindaian terhadap *barcode* dengan menggunakan kamera pada perangkat, serta dapat mencocokkan data hasil pemindaian dengan data pada *database* perusahaan terkait. Metode penelitian yang digunakan terdiri dari studi literatur dan analisis kebutuhan, perancangan Scrum dan desain aplikasi, implementasi, dan pengujian sistem. Hasil pengujian menunjukkan bahwa keseluruhan sistem berfungsi sebesar 98,54% sehingga dapat disimpulkan bahwa aplikasi berhasil dibangun dengan sangat baik.

**Kata kunci**—Stock Opname, Barcode, Kamera, Android

## I. PENDAHULUAN

Gudang merupakan salah satu tempat penting bagi perusahaan karena berbagai persediaan barang milik perusahaan diletakkan di sana. Ketersediaan barang yang ada di gudang perlu dipastikan agar kebenaran catatan dalam pembukuan dapat diketahui. Hal itu dapat meminimalisir penyimpangan terhadap jumlah persediaan barang, baik kekurangan maupun kelebihan, juga dapat dilakukan tindakan sesuai dengan perbedaan jumlah persediaan barang yang terjadi. Kegiatan yang dilakukan untuk memastikan ketersediaan barang di gudang adalah *stock opname*. *Stock opname* merupakan kegiatan membandingkan jumlah persediaan barang yang ada di gudang atau toko dengan yang ada di catatan pembukuan [1].

Pada proses *stock opname*, terdapat proses pendataan persediaan barang di gudang. Tujuan pelaksanaan kegiatan ini adalah untuk mendapatkan data mengenai jumlah persediaan barang yang ada di gudang. Data tersebut akan dibandingkan dengan data yang ada di pembukuan guna memastikan kebenaran dari catatan dalam pembukuan. Proses pencatatan persediaan barang dengan jumlah banyak cukup memakan waktu. Terlebih lagi jika perusahaan tersebut memiliki jenis barang yang banyak. Perusahaan yang sedang melaksanakan *stock opname* juga harus menghentikan pergerakan barang masuk dan keluar dari gudang. Hal tersebut dapat mengganggu kegiatan produksi dan menyebabkan kerugian jika terlalu lama dilakukan. Jadi, akan lebih baik jika proses *stock opname* selesai dalam waktu singkat. Proses pendataan persediaan barang merupakan kegiatan yang paling banyak menyita waktu. Jika proses tersebut dapat diselesaikan dengan lebih cepat, maka proses *stock opname* dapat selesai dengan lebih cepat pula.

Perkembangan teknologi saat ini dapat digunakan untuk mempermudah proses pendataan persediaan barang, contohnya adalah dengan menggunakan *barcode* untuk mengidentifikasi barang. *Barcode* dapat membuat proses lebih cepat dan efisien, dan juga memperkecil kemungkinan terjadinya kesalahan pencatatan barang. Kemudian, *smartphone* atau komputer tablet dapat digunakan sebagai media aplikasi yang memiliki pemindai *barcode* dengan menggunakan kamera dan melakukan pencatatan jumlah persediaan barang, serta membuat laporan hasil pendataan tersebut. Para pekerja yang bertugas melakukan pendataan barang dapat memasang aplikasi tersebut pada perangkat yang tersedia. Penggunaan *barcode* sebelumnya pernah digunakan oleh Roslan., Othman., dan Razak untuk melakukan verifikasi registrasi produk farmasi dan status registrasi dari produk tersebut di Departemen Kesehatan Malaysia [2]. Selain itu, terdapat beberapa penelitian terkait aplikasi untuk membantu proses *stock opname*. Penelitian yang pertama dilaksanakan oleh Yudha., Sudarma., dan Mertasana untuk membantu meningkatkan pelayanan pegawai Krisna Oleh-Oleh Khas Bali dalam memberi informasi ketersediaan barang [3]. Penelitian lainnya dilaksanakan oleh Setya Budi Santosa untuk membantu proses pembelian dan pengeluaran barang, serta membantu proses pembuatan laporan mutasi pada apotek Setya Budi [4].

Dengan menjadikan penelitian-penelitian yang telah disebutkan sebelumnya sebagai bahan acuan penelitian, Penulis ingin menjalankan sebuah penelitian yang bertujuan untuk menghasilkan aplikasi dengan fitur pemindai *barcode* berbasis Android yang dapat membantu proses pendataan jumlah persediaan barang di gudang barang jadi (*finished goods*) milik

salah satu perusahaan cat yang bertempat di Tangerang saat *stock opname* dilaksanakan. Aplikasi tersebut juga akan dirancang agar dapat mengunduh data dari sistem *database* yang sudah ada di perusahaan terkait dan mengirim hasil pendataan persediaan barang ke *server* tempat penyimpanan data hasil tersebut. Aplikasi ini diharapkan dapat menggantikan aplikasi yang sebelumnya dan mempermudah proses tersebut sehingga kegiatan *stock opname* pada perusahaan terkait dapat terselesaikan dengan waktu yang diharapkan.

## II. KAJIAN TEORI

### A. Stock Opname

*Stock opname* atau *stock taking* merupakan kegiatan membandingkan jumlah persediaan barang yang ada di gudang atau toko dengan yang ada di catatan pembukuan [1]. *Stock opname* biasanya dilaksanakan secara independen agar dapat memberikan hasil yang benar-benar akurat. Perusahaan yang sedang melaksanakan *stock opname* harus menghentikan pergerakan barang masuk dan keluar dari gudang. Proses *stock opname* dimulai dengan melakukan pendataan jumlah persediaan barang di gudang. Setelah itu, hasil dari pendataan persediaan barang di gudang akan dibandingkan dengan data pembukuan. Jika terdapat selisih saat proses perbandingan selesai, maka perlakuan selisih disesuaikan dengan kebijakan perusahaan.

### B. Proses Stock Opname pada Perusahaan Terkait

Waktu rata-rata yang diperlukan untuk melakukan *stock opname* pada perusahaan terkait adalah tiga hari. Terdapat dua cara yang digunakan untuk melakukan proses *stock opname* pada perusahaan tersebut, yaitu *Physical Inventory* dan *Cycle Counting*. *Physical Inventory* merupakan cara menghitung persediaan barang secara menyeluruh. Sedangkan *Cycle Counting* adalah cara penghitungan sebagian atau salah satu jenis persediaan barang. *Cycle Counting* lebih sering dilakukan karena waktu yang diperlukan lebih sedikit jika dibandingkan dengan menggunakan cara *Physical Inventory*. *Cycle Counting* juga dipilih agar kegiatan produksi tidak perlu dihentikan sepenuhnya melainkan hanya pada persediaan barang yang sedang dalam proses tersebut.

Proses *stock opname* pada perusahaan ini dimulai dengan melakukan pendataan persediaan barang di gudang. Setelah semua data persediaan barang selesai diisi, data tersebut akan disimpan ke *server*. Hasil dari pendataan persediaan barang di gudang kemudian diproses ke sistem aplikasi sebelum diposting. Sistem aplikasi yang dipakai adalah Oracle E-Business Suite R.12 (EBS R.12). Data dari proses pendataan persediaan barang di gudang akan dibandingkan dengan data pembukuan dengan bantuan sistem aplikasi tersebut. Proses posting merupakan akhir rangkaian dari siklus *stock opname* di sana.

### C. Barcode

*Barcode* adalah tampilan dari data yang bisa dibaca dengan menggunakan mesin optik yang berkaitan dengan benda yang dilampirkan [5]. *Barcode* digunakan untuk menyandikan informasi dalam pola visual, yang dapat digunakan untuk melacak produk, harga, dan tingkat stok untuk perekaman terpusat dalam sistem perangkat lunak [6]. Gambar 2.1 menunjukkan contoh dari *barcode*.



Gambar 2.1 Contoh *Barcode*

### D. Android

Android merupakan susunan *software* untuk perangkat seluler yang terdiri dari sistem operasi, *middleware*, dan aplikasi utama [7]. Android dikembangkan untuk perangkat *mobile* berbasis Linux [8]. Android pertama kali dikembangkan oleh Android Inc. lalu pada tahun 2005 dibeli oleh pihak Google. Open Handset Alliance (OHA) dibentuk pada tahun 2007 sebagai usaha pengembangan Android.

### E. Android Studio

Merupakan Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) legal yang digunakan untuk mengembangkan aplikasi berbasis Android, yang didasarkan pada IntelliJ IDEA [9]. Android Studio menyediakan beberapa fitur seperti Sistem *build* berbasis Gradle, Emulator, IDE yang dapat mengembangkan aplikasi untuk perangkat Android, Framework dan fitur pengujian, dan beberapa fitur lainnya.

### F. MySQL

MySQL adalah salah satu *database* yang dapat digunakan secara gratis [10]. *Database* ini dapat digunakan untuk aplikasi berbasis *website*. Penulis menggunakan MySQL untuk sistem *database* aplikasi yang akan dibangun, pada saat melakukan proses pendataan persediaan barang secara *online*.

### G. Java

Java merupakan bahasa pemrograman dan *platform* komputasi yang dirilis pada tahun 1995 oleh Sun Microsystems [11]. Bahasa pemrograman ini berorientasi objek atau disebut juga dengan istilah OOP (*Object Oriented Programming*) [12]. Java bersifat umum (*general purpose*), konkuren, berbasis kelas dan berorientasi objek [13]. Java mirip dengan bahasa C dan C++ tetapi dipasang sedikit berbeda. Sejumlah aspek dari bahasa C dan C++ dihilangkan dan beberapa ide dari bahasa pemrograman lain dimasukkan. Hal itu dimaksudkan agar Java dapat menjadi bahasa produksi, bukan bahasa penelitian.

### H. SQLite Database

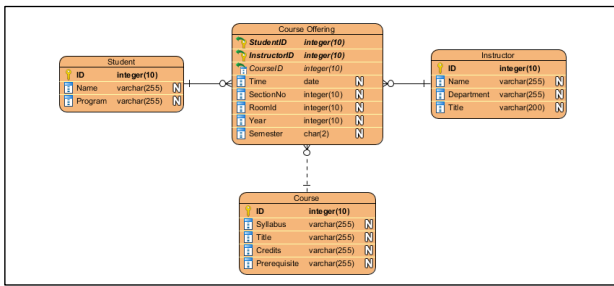
SQLite merupakan *library* bahasa pemrograman C yang mengimplementasikan mesin *database* SQL [14]. Sistem *database* ini mulai dikembangkan dari 9 Mei 2000. Kode yang digunakan untuk SQLite merupakan domain publik (*public domain*) dan bisa digunakan secara gratis untuk berbagai kepentingan, baik untuk komersial ataupun pribadi.

### I. Scrum

Scrum adalah *framework* yang memungkinkan orang yang menggunakannya untuk mengatasi masalah adaptif yang kompleks, selagi secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin [15]. Scrum dibangun dengan menggunakan teori empirisme. Scrum menggunakan strategi yang bertingkat dan kontinu guna memaksimalkan keterampilan dalam memprediksi dan mengendalikan risiko.

### J. ERD (*Entity Relationship Diagram*)

Diagram ini merupakan model yang digunakan untuk memperlihatkan relasi antara data yang berada di *database* dengan berdasar pada objek-objek dasar data yang memiliki hubungan antar relasi [16]. Tujuan utama dari ERD adalah menunjukkan objek data (*entity*) dan hubungan (*relationship*) yang ada pada *entity* berikutnya [17].



Gambar 2.2 Contoh Entity Relationship Diagram

K. UML (Unified Modeling Language)

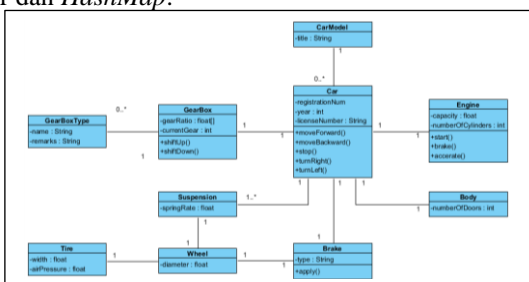
Unified Modeling Language merupakan bahasa yang berdasarkan grafik atau gambar yang digunakan untuk menggambarkan, mendokumentasikan, menjelaskan, dan membangun perangkat lunak yang berorientasi objek [18]. Tujuan dari UML adalah untuk menyediakan arsitektur sistem, software engineer, dan perangkat lunak pengembang dengan alat untuk analisis, desain, dan implementasi sistem berbasis perangkat lunak, serta untuk pemodelan bisnis (business modeling) dan proses serupa [19]. Terdapat beberapa jenis diagram dasar dalam UML yang digunakan untuk pemodelan berorientasi objek dan berbasis visual [20], yaitu Use Case Diagram, Class Diagram, Sequence Diagram, Statechart Diagram, dan Activity Diagram. Pada penelitian ini, Penulis menggunakan beberapa jenis UML pada pembangunan aplikasinya, yaitu sebagai berikut.

1. Use Case Diagram

Use Case diagram merupakan model fungsional sistem yang menggunakan aktor dan use case [20]. Setiap Use Case adalah suatu urutan (sequence) transaksi yang saling berhubungan dan dilakukan oleh aktor dan sistem dalam bentuk sebuah dialog. Diagram ini digunakan untuk membantu tim pengembangan menggambarkan kebutuhan fungsional sistem, termasuk hubungan antara aktor dengan proses penting, serta hubungan dengan berbagai use case [21]. Use case ditunjukkan dengan cara menggambar bentuk lonjong (oval) dengan disertai nama use case di dalamnya pada bagian tengah diagram. Aktor pada use case digambarkan dalam wujud stickman yang diletakkan pada bagian kanan atau kiri diagram. Stickman adalah gambar manusia yang berbentuk kepala, tangan, badan, dan kaki, dimana kepala diwakili oleh lingkaran, dan sisanya hanya dengan garis. Aktor dan use case biasanya dihubungkan dengan garis lurus.

2. Class Diagram

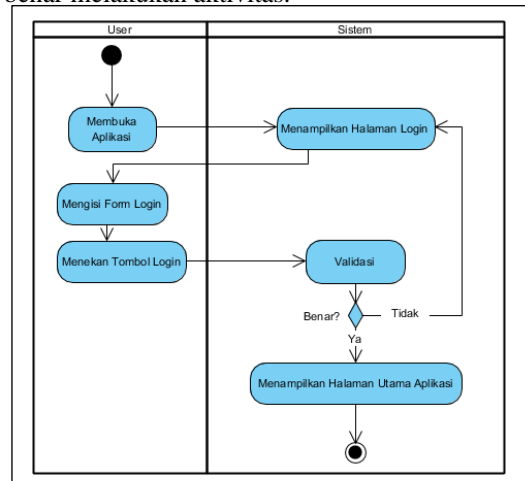
Class diagram menunjukkan bagaimana entitas yang berbeda (orang, benda, dan data) saling berhubungan [21]. Diagram ini menunjukkan struktur statis sistem. Class diagram dapat digunakan untuk menunjukkan implementasi kelas-kelas yang merupakan hal yang biasanya ditangani oleh programmer. Diagram yang digambarkan dengan menggunakan class diagram tidak akan memiliki atribut yang sama karena kemungkinan besar akan memiliki referensi ke hal-hal seperti Vektor dan HashMap.



Gambar 2.3 Contoh Class Diagram pada Sebuah Mobil

3. Activity Diagram

Activity diagram merupakan diagram yang menunjukkan aliran kontrol prosedural antar dua atau lebih objek kelas saat memroses suatu aktivitas [21]. Activity diagram dapat digunakan untuk memodelkan proses bisnis. Activity diagram dimulai dengan initial starting point yang terhubung ke aktivitas awal. Aktivitas dimodelkan dengan menggambarkan persegi panjang dengan sudut bulat dan melampirkan nama aktivitas di dalamnya. Aktivitas dapat dihubungkan ke aktivitas lain dengan menggunakan garis transisi, atau ke titik keputusan (decision point) yang terhubung ke berbagai akhivitas yang dijaga oleh decision point. Aktivitas yang menghentikan proses pemodelan terhubung ke termination point (titik terminasi). Secara opsional, aktivitas dapat dikelompokkan ke dalam swimlanes, yang digunakan untuk menunjukkan objek yang benar-benar melakukan aktivitas.



Gambar 2.4 Contoh Activity Diagram

L. JDBC (Java Database Connectivity Driver)

Java Database Connectivity (JDBC) driver merupakan komponen perangkat lunak yang digunakan sebagai penghubung antara aplikasi Java dengan sistem database [22].

M. ZXing

ZXing (dibaca zebra crossing) merupakan library pemrosesan gambar barcode yang diimplementasikan di Java, dengan port ke bahasa lain [23]. Library ini memiliki dukungan untuk produk 1D (satu dimensi), industrial 1D, dan barcode 2D (dua dimensi). ZXing digunakan oleh pencarian web (web search) untuk membuat barcode pada web indexable. ZXing juga membentuk dasar dari aplikasi barcode scanner dan terintegrasi ke dalam Google Product dan Book Search. Zxing dapat melakukan proses encode maupun decode barcode. Library zxing ini berfokus kepada penggunaan kamera pada smartphone sebagai pengganti dari alat scanner, namun library ini juga dapat dipakai pada desktop dan server.

N. JSON

JSON (Java Script Object Notation) merupakan format pertukaran data yang ringan dan mudah untuk dibaca dan ditulis oleh manusia [24]. Format ini juga mudah untuk diuraikan dan dihasilkan oleh mesin. JSON berdasar pada subset dari Standar Bahasa Pemrograman JavaScript ECMA-262 Edisi 3 – Desember 1999. JSON adalah format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familier bagi programmer beberapa bahasa pemrograman, seperti bahasa C, C++, C#, Java, JavaScript, Perl, Python, dan beberapa bahasa lainnya. Hal itu yang menjadikan JSON bahasa pertukaran data yang ideal.

O. HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa pemrograman yang digunakan sebagai pondasi dasar untuk membuat halaman *website* [25]. Dokumen HTML disimpan dengan ekstensi “.html”, dan dapat diakses menggunakan *web browser*.

P. CSS

CSS (*Cascading Style Sheets*) merupakan bahasa pemrograman yang digunakan untuk mendesain halaman *website* [25]. Bahasa pemrograman ini menggunakan penanda seperti id dan class untuk mengenali objek yang akan didesain. CSS dapat melakukan pengaturan *font*, mengatur *layout*, membuat halaman *website* yang *responsive*, dan melakukan beberapa kegiatan mendesain lainnya. Dokumen CSS disimpan dengan ekstensi “.css” dan dapat dihubungkan ke dalam file HTML atau PHP yang ingin didesain dengan dokumen tersebut dengan menggunakan tag <link>.

Q. PHP

PHP (singkatan rekursif untuk PHP: Hypertext Preprocessor) merupakan bahasa *scripting* yang memiliki tujuan umum dan bersifat *open source* serta banyak digunakan dan sangat cocok untuk pengembangan *web* dan dapat disematkan ke dalam HTML [26]. Kode PHP terlampir dalam instruksi pemrosesan awal dan akhir khusus, yaitu “<? Php” dan “?>”. Instruksi tersebut yang memungkinkan *programmer* untuk masuk dan keluar dari “mode PHP”. Yang membedakan PHP dari sesuatu seperti *client-side* JavaScript adalah pengekseskuan kode di server, lalu menghasilkan HTML yang kemudian dikirim ke klien. Klien akan menerima hasil dari menjalankan skrip tersebut, tetapi tidak akan tahu apa kode yang mendasarinya. PHP juga dapat digunakan untuk mengkonfigurasi *web server* untuk memproses semua file HTML, dan kemudian tidak ada cara bagi pengguna dari *web server* tersebut untuk mengetahui kode apa yang ada di dalamnya.

R. JavaScript

JavaScript merupakan bahasa pemrograman yang berguna dalam menciptakan interaksi atau menambahkan fitur situs dinamis ke dalam *website* [27]. Bahasa pemrograman ini pemrosesannya dilakukan oleh *client* (disebut juga *Client Side Programming Language*). *Client* yang dimaksud adalah *web browser*.

S. Black Box Testing

Pada penelitian ini, Penulis akan memakai *Black Box Testing* sebagai metode pengujian perangkat lunak. *Black Box Testing* atau *behavioral testing* (pengujian perilaku) merupakan pengujian yang berpusat pada syarat fungsional *software* [28]. Pengujian ini dilakukan untuk mencari kesalahan dalam beberapa bagian seperti: kesalahan tampilan, fungsi yang salah, kesalahan pada struktur data atau akses *database* eksternal, kesalahan perilaku atau kinerja, dan kesalahan inialisasi dan terminasi.

Tabel 2.1 Contoh Tabel *Black Box Testing*

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Mengosongkan kolom <i>Username</i> dan <i>Password</i> , lalu langsung klik tombol Login	Sistem akan menolak akses <i>Login</i> dan menampilkan pemberitahuan untuk mengisi <i>Username</i>	(Berhasil/Gagal)*

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
2	Mengisi kolom <i>Username</i> dan mengosongkan kolom <i>Password</i> , lalu menekan tombol Login	Sistem akan menolak akses <i>Login</i> dan menampilkan pemberitahuan untuk mengisi kolom <i>Password</i>	(Berhasil/Gagal)*
3	Mengosongkan kolom <i>Username</i> dan mengisi kolom <i>Password</i> , lalu menekan tombol Login	Sistem akan menolak akses <i>Login</i> dan menampilkan pemberitahuan untuk mengisi kolom <i>Username</i>	
4	Mengisi kolom <i>Username</i> dengan <i>username</i> yang tidak terdaftar dan mengisi kolom <i>Password</i> , lalu menekan tombol Login	Sistem akan menolak akses <i>Login</i> dan menampilkan pemberitahuan <i>Username</i> atau <i>Password</i> salah	

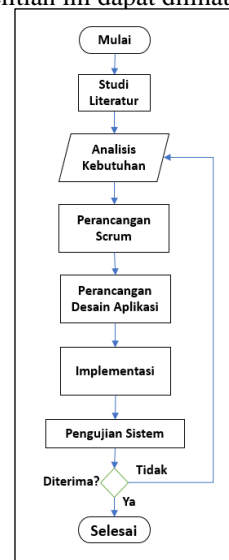
Keterangan:

Kesimpulan akan dinyatakan Berhasil saat pengujian menunjukkan hasil yang diharapkan, atau dinyatakan Gagal saat pengujian menunjukkan hasil yang tidak diharapkan.

III. IMPLEMENTASI PROGRAM

A. Tahapan Penelitian

Tahapan pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 3.1 Diagram Alur Penelitian

B. Existing System (Sistem yang Sudah Ada)

Sistem yang digunakan untuk melakukan proses *stock opname* pada perusahaan terkait sudah cukup bagus. Proses tersebut dilaksanakan dengan menggunakan bantuan aplikasi berbasis *website* untuk melakukan proses pendataan persediaan barang di lapangan (gudang) dan aplikasi *desktop* untuk memproses hasil dari pendataan tersebut ke sistem aplikasi sebelum diposting. Sistem aplikasi yang dipakai adalah Oracle E-Business Suite R.12 (EBS R.12). Proses posting merupakan akhir rangkaian dari siklus *stock opname* di sana. Perangkat yang digunakan selama proses pendataan persediaan barang di gudang adalah tablet yang memiliki sistem operasi Windows. Sistem operasi tersebut diperlukan karena memerlukan aplikasi Xampp untuk menjalankan *website* serta menyambungkannya ke *database*.

Terdapat beberapa langkah pada tahap pendataan persediaan barang di Gudang perusahaan terkait, yaitu sebagai berikut:

1. memilih gudang (*warehouse*) tempat inventarisasi barang dilakukan,
2. isi nomor rak atau lokasi barang,
3. mengisi nomor atau kode barang dengan nomor Lot, lalu
4. mengisi jumlah persediaan barang yang ada pada rak tersebut.

Ada tiga cara untuk mencatat nomor lokasi dan kode barang pada saat proses inventarisasi, yaitu mengetik seluruh kode secara manual, memilih tombol yang berbentuk penggalan-penggalan dari kode barang, dan menggunakan *barcode reader external* yang disambungkan ke tablet melalui kabel usb (*universal serial bus*). Setelah proses pendataan persediaan barang selesai, hasil dari proses tersebut akan dikonversi ke dalam format file *excel* dengan menggunakan aplikasi *desktop*. Laporan hasil dari beberapa orang yang bertugas untuk melakukan pendataan persediaan barang akan diproses dan digabungkan ke dalam satu *server*. Setelah itu, laporan-laporan tersebut akan dibandingkan dengan catatan pembukuan. Proses tersebut dilakukan dengan bantuan aplikasi Oracle.

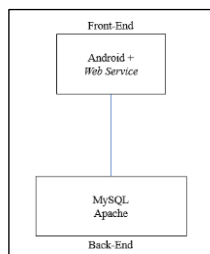
### C. Analisis Kebutuhan (*Requirement*)

Analisis kebutuhan (*requirement*) pada aplikasi yang akan dibangun dilakukan dengan berdiskusi dan mewawancarai salah satu pihak perusahaan terkait. Berdasarkan hasil wawancara dan diskusi bersama narasumber, ditetapkan beberapa ide untuk membangun aplikasi, yaitu sebagai berikut:

1. Aplikasi yang dibangun berbasis Android dan dapat digunakan secara *online* maupun *offline* pada saat proses pendataan persediaan barang, tergantung pada kondisi di gudang
2. Aplikasi digunakan untuk proses pendataan persediaan barang di gudang berdasarkan tahapan yang sudah diberitahukan sebelumnya, serta mengirimkan hasilnya ke *server* tempat penyimpanan laporan hasil pendataan persediaan barang
3. Aplikasi harus mampu mengambil beberapa data yang diperlukan untuk proses pendataan persediaan barang dari sistem *database* yang sudah ada di perusahaan
4. Aplikasi memiliki dua level user pada mode *online* dan satu level user pada mode *offline*.

### D. Struktur Aplikasi

Gambar berikut adalah struktur dari aplikasi yang akan dibangun pada penelitian ini.



Gambar 3.2 Struktur dari Aplikasi

Aplikasi ini memiliki bagian *Front-End*, yaitu Android yang *support web service*, sehingga aplikasi dari beberapa perangkat yang berbeda dapat berkomunikasi dan berbagi data dan layanan kapanpun dan dimanapun selama perangkat memiliki koneksi internet yang memadai. Pada bagian *Back-End*, MySQL berfungsi sebagai *database* dan Apache sebagai *web server* yang menghubungkan aplikasi dengan sistem *database*.

### E. Perancangan Scrum

*Product backlog* dari pembangunan aplikasi ditunjukkan oleh Tabel 3.1. Perkiraan dari penyelesaian proyek pembangunan aplikasi ini adalah 32 hari, akan tetapi proyek terselesaikan selama 45 hari. *Priority* (prioritas) menunjukkan pekerjaan yang lebih penting untuk diselesaikan terlebih dahulu. Kegiatan yang memiliki angka prioritas yang lebih kecil akan dikerjakan terlebih dahulu.

Tabel 3.1 *Product Backlog* dari Pembangunan Aplikasi

ID	Description (Deskripsi Backlog)	Initial Estimate (in Day)	Adjustment Estimate	Priority
1	Perancangan Desain Aplikasi	1	3	1
<b>Sprint-1</b>		1	3	
9	Perancangan Ulang Desain Aplikasi	1	2	2
10	Penambahan Proses Registrasi, Login, dan Logout pada Mode <i>Online</i>	1	1	3
11	Pembuatan Halaman Utama pada mode <i>Online</i>	1		3
12	Penambahan Proses Pengunduhan Data ke MySQL	3	4	4
13	Penambahan Proses Pendataan Barang Secara <i>Online</i>	3	5	5
14	Penambahan Fitur untuk Melihat Hasil Pendataan Secara <i>Online</i>	1	2	6
15	Penambahan Proses Upload Data ke <i>Server</i> pada Mode <i>Online</i>	3	3	7
16	Penambahan Fitur <i>Settings</i> pada Mode <i>Online</i>	1	2	8
17	Penambahan Role Admin pada Aplikasi	1	2	9
18	Pengubahan Beberapa Bagian Halaman pada mode <i>Online</i> dan Pembuatan Halaman Admin	2	1	10
19	Penambahan Fitur Forgot Password pada Mode <i>Online</i>	1	2	11
<b>Sprint-2</b>		18	24	
2	Penambahan Proses Registrasi, Login, dan Logout pada Mode <i>Offline</i>	1	2	12
3	Pembuatan Halaman Utama pada mode <i>Offline</i>	1		12
4	Penambahan Proses Pengunduhan Data ke Perangkat	3	4	13
5	Penambahan Proses Pendataan Barang Secara <i>Offline</i>	3	4	14
6	Penambahan Fitur untuk Melihat Hasil Pendataan Secara <i>Offline</i>	1	2	15
7	Penambahan Proses Upload Data ke <i>Server</i> pada Mode <i>Offline</i>	2	3	16
8	Penambahan Fitur <i>Settings</i> pada Mode <i>Offline</i>	1	1	17
9	Proses <i>Finishing</i> (Penyelesaian) Aplikasi	1	2	18
<b>Sprint-3</b>		13	18	
<b>Total</b>		32	45	

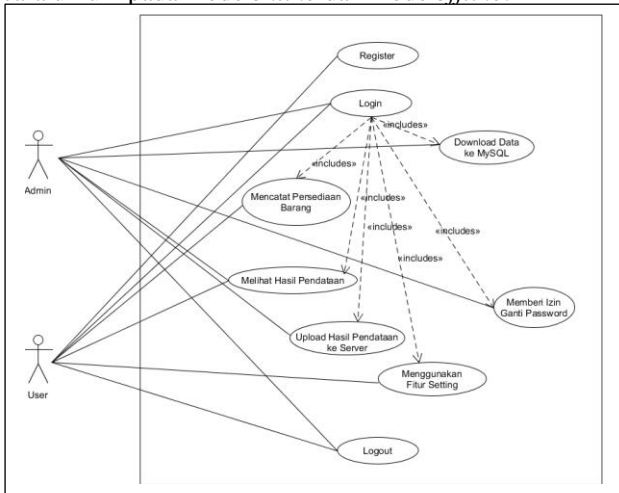
### F. Perancangan Desain Aplikasi

Pada tahap ini, dilakukan perencanaan dan pemecahan masalah yang didapat dari tahap sebelumnya untuk solusi perangkat lunak. Solusi tersebut akan disajikan dalam bentuk visual berupa *Use Case Diagram*, *Entity Relationship Diagram*, *Class Diagram*, *Activity Diagram*, dan desain *User Interface*.

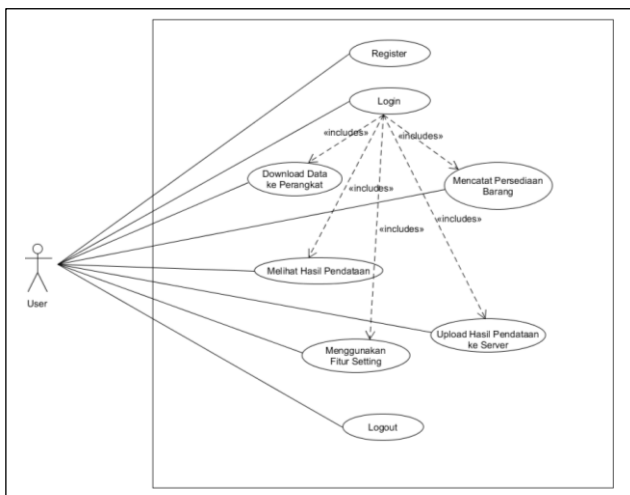


### 1. Use Case Diagram

Gambar 3.3 dan Gambar 3.4 menunjukkan *use case diagram* yang menggambarkan aktor dalam aplikasi yang akan dibangun beserta *use case* atau aksi yang dapat dilakukan oleh aktor secara umum pada mode *online* dan mode *offline*.



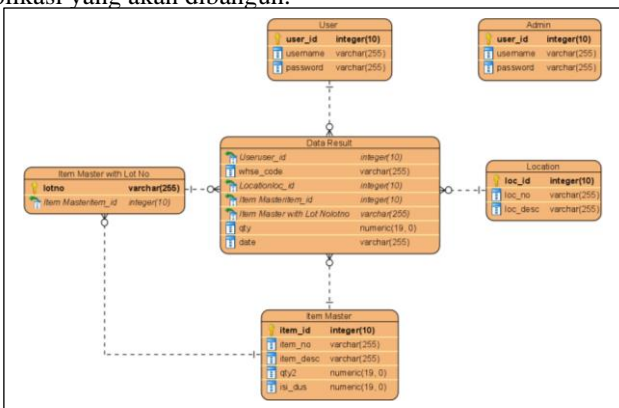
Gambar 3.3 Use Case Diagram Aplikasi pada Mode Online



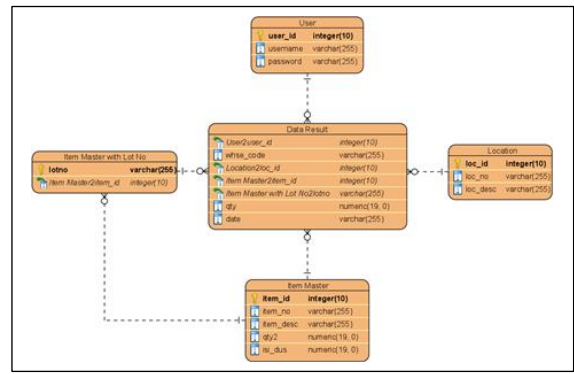
Gambar 3.4 Use Case Diagram Aplikasi pada Mode Offline

### 2. Entity Relationship Diagram (ERD)

ERD pada mode *online* berfungsi untuk menyimpan data User dan Admin, nomor rak atau lokasi barang, kode barang, nomor lot barang dan hasil dari pendataan persediaan barang di gudang secara *online*. Sedangkan pada mode *offline* tidak ada data Admin karena mode ini tidak memiliki *role* Admin. Berikut ini adalah gambaran dari *Entity Relationship Diagram* aplikasi yang akan dibangun.



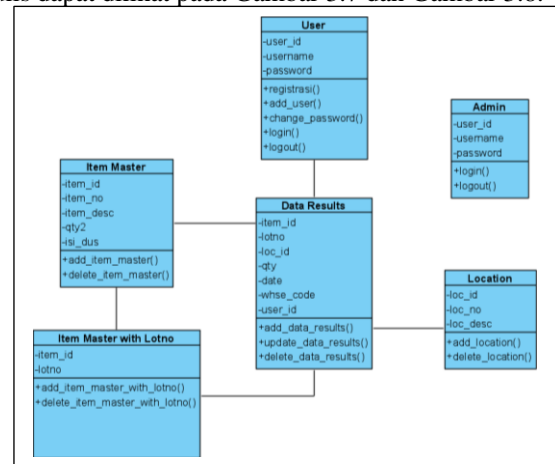
Gambar 3.5 ERD Aplikasi pada Mode Online



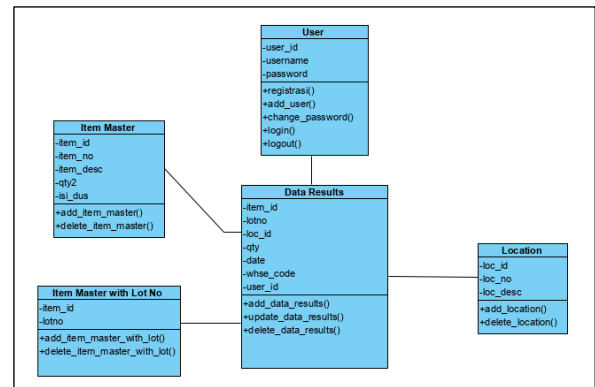
Gambar 3.6 ERD Aplikasi pada Mode Offline

### 3. Class Diagram

Desain *Class Diagram* pada aplikasi yang akan dibuat oleh Penulis dapat dilihat pada Gambar 3.7 dan Gambar 3.8.



Gambar 3.7 Class Diagram Aplikasi pada Mode Online



Gambar 3.8 Class Diagram Aplikasi pada mode Offline

### 4. Activity Diagram

Terdapat beberapa *Activity Diagram* pada aplikasi yang akan dibangun oleh Penulis, yaitu *Login*, *Registrasi*, *Mengunduh (Download) Data ke database MySQL*, *Pendataan Persediaan Barang Secara Online*, *Mengunduh (Download) Data ke perangkat*, *Pendataan Persediaan Barang Secara Offline*, dan *Mengunggah (Upload) Data*.

### 5. Desain Mock-Up atau User Interface

Terdapat beberapa desain UI yang dibuat untuk aplikasi yang akan dibangun, yaitu *Halaman Awal*, *Halaman Register* untuk tiap mode, *Halaman Login* untuk tiap mode, *Halaman Download Data ke MySQL*, *Halaman Download Data ke Perangkat*, *Halaman Count Qty* untuk tiap mode, *halaman Data Results* untuk tiap mode, *Halaman untuk proses Upload Data* pada tiap mode, dan *Halaman Settings* untuk tiap mode.

## G. Implementasi

Pada tahap ini, dilakukan pembangunan aplikasi dengan mengimplementasikan semua desain yang sudah dirancang pada tahap sebelumnya. Semua tahap implementasi akan dibahas sesuai dengan urutan pada *Product Backlog* yang sudah dirancang.

## H. Pengujian

### 1. Pengujian *Scanner* pada Aplikasi

Pengujian ini dilakukan untuk mengetahui kualitas kerja dari *scanner* pada aplikasi. Terdapat dua bagian pengujian yang dilakukan pada pengujian ini. Pengujian yang pertama adalah pengujian *scanner* dengan menggunakan kode yang tertera pada *barcode* sebagai pembanding. Pengujian ini dilakukan untuk melihat kinerja *scanner* dalam membaca *barcode* dan melakukan validasi data pada *barcode* terhadap data di *database*. Pengujian dilakukan dengan menggunakan metode pengujian *Black Box Testing*. *Black Box Testing* atau *behavioral testing* adalah pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Persentase keberhasilan fungsional *scanner* akan dihitung dengan membagi jumlah skenario pengujian yang berhasil (B) dengan jumlah skenario pengujian. Jika ada N jumlah skenario pengujian, maka Persentase Fungsional *Scanner* (PFS) dapat dihitung sebagai berikut.

$$PFS = \frac{B}{N} * 100$$

Pengujian yang kedua adalah pengujian jarak, ukuran, dan tingkat pencahayaan. Pengujian ini dilakukan untuk melihat apakah *scanner* dapat membaca *barcode* dengan ukuran terkecil dan terbesar pada barang atau rak perusahaan terkait, dengan jarak yang telah ditentukan, dan tingkat pencahayaan tertentu. Hasil dari pengujian akan disajikan dalam bentuk tabel. Penarikan kesimpulan dilakukan setelah mendapatkan hasil pengujian dengan berdasarkan pada hasil pengujian tersebut.

### 2. Pengujian Proses Count Qty

Pengujian proses Count Qty dilaksanakan dengan melakukan pendataan pada sepuluh *barcode* kode barang dengan ukuran tertentu. Pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan untuk menyelesaikan proses tersebut jika dilakukan dengan menggunakan *scanner*. Pengujian pada tiap *barcode* dilakukan sebanyak tiga kali untuk memastikan jumlah waktu yang diperlukan. Hasil dari pengujian akan disajikan dalam bentuk tabel. Setelah mendapatkan hasil pengujian, dilakukan penarikan kesimpulan dengan berdasarkan pada hasil pengujian tersebut.

### 3. Pengujian Keseluruhan Aplikasi

Pengujian ini bertujuan untuk menguji fungsi aplikasi secara keseluruhan dalam melakukan pendataan persediaan barang untuk proses *stock opname* perusahaan terkait. Pengujian dilakukan kepada setiap proses dan fitur halaman yang ada pada aplikasi dengan menggunakan metode pengujian *Black Box Testing*. Hasil pengujian ini juga akan dihitung persentase keberhasilannya. Persentase keberhasilan fungsional keseluruhan aplikasi akan dihitung dengan membagi jumlah skenario pengujian yang berhasil (B) dengan jumlah skenario pengujian. Jika ada N jumlah skenario pengujian, maka Persentase Fungsional Aplikasi (PFA) dapat dihitung sebagai berikut.

$$PFA = \frac{B}{N} * 100$$

## IV. PENGUJIAN DAN HASIL

### A. Pengujian *Scanner* pada Aplikasi

#### 1. Pengujian Fungsional *Scanner* pada Aplikasi

Pengujian Fungsional *scanner* dilakukan untuk mencari tahu kinerja *scanner* dalam melakukan pemindaian *barcode* dan melakukan validasi data pada *barcode* terhadap data di *database*. Tabel 4.1 menunjukkan daftar skenario pengujian beserta hasil pengujian pada pengujian fungsional *scanner* bagian pertama. Pengujian ini dilakukan terhadap *scanner* pada mode *online* aplikasi dan dengan kondisi tidak ada koneksi jaringan (*network*) pada perangkat. Koneksi jaringan (*network*) yang dimaksud adalah sambungan jaringan *database* MySQL terkait.

Tabel 4.1 Pengujian Fungsional *Scanner* pada Aplikasi (Bagian Pertama)

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Scan barcode</i> kode lokasi yang tidak terdaftar di <i>database</i>	Kode lokasi pada <i>barcode</i> terbaca tetapi sistem tidak melakukan validasi	Berhasil
2	<i>Scan barcode</i> kode lokasi yang terdaftar di <i>database</i>	Kode lokasi pada <i>barcode</i> terbaca tetapi sistem tidak melakukan validasi	Berhasil
3	<i>Scan barcode</i> kode barang yang tidak terdaftar di <i>database</i>	Kode barang pada <i>barcode</i> terbaca tetapi sistem tidak melakukan validasi	Berhasil
4	<i>Scan barcode</i> kode barang yang terdaftar di <i>database</i>	Kode barang pada <i>barcode</i> terbaca tetapi sistem tidak melakukan validasi	Berhasil

Daftar skenario pengujian beserta hasil pengujian pada pengujian fungsional *scanner* bagian kedua ditunjukkan oleh Tabel 4.2. Pengujian ini dilakukan terhadap *scanner* pada mode *online* dan dengan kondisi ada koneksi jaringan (*network*) pada perangkat. Koneksi jaringan (*network*) yang dimaksud adalah sambungan jaringan *database* MySQL terkait.

Tabel 4.2 Pengujian Fungsional *Scanner* pada Aplikasi (Bagian Kedua)

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<i>Scan barcode</i> kode lokasi yang tidak terdaftar di <i>database</i>	Sistem akan menampilkan pemberitahuan bahwa kode lokasi hasil <i>scan</i> tidak ditemukan	Berhasil
2	<i>Scan barcode</i> kode lokasi yang terdaftar di <i>database</i>	Sistem akan menampilkan kode lokasi hasil <i>scan</i> di kolom lokasi	Berhasil
3	<i>Scan barcode</i> kode barang yang tidak terdaftar di <i>database</i>	Sistem akan menampilkan pemberitahuan bahwa kode hasil <i>scan</i> tidak ditemukan	Berhasil
4	<i>Scan barcode</i> kode barang yang terdaftar dengan nomor lot tidak terdaftar di <i>database</i>	Sistem akan menampilkan kode barang beserta nama barang pada kolom kode barang, dan menampilkan nomor lot pada kolom nomor lot dengan keterangan bahwa nomor tersebut tidak terdaftar	Berhasil
5	<i>Scan barcode</i> kode barang dengan nomor lot yang terdaftar di <i>database</i>	Sistem akan menampilkan kode barang beserta nama barang pada kolom kode barang, dan menampilkan nomor lot pada kolom nomor lot	Berhasil

Tabel 4.3 menunjukkan daftar skenario pengujian beserta hasil pengujian pada pengujian fungsional *scanner* yang dilakukan terhadap *scanner* pada mode *offline*. Pengujian ini dapat dilakukan dengan atau tanpa koneksi internet. Hal itu disebabkan karena data pada *database* disimpan di dalam memori perangkat.

Tabel 4.3 Pengujian Fungsional Scanner pada Aplikasi (Bagian Ketiga)

No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Scan barcode kode lokasi yang tidak terdaftar di database	Sistem akan menampilkan bahwa kode lokasi hasil scan tidak ditemukan	Berhasil
2	Scan barcode kode lokasi yang terdaftar di database	Sistem akan menampilkan kode lokasi hasil scan di kolom lokasi dilengkapi keterangan lokasi	Berhasil
3	Scan barcode kode barang yang tidak terdaftar di database	Sistem akan menampilkan bahwa kode barang hasil scan tidak ditemukan	Berhasil
4	Scan barcode kode barang yang terdaftar dengan nomor lot tidak terdaftar di database	Sistem akan menampilkan kode barang beserta nama barang pada kolom kode barang, dan menampilkan nomor lot pada kolom nomor lot dengan keterangan bahwa nomor tersebut tidak terdaftar	Berhasil
5	Scan barcode kode barang dengan nomor lot yang terdaftar di database	Sistem akan menampilkan kode barang beserta nama barang pada kolom kode barang, dan menampilkan nomor lot pada kolom nomor lot	Berhasil

Dari ketiga tabel di atas, dapat dilihat bahwa scanner berjalan dengan baik dalam memindai kode pada barcode dan melakukan validasi hanya pada saat tertentu. Fungsional scanner pada aplikasi bekerja sesuai dengan yang diharapkan baik pada mode *online*, maupun mode *offline*. Hasil pengujian menunjukkan bahwa PFS (Persentase Fungsional Scanner) adalah 100%.

## 2. Pengujian Jarak, Ukuran, dan Tingkat Pencahayaan

Pengujian ini dilakukan dengan menggunakan jarak perangkat dengan barcode, ukuran barcode, dan tingkat pencahayaan di ruangan saat melakukan pemindaian barcode. Pengujian ini dilakukan dengan melakukan pemindaian terhadap barcode dengan ukuran yang terkecil dan terbesar pada barang atau rak perusahaan terkait. Jarak yang digunakan untuk pengujian ini ada empat, yaitu jarak pertama (10 cm), jarak kedua (20 cm), jarak ketiga (30 cm) dan jarak keempat (40 cm). Tingkat pencahayaan yang digunakan dalam pengujian ini adalah pencahayaan ruangan tanpa penerangan pada siang hari dan ruangan dengan penerangan pada siang hari. Tabel 4.4 menunjukkan daftar pengujian beserta hasil dari pengujian jarak, ukuran, dan tingkat pencahayaan.

Tabel 4.4 Pengujian Scanner Bagian Kedua

No	Kondisi	Barcode 1,5x3 cm	Barcode 2x13 cm
1	Jarak pertama, tanpa penerangan	Terbaca	Tidak Terbaca
2	Jarak pertama, dengan penerangan	Terbaca	Tidak Terbaca
3	Jarak kedua, tanpa penerangan	Terbaca	Terbaca
4	Jarak kedua, dengan penerangan	Terbaca	Terbaca
5	Jarak ketiga, tanpa penerangan	Tidak Terbaca	Terbaca
6	Jarak ketiga, dengan penerangan	Tidak Terbaca	Terbaca
7	Jarak keempat, tanpa penerangan	Tidak Terbaca	Tidak Terbaca
8	Jarak keempat, dengan penerangan	Tidak Terbaca	Tidak Terbaca

Berdasarkan tabel di atas, dapat dilihat bahwa jarak antar scanner dan barcode perlu diperhatikan agar pemindaian dapat berhasil dilakukan. Barcode dengan ukuran berbeda memiliki jarak minimum dan maksimum berbeda pula. Jarak pemindaian juga memengaruhi kecepatan pemindaian.

Proses pemindaian akan lebih cepat jika jarak antara scanner dan barcode tidak terlalu jauh. Jika jaraknya terlalu jauh, maka kamera akan kesulitan untuk memfokuskan targetnya sehingga diperlukan waktu lebih lama dalam memindai barcode. Dapat dilihat juga bahwa scanner dapat melakukan pemindaian barcode walaupun kondisi pencahayaan dalam ruangan redup tanpa pencahayaan dari lampu jika ruangan masih cukup terang bagi kamera. Akan tetapi, perlu diperhatikan agar barcode tidak tertutup oleh bayangan pada saat melakukan scan. Hal itu dapat mengganggu proses pemindaian barcode. Setelah melakukan pengujian ini, dapat disimpulkan bahwa jarak memiliki pengaruh dalam melakukan scan barcode. User perlu memperhatikan jarak antara perangkat (scanner) dengan barcode agar dapat melakukan proses pemindaian dengan lebih cepat.

## B. Pengujian Proses Count Qty

Pengujian ini memiliki tujuan untuk mengetahui waktu yang dibutuhkan untuk menyelesaikan proses Count Qty jika dilakukan dengan menggunakan fitur barcode scanner. Tabel 4.5 menunjukkan hasil pengujian pada proses Count Qty. Pengujian ini dilakukan dengan melakukan pendataan pada sepuluh barcode kode barang dengan ukuran tertentu. Pengujian ini dilakukan sebanyak tiga kali untuk setiap ukuran barcode.

Dari hasil pengujian, dapat dilihat bahwa waktu yang diperlukan untuk memindai sepuluh barcode bervariasi dan tidak pasti. Saat melakukan pengujian, Penulis melihat bahwa barcode perlu terlihat jelas oleh kamera scanner dan mendapat pencahayaan yang merata. Proses pemindaian tidak akan berhasil jika sebagian permukaan barcode tertutup oleh bayangan. Posisi yang tepat untuk scanner dan barcode sangat perlu diperhatikan agar proses pemindaian dapat selesai lebih cepat. Waktu tercepat yang didapat saat melakukan pengujian ini adalah 1 menit 31 detik. Hal itu berarti diperlukan waktu paling cepat sekitar 9 detik untuk melakukan pendataan pada satu barcode kode barang. Akan tetapi, tidak menutup kemungkinan untuk melakukannya dengan lebih cepat. Hal itu tergantung pada kemampuan User dalam memindai dan menghitung jumlah barang.

Tabel 4.5 Pengujian Waktu untuk Proses Count Qty

No	Ukuran Barcode	Percobaan Pertama	Percobaan kedua	Percobaan Ketiga
1	10 px (1x7,5 cm)	3 menit 48 detik	1 menit 45 detik	1 menit 58 menit
2	12 px (1,3x8,6 cm)	3 menit 5 detik	1 menit 46 detik	1 menit 45 detik
3	14 px (1,5x10 cm)	2 menit 45 detik	1 menit 31 detik	1 menit 45 detik
4	18 px (2x13 cm)	1 menit 50 detik	1 menit 31 detik	1 menit 46 detik

## C. Pengujian Keseluruhan Aplikasi

Pengujian keseluruhan aplikasi dilakukan untuk menguji fungsi aplikasi secara keseluruhan dalam melakukan pendataan persediaan barang untuk proses stock opname perusahaan terkait. Pengujian dilakukan kepada setiap proses dan fitur halaman yang ada pada aplikasi baik pada mode *online* maupun mode *offline*. Pengujian ini dilakukan dengan menggunakan metode *Black Box Testing*. Metode ini memberikan hasil penilaian terhadap kesesuaian fungsionalitas sistem pada aplikasi.



Tabel 4.6 Hasil Pengujian Keseluruhan Aplikasi

Pengujian	Jumlah	Jumlah	Jumlah
	Skenario	Berhasil	Gagal
Pengujian Proses dan Fitur pada Mode <i>Online</i>	134	131	3
Pengujian Proses dan Fitur pada Mode <i>Offline</i>	71	71	0
Total	205	202	3

Tabel 4.6 menunjukkan hasil pengujian keseluruhan aplikasi. Dari total delapan belas pengujian yang dilakukan pada keseluruhan aplikasi, terdapat 134 skenario pengujian pada proses dan fitur di mode *online* dan tujuh puluh satu skenario pengujian pada proses dan fitur di mode *offline*. Dari total 205 skenario pengujian, 202 skenario pengujian menunjukkan hasil Berhasil dan tiga skenario pengujian menunjukkan hasil Gagal. Persentase keberhasilan fungsional keseluruhan aplikasi adalah 98,54%. Skenario pengujian yang menunjukkan Gagal adalah pengujian fungsional sistem untuk menampilkan pemberitahuan saat tidak ada koneksi jaringan di dalam proses Download Data ke MySQL. Kegagalan fungsional sistem ini tidak terlalu mempengaruhi *business process* pada aplikasi. Jadi, dapat disimpulkan bahwa aplikasi Pendataan Persediaan Barang untuk proses Stock Opname pada perusahaan terkait berhasil dibuat dengan sangat baik.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, didapat beberapa kesimpulan sebagai berikut.

1. Pembuatan aplikasi pendataan persediaan barang dengan menggunakan *barcode* berbasis Android pada perusahaan terkait dimulai dengan studi literatur dan analisis kebutuhan. Kemudian dilanjutkan dengan perancangan Scrum dan desain aplikasi. Lalu, dilakukan implementasi terhadap hasil rancangan dari tahap sebelumnya. Setelah itu, pengujian dilakukan untuk melihat fungsional aplikasi yang telah dibuat.
2. Hasil pengujian menunjukkan bahwa *scanner* pada aplikasi dapat bekerja sesuai dengan harapan. Persentase keberhasilan fungsional *scanner* pada aplikasi pada pengujian yang dilakukan adalah sebesar 100%. Waktu yang diperlukan untuk melakukan pendataan persediaan barang tidak pasti tetapi kegiatan ini dapat diselesaikan dengan cepat jika User dapat memindai *barcode* dan menghitung jumlah barang secepat mungkin. Hasil pengujian juga menunjukkan tingkat keberhasilan fungsional keseluruhan aplikasi sebesar 98,54%. Hal ini menunjukkan bahwa aplikasi berhasil dibuat dengan sangat baik sehingga aplikasi ini dapat digunakan untuk menggantikan aplikasi yang sebelumnya digunakan pada perusahaan terkait.

### B. Kesimpulan

Terdapat beberapa saran untuk penelitian selanjutnya, yaitu sebagai berikut.

1. Memperbaiki kegagalan fungsional aplikasi di bagian proses Download Data ke MySQL dalam menampilkan pemberitahuan saat perangkat tidak terhubung dengan *database* MySQL.

2. Memperbaiki masalah *progress dialog* yang kurang sinkron pada saat sistem mengunduh data yang terlalu banyak pada proses Download Data.
3. Melakukan pengujian pengaruh penggunaan aplikasi terhadap kecepatan proses *stock opname* di perusahaan terkait.
4. Menambahkan fitur untuk memilih data yang ingin diunduh pada proses Download Data.
5. Mengembangkan aplikasi agar dapat mencocokkan letak atau lokasi persediaan barang berdasarkan *database*.
6. Mengembangkan aplikasi agar dapat melakukan perbandingan jumlah persediaan barang yang telah dicatat pada proses *stock opname* dengan data di *database*.
7. Membangun aplikasi dengan berbasis sistem operasi lain seperti iOS atau Windows.
8. Menggunakan QR Code untuk mengidentifikasi barang dengan menggunakan aplikasi yang telah dibangun karena QR Code dapat membantu menyimpan lebih banyak informasi dan aplikasi yang telah dibangun sudah dapat digunakan untuk memindai QR Code.
9. Memperbagus jaringan pada perusahaan terkait agar dapat lebih sering melakukan proses Count Qty secara *online* agar lebih menyederhanakan proses tersebut.

## DAFTAR PUSTAKA

- [1] C. F. Guswai, My Retail Formula, Jakarta: Gramedia Pustaka Utama, 2013.
- [2] F. N. M. Roslan, M. A. Othman dan M. A. A. Razak, "UPC Barcode Apps For Drug Registered Verification," *Jurnal Teknologi*, vol. 81, no. 6, pp. 31-37, 2015.
- [3] I. P. A. P. Yudha, M. Sudarma dan P. A. Mertasana, "Perancangan Aplikasi Sistem Inventory Barang Menggunakan Barcode Scanner Berbasis Android," *E-Journal SPEKTRUM*, vol. 4, no. 2, pp. 72-79, 2017.
- [4] S. B. Santosa, "Aplikasi Pengelolaan Stok Barang Menggunakan Zxing QR-Code pada Android Studi Kasus Apotek Setya Budi," pp. 1-87, 2018.
- [5] T. Lotlikar, R. Kankapurkar, A. Parekar dan A. Mohite, "Comparative study of Barcode, QR-code and RFID System," *Int.J.Computer Technology & Applications*, vol. 4, no. 5, p. 818, 2013.
- [6] P. Trujillo, "Wasp Buzz," 25 Maret 2014. [Online]. Available: <http://www.wasppbarcode.com/buzz/how-barcode-scanners-work/>. [Diakses 11 November 2019].
- [7] A. K. Saha, "A Developer's First Look At Android," dalam *Linux For You*, Open Source For You, 2018, p. 48.
- [8] E. Maiyana, "Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa," *Jurnal Sains dan Informatika*, vol. 4, no. 1, p. 57, 2018.
- [9] G. Developers, "Android Developers," [Online]. Available: <https://developer.android.com/studio/intro?hl=ID>. [Diakses 13 September 2019].
- [10] Anonim, "MySQL," Oracle, [Online]. Available: <https://www.mysql.com>. [Diakses 18 Februari 2020].
- [11] Java, "Java," [Online]. Available: [https://www.java.com/en/download/faq/whatis\\_java.xml](https://www.java.com/en/download/faq/whatis_java.xml). [Diakses 27 September 2019].
- [12] J. Enterprise, Mengenal Java dan Database, Jakarta: PT Elex Media Komputindo, 2015.
- [13] J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley dan D. Smith, "Oracle," 21 August 2019. [Online]. Available: <https://docs.oracle.com/javase/specs/jls/se13/jls13.pdf>. [Diakses 27 September 2019].

- [14] SQLite, "SQLite," [Online]. Available: <https://sqlite.org/index.html>. [Diakses 13 September 2019].
- [15] K. Schwaber dan J. Sutherland, "Scrum Guides," November 2017. [Online]. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Indonesian.pdf>. [Diakses 15 Februari 2020].
- [16] N. L. M. Sugiarni, D. Pramana dan N. N. H. Puspita, "Implementasi CRM (Customer Relationship Management) pada Sistem Informasi Travel X Berbasis Web," *Jurnal Sistem dan Informatika*, vol. 9, no. 2, p. 53, 2015.
- [17] E. W. Fridayanthie dan T. Mahdiati, "Rancang Bangun Sistem Informasi Permintaan ATK Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkas Bitung)," *Jurnal Khatulistiwa Informatika*, vol. 4, no. 2, p. 132, 2016.
- [18] Suendri, "Implementasi Diagram UML (Unified Modeling Language) pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 01, pp. 2-3, 2018.
- [19] O. M. Group, "OMG Unified Modeling Language Version 2.5.1," *OMG Unified Modeling Language Publication*, p. 1, December 2017.
- [20] Henderi, "Object Oriented Modeling with Unified Modeling Language (UML)," *Diktat Perancangan Sistem Informasi (SI-111/3 SKS)*, p. 6, June 2016.
- [21] D. Bell, "The Rational Edge," 2003. [Online]. Available: [https://developer.ibm.com/series/uml-basics/?mhsrc=ibmsearch\\_a&mhq=uml](https://developer.ibm.com/series/uml-basics/?mhsrc=ibmsearch_a&mhq=uml). [Diakses 17 September 2019].
- [22] D. C. Berg, K. S. Bhogal dan C. D. J. B. K. M. Nitin Gaur, "Self-Learning Java Database Connectivity (JDBC) Driver". United States of America Paten US 2015/0019584 A1, 15 January 2015.
- [23] G. O. Source, "Google Open Source," [Online]. Available: <https://opensource.google/projects/zxing>. [Diakses 8 October 2019].
- [24] D. Crockford, "JSON," [Online]. Available: <http://www.json.org/>. [Diakses 8 October 2019].
- [25] D. A. Hadi, "Ebook Belajar HTML dan CSS Dasar," 20 Oktober 2016. [Online]. Available: [www.malasngoding.com](http://www.malasngoding.com). [Diakses 29 September 2020].
- [26] T. P. Group, "PHP," 2001. [Online]. Available: <https://www.php.net/manual/en/intro-what-is.php>. [Diakses 9 October 2019].
- [27] Bernardhed dan M. Sulistiyono, *Modul Praktikum Web Programming*, Yogyakarta, 2014.
- [28] R. S. Pressman, *Software Engineering Practioner's Approach*, New York: McGraw-Hill, 2010.
- [29] I. K. Raharjana, *Pengembangan Sistem Informasi Menggunakan Metodologi Agile*, Yogyakarta: Penerbit Deepublish, 2017.
- [30] S. M. Ni Ketut Dewi Ari Jayanti dan S. M. Ni Kadek Sumari, *Teori Basis Data*, Yogyakarta: Penerbit ANDI, 2018.
- [31] K. E. Kendall dan J. E. Kendall, *Systems Analysis And Design*, New Jersey: Prentice Hall, 2011.
- [32] C. Kobryn, "Object Management Group," 2001. [Online]. Available: [https://www.omg.org/news/meetings/workshops/presentations/embedded-rt2002/02-1\\_Kobryn\\_UML\\_Tutorial.pdf](https://www.omg.org/news/meetings/workshops/presentations/embedded-rt2002/02-1_Kobryn_UML_Tutorial.pdf). [Diakses 18 September 2019].