

**PENGEMBANGAN APLIKASI ODOO DENGAN *COMMAND LINE INTERFACE*
TERHADAP *WAREHOUSE MANAGEMENT SYSTEM* MENGGUNAKAN MODEL
EXTREME PROGRAMMING
(STUDI KASUS: PT INDOGUNA UTAMA)**

Elis Sondang S.kom,M.M,M.Kom¹⁾ dan Brosky Samson Putra Halim²⁾

¹⁾ Staf Pengajar Studi Sistem Informasi

²⁾ Alumni Program Studi Teknik Informatika

Intitut Bisnis dan Informatika Kwik Kian Gie

Jl. Yos Sudarso Kav.87, Sunter Jakarta Utara 14350

Email: elis.sondang@kwikkiangie.ac.id

broskysamson@gmail.com

ABSTRACT

In the ongoing system, ERP Odoo has been used. However, the data input process become longer. This data input process based on GUI (Graphical User Interface). Interaction that occurs between user and computer is very much, thus slowing the input process. Therefore, a CLI-based system (Command Line Interface) is made to receives the main input in the form of barcode. In the system that has been created, researcher uses weight information that contained in the LOT code. User only have to scan LOT code using barcode scanner device, and system will process the data. The CLI-based system make the data input process become faster, because the weight of goods information already contained in the LOT code. User only need to input the LOT code and the system will convert it to weight for each box.

Keyword: Command Line Interface, Warehouse Management System, Enterprise Resource Planning, Extreme Programming.

1. Pendahuluan

Perkembangan teknologi dan informasi pada masa kini bergerak dengan cepat. Perkembangan ini tidak hanya terjadi di negara maju saja tetapi juga terjadi di negara berkembang seperti Indonesia. Dengan berkembangnya teknologi dan informasi, maka semua aspek kehidupan manusia menjadi lebih mudah dan lebih cepat.

Melihat perkembangan teknologi yang kian pesat, maka para pelaku bisnis juga mengikuti perkembangan tersebut. Para pelaku bisnis memanfaatkan perkembangan teknologi dan informasi ini untuk menjalankan proses bisnis di dalam perusahaan. Salah satu perusahaan yang memanfaatkan perkembangan

teknologi dan informasi ini adalah PT Indoguna Utama.

Sebagai salah satu perusahaan multi nasional yang bergerak di bidang impor dalam segmen pendistribusian makanan (*Food Supply Chain*), PT Indoguna Utama memiliki *warehouse* yang mendukung untuk menyimpan barang-barang yang telah diimpor sebelum nantinya didistribusikan.

Perlu diketahui bahwa setiap harinya, proses sistem pergudangan di dalam PT Indoguna Utama terjadi dengan cepat. Proses pergudangan itu meliputi proses Receiving, Internal Transfer, dan Delivery.

Di dalam *warehouse*, setiap pergerakan barang di dalam gudang dicatat pada lembar bernama "*tally*

sheet” oleh staff gudang. Selanjutnya data pada lembar *tally sheet* disimpan dalam dokumen Excel. Akan tetapi, pada prakteknya staff gudang terkadang mengambil barang tanpa melakukan perubahan data, dan memberitahukannya kepada bagian racking.

Hal ini mengakibatkan tidak akuratnya jumlah stok barang yang terdapat dalam sistem dengan kondisi nyata di dalam warehouse. Selain itu hal ini juga mengakibatkan sulitnya mencari barang di dalam warehouse.

Melihat permasalahan yang ada, PT Indoguna Utama sedang melakukan implementasi aplikasi ERP Odoo sebagai *Warehouse Management System* untuk mengelola pencatatan pergerakan barang di dalam gudang, akan tetapi sistem yang diterapkan tidak efisien karena proses *input* data menjadi lebih lambat. Selain proses input yang menjadi lebih lambat, input berat barang juga tidak akurat. Hal ini disebabkan karena *Human Error*. Semua permasalahan yang ada ini tidak bisa diatasi oleh sistem yang ada.

2. Tinjauan Pustaka

A. Sistem

Sistem adalah satu set komponen yang saling terkait, dengan batas yang jelas, bekerja sama untuk mencapai tujuan umum dengan menerima masukan dan menghasilkan output dalam transformasi yang terorganisir [4].

Sistem memiliki tiga fungsi dasar:

1. Input.

Merupakan masukan kedalam sistem yang selanjutnya diproses. Sebagai contoh bahan mentah, energi, data, dan usaha manusia harus diamankan dan diorganisir pengolahan.

2. Processing.

Merupakan suatu tahap untuk menghasilkan output. Contohnya adalah proses pembuatan, proses pernapasan manusia, atau perhitungan matematis.

3. Output.

Merupakan hasil akhir dari sistem. Setelah input diproses oleh sistem, maka selanjutnya sistem akan mengeluarkan output. Contohnya yaitu produk jadi, layanan manusia/jasa, dan informasi manajemen harus dikirimkan untuk pengguna mereka.

B. Sistem Informasi

Sistem Informasi merupakan gabungan antara orang (*brainware*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, sumber data, dan kebijakan dan prosedur untuk membuat, melihat, mengubah dan menghapus informasi di dalam suatu sistem. Orang mengandalkan sistem informasi modern untuk berkomunikasi satu sama lain menggunakan berbagai perangkat fisik (perangkat keras), instruksi dan prosedur pemrosesan informasi (perangkat lunak), saluran komunikasi (jaringan), dan data tersimpan (sumber data). Meskipun sistem informasi saat ini biasanya dianggap memiliki kaitan dengan komputer, namun kita telah menggunakan sistem informasi sejak awal peradaban. Bahkan saat ini kita sering menggunakan sistem informasi yang tidak ada hubungannya dengan komputer.

C. Aplikasi Odoo

Odoo adalah sebuah aplikasi bisnis open source yang ditulis dengan Python dan dirilis di bawah lisensi AGPL. Odoo memiliki fitur yang sangat lengkap mulai dari CRM, Sales Management, Purchase Management, Accounting, Finance, dan lain sebagainya.

Odoo adalah aplikasi ERP yang paling dikagumi dan memiliki perkembangan tercepat di dunia dan merupakan solusi perangkat lunak manajemen yang gratis. Sejak tahun 2009, Indonesia secara resmi telah membangun kerja sama dengan TinySPRL Belgia untuk mengembangkan OpenERP versi Indonesia. Odoo adalah

sebuah sistem manajemen yang sangat terkenal di dunia dan diunduh lebih dari 600 kali per hari (Tiny SPRL, 2015).

Odoo memiliki banyak modul yang siap untuk diunduh dan digunakan. Sebagian besar modulnya dapat didapatkan secara gratis. Beberapa modul yang ada didalam software Odoo antara lain Sales, Manufacturing, Purchase, Warehouse, Accounting, E-commerce dll.

Odoo (sebelumnya bernama OpenERP) lahir dengan nama TinyERP yang kemudian berubah menjadi OpenERP, dan selanjutnya bertransformasi menjadi Odoo.

D. Warehouse

Warehouse atau gudang (kata benda) merupakan bangunan yang dipergunakan untuk menyimpan barang dagangan. *Warehousing* atau pergudangan (kata kerja) ialah kegiatan menyimpan dalam gudang.

E. Warehouse Management System

Warehouse Management System (WMS) adalah solusi perangkat lunak yang digunakan untuk merencanakan, mengoptimalkan, dan menjalankan operasi gudang. WMS memberikan dukungan untuk kegiatan di dalam gudang seperti *orderpicking*, *order packing*, dan *put away*. WMS juga melaporkan status persediaan secara *real time* dan dapat digunakan untuk mengukur produktivitas pekerja gudang. Gudang mencakup banyak proses seperti penerimaan barang, *unpacking*, *put away*, *picking*, *full palet handling*, *packing*, *loading* dan *shipping*.

Alasan utama untuk menerapkan WMS adalah untuk meningkatkan efisiensi, mendapatkan lebih banyak kendali persediaan atau mencapai akurasi yang lebih baik.

Sistem manajemen gudang (WMS) juga dapat didefinisikan sebagai aplikasi perangkat lunak, yang dirancang untuk

mendukung manajemen gudang dan pusat distribusi dan staf. Mereka memfasilitasi manajemen dalam perencanaan harian mereka, mengatur, mengatur, mengarahkan, dan mengendalikan pemanfaatan sumber daya yang ada, untuk memindahkan dan menyimpan material ke dalam, di dalam, dan di luar gudang , sambil mendukung staf dalam pelaksanaan pergerakan material dan penyimpanan di dan sekitar sebuah gudang

WMS menggunakan database yang dikonfigurasi untuk mendukung operasi gudang, yang berisi detail yang menjelaskan berbagai elemen gudang standar termasuk:

1. Unit penyimpanan individual (SKU) yang ditangani dan disimpan, misalnya berat, dimensi, paket kasus, label ID otomatis (kode batang, dan lain-lain.), Dan inventaris berdasarkan lokasi dengan tanggal pembuatan, kode lot, dll. SKU dapat mencakup dasar bahan, bagian fabrikasi, rakitan, dan barang jadi industri dan konsumen, dll;

2. Lokasi penyimpanan gudang, misalnya, nomor lokasi individu, urutan pengambilan, jenis penggunaan (pemungutan, penyimpanan cadangan, dan lain-lain.), Jenis penyimpanan (masing-masing, kasus, palet), ukuran atau kapasitas lokasi, batasan penyimpanan (mudah terbakar, berbahaya, tinggi nilai bahan, outdoor, dll), dan lain-lain.

3. Pintu pintu, misalnya, nomor individu, dll; dan

4. Tingkat produktivitas tenaga kerja yang diharapkan berdasarkan fungsi atau aktivitas, misalnya, kasus yang dipilih per jam kerja, dll.

Dalam WMS ada beberapa konsep yang biasa digunakan, yaitu:

- FIFO (*First In First Out*), barang yang pertama masuk harus keluar pertama kali.

- LIFO (*Last In First Out*), barang yang terakhir masuk harus pertama kali keluar.

- FEFO (*First Expired First Out*), barang yang cepat kadaluarsa harus pertama kali keluar

F. User Interface

User Interface adalah interaksi antara user dengan operating sistem (OS). *User Interface* (UI) mengontrol bagaimana kita memasukan data dan instruksi serta bagaimana informasi ditampilkan pada layar[8].

Dua tipe dari *User Interface* adalah *Graphical* dan *Command Line*.

1. Graphical User Interface (GUI)

Sebagian besar user saat ini bekerja menggunakan *Graphical User Interface* (GUI). Dengan GUI, user dapat berinteraksi dengan menu dan gambar visual dengan menyentuh, pointing, tapping atau mengklik tombol atau objek lain untuk mengeluarkan perintah. Saat ini sistem operasi GUI menggabungkan fitur yang mirip dengan browser, seperti tautan dan tombol navigasi (yaitu tombol *Back* dan tombol *Forward*) saat menavigasi komputer atau media penyimpanan perangkat *mobile* untuk mencari file.

Graphical User Interface yang dirancang untuk input sentuhan terkadang disebut *Touch User Interface*. Beberapa sistem operasi untuk desktop dan laptop dan banyak sistem operasi untuk perangkat *mobile* memiliki antarmuka pengguna sentuh.

2. Command Line Interface (CLI)

Command Line Interface merupakan antarmuka yang banyak digunakan untuk mengkonfigurasi perangkat, mengatur sumber daya sistem, mengotomasi sistem, management sistem, dan memecahkan masalah jaringan, menggunakan

command-line interface. Pada command-line interface, user memasukan perintah yang ditunjukkan dengan kata kunci atau singkatan pendek atau menekan tombol khusus pada keyboard (seperti tombol fungsi atau kombinasi tombol) untuk memasukan data dan instruksi.

G. Enterprise Resource Planning (ERP)

Enterprise Resource Planning atau ERP merupakan sistem informasi yang mengintegrasikan MRP II dengan arus informasi di seluruh organisasi untuk dikelola dan mengkoordinasikan kegiatan perusahaan yang sedang berjalan, termasuk perencanaan produk, pembuatan dan pendistribusian, akuntansi dan keuangan, penjualan, sumber daya manusia, dan dukungan pelanggan[8].

H. Extreme Programming

XP (*eXtreme Programming*) merupakan metode yang paling dikenal dan paling banyak digunakan untuk metode *agile*. Nama ini diciptakan oleh Kent Beck (2000) karena pendekatan ini dikembangkan dengan practice yang bagus seperti *iterative development*, sampai ke level *extrem*. Contohnya di dalam XP, beberapa versi baru dari sistem dapat dikembangkan oleh programmer yang berbeda, diintegrasikan dan di test dalam satu hari.

XP sangat cocok untuk pengembangan proyek yang memerlukan adaptasi cepat dalam perubahan-perubahan yang terjadi selama pengembangan aplikasi. XP juga cocok untuk anggota tim yang tidak terlalu banyak dan berada pada lokasi yang sama dalam pengembangan sistem.

Para pengembang perangkat lunak banyak menggunakan metodologi *eXtreme Programming* untuk mengembangkan perangkat lunak dengan cepat.

1. Kerangka Kerja Extreme Programming.

Tahapan pengembangan perangkat lunak dengan XP meliputi:

- a. Planning/Perencanaan. Tahap ini dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan keluaran (output), fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, penentuan waktu dan biaya pengembangan aplikasi, serta alur pengembangan aplikasi.
- b. Design/Perancangan. Tahap ini menekankan pada desain aplikasi secara sederhana. Alat untuk mendesain pada tahap ini dapat menggunakan kartu CRC (Class Responsibility Collaborator). CRC digunakan untuk pemetaan (membangun) kelas-kelas yang akan digunakan pada diagram use case, diagram kelas, dan diagram objek.

CRC diperkenalkan oleh Kent Beck dan Ward Cunningham sekitar tahun 1989 sebagai kelengkapan pemrograman berorientasi objek. CRC sebagai cikal bakal yang menjadi kelas pada saat tahap analisis.

Deskripsi tentang kartu CRC:

- Nama kelas (Class Name): memberikan nama kelas.
- Kelas Induk (Superclass): merupakan kelas induk (orang tua) dalam konsep pewarisan yang akan dibuat CRC-nya.
- Kelas turunan (Subclass): merupakan kelas anak dalam konsep pewarisan yang akan dibuat CRC-nya.

- Tanggung Jawab (Responsibilities): atribut, operasi (methods) yang ada pada kelas yang dibuat CRC-nya.
 - Kelas terkait (Collaborators): kelas yang terkait dengan kelas yang dibuat CRC-nya tetapi bukan kelas induk (orang tua)/kelas anak (turunan).
- c. Coding/Pengkodean. Hal utama dalam pengembangan aplikasi dengan menggunakan XP adalah pair programming (dalam membuat program melibatkan 2 atau lebih programmer).
 - d. Testing/Pengujian. Tahap ini memfokuskan pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak ada kesalahan (error) dan aplikasi yang dibuat sesuai dengan proses bisnis pada klien (pelanggan).

2. Nilai Inti pada XP

Yang menjadi nilai inti dari pengembangan sistem dengan metodologi XP adalah:

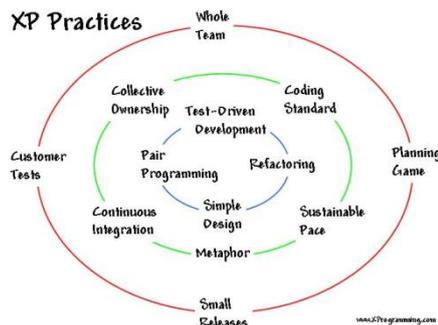
- a. Communication (Komunikasi). XP menekankan komunikasi antartim (programer), setiap anggota tim saling berbagi pengetahuan dalam pengembangan perangkat lunak.
- b. Simplicity (Kesederhanaan). Setiap anggota tim harus melakukan semuanya dengan sederhana, mencoba mencari solusi yang paling sederhana dan paling praktis.
- c. Feedback (Masukan). Feedback untuk mengetahui kemajuan dari

proses dan kualitas perangkat lunak yang dibuat. Setiap anggota tim dapat memberikan feedback.

- d. **Courage (Keberanian).** Untuk kesuksesan pengembangan aplikasi, setiap anggota tim harus memiliki keberanian, keyakinan, dan integritas dalam pekerjaannya.
- e. **Respect (Menghormati).** Setiap anggota tim sangat penting respect terhadap anggota tim lainnya, XP menerapkan siklus pendek dan integrasi continue.

3. Aspek Dasar XP

Aspek dasar yang perlu diperhatikan bila menggunakan XP dalam pengembangan aplikasiterdapat pada gambar 2.1.



Gambar 2.1
Aspek dasar eXtreme Programming (XP)

Nilai-nilai aspek dasar pada XP adalah:

- a. **The Planning Game.** Pada tahap ini XP mirip dengan metodologi RAD (Rapid Application Development). RAD menggunakan proses yang pendek dan cepat, mengutamakan aspek teknik. Proses ini juga menggunakan istilah game, menurut Kent Beck untuk menentukan

requirements dibutuhkan penggunaan teknik score card.

- b. **Small Releases.** Developer akan menyelesaikan setiap bagian dari perangkat lunak, kemudian dipresentasikan kepada perusahaan (klien). Apabila disetujui maka bagian tersebut diterapkan ke seluruh aplikasi, kemudian dilakukan tes awal pada penerapan aplikasi tersebut.
- c. **Metaphor.** Metaphor menggambarkan visi, tujuan pengembangan perangkat lunak secara luas. Metaphor juga bertujuan menjaga komunikasi antara klien dengan developer agar tetap baik.
- d. **Simple Design.** XP didesain sesederhana mungkin untuk menghindari pengembangan sistem yang sulit dimengerti. Bila terjadi perubahan pada desain, maka resiko kegagalan desain dapat diminimalkan.
- e. **Refactoring.** Refactoring merupakan suatu aspek khusus pada metodologi XP. Proses ini mengubah kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program. Sehingga developer tidak harus mengulang proses desain.
- f. **Testing.** Tim Developer harus membuat tes yang akan digunakan untuk menjalankan software dan mengembangkannya. Pada

- saat proses coding selesai, software tersebut akan diuji menggunakan model tes yang telah dibuat sebelumnya.
- g. Pair Programming. Merupakan aspek di mana dua orang programmer bersama-sama dalam satu tempat atau komputer, mereka saling berinteraksi dan berdiskusi tentang permasalahan dan penyelesaiannya.
 - h. Collective Ownership. Antar programmer saling berbagi informasi sehingga tidak ada ketergantungan terhadap satu orang programmer.
 - i. Coding Standart. Antar programmer harus sepakat dalam memahami script/program. Sehingga programmer yang terlibat dalam pengembangan proyek memiliki pemahaman yang sama.
 - j. Continous Integration. Developer diwajibkan untuk melakukan build agar kesalahan pada program dapat dideteksi dan diperbaiki secepat mungkin. Waktu maksimal yang dibutuhkan untuk build dalam XP adalah satu hari.
 - k. Coding Standart. Menurut pandangan Kent Beck, waktu ideal seorang programmer bekerja dalam satu hari adalah 8 jam dan 5 hari dalam satu minggu.
 - l. Customer Test. Bila menggunakan XP dalam pengembangan sistem maka sebaiknya melibatkan klien dalam proses build dan tes yang

dilakukan. Klien maka dapat memberikan masukan dan koreksi atas aplikasi yang dibuat sehingga aplikasi yang dibuat sesuai dengan kebutuhan klien.

- m. Sustainable Pace. Yang dimaksud dengan sustainable pace adalah proses pekerjaan yang dilakukandengankecepatan yang stabil dan nyaman.

I. Pemrograman Berorientasi Objek (PBO)

Object-orientation adalah pendekatan terhadap pengembangan sistem yang membantu menangani banyak masalah. Sesuai namanya, yang terpenting dalam object-orientation adalah object[2].

Pemrograman berorientasi objek (Inggris: object-oriented programming disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Konsep dasar dari OOP adalah

1. Kelas

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas

definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. Objek

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk

mengembangkan sebuah pengabstrakan.

4. Enkapsulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. Polimorfisme

Melalui pengiriman pesan, tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

J. Bar Code

Kode batang (*Bar Code*) adalah kode identifikasi yang terdiri dari satu set garis vertikal dan spasi dengan lebar yang berbeda atau pola dua dimensi titik, kotak, dan gambar lainnya. Kode batang mewakili data yang mengidentifikasi produsen dan item[8].

Bar code pertama kali muncul dan diperkenalkan oleh dua orang mahasiswa Drexel Institute of Technology Bernard Silver dan Norman Joseph Woodland pada tahun 1948. Mereka mempatenkan inovasi barcode tersebut pada tahun 1949 dan permohonan tersebut baru dikabulkan pada tahun 1952. Namun baru pada tahun 1996, penemuan mereka mulai digunakan dalam dunia komersial.

Bar code pada dasarnya adalah susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, sangat sederhana namun sangat berguna, dengan kegunaannya untuk menyimpan data-data spesifik misalnya seperti kode produksi,

tanggal kadaluwarsa, serta nomor identitas lingkaran konsentris atau tersembunyi dalam sebuah gambar.

Bar code dibaca dengan memakai sebuah alat baca optik yang disebut dengan *bar code* reader. Pada prinsipnya *bar codereader* hanyalah sebuah alat input biasa seperti halnya keyboard ataupun *scanner* namun peran manusia sebagai operator sangat minim dalam sisi *Point of Sale*. Penggunaan *bar code* sangatlah membantu dalam menganalisis data trend penjualan dengan sangat cepat.

1. Manfaat *bar code*

Ada beberapa manfaat dari adanya *bar code*, yaitu sebagai berikut:

- a. Proses input data lebih cepat, hal ini karena *bar code scanner* dapat membaca atau merekam data lebih cepat daripada melakukan proses input data secara manual.
- b. Proses input data lebih tepat, hal ini karena dengan teknologi *bar code* memiliki ketepatan yang tinggi dalam input data.
- c. Proses input lebih akurat mencari data, hal ini karena teknologi barcode memiliki akurasi serta ketelitian yang sangat tinggi.
- d. Mengurangi biaya, hal ini karena bisa menghindari kerugian dari kesalahan-kesalahan pencatatan data, serta dapat mengurangi pekerjaan yang dilakukan secara manual dan secara berulang-ulang.
- e. Peningkatan kinerja manajemen, hal ini dikarenakan dengan data yang lebih cepat, tepat serta akurat maka pengambilan keputusan yang dilakukan oleh manajemen akan jauh lebih baik dan lebih tepat, yang pada akhirnya akan sangat

berpengaruh dalam menetapkan kebijakan perusahaan.

- f. Kemampuan bersaing dengan perusahaan saingan atau kompetitor akan lebih terjaga.

2. Jenis-jenis Barcode

Terdapat beberapa jenis barcode, antara lain sebagai berikut:

a. *Linear Code* (Barcode 1 Dimensi)

Linear code terdiri atas:

- (1) *Code 39* (*code 3 of 9*), yaitu sebuah *bar code alphanumeric* (*bar code Full ASCII*) yang mempunyai panjang baris yang bervariasi.

Implementasi *bar code* jenis ini biasanya adalah untuk tanda pengenal identitas, *inventory*, dan *asset tracking*.

- (2) *Interleaved 2 of 5*, yaitu sebuah *bar code* berbentuk numerik dan mempunyai panjang baris yang bervariasi. *Bar code* jenis ini pada umumnya dipakai untuk industri dan laboratorium.

- (3) *Code 128*, yaitu suatu *bar code alphanumeric* (*Full ASCII*) yang mempunyai kerapatan (*density*) sangat tinggi serta dengan panjang baris yang bervariasi. Pemakaian barcode jenis ini biasanya

untuk sistem *shipping* dan *warehouse management*.

- (4) *UPC* (*Universal Product Code*), adalah sebuah *bar code numeric* dan memiliki panjang baris yang tetap (*fixed*). *UPC* banyak digunakan untuk pelabelan pada produk-produk berukuran kecil/ritel.

b. *Matrix Code* (Bar code 2 dimensi)

Matrix code (*bar code 2 dimensi*), lebih canggih dibandingkan dengan *Linear Code* (*bar code 1 dimensi*) hal ini karena *matrix code* dapat memuat ratusan digit karakter dan tampilannya juga berbeda dengan *linear code*. Pada *matrix code* (*bar code 2 Dimensi*), informasi atau data yang besar bisa disimpan dalam ruang (*space*) yang kecil. Contoh dari *matrix code* (*bar code 2 Dimensi*) antara lain yakni *PDF417* yang bisa menyimpan lebih dari 2000 karakter dalam sebuah *space* 4.

3. Jenis Bar code Scanner

Terdapat setidaknya 4 jenis *bar code reader* yang umum dan tersedia di pasaran, antara lain sebagai berikut:

a. *Pen Type Readers* atau *Bar Code Wands*

Tipe ini memiliki *photo diode* yang berada disamping ujung pena. Untuk membaca *bar code*, cukup menempatkan kode tersebut di ujung pena lalu digeser ke semua bar secara stabil, lalu kemudian

diode tersebut akan mengukur intensitas cahaya yang dipantulkan dari sumber cahaya dan menghasilkan gelombang yang sesuai dengan lebar dari bar serta spasi dalam kode tersebut. Sesudah itu *bar code reader* akan mengirimkan gelombang ke *decoder* lalu kemudian menterjemahkannya dan mengirimkannya ke komputer dalam format data yang sederhana.

b. *Laser Bar Code Scanner*

Intinya tipe *bar code reader* ini cara kerjanya sama dengan tipe pena akan tetapi barcode reader ini memakai sinar laser sebagai sumber cahayanya. Biasanya menggunakan cermin prisma atau kaca bolak-balik untuk memindai laser yang melintasi kode bar.

c. *Camera Based Bar Code Readers*

Bar code scanner tipe ini berbasis kamera video kecil untuk dapat menangkap gambar ke kode bar, kemudian memakai teknik pengolahan citra digital untuk dapat memecahkan kode bar tersebut.

d. *CCD Bar Code Scanners*

Bar code scanner tipe ini memakai array sensor cahaya berbentuk kecil yang berbaris sejajar pada ujung *bar code scanner*. Tegangannya berbentuk seperti gelombang yang sesuai dengan bar dan ruang dari *bar code* yang dihasilkan dan dikirim ke komputer. Perbedaan utama dari *scanner bar code* CCD dengan tipe *scanner bar code* pena dan *scanner laser bar code* ialah bahwa *bar code scanner* CCD akan mengukur bentuk cahaya yang dipancarkan dari kode bar

sementara pena atau *laser scanner bar code* mengukur dari pantulan cahaya dari frekuensi tertentu yang berasal dari *scanner* itu sendiri.

3. Metode Penelitian

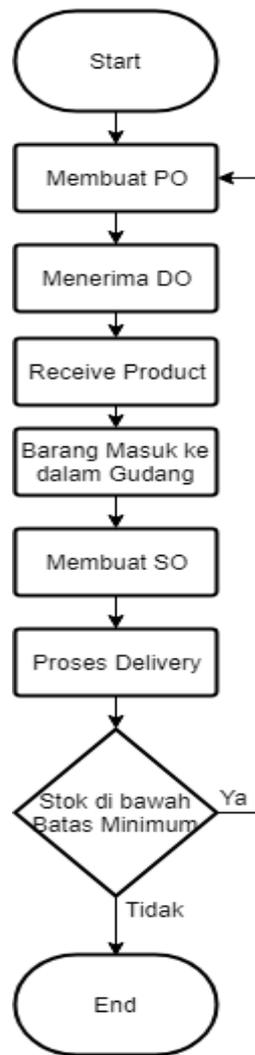
Latar Belakang Objek Penelitian

PT Indoguna Utama merupakan perusahaan yang bergerak di bidang distribusi bahan makanan (*food supply chain*). Setiap barang yang di importasikan masuk kedalam gudang terlebih dahulu.

Proses pergudang dimulainya tahap staff purchasing membuat RFQ (*Request for Quotation*) kepada beberapa vendor. Apabila harga dan qty disetujui, maka staff purchasing akan mengubah status document RFQ menjadi PO kepada vendor.

Setelah PO diterima, maka vendor akan melakukan proses *delivery order* (DO). Sesampainya di Indonesia, barang tersebut dikirimkan ke *warehouse* PT Indoguna yang berada di Pondok Bambu untuk dilakukan proses *receiving*. Pada tahap ini, akan di cocokkan antara qty yang diterima dengan dokumen DO. Apabila jumlah tidak sesuai, barang akan dikirimkan pada pengirim selanjutnya.

Setelah barang masuk kedalam *warehouse*, maka barang tersebut sudah boleh di jual oleh sales. Setelah customer membuat SO kepada sales, maka sales akan mengirimkan dokumen SO tersebut kepada gudang untuk dilakukan proses *delivery order* (DO)



Gambar 3.1
Sistem yang Berjalan

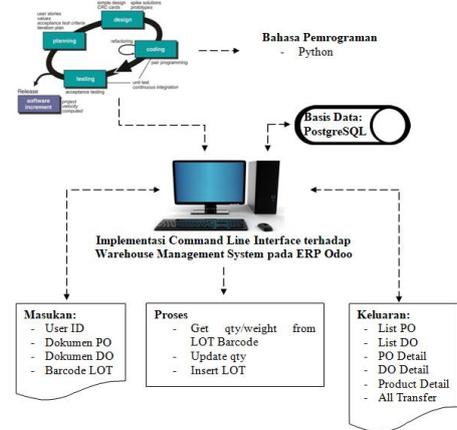
Teknik Pengumpulan Data

Salah satu komponen penting dalam penelitian adalah proses peneliti dalam mengumpulkan data. Data yang dikumpulkan sangat membantu peneliti dalam membangun sistem yang akan peneliti kembangkan.

Dalam penelitian ini, peneliti menggunakan teknik pengumpulan data, yaitu wawancara terstruktur, observasi tidak langsung, dan studi pustaka.

4. Hasil dan Pembahasan

Design Alur Pengembangan Aplikasi



Topologi Jaringan



Rancangan Alur Program

1. User Verification

Berguna untuk masuk ke dalam aplikasi.

Scan Barcode Kartu Identitas / Login / Ketik manual
 If User Validation = Valid:
 Succes Login
 Else
 Login Failed
 End If

2. Proses Receiving

Berguna untuk melakukan proses Receiving (barang masuk).

Pilih Menu Receive / Scan Barcode
 Scan Barcode Dokumen PO

While (Product >0)
 Scan Barcode Barang
 Input Digit Start
 Input Panjang Character
 Input Decimal Place
 While (Karton >0)
 Scan Kode LOT
 End While
 End While
 OK

3. Proses Delivery
 Berguna untuk melakukan proses Delivery (barang keluar).

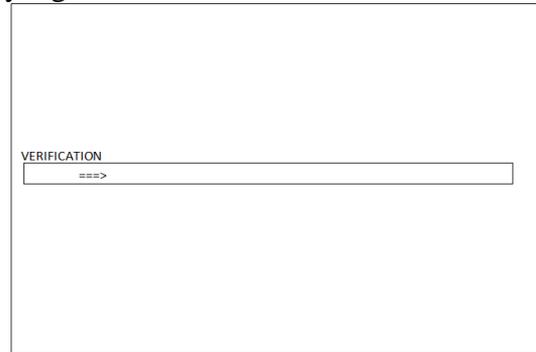
Pilih Menu Delivery / Scan Barcode
 Scan Barcode Dokumen DO
 While (Product >0)
 Scan Barcode Barang
 Input Digit Start
 Input Panjang Character
 Input Decimal Place
 While (Karton >0)
 Scan Kode LOT
 End While
 End While
 OK

4. Membuat file CSV
 Berguna untuk melakukan proses Receiving (barang masuk).

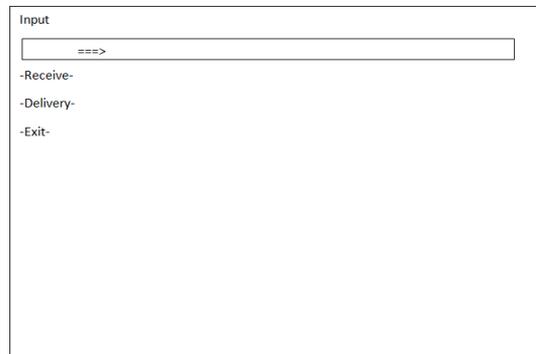
Pilih Menu Receive / Scan Barcode
 Scan Barcode Dokumen PO /DO
 While (Product >0)
 Scan Barcode Barang
 Input Digit Start
 Input Panjang Character
 Input Decimal Place
 While (Karton >0)
 Scan Kode LOT
 End While
 End While

Rancangan Antar Muka

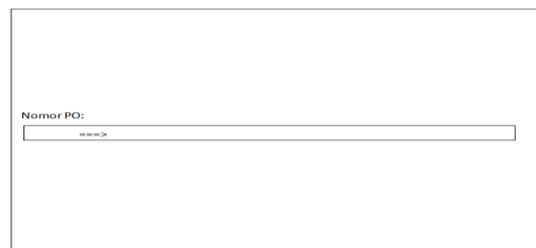
Berikut ini adalah rancangan antar muka dari aplikasi Command Line Interface yang dibuat.



Gambar 4.1 Rancangan Tampilan user Verification (Login)



Gambar 4.2 Rancangan Tampilan Menu Utama



Gambar 4.3 Rancangan Tampilan Search PO (Input Nomor PO)



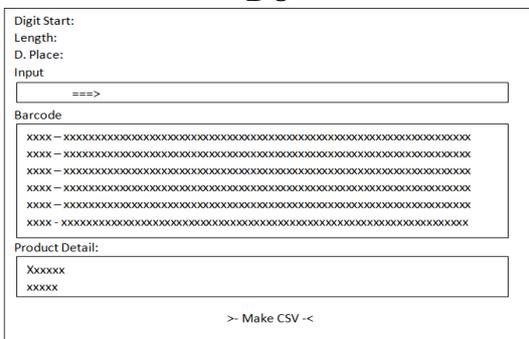
Gambar 4.4 Rancangan Tampilan Detail PO



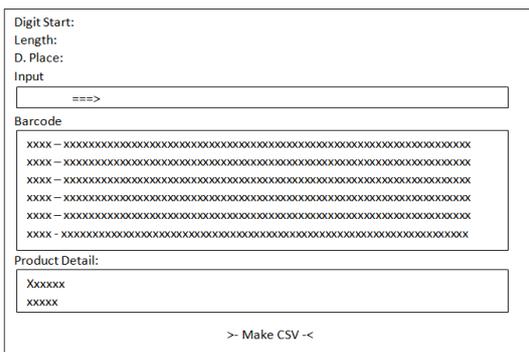
Gambar 4.5 Rancangan Tampilan Search DO (Input Nomor DO)



Gambar 4.6 Rancangan Tampilan Detail DO



Gambar 4.7 Rancangan Tampilan Input Barcode PO



Gambar 4.8 Rancangan Tampilan Input Barcode DO

5. Kesimpulan dan Saran

Berdasarkan penelitian yang telah penulis lakukan dan setelah aplikasi yang dibuat sudah selesai dan melalui proses testing dan evaluasi, maka penulis mengambil kesimpulan:

1. Proses pergudangan yang terjadi dapat berjalan dengan cepat, karena proses input data ke dalam *warehouse management system* yang lebih cepat dan efisien (sudah terotomisasi).
2. Staff gudang tidak perlu lagi melapor kepada bagian racking setelah melakukan quick count karena proses pergudangan sudah ditangani oleh sistem. Perpindahan barang hanya boleh terjadi bila melalui sistem. Bila tidak terdaftar pada sistem, maka tidak boleh terjadi perpindahan barang di dalam gudang.
3. Stock barang yang terdapat di dalam gudang menjadi akurat. Hal ini karena semua proses didata dan disimpan di dalam sistem.
4. Pencarian barang di dalam gudang menjadi lebih mudah karena lokasi barang sudah terdata pada sistem.
5. Input berat barang menjadi lebih akurat, karena informasi berat sudah terdapat pada barcode kode lot.

Saran yang dapat diberikan untuk pengembangan di kemudian hari berdasarkan hasil penelitian ini adalah:

1. Melakukan pengembangan aplikasi untuk dapat menangani proses *internal transfer* atau proses perpindahan antar gudang.
2. Melakukan penelitian lebih lanjut terhadap aplikasi yang

dibuat, termasuk menangani *unit of measure* (UOM) selain KG seperti LBS.

6. Daftar Pustaka

- [1] Artmant dan Jonsson. 2014. *Managing Change in the Warehouse*. Sweden.
- [2] Bennett, Simon. et al. 2010. *Object-Oriented Systems Analysis and Design Using UML. 4th Edition*. New York: McGraw-Hill.
- [3] I Gusti Ngurah Suryantara. 2017. *Merancang Aplikasi dengan Metodologi Extreme Programmings*. Jakarta: Elex Media Komputindo.
- [4] Marakas dan O'Brien. 2013. *Introduction to Information Systems, Sixteenth Edition*. McGraw Hill, New York – USA.
- [5] Pressman, R.S. 2012. *Rekayasa Perangkat Lunak: Pendekatan Praktisi*. Yogyakarta: Penerbit Andi.
- [6] Sommerville, Ian. 2009. *Software Engineering*. Pearson.
- [7] Sugiyono. 2016. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- [8] Vermaat et.al. 2016. *Discovering Computers 2016*. Cengage Learning.