

Pengembangan Sistem Inventory Control Perusahaan Berbasis Web Studi Kasus PT. XYZ

Muhammad Al Anis Faishal¹, Tri Astoto Kurniawan², Denny Sagita Rusdianto³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹maafaishal@gmail.com, ²triak@ub.ac.id, ³denny.sagita@ub.ac.id

Abstrak

Dalam menjalankan aktivitas logistik setiap harinya, PT. XYZ menggunakan 2 Sistem *Inventory Control*, yakni Sistem *Inventory Control* internal dan eksternal. Sistem *Inventory Control* internal adalah sebuah sistem yang dibeli dari pengembang perangkat lunak. Sedangkan, Sistem *Inventory Control* eksternal adalah sistem yang disewa dari sebuah perusahaan Jasa Pertukaran Data Elektronik (PDE) di Indonesia setiap bulannya agar dapat diakses oleh pihak Ditjen Bea dan Cukai, karena berdasarkan Peraturan Dirjen Bea dan Cukai Nomor per-09/BC/2014 BAB II Pasal 2 Ayat 1 setiap perusahaan yang menggunakan fasilitas Kawasan Berikat harus memiliki *IT Inventory* yang dapat diakses pihak Ditjen Bea dan Cukai. Karena PT. XYZ memiliki 2 Sistem *Inventory Control*, staf logistik harus memasukkan data yang sama sebanyak 2 kali ke kedua sistem tersebut. Jadi diperlukan sebuah Sistem *Inventory Control* yang dapat digunakan secara internal dan eksternal, yaitu sebuah Sistem *Inventory Control* berbasis Web hasil kombinasi Sistem *Inventory Control* internal dan eksternal. Sistem *Inventory Control* berbasis Web hasil kombinasi Sistem *Inventory Control* internal dan eksternal tersebut telah diuji dengan melakukan pengujian unit dan integrasi terhadap 3 sampel fungsi dari beberapa klas menggunakan metode *white-box testing* mendapatkan hasil 100% valid. Kemudian, pengujian sistem dengan menguji 85 kebutuhan fungsional menggunakan metode *black-box testing* mendapatkan hasil 100% valid.

Kata Kunci: *bea dan cukai, model waterfall, pengembangan berorientasi objek, sistem inventory control*

Abstract

In conducting logistics activities every day, PT. XYZ uses 2 Inventory Control Systems, that are internal and external Inventory Control Systems. The Internal Inventory Control System is a purchased system from software developers. Then, the External Inventory Control is a leased system of a Electronic Data Interchange (EDI) services company in Indonesia every month that can be accessed by the Directorate General (DG) of Customs and Excise, as there is a Regulation of the DG of Customs and Excise No. per-09 / BC / 2014 CHAPTER II Article 2 Section 1, every company that uses Bonded Zone facilities must have IT Inventory accessible to the DG of Customs and Excise. Because PT. XYZ has 2 Inventory Control Systems, the logistics staff should enter the same data twice into both systems. So we need a good Inventory Control System that can be used internally and externally, that is a Web-based Inventory Control System, the result of a combination of internal and external Inventory Control Systems to be able to solve the problems of PT. XYZ right now. Web-based Inventory Control System, the result of a combination of internal and external Inventory Control System has been tested by unit testing and integration testing of 3 sample methods from several classes using white-box testing method get 100% valid results. Then, system testing by testing 85 functional requirements using black-box testing method get 100% valid results.

Keywords: *customs and excise, waterfall model, object oriented, inventory control system*

1. PENDAHULUAN

Inventory adalah persediaan suatu barang atau sumber yang digunakan pada suatu

organisasi. *Inventory Control* adalah sekumpulan kebijakan dan pengendalian untuk memantau tingkat persediaan dan menentukan persediaan mana yang harus dipertahankan, kapan persediaan harus diisi ulang, dan seberapa

besar pesanan yang akan keluar (Jacobs & Chase, 2013). Pada industri manufaktur, persediaan manufaktur umumnya mengacu pada barang-barang yang menjadi bagian dari produk perusahaan. Persediaan manufaktur biasanya diklasifikasikan ke dalam bahan baku, produk akhir, dan komponen. Dalam distribusi persediaan, persediaan digolongkan sebagai *in-transit*, artinya persediaan dipindahkan di dalam sistem dan gudang atau pusat distribusi.

Manfaat dari *Inventory Control* meliputi (Pontius, 2017) :

- a. Meningkatkan profit
- b. Memiliki cukup persediaan sehingga tidak kehabisan
- c. Sistem secara otomatis mendeteksi produk yang lebih banyak
- d. Pengurangan atau penghapusan persediaan
- e. Kemampuan untuk melakukan audit lebih cepat dan efisien

PT. XYZ merupakan alias atau nama lain dari perusahaan tempat melakukan penelitian ini yang tidak ingin disebutkan di dalam penelitian ini. PT. XYZ merupakan suatu perusahaan multinasional yang bergerak di bidang *trading* atau perdagangan bahan-bahan kimia. Dalam menjalankan aktivitas bisnis setiap harinya, PT. XYZ menggunakan 2 (dua) Sistem *Inventory Control* dalam melakukan pengelolaan logistik yakni Sistem *Inventory Control* internal dan Sistem *Inventory Control* eksternal (Suwandaru, 2017).

Sistem *Inventory Control* internal merupakan sebuah sistem yang dibeli dari pengembang perangkat lunak. Sedangkan, Sistem *Inventory Control* eksternal merupakan sebuah sistem yang disewa dari sebuah perusahaan Jasa Pertukaran Data Elektronik di Indonesia untuk dapat diakses oleh pihak Ditjen Bea dan Cukai. Hal ini mengacu pada mengacu kepada Peraturan Direktur Jenderal Bea dan Cukai Nomor per-09/BC/2014 BAB II Pasal 2 Ayat 1 yang berbunyi "Perusahaan yang menggunakan fasilitas: Pembebasan, Pengembalian, Kawasan Berikat, Gudang Berikat, dan/atau Toko Bebas Bea diharuskan untuk memiliki dan mendayagunakan *IT Inventory*".

Karena PT. XYZ memiliki dua Sistem *Inventory Control* yang beroperasi dalam aktivitas logistik, staf logistik harus memasukan data yang sama sebanyak dua kali. Hal ini membutuhkan waktu yang banyak hanya untuk

melakukan kegiatan yang sama. Sehingga diperlukan sebuah Sistem *Inventory Control* yang dapat digunakan secara internal dan eksternal yaitu dengan melakukan pengembangan Sistem *Inventory Control* berbasis Web.

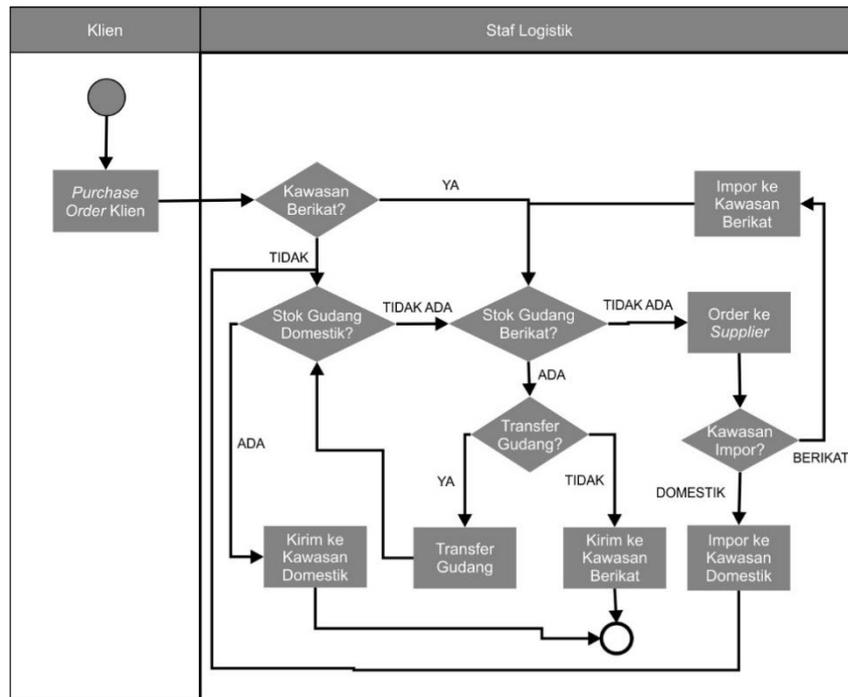
Dari ulasan diatas maka diperlukan sebuah Sistem *Inventory Control* perusahaan berbasis Web yang diharapkan meningkatkan efisiensi waktu dalam memasukan data dan biaya operasional perusahaan, sehingga tidak perlu lagi menyewa sistem pada perusahaan lain, mengurangi prosedur manualisasi, dan memudahkan dalam memantau setiap transaksi dokumen logistik. Pengembangan Sistem *Inventory Control* berbasis Web menggunakan model *Software Development Life Cycle* (SDLC) Waterfall. Model SDLC Waterfall tepat digunakan dalam pengembangan Sistem *Inventory Control* berbasis Web karena semua kebutuhan sudah dipahami dengan baik pada awal pengembangan, definisi sistem stabil dan paham dengan teknologi yang digunakan. Teknologi yang digunakan dalam pengembangan sistem tersebut yakni PHP dengan *framework Model-View-Controller* (MVC) digunakan untuk proses pengelolaan data dan *Database Management Systems* (DBMS) MySQL digunakan untuk mengelola basis data.

2. METODOLOGI PENELITIAN

Metodologi penelitian berisi tahapan dalam menyelesaikan pengembangan Sistem *Inventory Control* Perusahaan Berbasis Web Studi Kasus PT. XYZ. Tahapan-tahapan yang dilakukan dalam metodologi penelitian ini mengacu pada model SDLC Waterfall. Berikut merupakan diagram alir dari metodologi penelitian dapat dilihat pada Gambar 2.

Tahap pertama yang dilakukan pada metodologi penelitian adalah studi literatur. Tahap ini dilakukan untuk mempelajari beberapa dasar teori yang diperoleh dari berbagai referensi.

Tahap kedua adalah analisis kebutuhan. Tahap ini dilakukan dengan mengelitisasi dan menganalisis kebutuhan menjadi definisi kebutuhan dan spesifikasi kebutuhan, kemudian dikategorikan menjadi beberapa kebutuhan seperti fungsional dan nonfungsional dan mengidentifikasi aktor. Lalu membuat pemodelan kebutuhan *Use Case Diagram* dan *Use Case Scenario*.



Gambar 1. Proses bisnis *inventory control* PT. XYZ
 Sumber : Suwrandu (2017)

Tahapan ketiga adalah perancangan. Tahapan ini dilakukan dengan membuat pemodelan *Sequence Diagram* dari hasil pemodelan kebutuhan. Kemudian hasil beberapa *Sequence Diagram* dikelompokkan menjadi beberapa kelas pada pemodelan *Class Diagram* dan membuat algoritme masing- masing fungsi di setiap kelas. Lalu membuat model *Entity Relationship Diagram* (ERD) dan rancangan antarmuka.

Tahapan keempat adalah implementasi. Tahap ini mengimplementasikan sistem berdasarkan rancangan *Class Diagram*, algoritme, ERD, dan antarmuka.

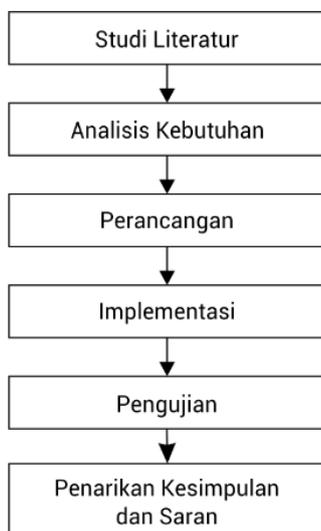
Tahapan kelima adalah pengujian. Tahap ini dilakukan dengan melakukan pengujian unit, integrasi, dan sistem.

Tahapan yang terakhir adalah penarikan kesimpulan dan saran. Tahapan ini dilakukan setelah semua tahapan dari studi literatur hingga pengujian telah dilakukan, kemudian dapat diperoleh kesimpulan dan saran dari hasil penelitian yang dilakukan.

3. ANALISIS KEBUTUHAN

Tahap analisis kebutuhan merupakan langkah awal dalam penelitian yang harus tercukupi sehingga hasil dari analisis kebutuhan dapat digunakan dalam perancangan sistem yang dilakukan menggunakan pendekatan OO.

Pertama kali melakukan elisitasi kebutuhan dengan melakukan identifikasi aktor terlebih dahulu, kemudian mendefinisikan kebutuhan sistem dan masing-masing definisi kebutuhan tersebut dijelaskan pada spesifikasi kebutuhan. Setelah elisitasi telah dilakukan, kemudian pemodelan *use case diagram* dari semua kebutuhan fungsional yang telah didefinisikan. *Use case diagram* dapat dilihat pada Gambar 3. Gambar 3 hanya menampilkan beberapa *use case* utama. Setelah membuat *use case diagram*, langkah selanjutnya ialah membuat *use case scenario*.



Gambar 2. Diagram Alir Metodologi Penelitian

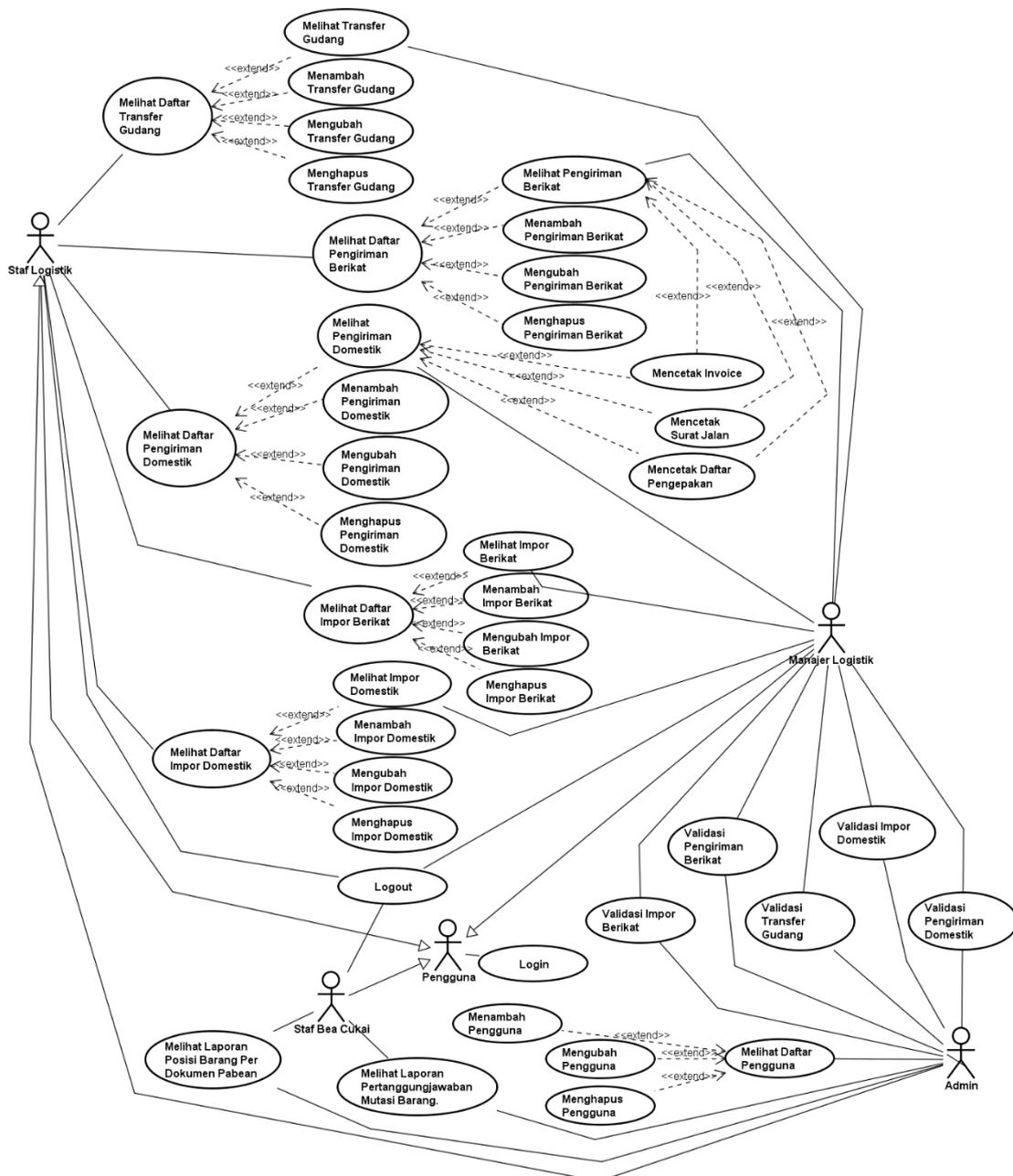
4. PERANCANGAN

Setelah tahap analisis kebutuhan selesai dilakukan, tahap selanjutnya ialah perancangan. Hasil dari analisis kebutuhan akan dijadikan acuan dalam melakukan perancangan. Terdapat lima tahap yang dilakukan dalam melakukan perancangan, yakni pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan algoritme, perancangan basis data, dan perancangan antarmuka.

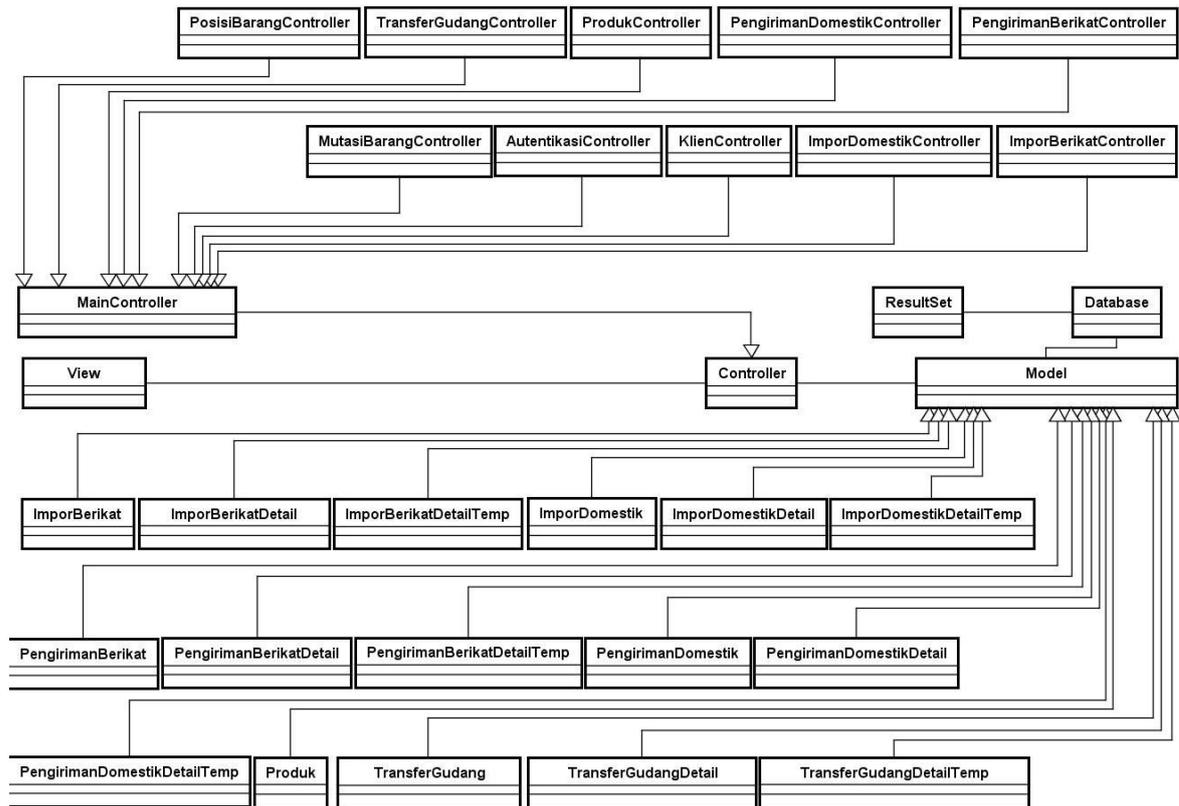
Pemodelan *sequence diagram* dilakukan untuk menggambarkan interaksi antar sejumlah objek. Kemudian pemodelan *class diagram*

dilakukan untuk menggambarkan struktur dan klas, serta hubungan antar klas berdasarkan hasil pemodelan *sequence diagram*. Hasil pemodelan *class diagram* dapat dilihat pada Gambar 5. Gambar 5 hanya menampilkan beberapa klas utama. Setelah melakukan pemodelan *class diagram*, kemudian melakukan perancangan algoritme. Perancangan algoritme dilakukan pada masing-masing fungsi yang terdapat pada masing-masing klas yang terdapat pada hasil pemodelan *class diagram*.

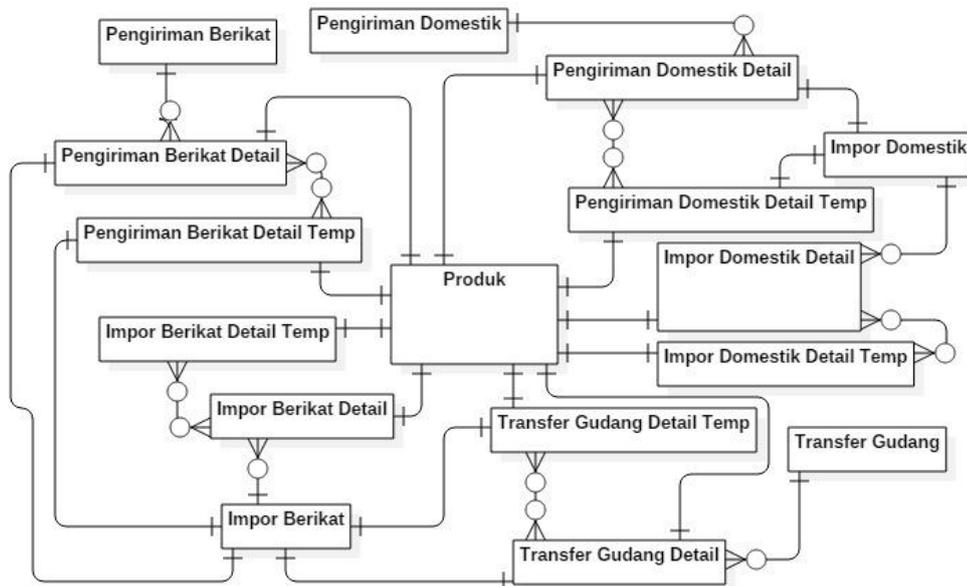
Selanjutnya melakukan perancangan basis data. Perancangan basis data dilakukan secara *Conceptual, Logical, dan Physical*.



Gambar 3. Use Case Diagram (Parsial)



Gambar 4. Class Diagram (Parsial)



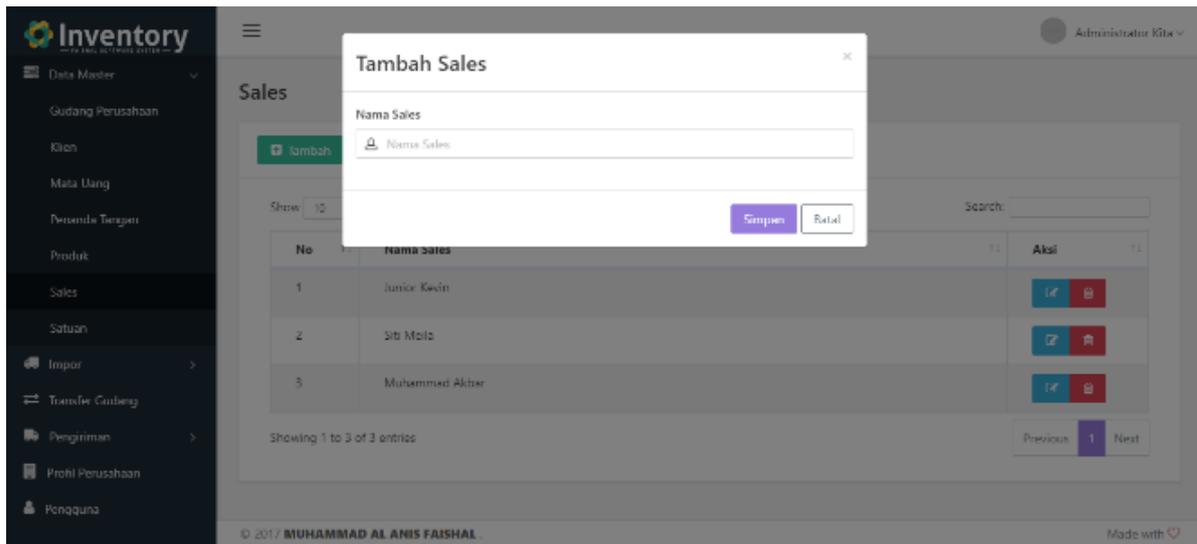
Gambar 5. Conceptual Data Model (Parsial)

Perancangan basis data secara *Conceptual* dapat dilihat pada Gambar 5. Gambar 5 hanya menampilkan beberapa entitas utama.

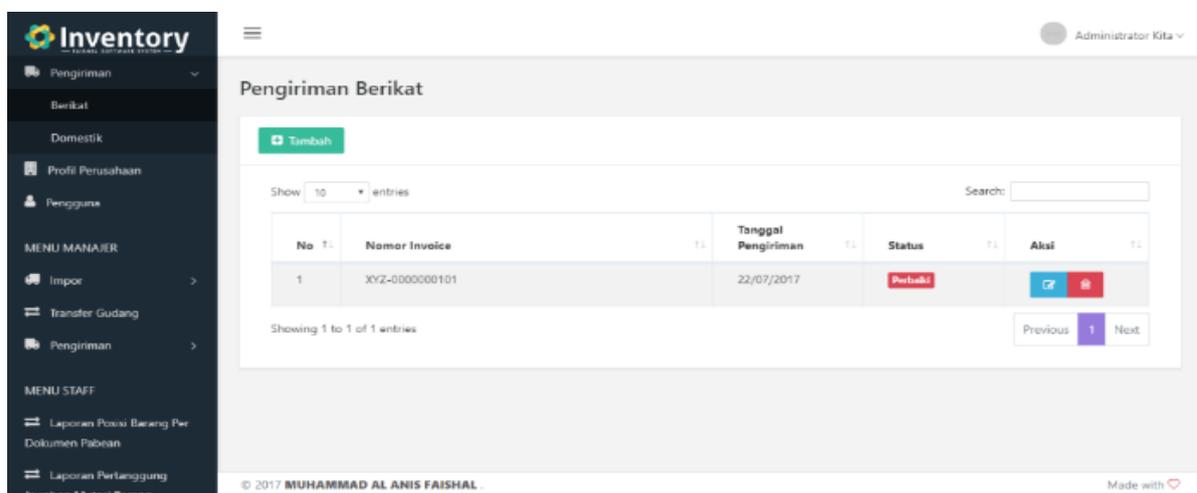
Kemudian tahap terakhir dalam melakukan perancangan adalah perancangan antarmuka. Perancangan antarmuka dilakukan dengan membuat rancangan dari tampilan sistem yang akan dibangun.

5. IMPLEMENTASI

Tahap implementasi dilakukan berdasarkan hasil analisis kebutuhan dan perancangan yang telah dilakukan sebelumnya. Pada tahap ini, rancangan *class diagram*, algoritme, basis data, dan



Gambar 6(a). Implementasi Antarmuka Menambah Sales



Gambar 6(b). Implementasi Antarmuka Pengiriman Berikat

antarmuka diimplementasi. Implementasi *class diagram* dibuat menjadi kode program berdasarkan dari rancangan algoritme.

Berikut beberapa hasil implementasi antarmuka, yakni antarmuka Menambah Sales pada Gambar 6(a) dan antarmuka Pengiriman Berikat pada Gambar 6(b). Pada Gambar 6(a) terdapat formulir untuk menambah Sales baru. Pada Gambar 6(b) terdapat daftar pengiriman berikat beserta status dokumen dan tombol untuk dapat menambah, mengubah, dan menghapus pengiriman berikat.

6. PENGUJIAN

Tahap pengujian dilakukan setelah tahap implementasi selesai dilakukan. Tahap ini bertujuan untuk melakukan pemeriksaan terhadap hasil implementasi apakah sesuai dengan analisis kebutuhan dan perancangan yang telah dibuat sebelumnya. Pengujian yang

dilakukan adalah pengujian unit, pengujian integrasi, dan pengujian sistem.

Pengujian unit merupakan pengujian yang dilakukan terhadap setiap operasi yang terdapat pada masing-masing klas. Pengujian unit dilakukan menggunakan *basis path testing* yang merupakan salah satu jenis dari *white-box testing*. Berikut merupakan langkah-langkah *basis path testing* :

- 1) Membuat *flow graph* dari hasil perancangan algoritme
- 2) Menghitung *cyclomatic complexcity* dari hasil *flow graph* yang dibuat
- 3) Menentukan jalur *independent path* dari hasil *flow graph* yang dibuat
- 4) Membuat kasus uji berdasarkan jalur *independent path*

Setelah diperoleh beberapa jalur *independent path*, maka hasil implementasi diuji

berdasarkan kasus uji yang dibuat. Pengujian unit dilakukan pada operasi login() pada kelas AutentikasiController, operasi add() pada kelas ImporBerikatController, dan operasi add() pada kelas PengirimanBerikatController. Setelah dilakukan pengujian unit, 100% semua kasus uji valid pada ketiga operasi tersebut.

Pengujian integrasi merupakan pengujian yang dilakukan terhadap setiap kelas yang memiliki hubungan satu dengan yang lainnya. Pengujian integrasi dilakukan menggunakan *basis path testing*. Pengujian integrasi dilakukan pada operasi add() pada kelas PengirimanBerikatController yang memanggil operasi getSort() pada kelas PenandaTanganModel, operasi getSort() pada kelas GudangPerusahaanModel, operasi getSort() pada kelas MataUangModel, operasi getWhere() pada kelas Klien, operasi getJoin() pada kelas ImporBerikat Model, operasi getJoin(), get(), dan delete() pada kelas PengirimanBerikatDetailTempModel, operasi insert() dan getWhere() pada kelas PengirimanBerikat Model, operasi insert() pada kelas PengirimanBerikat DetailModel. Setelah dilakukan pengujian integrasi, 100% semua kasus uji valid pada ketiga operasi tersebut.

Pengujian sistem merupakan pengujian yang dilakukan terhadap sistem untuk mengetahui apakah sistem yang telah dikembangkan telah sesuai dengan setiap kebutuhan yang telah ditentukan pada tahap analisis kebutuhan, yakni kebutuhan fungsional dan nonfungsional. Pengujian sistem dilakukan menggunakan teknik pengujian *Equivalence partitioning*, *Boundary Value Analysis*, dan *Decision Tables* yang merupakan salah satu jenis dari *black-box testing*. Setelah dilakukan pengujian sistem, 100% semua kasus uji valid pada 85 kebutuhan fungsional.

7. KESIMPULAN

Berdasarkan hasil penelitian mengenai “Pengembangan Sistem *Inventory Control* Perusahaan Berbasis Web Studi Kasus PT. XYZ” terdapat kesimpulan dari analisis kebutuhan, perancangan, implementasi, dan pengujian yang dilakukan.

Dari hasil analisis kebutuhan, terdapat 5 aktor dan 85 kebutuhan fungsional yang dihasilkan untuk dapat mengelola *Inventory* di PT. XYZ.

Dari hasil perancangan OO, menghasilkan terdapat 46 kelas yang terbagi dalam komponen

Model, *View*, dan *Controller* yang masing-masing operasi dari setiap kelas tersebut dijelaskan dalam bentuk algoritme, 26 tabel dari hasil perancangan basis data, dan antarmuka yang digunakan pada tahap implementasi sebagai acuan. Pada tahap implementasi menghasilkan beberapa fitur yang mendukung untuk melakukan *Inventory Control*.

Dari hasil pengujian, 100% pengujian yang dilakukan valid dalam Sistem *Inventory Control* Perusahaan PT. XYZ. Pengujian dilakukan dengan menggunakan *White-box Testing* dan *Black-box Testing*. *White-box Testing* dilakukan pada pengujian unit dan integrasi dan *Black-box Testing* dilakukan pada pengujian sistem.

DAFTAR PUSTAKA

- Jacobs, F Robert & Chase, Richard B. 2013. *Operations and Supply Chain Management: The Core*. New York: McGraw-Hill Education.
- Peraturan Direktur Jenderal Bea dan Cukai Nomor Per-09/Bc/2014. n.d.. Jakarta: Kementerian Keuangan Republik Indonesia.
- Pontius, Nicole, 2017. *What Is Inventory Control?*. Tersedia di: <<https://www.camcode.com/asset-tags/what-is-inventory-control/>> [Diakses 4 Januari 2018]
- Suwandaru, A., 2017. *Inventory Control System* Perusahaan. Diwawancara oleh Muhammad Al Anis Faishal. Bekasi, 15 Januari 2017