

Rancang Bangun Sistem Informasi Pembelian dan Penjualan Berbasis Client - Server Pada Toko XYZ

Kevin Gozali

Abstrak— Banyaknya perusahaan yang menggunakan sistem informasi dan aplikasi untuk membantu operasional mereka. Teknologi Informasi dapat membantu perusahaan-perusahaan untuk mengefisienkan, memudahkan, hingga mengotomatisasi operasional mereka. Selain itu, data-data yang tersimpan dapat digunakan untuk membuat keputusan-keputusan bisnis yang dapat meningkatkan efisiensi mulai dari penjualan, pembelian, produksi, dll. Penggunaan teknologi yang sama juga dibutuhkan untuk sebuah toko. Melihat masalah ini, merancang dan membangun aplikasi yang dapat menyesuaikan kebutuhan Toko XYZ adalah pilihan yang paling tepat untuk memenuhi tujuan penelitian ini yaitu untuk mengefisienkan operasional Toko XYZ secara keseluruhan. Metode-metode yang akan dilakukan dimulai dengan analisa dan observasi yang dapat menentukan kebutuhan dari Toko XYZ sekaligus memahami proses bisnisnya. Hasil dari penelitian ini adalah sebuah sistem dan aplikasi yang dapat memenuhi tujuan-tujuan dari penelitian berbasis *client-Server* yang dibangun dalam platform *Desktop Application*. Teknologi yang dimanfaatkan adalah *client-server*, yang merupakan teknologi yang dapat membantu mengintegrasikan data secara *real-time* dan data akan terpusat pada satu *server* dimana *client* akan dapat berbagi data secara mudah. Kesimpulan yang diperoleh adalah aplikasi yang dikembangkan telah mengintegrasikan data secara *real time* dan akan sangat membantu mengefisienkan operasional mereka.

Kata Kunci: *Client-Server, Aplikasi dekstop, Real-time, Data, Efisiensi Toko.*

Abstract— There are many companies use information systems to bestead their companies. Information technology increases the companies' efficiency, simplify the processes, and daily operations automations. Additionally, the companies can get some advantages through the data stored, which produces information and help the companies to make business decisions, which in the end increase the business efficiency, such as sales, purchases, production, and many more. As store, XYZ also looks forward to utilizes the same tools to enhance the company. A custom application tailored to suite XYZ store needs would be beneficiary, and promote the research goal, which is to increase the store operational efficiency. These first step in the methods to do analysis and observations to identify and determine the needs and the business processes understanding. The product of this

research is a dekstop application, based on client-server technology. The technology allows the integration of data in real time centered in a server, that allows the data shared with ease to all client workstations. As a result, the application allows the XYZ store to integrate their datas in real time and helps to promote the store's operational efficiency.

Keywords: *Client-Server, Desktop Application, Real-time, Store Efficiency.*

I. PENDAHULUAN

Dalam beberapa dekade yang lalu, perusahaan-perusahaan dengan taraf internasional melakukan operasional sehari-hari hanya dengan menggunakan kertas dan pensil. Alhasil, tumpukan kertas yang semakin banyak menjadi sebuah masalah yang sangat sulit diatasi, dan menjadi masalah lebih besar seiringnya waktu. Apabila hingga hari ini perusahaan tersebut masih mengandalkan kertas dan pensil, mungkin perusahaan tersebut akan tidak bisa berkompetisi dengan para kompetitornya dan kemungkinan kehilangan berbagai data penting adalah hal yang sangat mungkin terjadi. Agar bisa bertahan perusahaan tersebut harus memiliki sebuah sistem atau fondasi standar-standar yang sangat kuat agar dapat bersaing dengan berkembangnya jaman.

Sebuah sistem berbasis *software* tidak dapat dipungkiri merupakan hal yang sangat penting di era teknologi informasi ini dimana seiring dengan berkembang pesatnya kemajuan teknologi, sistem berbasis *software* sudah menjadi sebuah kebutuhan. Teknologi yang diterapkan pada perusahaan besar dapat diaplikasikan pada sebuah perusahaan pada kelas menengah maupun menengah-ke bawah. Secara sistem mereka memiliki kompleksitas dan fitur-fitur yang berbeda, namun memiliki tujuan yang sama yaitu membuat suatu usaha menjadi lebih efisien dan memberikan perhitungan yang jelas.

Penelitian ini bertujuan merancang sebuah sistem yang memungkinkan Toko XYZ untuk beroperasi secara efisien. Melalui penelitian ini semua data Toko XYZ mulai dari stok, penjualan, pembelian akan dapat diakses secara *real time*. Selain itu, melalui aplikasi yang dihasilkan, dimungkinkan untuk dihasilkan laporan dan hasil analisa yang dapat membantu pengelola toko untuk mengambil keputusan. Melalui penelitian ini aplikasi juga akan membantu pencatatan dan penghitungan transaksi yang terjadi dalam Toko XYZ.

Beberapa manfaat yang nantinya akan dapat diperoleh

¹ Mahasiswa, Jurusan Sistem Informasi Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60219 INDONESIA (telp:082141058000; e-mail: kgozali@student.ciputra.ac.id)

dari hasil penelitian ini. Aplikasi yang dihasilkan dapat mencatat dan mengintegrasikan data-data yang tersimpan secara digital, dan akan sangat membantu dalam penyimpanan data sekaligus *data retrieving*. Selain itu aplikasi dapat digunakan untuk mengintegrasikan data TokoXYZ secara *real time*. Manfaat lain adalah aplikasi akan dapat menghasilkan laporan-laporan yang akan sangat berguna untuk pengambilan keputusan bisnis, membantu memperhitungkan semua keuntungan dan segala sesuatu yang berhubungan dengan uang secara detil dan otomatis, serta membantu pertanggungjawaban semua transaksi yang terjadi pada Toko XYZ.

Penelitian ini dikembangkan dengan tahapan yang dimulai dengan observasi. Observasi yang dilakukan pada Toko XYZ bertujuan untuk menganalisa proses bisnis dari toko serta menentukan kelayakan fitur-fitur yang telah ditentukan, dan menilai apakah fitur tersebut diperlukan atau tidak. Hasil dari analisa proses bisnis akan dikumpulkan dan dirangkum, untuk kemudian membantu penentuan desain dari aplikasi. Setelah semua informasi yang dibutuhkan didapatkan, aplikasi akan dikembangkan dengan tahapan merancang *User Interface*, yang kemudian diuji kepada pengguna untuk mendapatkan umpan balik, kritik, dan saran. Selanjutnya adalah tahapan desain *Database*, yang akan menjadi fondasi dari aplikasi yang akan dibuat. Tahapan akan dilanjutkan dengan mengembangkan kode program untuk kemudian diuji coba. Berikutnya adalah tahap uji coba dan *bug fixing* yang dilaksanakan secara internal yang kemudian diikuti dengan uji coba dengan pengguna untuk mendapatkan umpan balik, sekaligus memastikan aplikasi dapat digunakan dan berjalan dengan *minor bug*. Pada tahapan berikutnya akan dilakukan entry data sebelum kemudian aplikasi dimanfaatkan.

II. LANDASAN TEORI

A. User Interface

Sebuah sistem komputer dengan desain yang bagus secara tidak langsung dapat memberikan perasaan positif yang menggambarkan kesuksesan, kompetensi, kemahiran, dan kejelasan pada pengguna. Ketika sistem didesain secara mahir, *user* akan secara otomatis akan mengikuti kemahiran sistem komputer itu sendiri, akan sangat cepat untuk beradaptasi atau belajar [1].

Menurut Johnson [1], untuk mencapai tujuan diatas, dia merancang prinsip-prinsip dasar yang harus dipenuhi, antara lain:

1. Memberikan *user* kebiasaan yang pasti. Mengenali akan lebih cepat dibandingkan mengingat-ingat, sehingga semua harus memiliki standar agar pengguna dapat mengenali secara cepat. Namun pada hal ini, desain tampilan untuk setiap pengguna tidak harus sama, dapat disesuaikan dengan kebiasaan masing-masing pengguna, tapi rutinitas harus diperhatikan.

Contoh: Tombol “Close” harus selalu di kanan atas meskipun desain setiap tampilan berbeda untuk setiap user, sehingga *user* dapat langsung mengenali fungsi tersebut tanpa harus belajar terlebih dahulu.

2. Mendesain tampilan yang memiliki kecepatan tinggi dan tidak berat secara *workload* adalah salah satu penilaian yang menentukan kenyamanan penggunaan tampilan itu sendiri.
3. Kurangi tulisan atau kata-kata yang terlalu panjang dalam satu tampilan, tampilan yang bagus tidak memerlukan banyak kata-kata untuk menjelaskan, pengguna seharusnya dengan cepat memiliki gambaran bagaimana cara mengoperasikan tampilan.
4. Setiap tampilan harus memiliki tujuan secara spesifik dimana tujuan tersebut harus dapat ditangkap pengguna secara cepat saat pertama melihat.

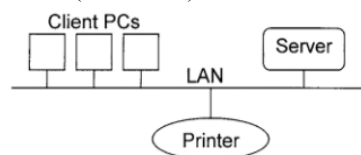
Semua prinsip ini sangat dibutuhkan dalam mendesain *User Interface* yang bagus agar penggunaannya tepat sasaran pengguna.

B. Client Server

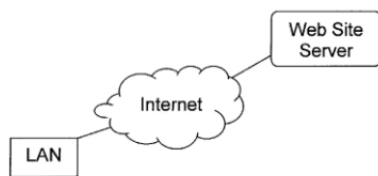
Client-Server adalah arsitektur jaringan yang paling umum dan yang paling sering digunakan untuk komunikasi data saat, dan popularitasnya dapat dilihat dengan ekspansi fenomenal dari *world wide web*. *Client* secara definisi adalah sebuah sistem atau program yang melakukan aktivitas permintaan pada sistem atau yang dapat dipanggil sebagai *server*. Sedangkan *server* adalah sebuah sistem atau program yang menerima permintaan dari satu atau banyak sistem yang disebut dengan *client* [2].

Agar *Client-Server* dapat berjalan, dibutuhkan sebuah *request* (permintaan) dan *return* (pengembalian), dimana *client* melakukan *request* dan kemudian *server* akan memberikan *return* berupa file. Dalam hal ini, semua file atau data tersimpan dalam perangkat yang ditetapkan sebagai *server* dan didistribusikan pada *client* sesuai dengan *request*.

Arsitektur dari model *Client-Server* sendiri dapat dibagi menjadi dua tipe, yaitu berbasis LAN (Gambar 1) dan berbasis Cloud (Gambar 2).



Gambar 1. Arsitektur Client-Server LAN



Gambar 2. Arsitektur Client-Server Cloud

Pada penelitian ini, peneliti akan menggunakan sistem *cloud-based server* dimana *server* tersebut tersimpan secara *online*. Hal ini bertujuan untuk pengembangan aplikasi dengan toko yang memiliki lebih dari 1 cabang.

C. DATABASE

Database adalah sebuah kumpulan koleksi elemen data yang tertata bertujuan untuk memenuhi kebutuhan perusahaan dan didesain secara spesifik untuk memenuhi kebutuhan banyak pengguna. Sistem database mengurangi redundansi data, menyatukan data sebuah organisasi, dan memungkinkan distribusi data kepada semua orang pada sebuah organisasi [3].

Database merupakan sebuah kumpulan data yang secara logika mempunyai relasi antara satu dengan lainnya. Data akan tersimpan pada struktur yang telah didesain.

1) DBMS/RDBMS

Database Management System (DBMS) dan Relational Database Management Systems (RDBMS) merupakan sebuah perangkat lunak yang menampilkan tampilan dimana tampilan tersebut menjembatani pengguna dengan sistem database itu sendiri. DBMS atau RDBMS merupakan sebuah tempat dimana pengguna dapat melakukan interaksi pada database dan dimana interaksi antara pengguna dan database terjadi [4].

Interaksi tersebut dapat dibagi menjadi 4 kelompok [3]: *Data Definition*, *Data Maintenance*, *Data Retrieval*, dan *Data Control*. *Data Definition* adalah interaksi dimana pengguna dapat membuat struktur database baru, menghapus struktur data dari database, sekaligus memodifikasi struktur dari data itu sendiri. *Data Maintenance* merupakan interaksi dimana pengguna dapat melakukan *Data Manipulation Language* (DML). DML sendiri terdiri dari pernyataan *insert*, *update*, dan *delete*. *Data Retrieval* merupakan interaksi dimana pengguna dapat mengekstrak data dari database ke program. *Data Control*. Merupakan interaksi dimana pengguna dapat mengatur dan memantau pengguna pada database, mengatur akses pada database, dan memonitor kinerja database itu sendiri.

2) MySQL

MySQL adalah database berbasis SQL (*Structured Query Language*) yang sudah dikembangkan sejak tahun 1995. MySQL adalah sebuah database bersifat *open-sourced* yang dimana database ini dikembangkan oleh komunitas. Terdapat beberapa alasan yang membuat MySQL adalah database dengan pengguna paling banyak di dunia untuk

pengembangan aplikasi.

Berbagai keuntungan dari penggunaan MySQL adalah bahwa *software* ini menggunakan lisensi *free* dan *Open-Source* yang berarti MySQL dapat dimanfaatkan tanpa harus membeli. Database ini menggunakan perintah SQL dimana yang merupakan salah satu bahasa perintah database yang paling banyak digunakan di dunia. Para penggunaannya juga akan mendapatkan dukungan komunitas yang baik, dimana hal ini sangat krusial dengan database yang berbasis *open source* sehingga dokumentasi penggunaannya lengkap. Database ini juga memiliki kecepatan tinggi dan tersedia pada berbagai sistem operasi seperti Windows, MacOS, Linux, Solaris, dan sebagainya. Penggunaan MySQL juga terjamin karena MySQL dikembangkan oleh salah satu anak perusahaan dari *Oracle Corporation* yang merupakan raksasa di bidang industri database.

D. Secure Shell -2 (SSH-2)

Secure Shell-2 atau yang biasa kita dengar dengan panggilan SSH-2 merupakan sebuah protokol koneksi yang digunakan oleh banyak aplikasi untuk memastikan keamanan dari koneksi antar mesin. SSH-2 sendiri merupakan pengembangan dari banyak protokol lainnya seperti telnet, rlogin, rsh, rdist, rcp, dan SSH-1 yang dimana telah kehilangan popularitasnya karena SSH-2 dapat menyediakan keamanan yang lebih dengan fitur autentikasi dan enkripsi [5]. Tentunya pada awalnya terdapat SSH-1 yang dimana protokol tersebut telah dimodifikasi menjadi lebih baik dan aman, yaitu SSH-2.

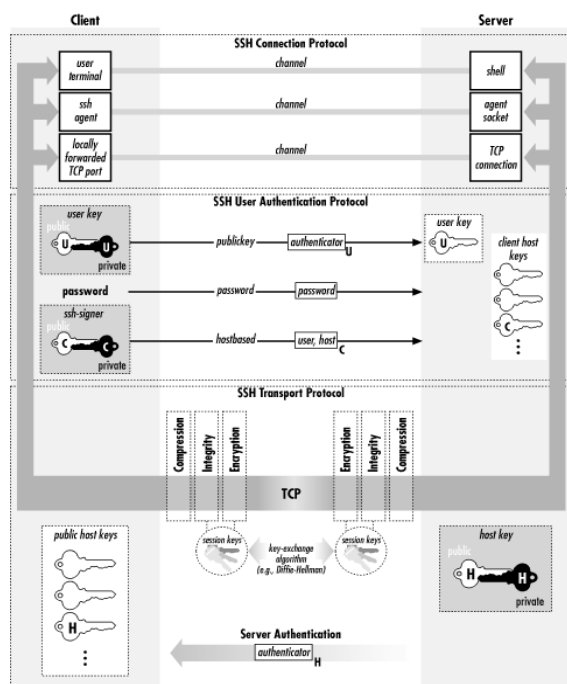
Sampai dengan saat ini, terdapat kunci algoritma terkuat, yaitu *Secret Key* dan *Public Key*. Algoritma *secret key* menggunakan sistem keamanan simetri, yang dimana satu file tersebut dapat mengenkripsi sekaligus mendekripsi. Tipe kedua *public key* menggunakan sistem keamanan asimetri, yang berarti file tersebut hanya digunakan untuk mengenkripsi, sedangkan untuk mendekripsi, sistem menggunakan sebuah *private key* sehingga data akan sangat aman jika *private key* tersebut disimpan dengan hati-hati [5]. Arsitektur dari SSH-2 yang akan digunakan dalam penelitian ini dapat dilihat pada Gambar 3.

E. UML

Unified Modeling Language (UML) merupakan sebuah bahasa grafis untuk memvisualisasikan, menjabarkan, merancang-bangun, dan mendokumentasi artefak sebuah software. UML merupakan sebuah standar dimana pengembang aplikasi dapat menulis *blueprint*, termasuk hal-hal yang masih dalam tahap konseptual seperti proses bisnis dan fungsi sistem, namun juga dapat digunakan untuk hal-hal yang pasti seperti pernyataan bahasa pemrograman, skema database, dan komponen dari perangkat lunak [6].

UML dapat dikategorikan dalam dua model sistem yang berbeda.

1. *Static/Structural View*. Menggambarkan struktur statis sistem menggunakan objek, atribut, operasi, dan relasi. Contoh: Class diagram, Composite Structure Diagram.
2. *Dynamic/Behavioral View*. Menggambarkan sifat dinamis sebuah sistem dengan menunjukkan kolaborasi dan korelasi antara objek. Contoh: Sequence diagram, Activity diagram, State Machine diagram.



Gambar 3. Arsitektur SSH-2

Dalam penelitian ini, peneliti menggunakan tiga tipe diagram yang terdapat pada UML, yakni: Use Case Diagram, Sequence Diagram, dan Activity Diagram. *Use Case* diagram adalah sebuah diagram yang digunakan untuk mendeskripsikan fungsi-fungsi yang tersedia dalam sistem, serta menghubungkannya dengan aktor-aktor, tujuan, dan korelasi antar *use case*. *Sequence diagram* menggambarkan bagaimana seluruh objek dalam sistem berinteraksi dengan sesama, sekaligus mengindikasikan masa hidup dari setiap objek dalam sistem. *Activity Diagram* akan menggambarkan setiap langkah dari proses bisnis secara detail dan menunjukkan cara kerja dan control dari keseluruhan sistem.

F. Income Statement

Income Statement merupakan laporan laba rugi merupakan sebuah laporan akuntansi yang melaporkan keuntungan atau kerugian sebuah perusahaan dalam suatu periode yang ditentukan [7].

Income Statement melakukan penjabaran pada pendapatan terlebih dahulu, dan kemudian diikuti oleh beban-beban perusahaan, dimana hasil dari kedua hal tersebut akan menghasilkan yang dinamakan *net income* atau laba bersih, atau *net loss* atau kerugian bersih. Ketika beban melebihi pendapatan, maka terjadilah *net loss* atau kerugian bersih.

Semua yang dianggap sebagai pendapatan adalah hasil penjualan produk atau jasa diluar pendapatan dana investasi, dan semua yang dianggap beban adalah semua pengeluaran perusahaan selain pembagian dividen oleh perusahaan. Rumus dasar untuk mencari hasil laba bersih dari laporan income statemen adalah sebagai berikut:

Laba Bersih = Pendapatan Kotor – Harga Pokok Penjualan – Beban

Rumus diatas merupakan perhitungan yang digunakan untuk mendapatkan hasil dari laba bersih. Peneliti akan menggunakan metode ini pada penelitian untuk menghasilkan laporan laba rugi.

G. Object-Oriented Programming

Object Oriented Programming (OOP) adalah sebuah paradigma dimana konsepnya adalah berbasis objek, diantaranya dapat berupa data, methods, class, instances, dan lain-lain. Tipe pemrograman berbasis OOP yang paling sering kita jumpai adalah paradigma berbasis class, yang dimana objeknya adalah class itu sendiri dan class memiliki tipe datanya sendiri [8].

Bahasa .NET sendiri seperti C# dan VB.NET tidak secara spesifik didesain untuk menjalankan metode pemrograman Object Oriented Programming, namun tidak dipungkiri bahwa metode tersebut dapat berjalan dengan struktur dari Bahasa .NET sendiri, meskipun beberapa fungsi tetap harus dikerjakan secara *procedural*.

Dengan memanfaatkan class sebagai objek, maka setiap pengulangan yang biasanya kita lakukan pada setiap sub untuk setiap class dapat kita hilangkan, dengan membuat satu class yang dapat digunakan untuk memenuhi semua class dalam satu solusi [8].

III. ANALISIS DAN DESAIN

A. Analisis Sistem

Sebuah sistem yang dikategorikan sebagai sistem yang layak dan selebihnya baik adalah sebuah sistem yang dapat memenuhi serta mengakomodasi kebutuhan penggunaanya. Hal ini dapat dicapai dengan melakukan analisa kebutuhan secara spesifik kepada calon pengguna, dan lebih dalam lagi mengetahui cara kerja bisnis pengguna secara keseluruhan. Peneliti menggunakan metode observasi

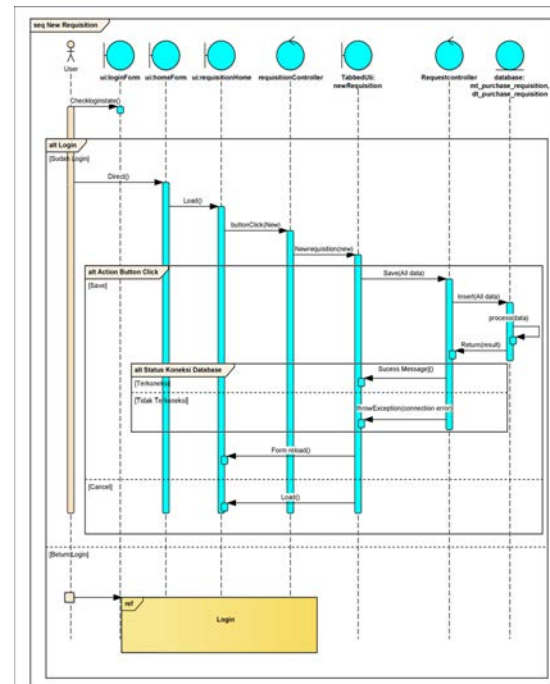
Fitur ini berfungsi untuk proses perencanaan pembelian (Gambar 4).

Dari hasil wawancara-wawancara tersebut, adapun kesimpulan yang diperoleh adalah bahwa Toko XYZ memiliki masalah dalam pencatatan penjualan, pembelian, stok yang tidak pernah cocok dengan aslinya sehingga menyebabkan kerugian. Adapun masalah lain adalah dalam standar prosedur operasional yang tidak dijalankan dengan benar sehingga menyebabkan kerugian-kerugian tersebut.

Observasi lapangan dilakukan dengan berkunjung ke salah satu cabang Toko XYZ bertujuan untuk menganalisa situasi dan operasional kerja dari Toko XYZ. Dengan melakukan observasi lapangan, informasi seperti kebiasaan dan budaya kerja dari Toko XYZ dapat dilihat dan digunakan sebagai referensi dalam pengembangan aplikasi.

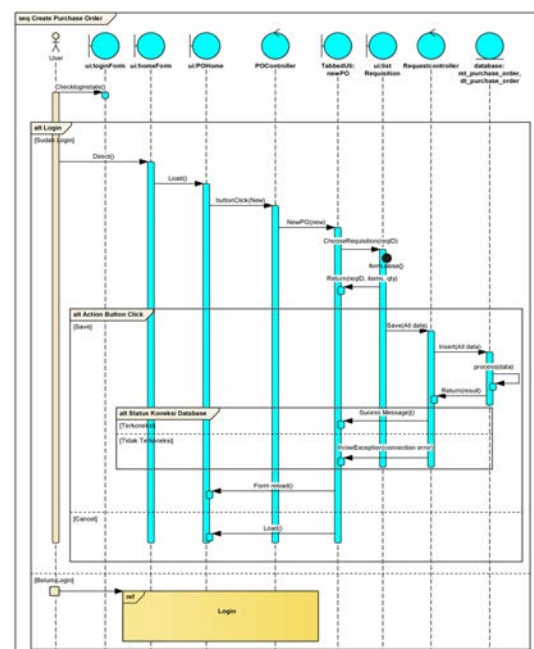
Tahapan Desain Sistem dikerjakan sesuai dengan kebutuhan sistem yang dihasilkan pada tahapan analisis untuk memenuhi kebutuhan-kebutuhan tersebut. Sistem ini diharapkan dapat mengefisiensikan sistem yang sudah berjalan pada industri pelayanan kesehatan khususnya klinik.

Modul Pembelian, digunakan untuk proses pembelian barang. Pembelian terdiri dari beberapa tahap diantaranya adalah:



Gambar 4. *Purchase Requisition*

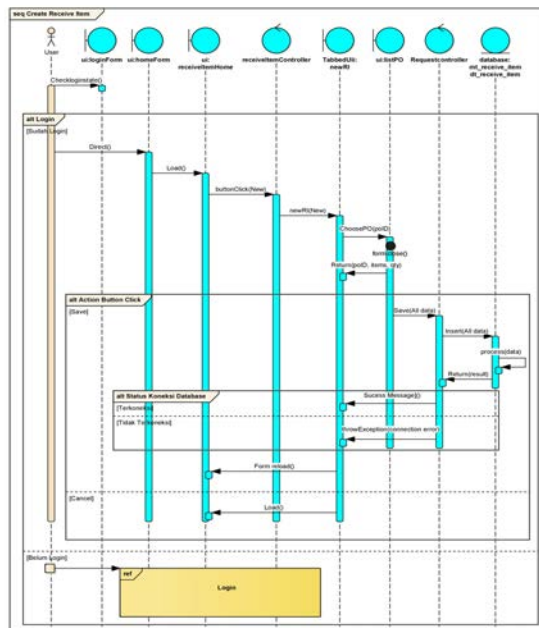
Fitur ini dipergunakan untuk proses *input* pembelian dengan mencantumkan harga berdasarkan *purchase requisition* (Gambar 5).



Gambar 5. Purchase Order

Receive Item

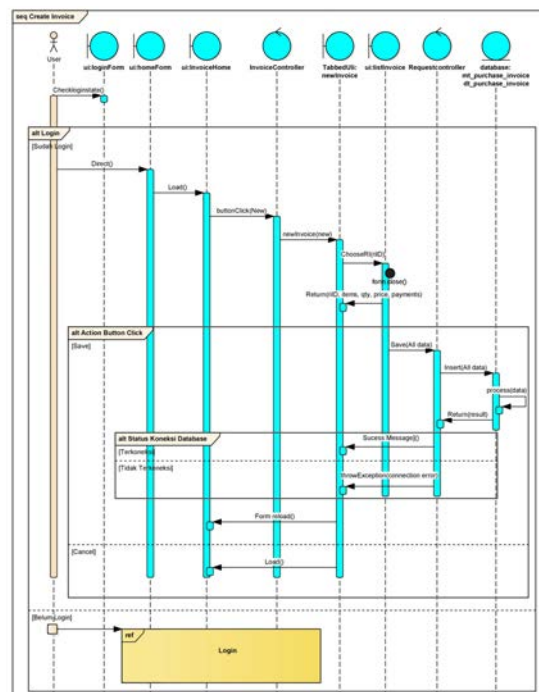
Fitur ini dipergunakan untuk mengkonfirmasi penerimaan barang berdasarkan *purchase order* (Gambar 6).



Gambar 6. Receive Item

Purchase Invoice

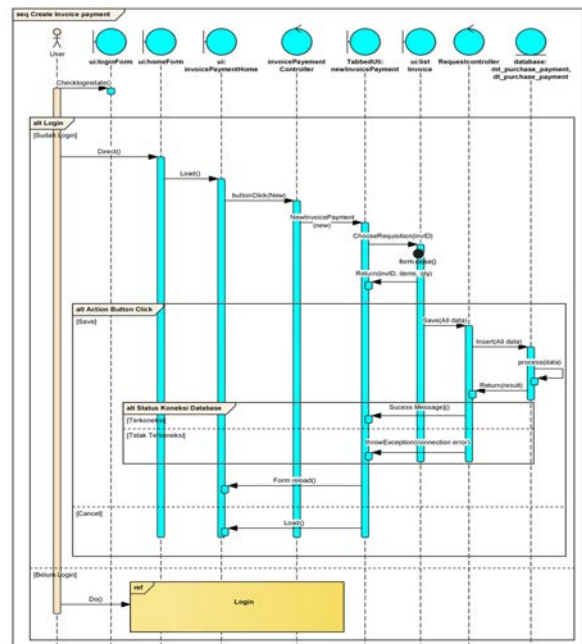
Dalam tahap ini pengguna dapat menerbitkan *invoice* berdasarkan tagihan dari *supplier* (Gambar 7).



Gambar 7. Purchase Invoice

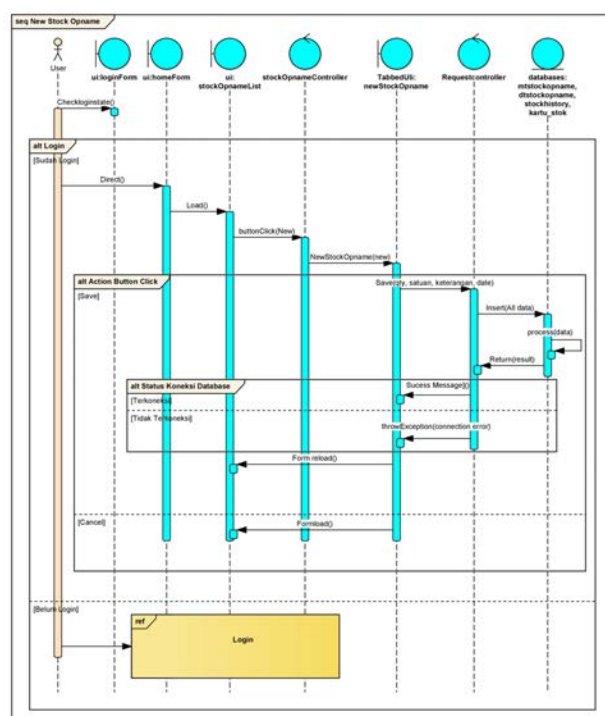
Purchase Payment

Fitur ini dipergunakan untuk proses pembayaran terhadap *invoice* yang telah diterbitkan (Gambar 8).



Gambar 8. Purchase Payment

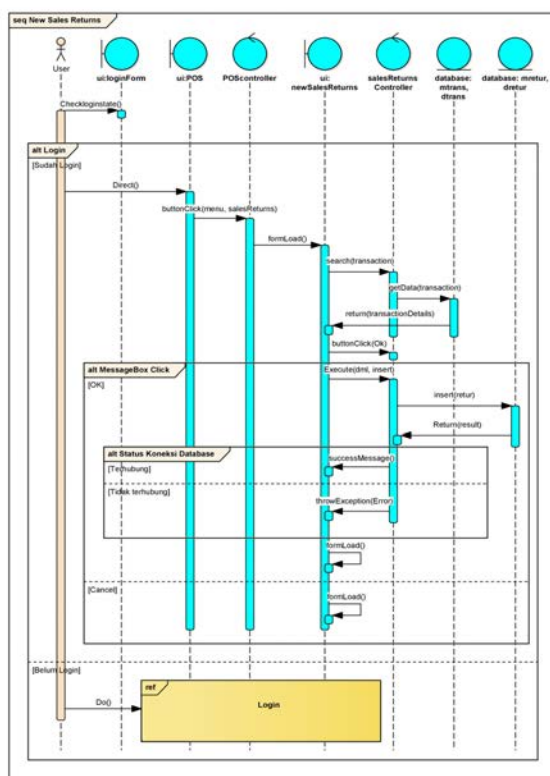
POS (*Point of sale*) Kasir. Fitur ini dapat dimanfaatkan untuk mencatat transaksi penjualan dari Toko XYZ. Penjualan mencakup modul diskon yang dapat dilakukan pada setiap transaksi penjualan (Gambar 9).



Gambar 9. Sales

Stock Opname. Fitur ini berfungsi untuk proses pengecekan stok yang dimana hasil dari *stock opname* dapat dibandingkan dengan stok secara system. Dengan adanya fitur ini, maka hasil selisih stok tersebut dapat dijadikan dasar untuk melakukan sangsi pada departemen-departemen yang bertanggung jawab pada Toko XYZ. Fitur ini merupakan salah satu fitur dari *inventory management* secara keseluruhan.

Retur Penjualan. Fitur ini digunakan untuk proses retur barang sesuai dengan batas hari yang telah ditetapkan. Fitur retur penjualan bertujuan untuk menggantikan barang yang sama terhadap barang rusak. Stok akan otomatis berkurang, dan stok barang rusak akan secara otomatis bertambah. Fitur ini adalah bagian dari *Point of Sales* secara keseluruhan (Gambar 10).



Gambar 10. Retur Penjualan

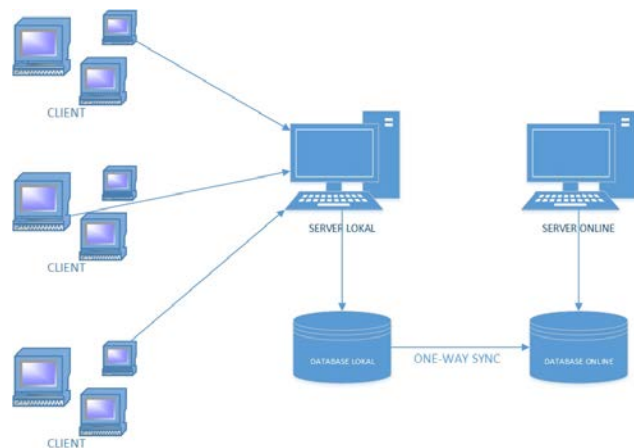
Sinkronisasi Data. Fitur ini merupakan salah satu fitur penting untuk melakukan sinkronisasi data antar server. Aplikasi *Point of Sales* yang dimana berjalan pada server lokal perlu di sinkronisasi ke server online guna mengintegrasikan data.

Backup. Fitur ini dapat dimanfaatkan untuk melakukan proses *backup* dimana sistem akan secara otomatis melakukan *backup* pada setiap skema database dengan interval waktu yang telah ditentukan. Dengan

demikian, data akan tersimpan secara aman dan memungkinkan sistem untuk melakukan *revert* data atau mengembalikan data apabila terjadi permasalahan.

2) Arsitektur Sistem

Rancangan arsitektur pada Gambar 11 merupakan arsitektur yang digunakan untuk koneksi lokal. “Database Lokal” akan kemudian tersinkronisasi dengan “Database Online” secara otomatis atau dapat dilakukan secara manual.



Gambar 11. Arsitektur Sistem

IV.IMPLEMENTASI

Aplikasi yang dikembangkan diimplementasikan dengan menggunakan bahasa pemrograman *Visual Basic / VB* yang dikembangkan oleh Microsoft. Selain itu, dimanfaatkan beberapa *service* yang dimanfaatkan untuk mendukung pengembangan aplikasi ini, yaitu:

- **Vultr**, layanan virtual private server sebagai server online yang digunakan untuk mengintegrasikan data-data Toko XYZ.
- **Bitvise SSH Client**, dipergunakan melakukan akses database menggunakan remote connection. Pada dasarnya aplikasi ini menyediakan tunneling yang mengenkripsi semua data dari komputer menuju server online. Selain aman, aplikasi ini juga membantu menstabilkan koneksi.
- **Serverpilot**, dipergunakan untuk setup server dan untuk maintenance server secara otomatis dan melakukan security update secara otomatis.
- **DbForge**, untuk proses sinkronisasi data dari server lokal menuju ke server online.

Pada proses pengembangan aplikasi, peneliti mengembangkan sebuah framework yang dimanfaatkan sebagai dasar dan acuan, sekaligus menjadi standar dari

kode program yang dikembangkan. Framework bertujuan untuk memudahkan peneliti untuk melakukan perbaikan saat tahap pengembangan maupun pada saat perawatan program

V. PENGUJIAN

A. Skenario Pengujian

Pengujian dilakukan dengan menggunakan skenario yang terdiri dari 6 tahapan atau langkah seperti dijabarkan berikut ini:

1. Login

Pada tahapan ini pengguna mencoba melakukan login melalui form login. *User* memasukan *username* dan *password* dan menekan tombol *Login* akan secara langsung terdiksi ke *form home*. Pada tahap ini, sistem akan membatasi level akses pengguna berdasarkan *setting* level akses setiap pengguna yang telah diatur sebelumnya.

2. Pembelian

Pada tahap ini, pengguna memulai dengan melakukan *purchase requisition*, dan dilanjutkan dengan menyimpan dan memiliki opsi untuk melakukan *print*. Setelah berhasil melakukan *purchase requisition*, pengguna melanjutkan dengan membuat *purchase order* dengan merefrensikan *purchase requisition*. Pada tahap ini, pengguna dapat melakukan *input* harga sesuai dengan yang disetujui dengan *supplier*. Pengguna melanjutkan melakukan penerimaan barang/*receive item* yang dimana pengguna mencatat penerimaan barang sesuai dengan merefrensikan *purchase order*. Setelah berhasil menyimpan, stok pada sistem akan otomatis bertambah. Pengguna pada tahap ini hanya perlu mengkonfirmasi jumlah penerimaan barang. Dilanjutkan dengan menerbitkan *invoice* dimana pengguna melakukan *input* kembali terhadap harga setiap barang sesuai dengan tagihan *supplier*. Pada tahap ini, harga yang diberikan oleh *supplier* dapat berubah sesuai kondisi sehingga perlu dilakukan kembali *input* terhadap harga setiap barangnya. Setelah *invoice* terbit, maka pengguna dapat melanjutkan dengan melakukan pembayaran pada *invoice* tersebut. Pembayaran dapat dilakukan dengan beberapa metode pembayaran, diantaranya adalah; *Cash*, *Giro*, *Cek*, dan *Transfer bank*. Pembayaran akan otomatis mengurangi saldo akun secara otomatis pada sistem.

3. Transaksi Penjualan Baru

Pengguna memulai dengan menambahkan barang dan menambahkan kuantitasnya sesuai dengan kuantitas penjualan. Pengguna dapat menambahkan diskon pada setiap barang yang ditambahkan sebelumnya. Setelah selesai memilih barang, pengguna dapat melanjutkan dengan melakukan pembayaran dan memilih metode

pembayaran. Metode pembayaran diantara lain adalah; *Cash*, *Credit Card*, *Debit Card*. Pembayaran menggunakan metode selain *Cash* akan secara otomatis memenuhi nominal pembayaran. Setelah memilih metode pembayaran, pengguna dapat melanjutkan dengan menambahkan diskon pada transaksi. Dengan melanjutkan, nota penjualan akan secara otomatis tercetak dan memberikan kode unik pada setiap transaksi yang dimana nantinya dapat digunakan sebagai bukti pembelian untuk melakukan retur.

4. Stock Opname

Pada *stock opname* pengguna dapat langsung memilih menu *stock opname* yang telah tersedia dan mengisi jumlah kalkulasi baru pada setiap barang dan menekan tombol *save*. Barang yang muncul pada *stock opname* hanya barang yang dimana sebelumnya telah dibeli pada modul pembelian, dikarenakan secara akuntansi, dibutuhkan nilai persediaan pada setiap barang sehingga barang yang belum dibeli tidak dapat diubah jumlahnya. Selain itu, secara sistem, cara satu-satunya barang dapat tercatat pada sistem adalah dengan melakukan penerimaan barang/*receive item* pada modul pembelian.

5. Retur Penjualan

Pengguna dapat memulai melakukan retur penjualan dengan melakukan pencarian terhadap nota penjualan. Batas pengembalian barang/retur dapat diubah berdasarkan *setting* yang telah disediakan. Setelah berhasil melakukan pencarian terhadap nota, pengguna melanjutkan dengan melakukan memilih barang dan melakukan *input* jumlah barang. Pengguna melanjutkan dengan menekan tombol *save*. Barang akan secara otomatis bertambah pada stok gudang rusak, dan secara otomatis akan juga mengurangi stok gudang karena retur bekerja dengan cara menukarkan barang.

6. Melakukan Sinkronisasi Data

Pengguna hanya perlu menekan tombol *sync* untuk melakukan sinkronisasi data. Cara sinkronisasi data bekerja adalah dengan membandingkan skema asal dan skema tujuan secara *field* serta *record* dengan membandingkan *primary key* pada tabel tujuan. Tabel asal akan melakukan *input* ke tabel tujuan dimana *primary key* pada tabel tujuan belum ada. Setelah itu, sistem akan mengubah semua *record* yang berbeda pada tabel tujuan.

B. HASIL SURVEY

Berdasarkan hasil survei *user acceptance test* (UAT (TABEL 1), dapat disimpulkan bahwa *user* cukup puas. Semua *user* merasa bahwa aplikasi sangat mudah digunakan dan hampir semua merasa bahwa aplikasi sangat nyaman untuk digunakan. Namun dalam perhitungan keuangan, hanya 64% *user* merasa bahwa *user*

belum begitu yakin terhadap hasil laporan keuangan pada aplikasi. Hal ini disebabkan oleh kurangnya waktu untuk melakukan pengujian terhadap modul keuangan. Menurut sebagian besar dari *user*, aplikasi dapat membantu mengefisienkan operasional sehari-hari, sehingga tujuan penelitian dapat dikatakan berhasil dicapai. Aplikasi dapat dilihat mampu mengintegrasikan data secara *real-time* menurut semua *user* yang telah mencoba dan akurasi data-data tersebut dapat terbilang lebih dari cukup sehingga *user* tersebut merasa menggunakan aplikasi lebih menguntungkan dibandingkan proses bisnis konvensional.

Tabel 1 Penilaian Hasil Survey

No	Pernyataan	Nilai
1	Aplikasi mudah digunakan	100%
2	Aplikasi nyaman digunakan	96%
3	Aplikasi dapat menghasilkan laporan-laporan yang bermanfaat serta berguna untuk digunakan sebagai dasar pengambilan keputusan	76%
4	Aplikasi dapat membantu perhitungan keuangan	64%
5	Aplikasi sangat membantu dalam mengefisienkan operasional sehari-hari serta mempercepat operasional	88%
6	Aplikasi dapat mengintegrasikan data-data secara real time	100%
7	Aplikasi dapat menyediakan data-data yang akurat	96%
8	Manfaat-manfaat menggunakan aplikasi lebih menguntungkan dibandingkan proses bisnis konvensional	96%

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian aplikasi yang telah dilakukan, maka berikut ini adalah hal-hal yang dapat disimpulkan:

- Aplikasi mampu membantu Toko XYZ menjadi lebih efisien, dengan meninggalkan cara-cara konvensional.

- Aplikasi mampu mengintegrasikan data-data Toko XYZ secara real time.
- Aplikasi mampu memberikan data-data dan laporan-laporan yang dapat membantu pengambilan keputusan berdasarkan hasil survey user acceptance testing (UAT)
- Aplikasi dapat membantu memperhitungkan dan mempertanggungjawabkan semua transaksi yang terjadi pada Toko XYZ.

B. Saran

Adapun saran untuk pengembangan aplikasi ke depannya:

- Pengembangan *mobile reporting* dan *mobile dashboard* sehingga pemilik dapat memantau Toko XYZ secara *real time*.
- Pengembangan modul *mobile application* untuk melakukan pesanan secara *online*. Pemilik Toko XYZ telah menunjukkan ketertarikan ke jalur tersebut.
- Menambahkan modul akuntansi lebih lengkap, sehingga semua asset, hutang, piutang usaha dapat dikalkulasi.

VII. DAFTAR PUSTAKA

- [1] J. Johnson, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, Waltham: Morgan Kaufmann, 2014.
- [2] D. Hanson, "The Client/Server Architecture," in *Server Management*, Boca Raton, CRC Press LLC, 2000, pp. 1-20.
- [3] S. Lee, "Unified Modeling Language (UML) for Database Systems and Computer Applications," *International Journal of Database Theory and Application*, pp. 157-163, 2012.
- [4] R. J. The Unified Modeling Language Reference Manual Second Edition, Boston: Addison-Wesley, 2004.
- [5] S. Licking, "SSH," San Jose, 2005.
- [6] B. Padmanabhan, "Unified Modeling Language Overview," Kansas, 2012.
- [7] J. Weygandt, "Intermediate Accounting," in *Intermediate Account Volume 1*, Denver, John Wiley & Sons, Inc., 2011, pp. 142-166.
- [8] E. Kindler, "Object-oriented Simulation of Systems With Sophisticated Control," *International Journal of General Systems*, pp. 313-343, 2011.