BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisa Masalah

Masalah utama dari tugas akhir ini yaitu melakukan implementasi Metode Enkripsi Rivest Code 6 (RC6) dan Triple DES (3DES) pada *output/ backup data* yang berupa *text file* untuk konsolidasi data inventory dari beberapa outlet ke *data center* di kantor pusat (HQ), khususnya pada perusahaan F&B. Enkripsi tersebut sangat diperlukan mengingat data inventory suatu perusahaan bersifat rahasia (*confidential*). Konsolidasi data dari outlet ke kantor pusat (HO) menggunakan *File Transfer Protocol* (FTP)

3.3.1 Analisis Algoritma RC6

Secara lengkap, algoritma RC6 ditulis sebagai berikut :

RC6-w/r/b

Algoritma RC6 merupakan algoritma yang memiliki parameter sebagai input, parameter tersebut terdiri dari panjang word (w), jumlah iterasi (r) dan panjang kunci (b). Dengan adanya parameter tersebut dalam melakukan implementasi algoritma RC6 dapat disesuaikan kebutuhan yang diinginkan, apabila yang diinginkan adalah algoritma enkripsi yang cepat maka jumlah iterasi yang dilakukan dapat dikurangi, walaupun dengan mengurangi jumlah iterasi dapat mengurangi kekuatan algoritma RC6. Sebaliknya, jika menambahkan jumlah iterasi, maka akan dihasilkan algoritma kriptografi yang kuat, namun membutuhkan waktu yang lama dalam melakukan enkripsi. Dan untuk menyesuaikan implementasi dengan mesin yang menjadi target implementasi dapat dilakukan dengan melakukan perubahan pada panjang *word*. Dengan karakteristik seperti ini, algoritma RC6 akan dapat diimplemetasikan secara fleksibel pada berbagai jenis platform dan mesin dengan prosesor yang beragam.

Panjang *word* (b) yang akan digunakan pada perangkat lunak yang akan dibangun adalah 32 bit, berarti panjang blok yang akan digunakan adalah 128 bit, Panjang word tersebut dipilih karena pada dasarnya algoritma RC6 memang ditujukan untuk menggunakan panjang word tersebut dan mudah untuk diimplementasikan karena algoritma RC6 menggunakan operasi *integer modulo* sebesar panjang *word* dan tipe integer sebagian besar *compiler* yang beredar sekarang ini memiliki panjang 32 bit. Untuk jumlah rotasi (r), akan terdapat menu pilihan pada perangkat lunak yang akan dibangun untuk menentukan jumlah rotasi yang akan digunakan. Panjang kunci yang akan digunakan akan beragam dari 0 sampai 255 karakter, hal ini sudah menjadi sifat algoritma RC6 yang menerima kunci dengan panjang kunci yang beragam.

3.3.2 **Analisis Algoritma 3DES**

3DES (*Triple Data Encryption Standard*) merupakan suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama, hanya pada 3DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali. 3DES memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES.

Mengenkripsi plaintext dengan satu kunci DES dan kemudian mengenkripsinya lagi dengan kunci DES kedua sama tidak amannya dengan enkripsi menggunakan satu kunci DES. Kelihatan bahwa jika kedua kunci memiliki n bit, serangan brute force untuk mencoba semua kunci yang mungkin akan membutuhkan 2^n x 2^n x 2^n kombinasi yang berbeda. Merkle dan Hellman menunjukkan bahwa plaintext yang diketahui, serangan *Man in the Middle* dapat memecahkan enkripsi ganda pada 2_{n+1} kali percobaan. Tipe serangan ini dicapai dengan mengenkripsi dari akhir, dan dekripsi dari yang lainnya, dan membandingkan hasilnya di tengah. Karena itu, Triple DES digunakan untuk mendapatkan enkripsi yang lebih kuat.

Triple DES mengenkripsi pesan tiga kali. Enkripsi ini dapat dicapai dengan beberapa cara. Sebagai contoh, pesan dapat dienkripsi dengan kunci 1, dekripsi dengan kunci 2 (pada dasarnya enkripsi yang lain), dan dienkripsi lagi dengan kunci 1:

[E{D(M,K1),K2},K1]

Enkripsi Triple DES dengan cara ini dikenal sebagai DES-EDE2. Jika tiga enkripsi dijalankan menggunakan dua kunci, dikenal sebagai DES-EEE2:

[E{E[E(M,K1)],K2},K1]

Sama dengan diatas:

[E{E[E(M,K1)],K2},K3]

menggambarkan enkripsi triple DES-EEE3 dengan tiga kunci yang berbeda. Enkripsi ini adalah bentuk yang paling aman dari Triple DES.

Untuk mendapatkan plainteks tanpa mengetahui kuncinya, jumlah kombinasi kemungkinan kunci yang harus dicoba adalah sebanyak $3{,}741 \times 10^{50}$ kali.

Waktu yang diperlukan untuk mencoba seluruh kemungkinan kunci oleh serangan *brute force* adalah $1{,}183 \times 10^{43}$ tahun.

3.3.3 **Analisis Text File Sebagai Output Data**

Output data berupa text file karena ukuran text file relative kecil dan dapat dikompres (rar) hingga beberapa puluh kali lebih kecil. Sebagai contoh ukuran normal text file sebesar 1.16 MB, setelah dikompress (rar) ukurannya tinggal 31KB. Text file yang digunakan bertipe XML (*eXtensible Markup Language*) karena XML mempu menyimpan data secara ringkas dan mudah diatur. Kelebihan lain yang dimiliki XML adalah bahwa informasi bisa di pertukarkan dari satu system ke system lain yang berbeda platform. Akan tetapi yang paling penting, XML mudah di-import ke dalam *database* karena hampir semua *database* sudah menyediakan function untuk import/ export data dari dan ke file XML. Begitu juga jika import data dilakukan secara manual tidak memerlukan parsing kata yang rumit karena struktur XML sudah terstandar sehingga banyak bahasa pemrograman yang sudah menyediakan *class* atau *function* khusus untuk menangani XML.

3.3.4 Analisis Penerapan Enkripsi Data Inventory

Algoritma RC6 merupakan algoritma yang sederhana, fungsi yang digunakan merupakan fungsi yang sederhana dan hanya mengandalkan prinsip *iterated cipher* untuk keamanan. Sehingga, dalam implementasi untuk melakukan enkripsi data inventory tidak diperlukan adanya penanganan khusus, yang perlu diperhatikan dalam melakukan implementasi algoritma RC6 pada data inventory adalah semakin besar jumlah rotasi pada algoritma RC6, maka tingkat keamanan akan semakin baik, namun waktu yang diperlukan untuk melakukan enkripsi dan dekripsi akan semakin besar. Hasil dari enkripsi RC6 selanjutnya akan dienkripsi lagi menggunakan metode 3DES.

3.3.5 Analisis Pengiriman Data Inventory

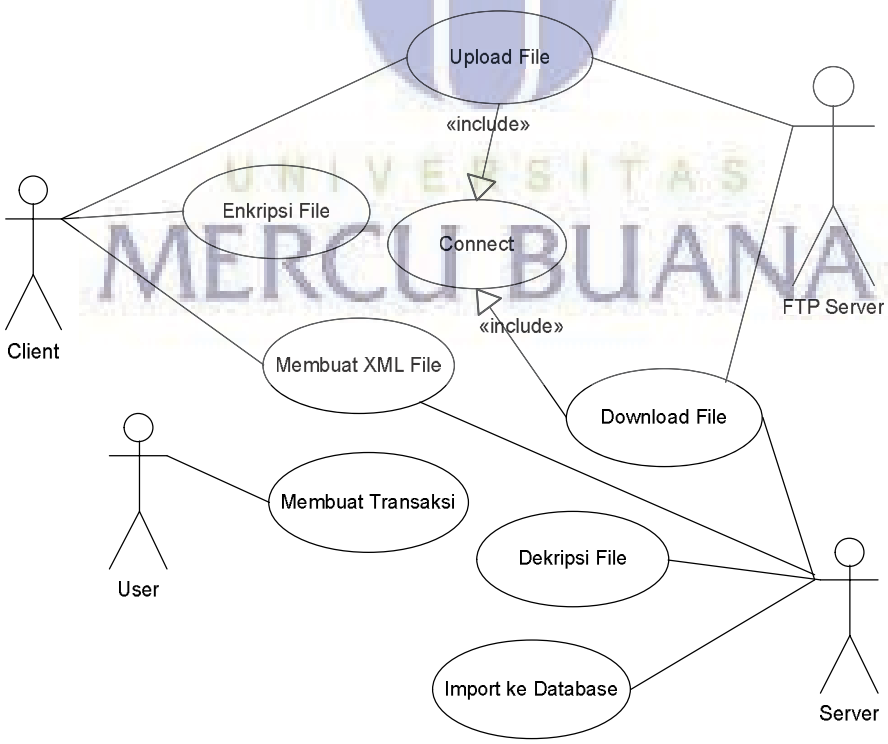
Untuk pengiriman data inventory dari outlet ke kantor pusat (HO) digunakan FTP (*File Transfer Protocol*). FTP adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (file) komputer antar mesin-mesin dalam sebuah internetwork. FTP menggunakan protokol *Transmission Control Protocol* (TCP) untuk komunikasi data antara klien dan server, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum transfer data dimulai. Sebelum membuat koneksi, port TCP nomor 21 di sisi server akan "mendengarkan" percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai port pengatur (control port) untuk (1) membuat sebuah koneksi antara klien dan server, (2) untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada server dan juga (3) mengembalikan respons server ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka server akan mulai membuka port TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mentransfer data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

3.2 Perancangan Sistem

Tujuan dari perancangan sistem yaitu untuk **mengurangi fungsi-fungsi yang terduplikasi**, baik dalam hal tujuan, operasi, data, form-form, serta laporannya. Juga menghindarkan adanya prosedur-prosedur yang tak perlu. Juga dalam hal aliran data, laporan-laporan, dan fungsi-fungsi. Analisa yang dilakukan adalah memahami dan mencari permasalahan yang terjadi pada proses menyimpan transaksi inventory kemudian mengenkripsi, *upload/ download* FTP, dekripsi dan *import* ke *database server*.

3.3.1 Perancangan *Use Case* Diagram

Use case merupakan gambaran umum dari rancangan sistem yang akan dibuat. Dalam perancangan *use case* system inventory ada 4 *actor* yang terlibat yaitu *user, client, server* dan FTP Server. *client, user* dan *FTP server* disini merupakan *actor* yang berupa sistem.



Gambar 3.1 *Use Case Diagram*

1. ***Actor Specification***

   a. User : adalah seseorang yang menggunakan software
   b. Client : adalah sistem pada sisi client
   c. Server : adalah sistem pada sisi server
   d. FTP Server : adalah sistem untuk transfer file menggunakan Protokol Internet

2. ***Use Case Spesification***

   a. USE CASE : MEMBUAT TRANSAKSI

   Overview

   Tujuan utama *use case* ini adalah melihat dan memanipulasi (tambah, ubah dan hapus) data inventory

   Primary Actor

   User

   Secondary Actor

   None

   Starting Point

   Use case ini dimulai ketika aktor hendak membuat transaksi inventory

   Ending Point

   Data transaksi berhasil disimpan atau gagal

   Measurable Result

   Data transaksi berhasil disimpan

Flow of Events

Aktor melakukan transaksi dengan mengisi kuantiti/ jumlah item barang

Alternative Flow of Events

None

Use Case Extensions

None

Outstanding Issues

None

b. USE CASE : ENKRIPSI FILE

Overview

Tujuan utama *use case* ini adalah enkripsi backup file menggunakan algoritma 2 tingkat yaitu RC6 dan 3DES

Primary Actor

User

Secondary Actor

None

Starting Point

Use case ini dimulai ketika aktor hendak enkripsi file

Ending Point

Data transaksi berhasil di enkripsi 2 tingkat atau gagal

Measurable Result

Data transaksi berhasil dienkripsi

Flow of Events

Aktor melakukan menerima file text XML dan dienkripsi menggunakan metode RC6, kemudian cipher text RC6 dienkripsi lg menggunakan 3DES

Alternative Flow of Events

None

Use Case Extensions

None

Outstanding Issues

None

c. USE CASE : UPLOAD FILE

Overview

Tujuan utama *use case* ini adalah mengunggah data ke FTP Server

Primary Actor

User

Secondary Actor

None

Starting Point

Use case ini dimulai ketika aktor hendak mengunggah data ke FTP Server

Ending Point

Data transaksi berhasil diunggah ke FTP Server

Measurable Result

Data transaksi berhasil diunggah

Flow of Events

Aktor melakukan cek koneksi internet, jika ada koneksi unggah data ke FTP Server.

Alternative Flow of Events

None

Use Case Extensions

None

Outstanding Issues

None

d. USE CASE : DOWNLOAD FILE

Overview

Tujuan utama *use case* ini adalah mengunduh data dari FTP Server

Primary Actor

User

Secondary Actor

None

Starting Point

Use case ini dimulai ketika aktor mengecek FTP Server dan ada file yang harus di *download*

Ending Point

Data transaksi berhasil di unduh dari FTP Server

Measurable Result

Data transaksi berhasil diunduh

Flow of Events

Aktor melakukan cek koneksi internet, jika ada koneksi cek FTP Server apakah ada file yang diterima. Jika ada unduh file dan simpan di lokal komputer.

Alternative Flow of Events

None

Use Case Extensions

None

Outstanding Issues

None

e. USE CASE : DEKRIPSI FILE

Overview

Tujuan utama *use case* ini adalah dekripsi file hasil unduhan dari FTP Server

Primary Actor

User

Secondary Actor

None

Starting Point

Use case ini dimulai ketika aktor berhasil mengunduh file dari FTP Server

Ending Point

Data transaksi berhasil di dekripsi atau gagal

Measurable Result

Data transaksi berhasil di dekripsi

Flow of Events

Aktor melakukan dekripsi file dengan metode 3DES kemudian RC6 menggunakan kunci yang sama seperti pada waktu enkripsi.

Alternative Flow of Events

None

Use Case Extensions

None

Outstanding Issues

None

f. USE CASE : IMPORT DATABASE

Overview

Tujuan utama *use case* ini adalah import file hasil dekripsi ke dalam *database*

Primary Actor

User

Secondary Actor

None

Starting Point

Use case ini dimulai ketika aktor menerima file hasil dekripsi

Ending Point

File hasil dekripsi berhasil di import ke dalam *database* atau gagal

Measurable Result

File hasil dekripsi berhasil di import ke dalam *database*

Flow of Events

Aktor membaca file hasil deskripsi kemudian di import ke dalam *database*.

Alternative Flow of Events

None

Use Case Extensions

None

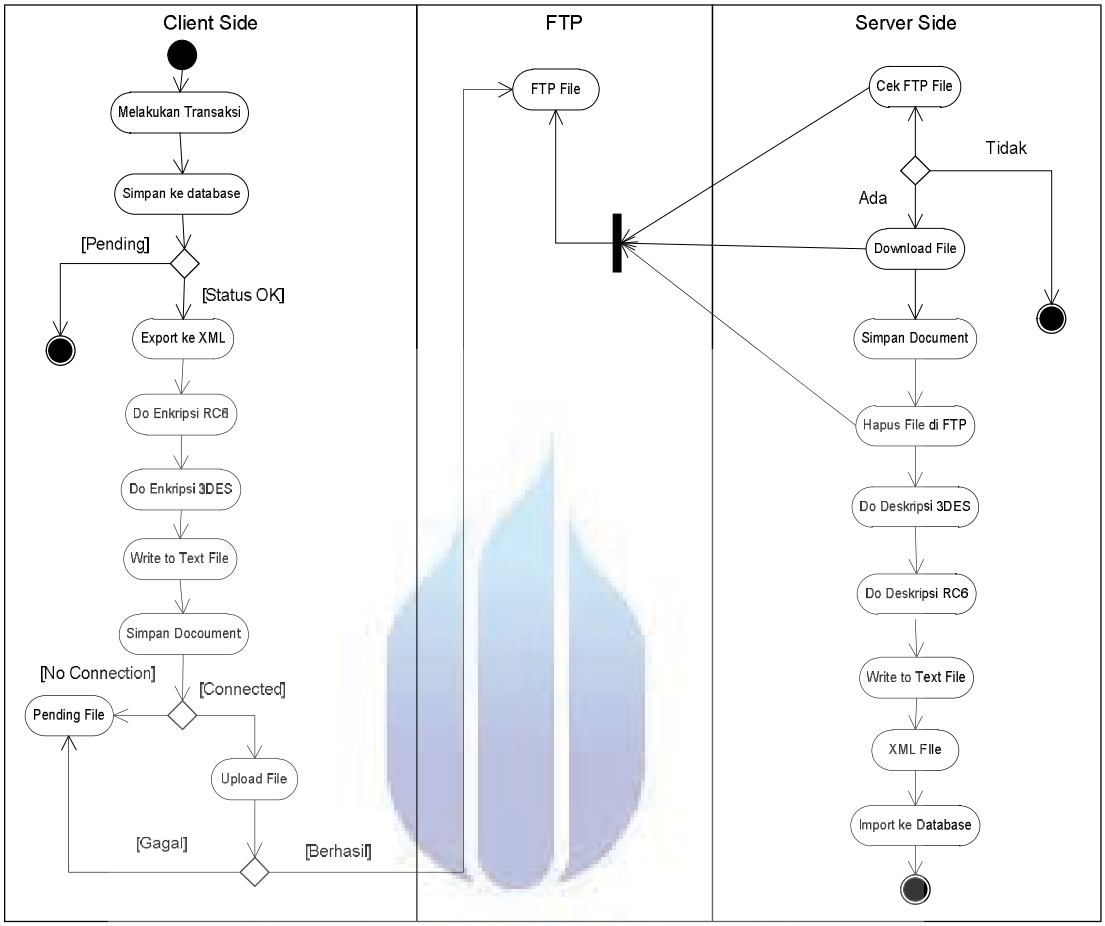
Outstanding Issues

None

3.3.2 Perancangan *Activity* Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

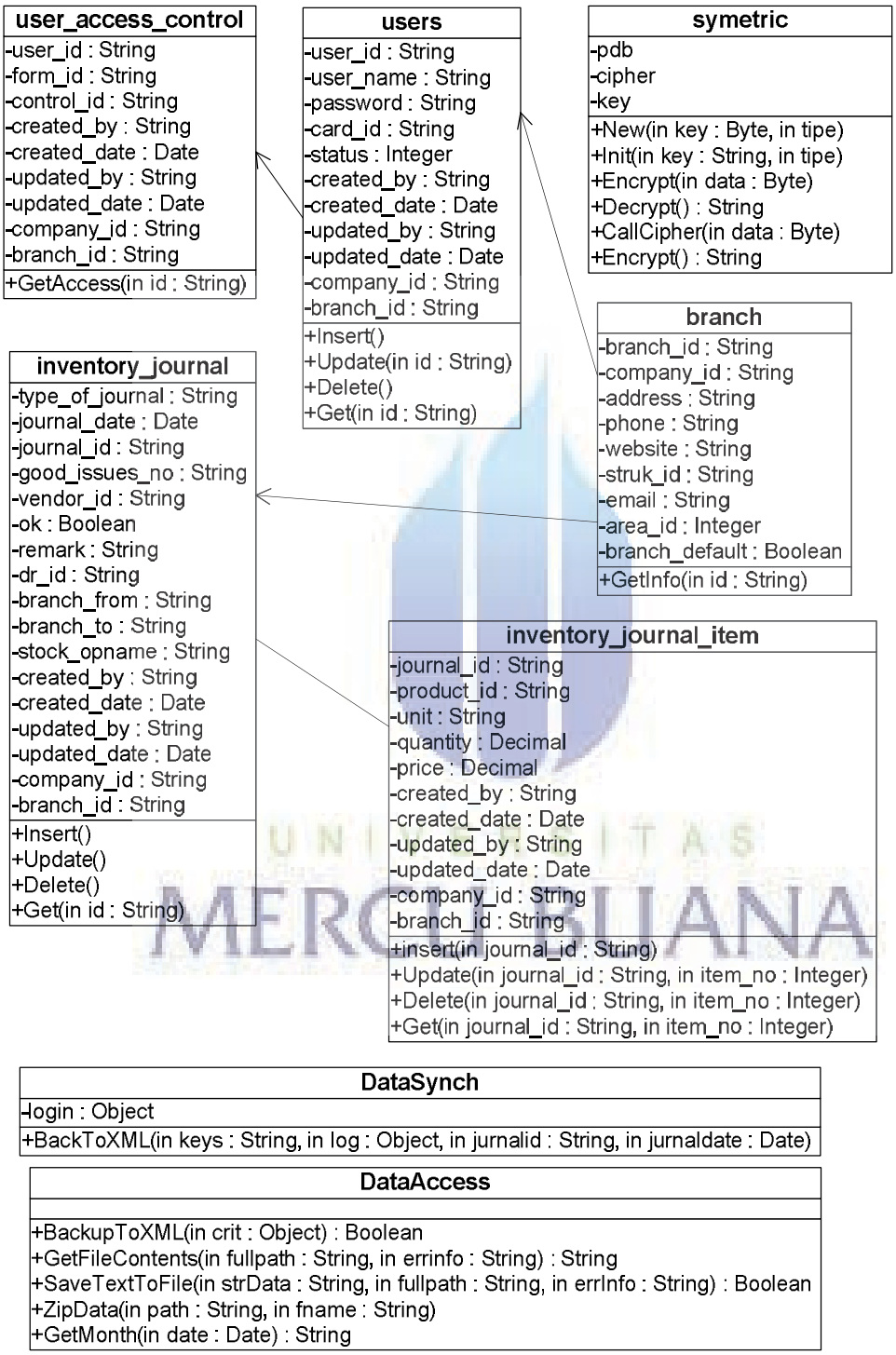
Gambar 3.6 menjelaskan tentang alur kerja sistem untuk melakukan konsolidasi data dari outlet ke HO. Setelah *user* di outlet melakukan transaksi, data akan disimpan di *database* local. Untuk konsolidasi data ke HO, data akan diexport ke XML lalu dienkripsi 2 tingkat. Enkripsi yang pertama, *plain text* yang berupa XML akan di enkripsi menggunakan metode RC6. Dari output enkripsi RC6 selajutnya di enkripsi kembali menggunakan menggunakan metode 3DES. Hasil dari enkripsi tersebut akan disimpan di computer local dan dianggap sebagai *file pending.* Secara periodik, sistem akan mengirim *file pending* tersebut ke FTP Server tentunya setelah mengecek ada tidaknya koneksi internet. Pada sisi kantor pusat (HO), sistem secara periodic akan mengecek ada tidaknya file yang masuk ke FTP server. Jika ada file yang masuk, file akan di *download* dan disimpan di komputer lokal. File yang sudah di *download* akan langsung di hapus dari FTP *Server*. Satu per satu file yang di *download* akan di proses untuk di deskripsi 2 tingkat, yaitu 3DES kemudian RC6. Setelah *plain text* di dapat, yaitu berupa format XML akan di simpan/ *import* ke dalam database. Maka proses konsolidasi data dari outlet ke HO selasai dilakukan.



Gambar 3.2 Konsolidasi Data *Activity* Diagram

3.3.3 Perancangan *Class* Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu *system*.

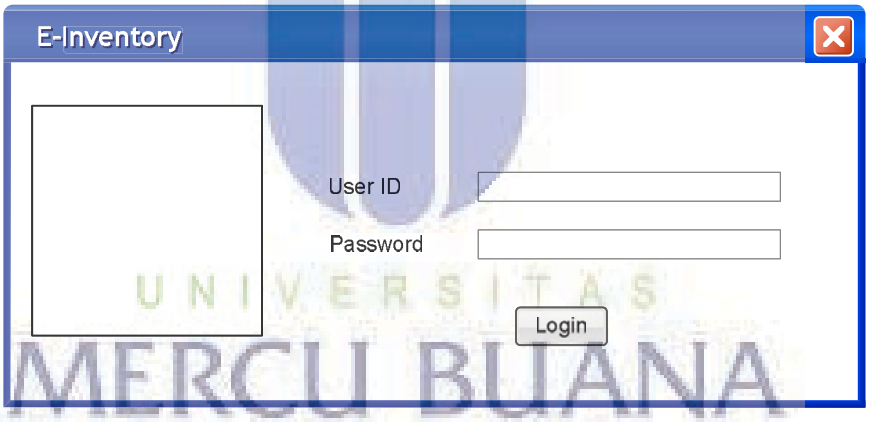


Gambar 3.3 Inventory *Class* Diagram

3.3 Perancangan Antarmuka

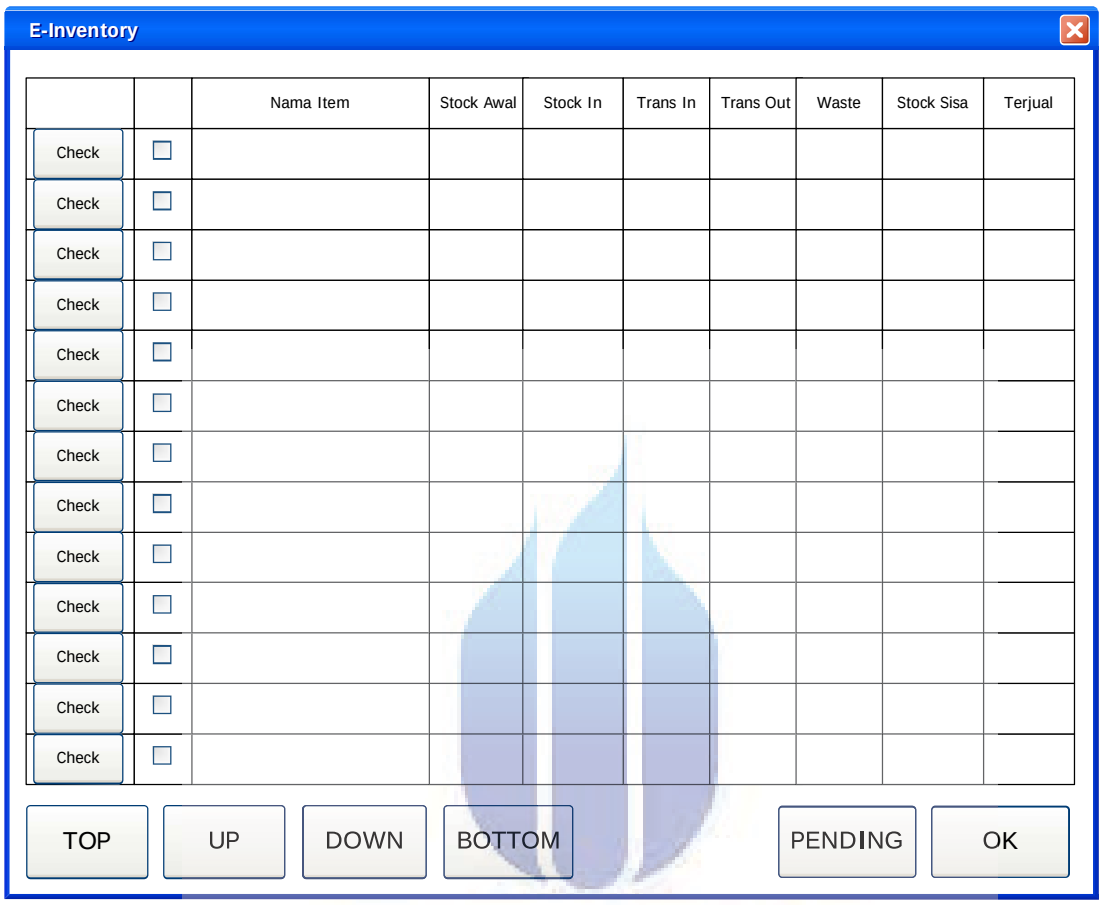
Merancang antarmuka merupakan bagian yang paling penting dari merancang sistem. Biasanya hal tersebut juga merupakan bagian yang paling sulit, karena dalam merancang antarmuka harus memenuhi tiga persyaratan: sebuah antarmuka harus sederhana, sebuah antarmuka harus lengkap, dan sebuah antarmuka harus memilki kinerja yang cepat. Alasan utama mengapa antarmuka sulit untuk dirancang adalah karena setiap antarmuka adalah sebuah bahasa pemrograman yang kecil: antarmuka menjelaskan sekumpulan objek-objek dan operasi-operasi yang bisa digunakan untuk memanipulasi objek.

3.3.1 *Form Login.* Fom *login* digunakan oleh *user* dan admin untuk masuk ke dalam sistem inventory. Tampilan halaman *login* dapat dilihat pada Gambar 3.11.



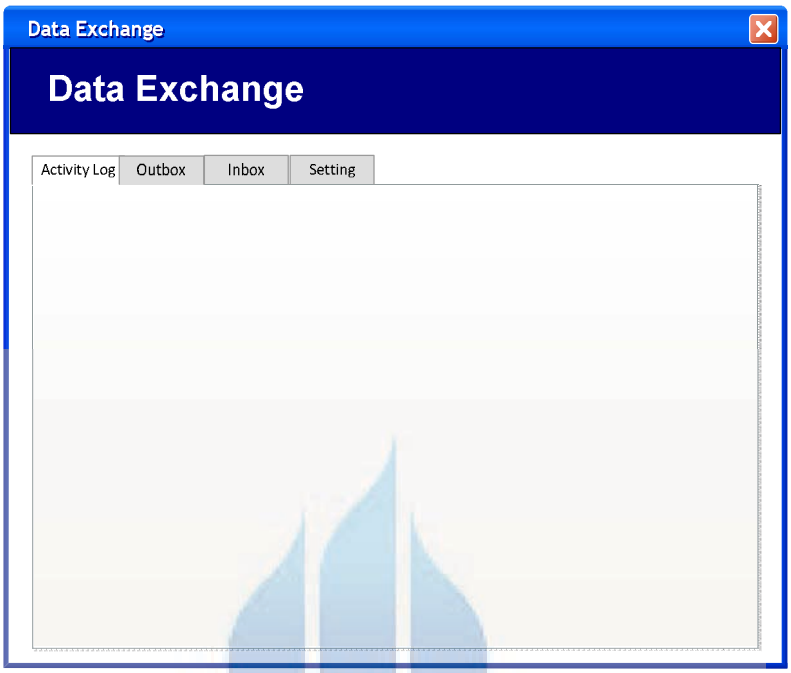
Gambar 3.4 Desain Halaman *Login*

3.3.2 *Form Transaksi.* Form transaksi digunakan user untuk melakukan transaksi inventory. Pada form ini ada 5 tombol yaitu *top, up, down, bottom, pending* dan OK. Pada saat klik tombol OK inilah proses backup data dan enkripsi dilakukan. Tampilan form transaksi dapat dilihat pada Gambar 3.12.

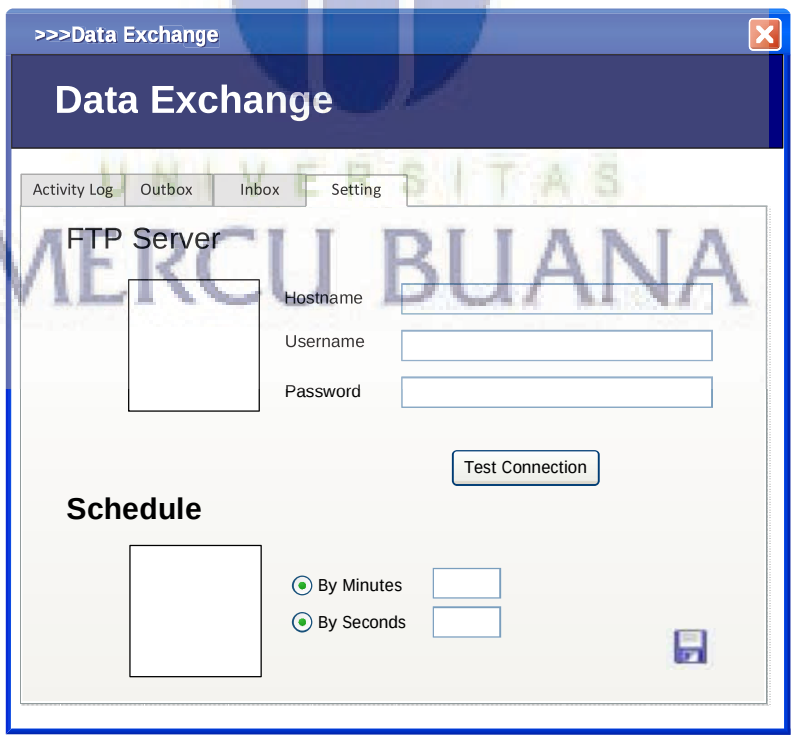


Gambar 3.5 Desain Form Transaksi

3.3.3 *Form Data Exchange*. Form data exchange digunakan untuk proses unggah dan unduh data dari dan ke FTP Server hingga proses dekripsi dan import database. Pada form ini ada 4 tab yaitu *Activity Log*, *Outbox*, *Inbox* dan *Setting*. *Tab Activity Log* berisi *log* aktivitas yang dilakukan sistem, tab *outbox* berisi daftar file backup yang masih belum terkirim ke FTP Server, tab *Inbox* berisi file hasil unduhan yang belum diimport ke dalam database, sedangkan tab setting digunakan untuk pengaturan FTP server dan periode pengecekan ada tidaknya file yang diunggah maupun unduh ke FTP Server. Tampilan form data exchange dapat dilihat pada gambar 3.13 dan 3.14.



Gambar 3.6 Desain *Form Activity Log*



Gambar 3.7 Desain *Form Setting*