

## **APLIKASI PENJUALAN BARANG BERBASIS WEB DENGAN *FEATURE DATABASE SYNCHRONIZER***

**Bambang Siswoyo**

Sekolah Tinggi Teknologi Informatika Sony Sugema, Bandung

### ***Abstract***

*Dr.emi creative design is a software house developer of web -based applications that are located in Bandung, West Java. Production activities include the development of a web -based application projects, retail software sales in the form of a CD master, merchandise and some goods such as tools - office stationery. Dr.emi creative design has had two (2) branches outside the city, namely Subang, West Java and Central Java Solo. To make the process of buying and selling goods, now using a web -based application, but the application is proven to take quite a long time will do searching data. This is because the client computer in the branch to check online and directly to the server at the center. The problem gets more complex when the sales data that have not been saved to the database server, go away because the connection is lost. To overcome these problems, the sale of goods made web-based application that has the feature database synchronizer, to analyze and understand the system that existed before. So expect this software will further accelerate the process transaction sales and simplify the process of selling goods report.*

**Keywords:** *Sales of Goods, Database Synchronizer, Xml*

### **Abstrak**

Dr.emi creative design merupakan salah satu software house pengembang aplikasi berbasis web yang berlokasi di Bandung, Jawa Barat. Kegiatan produksinya meliputi pengembangan proyek aplikasi berbasis web, penjualan software retail dalam bentuk CD master, merchandise dan beberapa barang berupa alat - alat tulis kantor. Dr.emi creative design telah memiliki 2 (dua) cabang di luar kota, yakni Subang Jawa Barat dan Solo Jawa Tengah. Untuk melakukan proses transaksi jual beli barang, saat ini sudah menggunakan aplikasi berbasis web, namun aplikasi yang dimaksud terbukti memakan waktu yang cukup lama saat akan melakukan pencarian data barang. Hal ini dikarenakan komputer client di cabang melakukan pengecekan secara online dan langsung ke server di pusat. Masalah akan lebih kompleks ketika data penjualan yang belum sempat disimpan ke dalam database server, hilang begitu saja karena koneksi terputus. Untuk mengatasi masalah tersebut, maka dibuatlah aplikasi penjualan barang berbasis web yang memiliki feature database synchronizer, dengan menganalisa dan memahami sistem yang ada sebelumnya. Sehingga diharapkan perangkat lunak ini akan lebih mempercepat proses transaksi penjualan dan mempermudah proses laporan penjualan barang.

**Kata Kunci:** Penjualan Barang, Database Synchronizer, Xml

## 1. PENDAHULUAN

Dr.emi creative design memiliki 2 (dua) cabang, di Solo dan Subang. Sedangkan pusatnya berada di Bandung. Untuk melakukan proses transaksi jual beli barang, saat ini sudah menggunakan aplikasi berbasis web, namun aplikasi yang dimaksud terbukti memakan waktu yang cukup lama saat akan melakukan pencarian data barang. Hal ini dikarenakan komputer *client* melakukan pengecekan secara *online* dan langsung ke *server*.

Dalam penggunaannya, aplikasi yang telah ada saat ini mengharuskan *administrator* di toko cabang untuk terus terkoneksi dengan internet, padahal sebetulnya koneksi internet hanya diperlukan saat mengirimkan laporan penjualan ke *server*. Biasanya dilakukan menggunakan rekam dalam format *Ms. Excel* dan dikirim melalui *email*.

## 2. MODEL, ANALISA, DESAIN, DAN IMPLEMENTASI

Penelitian ini dilakukan dengan metode studi pustaka dan observasi.

Dalam penelitian ini, diperoleh alur sistem yang sedang berjalan dapat dijelaskan secara rinci sebagai berikut:

1. Pihak penyedia barang di pusat menawarkan barang untuk dijual di cabang dengan harga khusus yang dapat diubah sesuai keinginan pihak cabang.
2. Kemudian pihak cabang mengambil barang ke pusat, yang dapat dilakukan dengan datang langsung ke pusat.
3. Pihak pusat menyediakan pendataan data barang yang diambil oleh pihak cabang, untuk dijadikan catatan awal. Dengan demikian, data awal pengambilan barangnya sama, baik di pusat maupun cabang.
4. Pihak cabang melakukan transaksi penjualan dengan konsumen di lokasi cabang berada. Dimana dalam proses transaksi ini menggunakan sistem aplikasi penjualan yang lama, dan dapat diakses secara online melalui sebuah *URL* yang telah disediakan pihak pusat.
5. Jika stok barang di cabang telah habis, maka pihak cabang akan melakukan pencatatan secara manual, untuk dilaporkan ke pihak pusat pada waktu tertentu. Proses melaporkan stok barang habis ini tidak dilakukan secara langsung setelah mengetahui stok barang habis, namun menunggu waktu tertentu terlebih dahulu, untuk kemudian disampaikan secara bersamaan dengan laporan penjualan ke pihak pusat.
6. Laporan penjualan dilakukan secara manual dengan format *file Excel*.
7. Bukti transaksi tidak dapat dicetak pada aplikasi penjualan yang lama.
8. Khusus untuk data barang, pihak cabang dapat merubah harga sesuai kebutuhan pasar di lokasi pihak cabang berada.

Dari serangkaian alur sistem yang sedang berjalan, menimbulkan masalah yang diataranya sebagai berikut:

1. Saat terjadi gangguan pada *server* di pusat, maka proses transaksi di *cabang* akan terhambat. Dengan demikian jika kebetulan sedang ada konsumen yang bertransaksi, maka data transaksi sementara tersimpan langsung di *database server* di pusat. Sementara pihak cabang tidak dapat *me-recovery* data tersebut.

2. Waktu tunggu untuk setiap kali bertransaksi berkisar 5-10 detik untuk satu kali *scanning* kode *barcode* pada *item* barang. Hal ini cukup menghambat proses transaksi, dimana seharusnya berjalan dengan cepat dan akurat.
3. Stok yang masih terdata dengan manual oleh pihak cabang, mengakibatkan ketidakakuratan data laporan ke pihak pusat. Sehingga sering terjadi barang yang sudah habis di pihak cabang malah tidak di order untuk dipesan ke pihak pusat.
4. Pihak cabang diharuskan terkoneksi internet terus menerus dalam melaksanakan transaksi ke *server* di pusat.

**2.1 Kebutuhan Fungsional**

**Administrator Server Pusat**

Tabel 2.1 Deskripsi Kebutuhan Administrator Pusat

No	Deskripsi
1	Input, Edit, Delete <i>schedule synchronize database</i>
2	Input, Edit, Delete dan Cetak data barang/ <i>item</i> dan kategorinya
3	Input, Edit, Delete data <i>server cabang</i> seperti IP dan nama komputer
4	Cetak laporan keseluruhan penjualan dari <i>server cabang</i>
5	Cetak laporan data barang

**2.2 Kebutuhan Fungsional**

**Administrator Server Cabang**

Tabel 2.2 Deskripsi Kebutuhan Administrator Pusat

No	Deskripsi
1	Input transaksi barang
2	Cetak laporan penjualan pada <i>server cabang</i> terkait
3	Cetak laporan data barang
4	Input, Edit, Delete <i>schedule synchronize database</i>

**2.3 Analisis User**

Tabel 2.3 Karakteristik Pengguna Aplikasi (Administrator Pusat)

Pengguna	Administrator Server Pusat (Pegawai)
Tanggung Jawab	Mengelola data master
Hak Akses	Dapat mengelola data barang, menentukan jadwal <i>synchronize database</i> dan dapat mengelola data informasi <i>server cabang</i> .
Keterampilan	Dapat mengoperasikan komputer

Tabel 2.4 Karakteristik Pengguna Aplikasi (Administrator Cabang)

Pengguna	Administrator Server Cabang (Pegawai)
Tanggung Jawab	Mengelola data transaksi
Hak Akses	Dapat mengelola data transaksi.
Keterampilan	Dapat mengoperasikan komputer

Tabel 2.5 Karakteristik Pengguna Aplikasi (Owner)

Pengguna	Owner
Tanggung Jawab	Mengelola data laporan
Hak Akses	Dapat mengelola data laporan.
Keterampilan	Dapat mengoperasikan komputer

## 2.4 Analisis Perangkat Lunak

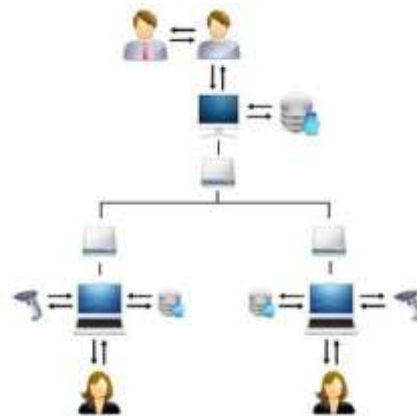
1. Untuk *server* pusat menggunakan sistem operasi *Ubuntu 9.04 (Jaunty Jackalope)*
2. Untuk *server* cabang menggunakan system operasi Windows XP.
3. *Apache web server* versi 2.2.14 (Ubuntu)
4. *Database* menggunakan *mySQL* versi 5.1.41-3ubuntu12.8
5. *PHP script language* versi 5.3.2-1ubuntu4.5
6. Browser menggunakan Mozilla Firefox/Internet Explorer.

## 2.5 Analisis Perangkat Keras

- a. *Server* pusat
  1. *Processor: XEON 5260 Quad Core*
  2. Memori 2 GB (2 DIMMs)
  3. *Harddisk 250 GB Serial ATA/150 7200 RPM by Seagate*
  4. *DELL Poweredge T610*
- b. *Server* cabang
  1. *Processor: AMD Phenom II X4 965 3.4GHz Black Edition*
  2. Memori 2 GB (2 DIMMs)
  3. *Harddisk 100 GB Serial ATA/150 7200 RPM by Seagate*
  4. *Ethernet Card, yaitu Generic Marvell Yukon 88E8056*
  5. *Barcode Scanner, dengan dukungan beberapa macam tipe barcode*

## 2.6 Arsitektur Sistem

Arsitektur sistem yang akan dibangun dapat dilihat pada gambar berikut ini.



Gambar 2.1 Arsitektur Sistem

Keterangan:

Tabel 2.6 Keterangan Icon/gambar Arsitektur Sistem

Icon/gambar	Deskripsi
	<i>Server Pusat</i>
	<i>Server Cabang</i>
	<i>Database Pusat</i>
	<i>Database Cabang</i>
	Modem
	Owner
	<i>Administrator Pusat</i>
	<i>Administrator Cabang</i>

## 3. HASIL DAN DISKUSI

Pada aplikasi ini *web server* dan *database server* akan diinstal di dua lokasi berbeda yakni pusat dan cabang. Database di cabang akan difungsikan sebagai database yang hanya *handle* data penjualan dan pendataan stok barang. Sedangkan data barang yang ada di cabang akan diambil dari referensi sumber data barang di pusat.

Database di pusat akan difungsikan

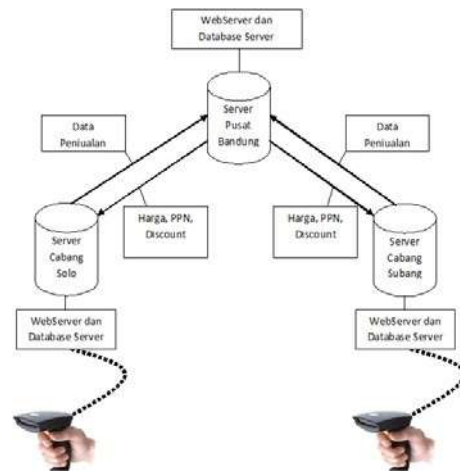
sebagai database yang *handle* kebutuhan data teknis di cabang, data barang, dan penjadwalan. Penjadwalan yang dimaksud adalah proses menentukan kapan server pusat / server cabang melakukan proses sinkronisasi data. Hal ini bertujuan agar data yang akan disinkronkan dapat diproses dengan tidak mengganggu proses transaksi pada server cabang.

Dapat diambil contoh bahwa, jika sinkronisasi data harga dari server pusat ke server cabang berlangsung saat server cabang melakukan transaksi, tentu saja akan ada selisih harga, dan berpengaruh pada total penjualan barang saat sebelum disinkronkan dengan setelah disinkronkan.

Sehingga penjadwalan akan dilakukan saat proses transaksi tidak sedang berlangsung. Penjadwalan tersimpan dalam table jadwal di *database* yang akan dibaca sebelum sinkronisasi dilakukan dan pembuatan jadwal ini dilakukan oleh administrator masing - masing server.

Penjadwalan ini tidak harus dilakukan oleh masing masing server, namun hanya berfungsi untuk membantu dalam proses otomatisasi proses sinkron. Sehingga sinkronisasi database ini dapat dilakukan dengan cara manual melalui sistem aplikasi yang sama.

Dengan demikian proses sinkronisasi *database* ini dapat digambarkan sebagai berikut:



Gambar 3.1 Diagram alir tahapan proses sinkronisasi *database*

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

1. Aplikasi Penjualan dengan *Feature Database Synchronizer* yang di bangun telah mempermudah dalam proses *update* informasi data penjualan dari *server* cabang ke *server* pusat, maupun *update* informasi data barang dari *server* pusat ke *server* cabang. Dimana *server* cabang, tidak harus terkoneksi secara terus menerus dalam melakukan proses sinkronisasi database.
2. Aplikasi ini juga mempercepat proses transaksi penjualan pada *server* cabang.
3. Dibuatnya aplikasi web based ini Gambar 3.1 Diagram alir tahapan proses sinkronisasi *database*

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

1. Aplikasi Penjualan dengan *Feature Database Synchronizer* yang di bangun telah mempermudah dalam proses *update* informasi data penjualan dari *server* cabang ke

*server* pusat, maupun *update* informasi data barang dari *server* pusat ke *server* cabang. Dimana *server* cabang, tidak harus terkoneksi secara terus menerus dalam melakukan proses sinkronisasi database.

2. Aplikasi ini juga mempercepat proses transaksi penjualan pada *server* cabang.
3. Dibuatnya aplikasi web based ini membantu administrator *server* pusat maupun *server* cabang untuk menentukan *schedule* *syncronize* *database*.

##### 5. DAFTAR PUSTAKA

- [1] Adi Nugroho, (2010), *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*, Andi Publisher, Yogyakarta.
- [2] Andi Sunyoto, M. Kom., (2007), *AJAX Membangun Web dengan teknologi Asynchronous Javascript & XML*, Andi Publisher, Yogyakarta.
- [3] Andri Kristianto (2004), *Rekayasa Perangkat Lunak Konsep Dasar*, Gava Media, Yogyakarta.
- [4] Betha, Husni I Pohan, (2001), *Pemrograman WEB dengan HTML*, Informatika, Bandung.
- [5] Budi Rahardjo, (2004), *Keamanan Sistem Informasi berbasis Internet*, Insan Infonesia, Bandung.
- [6] Henky Prihatna, (2005), *Kiat Praktis Menjadi Webmaster Professional*, Elex, Media, Komputindo, Jakarta.
- [7] <http://www.holub.com/goodies/uml/>, (2007), Alen's Holub UML Quick Reference, HOLUB, Berkeley California.
- [8] <http://www.ida.liu.se/~TDDDB62/uml.en.shtml>, (2005), *Introduction to UML and Rational Rose*, Department of Computer and Information Science of Linköping University, Linköping Swedia.
- [9] Jack Febrian, (2007), *Google & Yahoo Secrets!*, Informatika, Jakarta.
- [10] Ridwan Sanjaya, SE., S. Kom, (2004), *Trik Memperindah Website dengan Menu Dinamis*, Elex Media Komputindo, Jakarta.
- [11] Rosa A. S, M. Salahuddin, (2010), *Pemrograman Berorientasi Objek*, Modula, Bandung.