

PERANCANGAN SISTEM INTEGRASI APLIKASI TERDISTRIBUSI PADA TRANSAKSI DATA PESANAN MENGGUNAKAN *WEB SERVICE*

Nasir Suruali

Dosen Program Studi Teknik Mesin, Fakultas Teknik, Universitas Pattimura Ambon
e-mail : nsuruli@yahoo.com

ABSTRAK

Data pesanan merupakan kebutuhan utama dalam supply chain management pada sebuah perusahaan dalam rangka mencapai kelancaran aliran data dalam manajemen perusahaan. Masalah utama yang muncul adalah aliran data pesanan yang tidak terkontrol, proses produksi yang tidak sesuai lead time, pemborosan inventori serta lokasi yang berada pada jarak terpisah. Faktor lain yang muncul adalah keterbatasan mekanisme integrasi aplikasi serta interoperabilitas antar sistem. Perkembangan internet sebagai salah satu media komunikasi dalam e-business merupakan salah satu peluang dalam melakukan integrasi antar aplikasi tersebut. Salah satu teknologi yang berkembang saat ini dan ditujukan untuk mengakomodasi keanekaragaman dan masalah tersebut adalah Web Service. Pada penulisan artikel ini, dilakukan perancangan dan sistem Web Service pada kasus integrasi aplikasi dalam perusahaan antara bagian marketing dengan bagian produksi. Implementasi Web Service dilakukan meliputi proses transaksi dan penjejukan data pesanan. Perancangan dilakukan dengan menggunakan model UML dan implementasi menggunakan perangkat lunak platform .NET dari Microsoft. Hasil penelitian yang dilakukan menunjukkan bahwa Web Service merupakan teknologi yang layak diterima untuk efisiensi dan efektivitas di perusahaan garmen.

Kata kunci : integrasi aplikasi, Web Service, aplikasi terdistribusi, data pesanan

ABSTRACT

Order data is the main need in supply chain management in a company in order to achieve a smooth flow of data in the company's management. The main problem that arises is the data flow of orders that are not controlled, the production process that is not appropriate lead time, inventory and location of waste that are at a distance apart. Another factor that emerged was the limited mechanisms of integration of applications and interoperability between systems. The development of the Internet as a medium of communication in e-business is one chance in doing integration among these applications. One of the technologies developed at this time and is intended to accommodate diversity and that problem is the Web Service. At the writing of this article, to design and Web Service system in the case within the enterprise application integration between the marketing with production parts. Implementation of Web Service performed include transaction processing and tracking orders data. The design is done using UML models and implementations using software from Microsoft's .NET platform. Results of research conducted indicates that the Web Service is a technology that deserves to efficiency and effectiveness in a garment company.

Keywords: application integration, Web services, distributed applications, orders data

PENDAHULUAN

Dalam lingkungan bisnis yang kompetitif saat ini, efisiensi dan efektivitas dari *supply chain* adalah masalah yang penting agar perusahaan tetap bertahan dan pada umumnya hal ini dipengaruhi oleh sistem informasi yang mendukungnya. *Supply chain* mengacu pada aliran material, informasi, pembayaran, layanan, pengadaan bahan baku dari *supplier* yang melewati *factory*, *warehouse* menuju ke pelanggan.

Kecenderungan dunia bisnis untuk mengatasi masalah dalam *supply chain* adalah menata kembali proses yang terjadi dalam perusahaan yang melibatkan proses produksi, pengiriman yang tepat waktu (*just-in-time delivery*), penghematan inventori dan peningkatan efisiensi dalam setiap proses untuk mencapai keuntungan yang lebih besar. Karena adanya ketergantungan dalam aktivitas *supply chain*, masalah integrasi sistem menjadi hal yang penting terutama dalam fungsionalitas dari sistem yang

terlibat. Sedangkan jika ditinjau dari gambaran baru bisnis saat ini bahwa keputusan *supply chain* yang melibatkan proses perencanaan produksi dan keputusan *marketing* dapat berfungsi secara bersamaan. Salah satu teknologi yang berkembang saat ini dan ditujukan untuk mengakomodasi keanekaragaman sistem dalam hal *platform* yang digunakan adalah *Web Service*. Hal ini karena *Web Service* memiliki beberapa keunggulan terutama dalam hal kemampuannya menembus *firewall*, kompleksitas protokol yang digunakan serta kemampuan integrasi dengan *platform* yang berbeda. Berdasarkan dari keunggulan yang dimiliki *Web Service* dianggap sesuai untuk menjawab kebutuhan pengontrolan dan penyediaan data dengan melakukan pesanan data di sisi *marketing*, kemudian diimplementasikan *Web Service* untuk melakukan integrasi aplikasi dengan arsitektur berorientasi layanan.

TINJAUAN PUSTAKA

2.1 Komputasi Terdistribusi Berbasis Komponen

Ide komputasi terdistribusi ini telah berkembang dalam kurun waktu yang cukup lama dan telah dikembangkan dalam beragam teknologi komunikasi yang membolehkan pendistribusian dan penggunaan kembali aplikasi logik. Komputasi terdistribusi merupakan bagian dari logik aplikasi yang dibagi ke dalam unit-unit yang didistribusikan secara fisik antar dua komputer atau lebih dalam sebuah jaringan.

2.2 Web Service

Web Service merupakan proses standar untuk mengintegrasikan aplikasi-aplikasi berbasis web dengan menggunakan standar XML, SOAP, WSDL dan UDDI melalui *backbone* protokol internet. XML digunakan sebagai *tag* atau tanda pengenal dari data yang digunakan, SOAP digunakan untuk mengirimkan data, WSDL digunakan untuk mendeskripsikan layanan yang tersedia, dan UDDI digunakan untuk menampilkan daftar layanan yang tersedia. Dengan tujuan utama agar berbagai bisnis dapat saling berkomunikasi satu dengan lainnya maupun dengan para *client*-nya, *Web Service* memungkinkan organisasi-organisasi tersebut saling mempertukarkan data tanpa perlu mengetahui sistem yang ada di belakangnya sekali pun sistem tersebut berada di belakang *firewall*. [Webopedia].

2.2.1 Hyper Text Transfer Protocol (HTTP)

Kecenderungan kesuksesan internet adalah berdasarkan *Hypertext Transfer Protocol* (HTTP) dan *Simple Mail Transfer Protocol* (SMTP) ketika digabungkan dengan HTML. HTTP adalah sebuah protokol khusus yang dirancang untuk mendefinisikan pola pesan berdasarkan *request/respon* yang berada pada *layer* atas TCP/IP.

2.2.2 Extensible Markup Language (XML)

Extensible Markup Language (XML) merupakan bahasa dengan format teks yang fleksibel dan sederhana yang diturunkan dari *Standard Generalized Markup Language* (SGML), yaitu sebuah standar internasional dalam pendefinisian cara-cara merepresentasikan teks. XML di desain untuk memenuhi kebutuhan dalam hal semakin meningkatnya kebutuhan akan publikasi data elektronik dalam skala yang luas.

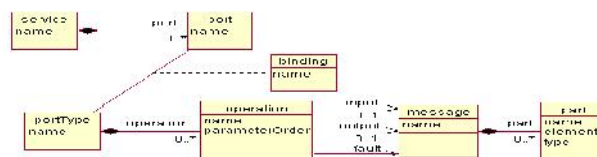
2.2.3 Simple Object Access Protocol (SOAP)

SOAP merupakan protokol berbasis XML untuk pengiriman pesan (*messaging*) dan untuk melakukan *Remote Procedure Call* (RPC). Protokol ini antara lain menyediakan standar untuk mengemas pesan (*message*) dan aturan pengkodean (*encoding*) untuk melakukan serialisasi (*serialize*) dan deserialisasi (*deserialize*) data selama pengiriman (*transmission*). Tujuan lainnya adalah menyediakan model bagi implementasi operasi RPC.

2.2.4 Web Service Description Language (WSDL)

Untuk menjamin agar fungsi yang ditawarkan oleh sebuah *Web service* dapat ditemukan dan digunakan oleh pengguna, deskripsi dari *Web service* dinyatakan dalam sebuah format XML yang disebut WSDL. Format ini mendukung sepenuhnya penjelasan mengenai operasi-operasi dan fungsi-fungsi apa saja yang tersedia pada sebuah layanan *Web Service* beserta parameter-parameter apa saja yang dibutuhkan untuk menggunakan *Web Service* tersebut.

Gambar 2.1 berikut ini adalah model UML yang menggambarkan bagaimana definisi skema untuk menjelaskan layanan dalam *Web Service* dalam WSDL.



Struktur definisi WSDL dalam model UML

2.2.5 *Universal Description, Discovery and Integration (UDDI)*

Universal Description, Discovery and Integration (UDDI) adalah sebuah *registry* global yang dikelola oleh *Microsoft*, *IBM* dan *Hewlett Packard* yang memungkinkan suatu organisasi bisnis untuk menemukan dan ditemukan oleh organisasi bisnis lain mana saja di seluruh dunia sehingga dapat dilakukan proses transaksi ekonomi lebih efisien.

2.3 *Web Service dan Konsep Arsitektur Berorientasi Layanan*

Jika *Web Service* kita pandang sebagai integrasi aplikasi yang digunakan dalam aplikasi masa depan maka *Web Service* akan lebih tepat ditinjau dengan posisinya dalam arsitektur berorientasi layanan. Adapun kemampuan yang dimiliki *Web Service* dalam mengakomodasi kebutuhan penting yang dimiliki antara lain, dalam teknologi *Web Service* ada pemisahan pembuatan *inteface* dengan pemrograman, *Web Service* memiliki *interface* yang fleksibel, *Web Service* menggunakan internet sebagai media, *Web Service* menggunakan bahasa *interface XML* dan *WSDL*.

2.4 *Konsep Supply Chain Management*

Penerapan konsep *SCM* pada saat ini bertujuan untuk mengurangi resiko dan ketidakpastian di dalam *supply chain*, yang mempengaruhi level-level inventori, siklus hidup, proses bisnis dan layanan pelanggan dalam rangka meningkatkan keuntungan dan daya saing. Dalam lingkungan bisnis yang kompetitif pada saat ini masalah efisiensi dan efektivitas *supply chain* dalam beberapa organisasi adalah masalah yang sangat diperhatikan agar perusahaan mampu bertahan dan tentunya hal ini didukung juga oleh sistem informasi yang ada.

2.5 *Tinjauan Aspek Keamanan Sistem Web Service*

Kerentanan yang ada pada *Web Service* yang berbasis *XML* menjadi pertimbangan utama bagi para pengguna untuk mengimplementasikan *Web Service*. Oleh karena itu diperlukan standar teknologi keamanan *Web Service* untuk mendukung keamanan penerapan *Web Service* secara luas. Meskipun demikian, standar yang ada pada saat ini seperti *Lightweight Directory Access Protocol (LDAP)*, *Public Key Infrastructure (PKI)*, *Secure Socket Layer* dan *Transport Layer Security (SSL/TLS)* tetap memegang peranan penting untuk mengamankan *Web Service* dalam konteks *point-to-point*.

ANALISA DAN PERANCANGAN SISTEM

3.1 *Deskripsi Dan Persyaratan Sistem*

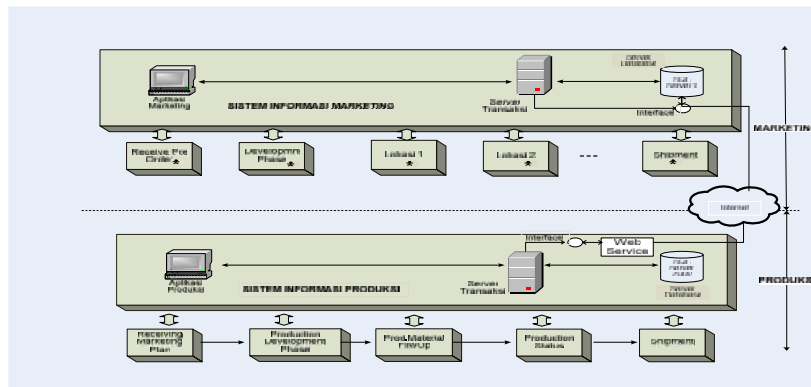
Perusahaan mendasarkan pada konsep *supply chain management* dimana antar sistem yang terlibat dalam perusahaan harus saling berhubungan dan menjaga aliran data melalui sistem yang terdistribusi, hal ini juga dilakukan untuk meningkatkan kinerja perusahaan, mencapai kompensasi yang memuaskan serta tuntutan persaingan pasar yang semakin tinggi. Tahap analisis dalam perancangan sebuah sistem bertujuan untuk mengetahui kerangka dasar dari sistem. Langkah awal adalah bagaimana mendeskripsikan sistem yang berjalan dan sistem yang akan dibangun beserta spesifikasi yang mendasari perancangan. Langkah selanjutnya adalah analisis sistem yang menjelaskan kebutuhan sistem dan analisisnya menggunakan metode berorientasi obyek.

3.2 *Deskripsi Sistem Yang Akan Dibangun*

Perumusan permasalahan akan dilakukan integrasi dua sistem yaitu aplikasi *marketing* dengan aplikasi produksi di sebuah perusahaan. Arsitektur mekanisme integrasi yang dilakukan yaitu akan mengubah prosedur penyediaan dan pengiriman data pesanan yang dilakukan dengan menggunakan *VPN* akan diganti dengan menggunakan *Web Service* yang memungkinkan operator *marketing* dapat melakukan transaksi data secara otomatis ke bagian produksi selanjutnya melakukan penjejukan data pesanan yang telah dikirim dan di simpan di *server* produksi. Gambaran umum arsitektur integrasi yang akan diimplementasikan dapat dilihat pada Gambar 3.1

3.3 *Persyaratan Sistem*

Sistem yang akan dibangun tetap mengikuti proses bisnis dan aturan bisnis yang berlaku, perubahan yang terjadi hanya mengganti *VPN* dengan menggunakan jaringan internet dan *Web Service* sebagai penghubung komunikasi antar aplikasi yang terintegrasi untuk proses transaksi dan penjejukan untuk setiap data pesanan yang dikirimkan ke bagian produksi. Persyaratan sistem menyangkut fungsi, batasan, masukan, keluaran dan performansi sistem.



Arsitektur mekanisme integrasi aplikasi yang akan dibangun

3.4 Analisa Sistem

Analisa sistem dilakukan dengan metoda analisa berorientasi obyek dengan tujuan untuk membuat pemetaan awal mengenai kebutuhan yang akan dirancang, pemetaan tersebut dibagi menjadi elemen-elemen analisis. Elemen analisis kemudian akan dikembangkan menjadi elemen model desain. Analisis kebutuhan sistem dijabarkan berupa model *use case* sistem dan analisis domain yang merupakan penjabaran dari tiap *use case* dalam model *class diagram*.

Pada Gambar 3.2 menunjukkan *class diagram database logic* sebagai integrasi antara aplikasi marketing, *Web Service* dan aplikasi produksi.

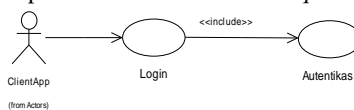


Class diagram database logic sebagai integrasi aplikasi marketing dengan produksi

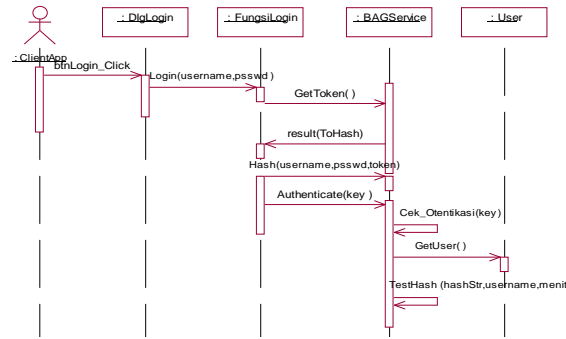
3.4.1 Analisa Use Case Menggunakan Sequence Diagram

1. Use Case Login

Use case list pertama yang dilakukan oleh semua aktor untuk masuk ke sistem dan melakukan transaksi adalah *login* ke sistem. Fungsi *login* ini menggunakan enkripsi *password*, *username* dikirim dalam *clear text*, digunakan untuk mengenkripsi *password*. Gambar 3.3 menunjukkan *use case diagram login* termasuk autentikasi. Sedangkan pada Gambar 3.4 adalah *sequence diagram* untuk fungsi *login*.



Gambar 3.3 *Use case diagram* untuk login



Sequence diagram untuk use case login

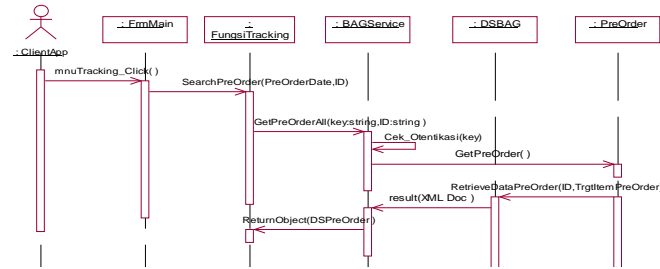
Fungsi *login* ini akan digunakan untuk setiap transaksi dan penjejukan oleh *client* yang akan mengakses layanan-layanan yang disediakan oleh *Web Service*.

2. Use case untuk fungsi-fungsi transaksi

Use case transaksi ini dilakukan oleh operator di sisi *marketing* untuk melakukan transaksi data pesanan yang meliputi modul *pre order*, *post order*, *development phase* serta modul *production* untuk proses transaksi *load* data yang sudah tersimpan di sisi *server*, *insert* data, *update* data serta *delete* data yang sudah tidak diperlukan lagi.

3. Use case untuk fungsi tracking

Use case tracking ini dilakukan oleh operator di sisi *marketing* untuk melakukan penjejukan data pesanan yang meliputi modul *pre order*, *post order*, *development phase* serta modul *production* untuk proses penjejukan data yang sudah tersimpan di sisi aplikasi produksi. Pada Gambar 3.5 menunjukkan interaksi antara modul *Track data pre order*, *Web Service* dan aplikasi *marketing* dengan menggunakan *sequence diagram*.



Sequence diagram untuk use case track data pre order

4. Use case untuk fungsi user administration

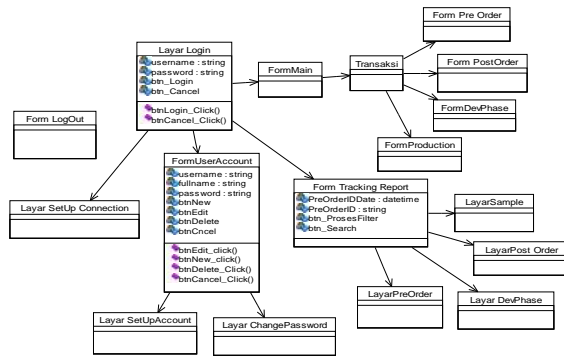
Menunjukkan interaksi antara fungsi *user administration*, *Web Service* dan aplikasi di sisi *marketing*. *Use case* ini dilakukan oleh operator di sisi *marketing* untuk penambahan *user* baru, menghapus *user* yang sudah tidak aktif, atau penggantian *username* atau *password* dari *user*.

5. Class Diagram User Interface

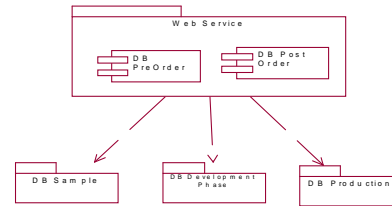
Class Diagram untuk level *user interface* yang dalam kasus ini adalah fungsi layanan *Web Service* di bagian *marketing* yang diakses oleh operator produksi adalah dalam bentuk modul-modul yang mengacu pada arsitektur *three-tier*, dalam dilihat pada Gambar 3.6

3.4.2 Package dan Deployment Diagram

Package Web Service yang terlibat integrasi aplikasi terdistribusi ini terdiri dari komponen *database pre order* dan *database post order*. Sedangkan *package* aplikasi produksi terhubung dengan *database sample*, *development phase* dan *production* pada Gambar 3.7.



Class diagram untuk user interface di sisi client



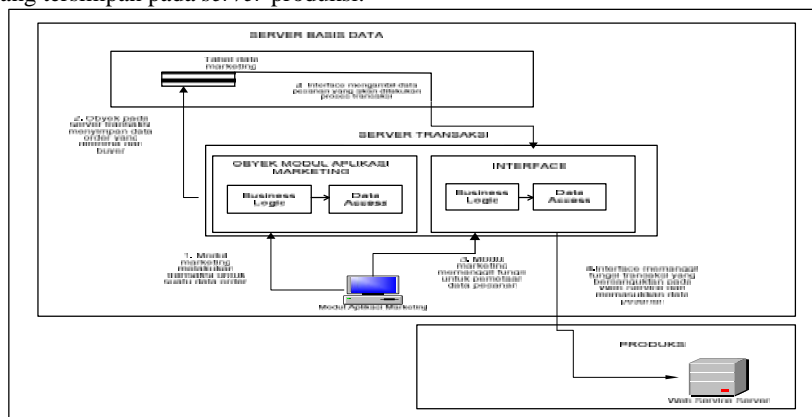
Package Web Service yang terhubung dengan basis data produksi

3.5 Perancangan Pengembangan Web Service

3.5.1 Perancangan Implementasi Web Service

Perancangan *Web Service* sebagai pengintegrasian antara aplikasi *marketing* dengan aplikasi produksi menggunakan mekanisme *library file*. *Web Service* dalam hal ini akan mengakses *library file* yang memuat obyek yang berisi *business logic* dari semua fungsi penjejak dan fungsi transaksi yang terdapat pada aplikasi pada produksi di sisi *marketing*. Sehingga *Web Service* maupun aplikasi produksi yang berada di sisi *marketing* akan mengakses *library file* yang sama, pada kondisi ini *file library* pada *server* akan mengatur cara akses dan penulisan data dari dan ke *server* basis data.

Desain yang dibuat dalam studi kasus ini sebagai mekanisme integrasi antar aplikasi *marketing* dengan produksi juga meliputi bagaimana skenario pergerakan data menuju sistem tujuan Gambar 3.8 menggambarkan skenario pergerakan data untuk transaksi data pesanan untuk pengambilan data dari basis data yang tersimpan pada *server* produksi.



Skenario pergerakan data untuk kasus transaksi data pesanan

3.4.2 Perancangan Mekanisme Pengujian

Pengujian terhadap mekanisme integrasi aplikasi yang telah dibuat khususnya pengujian secara fungsional terhadap fungsi-fungsi yang disediakan oleh *Web Service* yang dibangun di sisi produksi dilakukan untuk memastikan bahwa telah diimplementasikan mekanisme integrasi, yaitu integrasi antara aplikasi *marketing* dengan aplikasi produksi.

3.4.3 Perancangan Mekanisme Sistem Keamanan Yang Diterapkan

Dalam *Web Service*, ada banyak skenario keamanan *Web Service* yang dapat digunakan. Beberapa skenario yang umum antara lain:

- Otentikasi *Web Service* secara sederhana dengan *SOAP headers*. Akan tetapi pada skenario ini *username* dan *password* dikirim dalam *clear text* dan tidak ada enkripsi untuk data.

- Menggunakan Modul WSE (*Web Service Enhancement*) yang ditawarkan *Microsoft*. Hal ini mungkin solusi komplit, akan tetapi terlalu kompleks untuk jenis *Web Service* sederhana. Pada skenario ini, *username* dapat dikirim dalam *clear text*, akan tetapi mungkin juga untuk menggunakan *Password Digest* untuk mengenkripsi *password*. *SOAP message* dienkripsi dengan menggunakan spesifikasi *XML Encryption*.

Solusi yang digunakan adalah solusi tengah. *Username* dikirim dalam *clear text*, akan tetapi MD5 digunakan untuk mengenkripsi *password*. Diasumsikan data yang dikirim tidak terlalu sensitif, sehingga data yang dikembalikan oleh *Web Service* tidak dienkripsi. Hal ini dilakukan untuk mengambil jalan tengah antara *security* dan *performance*.

PENGUJIAN APLIKASI DAN ANALISA HASIL PENGUJIAN

4.1 Pengujian Dan Mekanisme Skenario Pengujian

Dalam rangka pengujian *Web Service* sebagai alat bantu dalam mekanisme integrasi aplikasi, dibangun sebuah aplikasi *client* di sisi produksi, dimana aplikasi *client* yang dibangun dengan merepresentasikan keadaan pada saat suatu transaksi data pesanan tertentu dilakukan. Pengujian yang dilakukan meliputi beberapa hal yaitu :

- pengujian keberhasilan penggunaan internet sebagai media komunikasi dalam integrasi aplikasi pada perusahaan;
- pengujian fungsional untuk setiap fungsi *use case* yang diperlukan dalam mekanisme integrasi aplikasi yang dirancang. Pengujian ini dilakukan terhadap setiap komponen dari arsitektur mekanisme integrasi, serta terhadap interaksi antar komponen dalam mendukung keberhasilan implementasi arsitektur integrasi yang telah dirancang.

4.2 Hasil Pengujian

Pengujian implementasi *Web Service* telah dilakukan pada lingkungan lokal pada domain *Local Area Network (LAN)* dan penggunaan jaringan internet. Masing-masing domain pengujian dilakukan untuk menguji fungsional dari setiap fungsi *use case* yang dimiliki oleh masing-masing komponen dan pengujian membahas interaksi dan integrasi aplikasi antara komponen di sisi *marketing* dengan komponen di sisi produksi dalam domain LAN dan internet.

4.2.1 Hasil Pengujian Fungsi Transaksi

Tabel 4.1 sampai dengan Tabel 4.7 menunjukkan hasil pengujian untuk fungsi-fungsi pada *interface* untuk setiap transaksi berdasarkan fungsi *use case* yang telah dibuat pada tahap perancangan.

1. Pengujian Untuk Login Ke Sistem

Hasil pengujian untuk login

| No | Fungsi Yang Diuji | Nama Fungsi | Hasil | | Kriteria Keberhasilan |
|----|-------------------|-------------|----------|-------|--|
| | | | Berhasil | Gagal | |
| 1. | Login ke sistem | Login | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon login memberi <i>token</i> dan meminta <i>key</i> untuk melakukan <i>hashing</i> ▪ Server menerima <i>key</i> dari klien dan mensplitnya berdasarkan <i>delimiter</i> ▪ Sistem akan menampilkan menu |

2. Pengujian Transaksi Modul Pre Order

Hasil pengujian untuk modul Pre Order

| No | Fungsi Yang Diuji | Nama Fungsi | Hasil | | Kriteria Keberhasilan |
|----|--|------------------|----------|-------|---|
| | | | Berhasil | Gagal | |
| 1. | Fungsi pemasukan data transaksi ke dalam tabel <i>pre order</i> | Insert Pre Order | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon terhadap <i>insert data pre order</i> ▪ Sistem memasukkan data baru |
| 2. | Fungsi perubahan status/maintenance data ke tabel <i>pre order</i> | Update Pre Order | ✓ | | <ul style="list-style-type: none"> ▪ Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi form yang telah diisi ▪ Melakukan koneksi dengan <i>web service</i> ▪ Melakukan pengiriman data terhadap <i>web service</i> |
| 3. | Fungsi penghapusan data yang tidak digunakan lagi | Delete Pre Order | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> akan menghapus data pre order yang dimaksud ▪ Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
| 4. | Fungsi pemasukan data transaksi ke dalam tabel sampel | Insert Sampel | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon terhadap <i>insert data sampel</i> ▪ Sistem memasukkan data sampel baru |
| 5. | Fungsi perubahan data pada tabel sampel | Update Sampel | ✓ | | <ul style="list-style-type: none"> ▪ Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi form yang telah diisi ▪ Melakukan koneksi dengan <i>web service</i> |

| | | | | | |
|----|--|--------------|---|--|---|
| 6. | Fungsi penghapusan data sampel pada tabel sampel | DeleteSampel | ✓ | | <ul style="list-style-type: none"> ▪ Melakukan pengiriman data terhadap <i>web service</i> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> akan menghapus data sampel yang dimaksud ▪ Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
|----|--|--------------|---|--|---|

3. Modul Post Order

Hasil pengujian fungsi *interface* untuk modul *Post Order*

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|----|--|---------------------------------|----------|-------|---|
| | | | Berhasil | Gagal | |
| 1. | Fungsi pemasukan data transaksi ke dalam tabel <i>post order</i> | Insert Post Order | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon terhadap <i>insert</i> data <i>post order</i> ▪ Sistem memasukkan data <i>post order</i> baru |
| 2. | Fungsi perubahan status/modifikasi data ke tabel <i>post order</i> | Update Post Order | ✓ | | <ul style="list-style-type: none"> ▪ Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi form yang telah diisi ▪ Melakukan koneksi dengan <i>web service</i> ▪ Melakukan pengiriman data terhadap <i>web service</i> |
| 3. | Fungsi penghapusan data yang tidak digunakan lagi | Delete Post Order | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> akan menghapus data pelanggan yang dimaksud ▪ Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
| 4. | Fungsi <i>load</i> data <i>post order</i> yang telah tersimpan di basis data | GetPostOrder ByParam | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> mengembalikan hasil berupa data sesuai dengan <i>query string</i> ID ▪ Sistem dapat menampilkan data <i>post order</i> sesuai dengan ID yang diinginkan ▪ Sistem menampilkan tombol "Cancel" |

4. Modul Development Phase

Hasil pengujian fungsi *interface* untuk Modul *Development Phase*

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|----|---|---------------------------------|----------|-------|--|
| | | | Berhasil | Gagal | |
| 1. | Fungsi pemasukan data transaksi ke dalam tabel DevPhase | Insert DevPhase | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon terhadap <i>insert</i> data <i>devphase</i> ▪ Sistem memasukkan data <i>devphase</i> baru ▪ Data sudah bertambah dan tersimpan di basis data |
| 2. | Fungsi perubahan status/ <i>maintenanc e</i> data ke tabel DevPhase | Update DevPhase | ✓ | | <ul style="list-style-type: none"> ▪ Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi form yang telah diisi ▪ Melakukan koneksi dengan <i>web service</i> ▪ Melakukan pengiriman data terhadap <i>web service</i> |
| 3. | Fungsi penghapusan data yang tidak digunakan lagi | Delete DevPhase | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> akan menghapus data pelanggan yang dimaksud ▪ Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
| 4. | Fungsi <i>loading</i> data DevPhase yang sudah dimasukkan dan tersimpan di sisi <i>client</i> | GetDevPhase byParam | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> mengembalikan hasil berupa data sesuai dengan <i>query string</i> ID ▪ Sistem menampilkan data <i>devphase</i> |

5. Modul Production Follow Up

Hasil pengujian fungsi *interface* untuk modul *Production Follow Up*

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|----|---|---------------------------------|----------|-------|--|
| | | | Berhasil | Gagal | |
| 1. | Fungsi pemasukan data transaksi ke dalam tabel <i>production</i> | Insert Production | ✓ | | <ul style="list-style-type: none"> ▪ Sistem melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> memberikan respon terhadap <i>insert</i> data <i>production</i> ▪ Sistem memasukkan data <i>production</i> baru ▪ Data sudah bertambah dan tersimpan di basis data |
| 2. | Fungsi perubahan status/ <i>maintenanc e</i> data ke tabel <i>production</i> | Update Production | ✓ | | <ul style="list-style-type: none"> ▪ Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi <i>form</i> yang telah diisi ▪ Melakukan koneksi dengan <i>web service</i> ▪ Melakukan pengiriman data terhadap <i>web service</i> |
| 3. | Fungsi penghapusan data yang tidak digunakan lagi | Delete Production | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> akan menghapus data pelanggan yang dimaksud ▪ Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
| 4. | Fungsi <i>load</i> data <i>Production</i> yang sudah dimasukkan dan tersimpan | GetProduction byParam | ✓ | | <ul style="list-style-type: none"> ▪ Sistem akan melakukan koneksi dengan <i>web service</i> ▪ Sistem <i>web service</i> mengembalikan hasil berupa data sesuai dengan <i>query string</i> ID <i>devphase</i> ▪ Sistem dapat menampilkan data <i>devphase</i> sesuai dengan <i>id</i> |

6. Pengujian Pada Fungsi User Administration

Hasil pengujian fungsi *interface User Administration*

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|----|--|---------------------------------|----------|-------|--|
| | | | Berhasil | Gagal | |
| 1. | Fungsi pemasukan data <i>user</i> baru yang akan melakukan akses program | <i>Insert User</i> | ✓ | | <ul style="list-style-type: none"> Sistem melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> memberikan respon terhadap <i>insert data user</i> Sistem memasukkan data <i>user</i> baru Data sudah bertambah dan tersimpan di basis data |
| 2. | Fungsi perubahan data <i>user</i> | <i>UpdateUser</i> | ✓ | | <ul style="list-style-type: none"> Pada tombol "Edit" sistem dapat merubah data sesuai dengan isi form yang telah diisi Melakukan koneksi dengan <i>web service</i> Melakukan pengiriman data terhadap <i>web service</i> Data <i>user</i> berupa <i>username, password, fillname</i> telah diubah |
| 3. | Fungsi penghapusan data <i>user</i> yang tidak aktif lagi | <i>DeleteUser</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> akan menghapus data <i>user</i> yang dimaksud Sistem akan memberikan <i>warning</i> terhadap data yang akan dihapus |
| 4. | Fungsi menampilkan data seluruh <i>user</i> | <i>GetUserAll</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data <i>user</i> Sistem dapat menampilkan data <i>user</i> keseluruhan |

4.2.2 Hasil Pengujian Fungsi Tracking

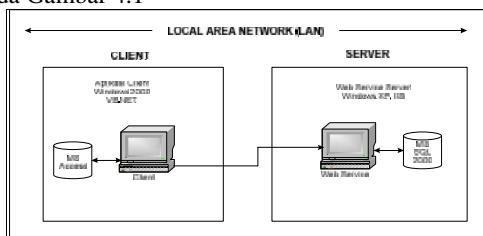
Tabel 4.7 adalah tabel hasil pengujian fungsional untuk penjejukan data yang berada pada *server* produksi dengan menggunakan *Web Service*.

Hasil pengujian fungsi *tracking*

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|----|--|---------------------------------|----------|-------|---|
| | | | Berhasil | Gagal | |
| 1. | Fungsi penjejukan data <i>pre order</i> | <i>GetPreOrder All</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data yang telah di-track sesuai dengan <i>query string pre orderID</i> dan <i>pre order date</i> Sistem menampilkan data <i>pre order</i> hasil penjejukan |
| 2. | Fungsi penjejukan data sampel | <i>GetSampelAll</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data yang telah di-track sesuai dengan <i>query string pre orderID</i> dan <i>pre order date</i> Sistem menampilkan data <i>sample</i> hasil penjejukan berdasarkan <i>query</i> |
| 3. | Fungsi penjejukan data <i>post order</i> | <i>GetPostOrder All</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data yang telah di-track sesuai dengan <i>query string pre orderID</i> dan <i>pre order date</i> Sistem menampilkan data <i>sample</i> hasil penjejukan berdasarkan <i>query</i> |
| 4. | Fungsi penjejukan data <i>development phase</i> | <i>GetDevPhase All</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data yang telah di-track sesuai dengan <i>query string pre orderID</i> dan <i>pre order date</i> Sistem menampilkan data <i>development phase</i> hasil penjejukan berdasarkan <i>query</i> |
| 5. | Fungsi Penjejukan Data <i>Production Follow Up</i> | <i>GetProductionAll</i> | ✓ | | <ul style="list-style-type: none"> Sistem akan melakukan koneksi dengan <i>web service</i> Sistem <i>web service</i> mengembalikan hasil berupa data yang telah di-track sesuai dengan <i>query string pre orderID</i> dan <i>pre order date</i> Sistem menampilkan data <i>production follow up</i> hasil penjejukan berdasarkan <i>query</i> |

4.2.3 Pengujian Interaksi Antar Komponen dengan Media Local Area Network (LAN)

Konfigurasi pengujian interaksi antar komponen *Web Service client* dengan *Web Service server* dengan media LAN dapat dilihat pada Gambar 4.1



Konfigurasi pengujian interaksi antar komponen dengan media LAN

Pada Tabel 4.8 dibawah ini menunjukkan hasil pengujian untuk interaksi antara komponen di sisi produksi sebagai *client* dengan komponen di sisi marketing sebagai *Web Service server* menggunakan media LAN.

Hasil pengujian fungsional untuk interaksi antar komponen *Web Service client* dengan *Web Service server* dengan media LAN

| No | Fungsi Yang Diuji | Nama Fungsi/ Store Procedure | Hasil | | Kriteria Keberhasilan |
|-----|--|--|----------|-------|---|
| | | | Berhasil | Gagal | |
| 1. | Instansiasi <i>Web Service</i> yang terpisah secara fisik dengan komputer <i>client</i> sebagai <i>local object</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Sistem dapat melakukan koneksi dengan baik |
| 2. | Visibilitas fungsi beserta parameter masukan untuk setiap fungsi yang disediakan <i>Web Service</i> terhadap aplikasi di sisi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Sistem mampu mengembalikan keluaran sesuai dengan fungsi dan parameter masukan yang disediakan <i>Web Service</i> |
| 3. | Pengecekan validitas <i>user</i> yang akan melakukan transaksi | - | ✓ | | <ul style="list-style-type: none"> ▪ User yang telah memasukkan <i>username</i> dan <i>Id</i> yang sesuai dapat mengakses menu pada program ▪ Ada pesan kesalahan jika user yang dimasukkan tidak valid |
| 4. | Pengaksesan fungsi pada modul <i>Pre Order</i> dari aplikasi <i>client</i> | <i>InsertPreOrder</i> <i>UpdatePreOrder</i> <i>DeletePreOrder</i> | ✓ | | <ul style="list-style-type: none"> ▪ Sistem memasukkan data baru, modifikasi data dan menghapus data pada <i>table pre order</i> ▪ Sistem akan menampilkan <i>warning</i> jika ada kesalahan |
| 5. | Penerimaan keluaran dari fungsi pada modul <i>Pre Order</i> yang dihasilkan <i>Web Service</i> ke aplikasi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Data keluaran ditampilkan pada form modul <i>pre order</i> aplikasi <i>client</i> |
| 6. | Pengaksesan fungsi pada modul sampel dari aplikasi <i>client</i> | <i>InsertSample</i> <i>UpdateSample</i> <i>DeleteSample</i> <i>GetSampleby</i> <i>ParamGet</i> <i>SampleAll</i> | ✓ | | <ul style="list-style-type: none"> ▪ Sistem memasukkan data baru, <i>load</i> data, modifikasi data dan menghapus data pada <i>table sampel</i> ▪ Sistem akan menampilkan <i>warning</i> jika ada kesalahan |
| 7. | Penerimaan keluaran dari fungsi pada modul sampel yang dihasilkan <i>Web Service</i> ke aplikasi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Data keluaran ditampilkan pada form modul <i>sampel</i> aplikasi <i>client</i> |
| 6. | Pengaksesan fungsi pada modul <i>Post Order</i> dari aplikasi <i>client</i> | <i>InsertPostOrder</i> <i>UpdatePostOrder</i> <i>DeletePostOrder</i> <i>GetPostOrderby</i> <i>Param</i> | ✓ | | <ul style="list-style-type: none"> ▪ Sistem memasukkan data baru, modifikasi data dan menghapus data pada <i>table post order</i> ▪ Sistem melakukan pencarian berdasarkan <i>query</i> yang diinginkan |
| 7. | Penerimaan keluaran dari fungsi pada modul <i>Post Order</i> yang dihasilkan <i>Web Service</i> ke aplikasi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Data keluaran ditampilkan pada form modul <i>post order</i> aplikasi <i>client</i> |
| 8. | Pengaksesan fungsi pada modul <i>Development Phase</i> dari aplikasi <i>client</i> | <i>InsertDevPhase</i> <i>UpdateDevPhase</i> <i>DeleteDevPhase</i> <i>GetDvPhaseby</i> <i>Param</i> | ✓ | | <ul style="list-style-type: none"> ▪ Sistem memasukkan data baru, modifikasi data dan menghapus data pada <i>table devphase</i> ▪ Sistem melakukan pencarian berdasarkan <i>query</i> yang diinginkan |
| 9. | Penerimaan keluaran dari fungsi pada modul <i>Development Phase</i> yang dihasilkan <i>Web Service</i> ke aplikasi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Data keluaran ditampilkan pada form modul <i>development phase</i> aplikasi <i>client</i> |
| 10. | Pengaksesan fungsi pada modul <i>Production Follow Up</i> dari aplikasi <i>client</i> | <i>InsertProduction,</i> <i>UpdateProduction</i> <i>DeleteProduction,</i> <i>GetProductionby</i> <i>Param</i> | ✓ | | <ul style="list-style-type: none"> ▪ Sistem memasukkan data baru, modifikasi data dan menghapus data pada <i>table production</i> ▪ Sistem melakukan pencarian berdasarkan <i>query</i> yang diinginkan |
| 11. | Penerimaan keluaran dari fungsi pada modul <i>ProductionFollow Up</i> yang dihasilkan <i>Web Service</i> ke aplikasi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Data keluaran ditampilkan pada <i>form</i> modul <i>production followup</i> aplikasi <i>client</i> |
| 12. | Interaksi antara sistem operasi <i>Microsoft Windows XP</i> dengan <i>Microsoft Windows 2000</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Interaksi berjalan dengan baik tanpa ada pesan kesalahan |
| 13. | Interaksi antara aplikasi <i>client</i> yang dibangun dengan bahasa pemrograman <i>VB.NET</i> dengan <i>Web Service</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Koneksi dan interaksi antar bahasa pemrograman berjalan dengan baik |
| 14. | Interaksi antara <i>Web Service</i> basis data <i>Microsoft SQL Server 2000</i> di sisi <i>server</i> dan <i>MS Access</i> di sisi <i>client</i> | - | ✓ | | <ul style="list-style-type: none"> ▪ Koneksi dan interaksi <i>MS SQL 2000</i> dengan <i>MS Access</i> berjalan dengan baik |

4.2. Analisa Hasil Pengujian

4.2.1 Analisa Fungsional

Analisa komponen melibatkan komponen yang ada pada bagian *Web Service server* yang berada pada sisi *marketing* dan bagian *Web Service client* yang berada pada aplikasi produksi.

1. Analisa Fungsional di Sisi *Marketing*

Dari hasil pengujian yang dilakukan, masing-masing fungsi *use case* berjalan dengan baik dan sesuai dengan kebutuhan. Dari hasil pengujian terhadap akses fungsi yang dilakukan oleh aplikasi *client*, pembuat aplikasi marketing yaitu dengan menambahkan *reference* pada *coding* aplikasi *marketing* terhadap *library filenya*, selanjutnya mengakses fungsi yang terdapat dalam obyek tersebut. Masing-masing fungsi pada setiap modul aplikasi *client* dapat melakukan transaksi untuk *loading* data, pemasukan data baru, modifikasi data serta menghapus data yang tidak aktif lagi berfungsi dengan baik. Sedangkan untuk fungsi penjejakan, masing-masing fungsi dapat melakukan *query* sesuai dengan kriteria yang diminta.

2. Analisa Fungsional di Sisi Produksi

Analisa yang dilakukan terhadap fungsional komponen di sisi produksi pada penelitian ini bertujuan untuk menunjukkan kemampuan dan kemungkinan Web Service melakukan interaksi dan komunikasi dengan *business logic layer* dan basis data yang telah ada pada saat ini pada sisi produksi. Dari hasil pengujian, *Web Service* memiliki kemampuan untuk mengakses fungsi yang terdapat pada *library filenya*. Perangkat lunak *Visual Studio .NET* memungkinkan *Web Service* untuk mengakses aplikasi yang berupa *file* yang sudah di eksekusi

4.2.2 Analisa Interaksi Antar Komponen Dengan Web Service

Dari pengujian yang telah dilakukan terhadap interaksi antara komponen di sisi produksi, *Web Service* memungkinkan terjadinya komunikasi antara aplikasi dengan menggunakan media internet sebagai komunikasi yang bisa dijadikan alternatif sebagai pengganti VPN yang digunakan perusahaan selama ini.

Web Service memungkinkan terjadinya interaksi antar aplikasi atau komponen perangkat atau komponen perangkat lunak terlepas dari *platform* yang digunakan, Penggunaan bahasa pemrograman yang berbeda, sistem operasi yang berbeda, serta konfigurasi jaringan yang berbeda-beda, tidak mempengaruhi pengaksesan dan interaksi antara aplikasi *marketing*, aplikasi produksi dengan *Web Service*.

4.2.3 Analisa Sistem Keamanan

Setiap fungsi yang digunakan dapat dienkripsi dengan baik ditinjau dari *key* yang digunakan. Analisa sistem keamanan digunakan dalam pengimplementasian *Web Service* dalam rangka proses transaksi dan penjejakan data pesanan dalam penelitian ini menggunakan algoritma kriptografi MD5 Hash. Mekanisme yang digunakan adalah melakukan enkripsi *password* yang berbentuk *clear text*. Dimana *client* mengambil *token* dari *Web Service*, kemudian dilakukan *hashing* di sisi aplikasi *client* yang selanjutnya dikirim kepada *Web Service* berupa *key*. *Web Service* akan melakukan otentikasi dengan cara membandingkan *key* yang diterima dari *client* dengan *key* yang ada pada *Web Service*. Jika *key* yang digunakan sama maka setiap transaksi baru dapat dilakukan.

PENUTUP

5.1 Kesimpulan

Berdasarkan analisa hasil pengujian maka dapat ditarik beberapa kesimpulan diantaranya adalah :

1. *Web Service* merupakan sebuah teknologi yang layak diterima dan digunakan sebagai suatu alternatif penyelesaian masalah integrasi aplikasi perusahaan dalam kasus transaksi dan data pesanan secara lebih efektif dan efisien;
2. *Web Service* mampu mengakomodasi masalah interoperabilitas sistem pada jaringan internet, hal ini didukung dengan bahasa XML sebagai standar pengkodean yang diterima sebagai bahasa *universal*, SOAP sebagai standar pemaketan pesan untuk melakukan komunikasi antar sistem, serta WSDL untuk mendeskripsikan layanan *Web Service* yang keseluruhannya berbasis XML; dari hasil pengujian terbukti bahwa *Web Service* mampu menembus *firewall* karena *Web Service* menggunakan HTTP sebagai protokol standar;
3. Enkripsi dengan MD5Hash pada level pengamanan *password* merupakan solusi penerapan sistem keamanan yang cukup baik diterapkan dalam implementasi *Web Service*;
4. Implementasi integrasi aplikasi pada perusahaan antara sisi produksi dengan sisi *marketing* membutuhkan usaha yang relatif kecil baik dari sumber daya manusia maupun infrastruktur, salah satu alasan memilih teknologi *Web Service* ini untuk integrasi aplikasi pada perusahaan ini adalah sumber daya, dimana dalam kondisi nyata antara bagian *marketing* dan bagian produksi telah

memiliki infrastruktur yang memungkinkan penerapan teknologi ini seperti tersedianya akses internet, perangkat keras maupun perangkat lunak yang *support* terhadap teknologi *Web Service*;

5.2. Saran

Saran yang diberikan untuk pengembangan lebih lanjut adalah sebagai berikut :

1. Rancangan dapat dikembangkan tidak hanya untuk kasus internal perusahaan tetapi juga *supplier* dan *customer* serta dengan perusahaan rekanan lainnya untuk menciptakan *supply chain management* yang lebih baik;
2. Untuk menghindari kerentanan karena menggunakan jaringan internet yang bersifat terbuka dan protokol SOAP yang berbasis XML, dibutuhkan sistem keamanan yang lebih andal dalam pengimplementasian *Web Service* selanjutnya, baik dari format datanya, keamanan pada level *application layer* yang bersifat *end-to-end* serta bagaimana pengaruhnya terhadap kinerja serta kecepatan pengiriman data terhadap sistem keamanan yang digunakan.

DAFTAR PUSTAKA

1. Basiura, Russ, Batongbacai, Mike, Bohling, Brandon, *Professional ASP .Net Web Service*, WROX Press Ltd, 2001
2. Chappel, David, *Understanding .NET A, Tutorial and Analysis*, Addison-Wesley, 2002.
3. Fred A Cummins, *Enterprise Integration*, OMG Press, 2002.
4. I Made Artawan, *Web Service dan Penerapannya Untuk Integrasi Aplikasi Terdistribusi*, Departemen Teknik Elektro, Institut Teknologi Bandung, 2003.
5. Rafmawan, Ahmad, *Perancangan Sistem Informasi Layanan Pelanggan Perusahaan Asuransi Yang Terdistribusi, Suatu Aplikasi Web Service*, Departemen Teknik Elektro, Institut Teknologi Bandung, 2004.
6. Riyanto T, Bambang, Ni Nyoman Victoria, *Paper Web Service, Impelementasi Integrasi Aplikasi Terdistribusi dengan Web Service*, Departemen Teknik Elektro, Institut Teknologi Bandung, 2003.
7. Short, S, *Building XML Web Services for the Microsoft .NET Platform*, Microsoft Press, 2002.
8. Alan Brown, Simon Johnston, Kevin Kelly, *Using Service-Oriented Architecture and Component-Based Development to Build Web Service Application*, 2003, available at : <http://www.ibm.rational/>
9. Minder Chen, *Building XML Web Services for E-Business Applications with .NET Framework and Visual Studio.NET*, 2002, available at : <http://www.msdn.com/>
10. Scott Borg, *Real-Time Profit Optimization : Coordinating Demand and Supply Chain Management* available at : <http://www.cisco.system/>
11. Maria Tzvetanova, *Security for Web Service*, Master Thesis, University of Constance, May 2003, available at: http://www.inf.uni-konstanz.de/~tzvetano/assets/docs/security_for_webservices.pdf