

XML

by Soetam Rizky

Submission date: 25-May-2022 08:37PM (UTC-0400)

Submission ID: 1844262161

File name: 16940-17447-2-PB.pdf (306.21K)

Word count: 3514

Character count: 21097

2
Seediscussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/233946910>

IMPLEMENTASI EXTENSIBLE MARKUP LANGUAGE WEB SERVICE PADA SECURE REMOTE FORM FIELD FILLIN DALAM DOKUMEN MICROSOFT WORD WEB EXTENSIBLE MARKUP LANGUAGE IMPLEME....

Article · June 2008

CITATION

1

READS

276

11 authors, including:



Soetam Rizky

Ma Chung University

91 PUBLICATIONS 37 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Ujian Akhir Semester - Tutorial IT For Human Resource - Universitas Ma Chung ² [View project](#)

IMPLEMENTASI *EXTENSIBLE MARKUP LANGUAGE* *WEB SERVICE* PADA *SECURE REMOTE FORM* *FIELD FILL* DALAM DOKUMEN *MICROSOFT WORD*

Soetam Rizky Wicaksono
Program Studi Sistem Informasi STIKOMP SURABAYA

WEB EXTENSIBLE MARKUP LANGUAGE IMPLEMENTATION SERVICE ON SECURE REMOTE FORM FIELD FILL IN MICROSOFT WORD DOCUMENT

Abstract: Remote form field fillin usually being done by institutions for many purposes. Common method that has been used for remote form field fillin in a web page. However, in many cases, there are abundant difficulties to do this job such as : slow or limited internet connection. Therefore some users can not fill the form easily. To resolve this problem, this research tried to solve remote form field fillin with secure login using Extensible Markup Language (XML) Web Service. XML Web Service itself is being injected in Visual Basic for Application at Microsoft Word document. Microsoft Word document was chosen because its flexibility and its widespread usage in Indonesian users, and also the consideration of easiness in applying disconnected concept in it. While XML Web Service can do submitting process for remote form field fillin, it also can server severe security in data transfer process.

Keywords: Form Field Fillin, Extensible Markup Language (XML) Web Service, Microsoft Word

Saat salah satu penyedia bursa kerja di internet menawarkan pengisian formulir dengan jumlah *field* isian yang cukup besar, bahkan hingga lebih dari 30 isian, maka seorang pengguna sempat bertanya sekaligus mengeluh tentang lamanya waktu koneksi serta banyaknya *field* yang harus diisi. Selain itu, beberapa formulir membutuhkan pemikiran yang lama karena butuh pencarian data pribadi, misalnya

pengisian mengenai nomor identitas, nomor ijazah, ataupun mengenai hasil pendidikan informal, yang seringkali membutuhkan pencarian data ulang secara personal.

Karena proses yang sedemikian besar dan lama, maka pengisian formulir seringkali gagal dilakukan karena terbatasnya koneksi internet. Dari salah satu kasus tersebut, dibutuhkan sebuah metode pengisian

Alamat Korespondensi:

Soetam Rizky Wicaksono, Program Studi Sistem Informasi, STIKOMP SURABAYA Jl. Raya Kedung Baruk No. 98, Surabaya. Telp: (031) 8413889, Fax: (031) 8436308, Email: soetam@stikom.edu

formulir secara jarak jauh dengan metode koneksi seminim mungkin, tetapi tetap dengan *level* keamanan tertentu untuk menjaga privasi si pengisi. Hal ini diterapkan, khususnya bagi pengguna yang melakukan pengisian formulir di tempat publik, seperti warung net ataupun di area kantor, sehingga untuk pengisian formulir yang dianggap rahasia bagi personal sangat tidak mungkin diisi dengan waktu yang lama.

Kerangka berpikir yang diterapkan untuk mengatasi masalah tersebut adalah dengan menerapkan konsep pengisian formulir yang dapat menyebabkan pengguna tidak terlalu lama melakukan koneksi internet, dan merasa nyaman melakukan pengisian formulir dengan koneksi *-semi online* atau lazim disebut sebagai konsep *disconnected*. Untuk itu, terdapat dua pilihan solusi yaitu dengan menggunakan teknologi *Asynchronous JavaScript and XML (AJAX)* (telah dilakukan di penelitian yang lain, dan menggunakan teknik *online* penuh) dan dengan memanfaatkan solusi *XML Web Service* yang dipadukan dengan aplikasi perkantoran, dalam kasus ini dipilih aplikasi *Microsoft Office* (khususnya *Microsoft Word*) dengan asumsi paling banyak digunakan dan memiliki *Visual Basic for Applications (VBA)* sebagai *embedded programming* di dalamnya.

Penggunaan *Microsoft Word* dalam penelitian ini menggunakan perangkat bantu *VBA* di dalamnya. Sejak *Microsoft Office* versi *XP*, *VBA* telah memiliki utilitas yang mampu mempermudah proses pengambilan nilai dari *XML Web Service*, yaitu *Office Web Services Toolkit*. Bahkan sejak *Microsoft Office 2007*, pemanfaatan *XML Web Service* telah menyatu sebagai fasilitas *built-in*.

Dengan adanya utilitas ini, maka *XML Web Service* yang diambil akan dibangkitkan menjadi *proxy class*, sehingga pemanggilan fungsi dapat menjadi lebih mudah dan cepat.

Pembuatan *form field fillin* dalam eksperimen ini memanfaatkan fitur *form control* dan *control toolbox* yang diintegrasikan dengan *VBA* yang akan memanggil fungsi dari *proxy class* hasil dari *XML Web Service*. Dalam *form field* ini, hasil pengisian formulir yang telah diproteksi dengan *user login* dan *password* akan menjadi nilai umpan balik yang selanjutnya akan diproses lebih lanjut di *web server*, baik intranet maupun internet.

Dari paparan latar belakang masalah yang telah disebutkan, jelaslah bahwa perlu dilakukan implementasi *secure remote form field fillin* dengan menggunakan dokumen *Microsoft Word* sebagai “jembatan” untuk mengisikan data ke dalam *web server* serta melakukan implementasi proses *fetching* data dengan menggunakan validasi dari dokumen *Microsoft Word* dengan menggunakan akses *XML Web Service* di dalam *VBA*.

Dalam eksperimen, *server* basis data yang digunakan adalah *Structured Query Language (SQL) Server* yang diasumsikan sebagai *server* basis data yang terletak di *web server*, meskipun pada implementasi lebih lanjut dapat menggunakan *server* basis data lain, misalnya: *Oracle*, *MySQL* ataupun *Firebird*. Sedangkan, bahasa pemrograman yang digunakan untuk membuat *XML Web Service* adalah *Visual Basic .NET 2005*.

Konsep aplikasi basis data yang terputus atau *disconnected* lahir seiring dengan lahirnya *ActiveX Data Objects (ADO) .NET* yang menjadi pengjawantahan koneksi basis data dalam lingkup *.NET*

Framework. Di dalam *ADO .NET* sendiri, koneksi basis data dibagi menjadi dua bagian yaitu *connected* dan *disconnected*. Konsep *disconnected* sendiri merupakan konsep yang memperbolehkan data untuk diambil dari sumber data dan kemudian dimanipulasi, sedangkan proses rekonsiliasi atau sinkronisasi data dapat dilakukan kemudian (Hamilton & MacDonald, 2003).

Dengan kata lain, konsep *disconnected* merupakan konsep koneksi *basis data* yang mengambil sumber data, kemudian menerapkannya secara *offline*. Sehingga, pada saat proses manipulasi data tidak diperlukan koneksi langsung ke sumber data yang asli. Koneksi data hanya akan diperlukan lagi pada saat akan terjadi proses sinkronisasi data, baik berupa penyimpanan data ataupun pemanggilan ulang data (Sceppa, 2002).

Dalam implementasi konsep *disconnected* dengan menggunakan *ADO .NET*, obyek utama yang dibutuhkan adalah *dataset*. Dengan menggunakan *dataset* yang sesungguhnya merupakan representasi dari dokumen *XML*, maka konsep *disconnected* dapat diimplementasikan dengan lebih baik. Salah satu kelebihan dari *dataset* ini adalah adanya kemampuan untuk mengkombinasikan berbagai sumber data yang berbeda di dalam satu aplikasi. Sehingga, dapat terjadi *unification* data dalam satu aplikasi kelas *enterprise* meski sumber data yang diperoleh memiliki format yang berbeda (Joubert & Ryan, 2003).

XML Web Service adalah fungsi logika yang dapat diakses dengan menggunakan protokol standar yaitu *HTTP*. *XML Web Service* juga merupakan sebuah fungsi yang dapat mengintegrasikan berbagai sistem dari bermacam bahasa pemrograman serta *basis data* yang berbeda (Rizky, 2008).

Di dalam referensi lainnya, *Web Service* didefinisikan sebagai sebuah layanan yang tersedia via internet dengan menggunakan *XML* sebagai sistem dalam menyampaikan pesan dan tidak tergantung kepada bahasa pemrograman tertentu (Cerami, 2005).

Khususnya di dalam lingkup *.NET Framework* yang dikembangkan oleh *Microsoft*, *Web Service* menjadi pokok bahasan utama dalam pengembangan aplikasi kelas *enterprise* yang melibatkan aplikasi berbasis web dan aplikasi berbasis *desktop* dalam satu layanan yang sama. Dalam konteks tersebut, *XML Web Service* menjadi “jembatan” utama bagi *layer* yang berada dalam lingkup aplikasi yang terintegrasi tersebut (Foggon, 2004).

Secara khusus, *XML Web Service* memiliki beberapa karakteristik penting yang wajib untuk dipahami yaitu (Rizky, 2008): (1) standar dari *XML Web Service* adalah standar industri yang independen, (2) *XML Web Service* tidak menyediakan antar muka, tetapi hanya menyediakan fungsi, (3) *XML Web Service* menggunakan *HyperText Transfer Protocol (HTTP)* sebagai protokol standar, (4) *XML Web Service* menggunakan model *request-response* yang sama seperti pada aplikasi web, (5) *XML Web Service* ditempatkan di dalam sebuah web server, (6) *XML Web Service* lebih ditujukan untuk kepentingan dari *programmer* ke *programmer*, sehingga pengguna *XML Web Service* yang utama bukanlah pengguna akhir tetapi *programmer* yang menggunakan bahasa pemrograman/basis data/platform yang berbeda, (7) hasil dari *XML Web Service* merupakan *plain text* yang berupa berkas *XML*, (8) fungsi dari sebuah *XML Web Service* bukanlah semacam sulap yang dapat langsung “muncul” dalam sebuah *interface* aplikasi, tetapi

perlu sebuah pengolahan lebih lanjut, (9) secara umum, hampir semua bahasa pemrograman yang mampu melakukan koneksi ke internet dan membaca berkas *XML*, akan mampu pula “mengonsumsi” *XML Web Service*, tapi tidak semua bahasa pemrograman tersebut mampu “memproduksi” *XML Web Service*, (10) dalam perkembangannya, beberapa hasil implementasi dari *XML Web Service* tidak lagi dapat dikonsumsi secara bebas oleh banyak bahasa pemrograman, bergantung kepada model dan jenis nilai balik yang ada dalam *XML Web Service* tersebut, misalnya: hasil *XML Serialization*, *XML Schema* dan lainnya.

VBA merupakan anggota dari keluarga *Visual Basic (VB)* yang posisinya berada di antara *VB* dan *Visual Basic Script (VB Script)* secara fungsional (Simon, 2002). Meski *VBA* mewarisi banyak fitur dari *VB* seperti penggunaan *form*, *controls*, *object* dan *modules*, tetapi *VBA* memiliki satu perbedaan yang sangat mendasar dengan *VB*. Perbedaan tersebut adalah bahwa *VBA* hanya akan mendukung lingkup aplikasi yang ditempatinya (*host environment*). Sebagai contoh adalah: *VBA* yang ada di dalam *Microsoft Word* akan berbeda dengan *VBA* yang ada di dalam *Microsoft Excel* dan sebaliknya.

VBA sesungguhnya juga berbeda dengan *Macro* yang biasa dikenal di lingkup *Microsoft Office*, karena *Macro* adalah bagian dari *VBA* yang merangkum kumpulan perintah dan fungsi, sehingga dapat membantu pengguna untuk lebih cepat mengerjakan tugasnya. Dan apabila *Macro* ingin dikembangkan, maka *Macro* tersebut harus dikerjakan ulang dengan menggunakan *VBA*. Dengan kata lain, *Macro* sebenarnya berfungsi seperti halnya prosedur atau fungsi di dalam *VBA* (Walkenbach, 2007).

Sedangkan, pengembangan *VBA* secara umum (tanpa memandang *host environment*) selalu ditandai dengan implementasi fungsi di dalamnya. Dari fungsi inilah yang akan menjadi penghubung utama antara aplikasi yang ditempati (dalam penelitian ini adalah *Microsoft Word*) dengan *VBA*. Dan dari fungsi itu pula yang nanti akan menjadi *otak* utama dalam pengembangan pemrograman dalam *host environment* pada *VBA* (Rizky, 2007)

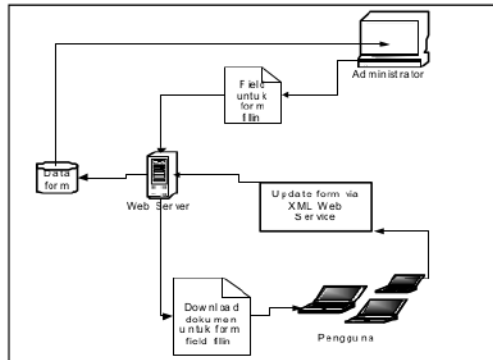
Salah satu fitur utama pengembangan *VBA* di dalam lingkup *Microsoft Office* (sejak *Microsoft Office XP* dengan adanya utilitas *Office Web Services Toolkit*, dan sejak *Microsoft Office 2007* telah terintegrasi sebagai satu bagian) adalah adanya dukungan terhadap penggunaan *XML Web Service*, sehingga mampu melakukan konsumsi atau akses terhadap *XML Web Service* dengan menggunakan *proxy class*. Hal ini berarti bahwa akses dokumen dari *Microsoft Office* dapat langsung diintegrasikan ke dalam sebuah aplikasi web dengan menggunakan *XML Web Service* (Wicaksono, 2007).

METODE

Aplikasi yang akan dibuat menggunakan masukan tunggal yang berupa dokumen atau berkas *office* yang akan dimodifikasi di dalamnya serta dilakukan injeksi *VBA* dengan memanfaatkan *XML Web Service* di dalamnya. Di dalam penelitian situs yang sesungguhnya dianggap sebagai sebuah situs *dummy* untuk uji coba hasil masukan dari dokumen *Microsoft Word*.

Alur arsitektur sistem diawali dari pihak *web-master* atau *administrator* yang menentukan beberapa *field* apa saja yang akan dilibatkan di dalam pengisian formulir. Setelah analisa tersebut selesai dilakukan, maka *administrator* akan menerbitkan dokumen

Microsoft Word yang di dalamnya telah termodifikasi dengan beberapa *field* isian yang diperlukan. Selanjutnya, dokumen tersebut diinjeksi dengan *VBA* untuk koneksi ke *XML Web Service* yang mengacu ke basis data yang berada di dalam *web server*. Arsitektur sistem dapat dilihat pada Gambar 1.



Gambar 1 Arsitektur Sistem

Dokumen *Microsoft Word* tersebut, kemudian menjadi *template* bagi pengguna untuk kemudian diisi dengan data yang dibutuhkan. Pengguna diasumsikan telah memiliki *user id* dan *password* sebelumnya tanpa harus melakukan registrasi terlebih dahulu di dalam situs registrasi. *User id* dan *password* tersebut, akan menjadi kunci keamanan dalam dokumen *Microsoft Word* (yang telah terinjeksi oleh *VBA*) saat akan melakukan proses *submit* data ke basis data yang berada di dalam *web server*.

XML Web Service yang dibuat, terdiri dari dua bagian utama yaitu bagian otorisasi dan bagian penyimpanan data ke dalam basis data. Kedua bagian tersebut, akan mengembalikan nilai yang berbeda, sebagai umpan balik terhadap pengguna pada saat proses koneksi pertama kali dilakukan dan pada saat proses penyimpanan data berhasil dijalankan.

Bagian yang pertama yaitu bagian otorisasi, akan menggunakan tabel yang berisi *user id* dan *password* untuk otorisasi. Selanjutnya, dari fungsi tersebut dilakukan *query* dengan menggunakan *data reader* untuk kemudian mengembalikan nilai balik bertipe *boolean* yang menyatakan keberhasilan proses otorisasi dalam dokumen *Microsoft Word*.

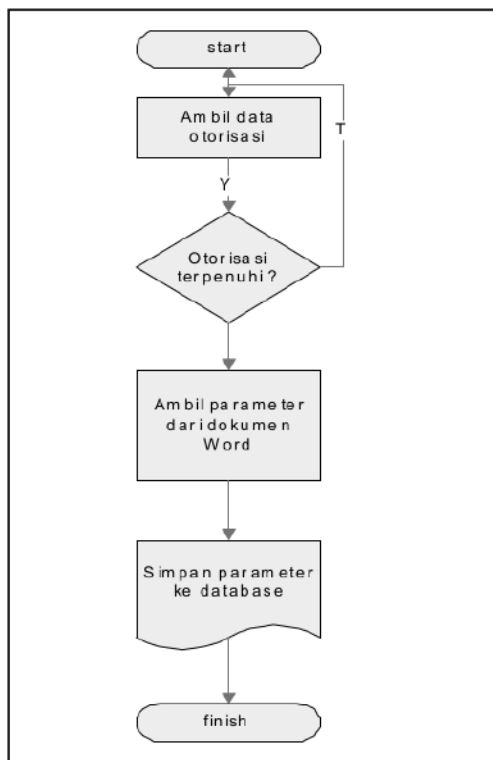
Dari hasil keberhasilan proses otorisasi tersebut, lalu disimpan ulang ke dalam tabel yang berisi *user id* dan *password* mengenai keberadaan *user* tersebut (sedang dalam status pengisian data), sehingga nantinya pada saat proses penyimpanan data tidak akan terjadi kesalahan *concurrency data*.

Sedangkan, bagian yang kedua yaitu bagian penyimpanan data, pada saat awal akan membaca hasil dari bagian yang pertama. Apabila pada bagian pertama telah menghasilkan nilai balik *true*, maka proses akan dilanjutkan. Sedangkan, jika nilai balik yang dihasilkan adalah *false*, maka proses akan dihentikan dan akan dilemparkan pesan kesalahan kepada pengguna. Pada saat proses dihentikan, maka alur program akan dikembalikan ke proses otorisasi.

Jika proses otorisasi berhasil dilalui, maka langkah berikutnya adalah melakukan pengambilan parameter dari dokumen *Microsoft Word* yang dilakukan dengan menggunakan bantuan *VBA*. Dari hasil pengambilan parameter itu, maka fungsi di bagian kedua *XML Web Service* akan mengambil alih tugas selanjutnya, yaitu menyimpan parameter yang berhasil diambil ke dalam basis data yang berada di *web server*.

Pada saat proses penyimpanan data di *web server* selesai dilakukan, maka pengguna akan diberikan nilai balik ulang dari *web server* mengenai keberhasilan proses. Dan pada titik ini, proses yang

ada dianggap selesai dilakukan. Untuk lebih jelas memahami mengenai proses yang terjadi di dalam *XML Web Service*, dapat dilihat pada diagram alir Gambar 2.



Gambar 2 Diagram Alir *XML Web Service*

Injeksi *VBA* ke dalam dokumen *Microsoft Word* pada penelitian ini, membutuhkan dua syarat penting yaitu *setting* sekuritas *Microsoft Word* yang diset menjadi *enable* agar fungsi yang ada dalam *VBA* dapat dieksekusi, serta kapabilitas koneksi terhadap *XML Web Service* yang berarti kemampuan untuk melakukan koneksi ke *web server* yang di dalamnya telah terdapat fungsi *XML Web Service* yang siap untuk digunakan.

Fitur utama dari *XML Web Service* yang dikembangkan oleh *Microsoft*, khususnya dalam lingkup *.NET Framework* adalah kemampuan untuk membuat *proxy class* atau fungsi temporer yang mampu menghasilkan implementasi konsep *dis-connected* di dalamnya. Pembuatan *proxy class* sendiri membutuhkan langkah awal yang dinamakan *discovery* yang berarti pencarian alamat *XML Web Service* yang valid. Jika proses pencarian telah berhasil dilakukan, fungsi yang ada di dalam *XML Web Service* bisa dieksekusi melalui obyek baru yang dideklarasikan di dalam fungsi *VBA*.

Dalam penelitian ini, *VBA* dibagi menjadi dua bagian utama yang mirip dengan bagian yang terdapat dalam *XML Web Service* yaitu bagian otorisasi dan bagian penyimpanan data. Pada saat awal bagian otorisasi, fungsi dalam *VBA* mengeksekusi fungsi *XML Web Service* melalui deklarasi obyek baru. Pendeklarasian tersebut, sekaligus melakukan dua hal penting yaitu proses *discovery XML Web Service* dan pembentukan *proxy class* untuk pemanggilan fungsi *XML Web Service* yang lain.

Apabila *proxy class* berhasil dibentuk, parameter otorisasi akan dilempar melalui fungsi *XML Web Service* untuk dieksekusi di *web server*. Lalu, fungsi di *VBA* mengambil nilai balik dari fungsi otorisasi yang berupa nilai balik bertipe *boolean*. Dari hasil ini akan menjadi dasar utama bagi pemanggilan fungsi *XML Web Service* selanjutnya.

Saat nilai balik bernilai *true*, maka bagian kedua dari fungsi *VBA* mulai dijalankan. Bagian kedua yang berfungsi untuk melakukan proses *submit* data ke *web server*, pada dasarnya merupakan fungsi yang mengambil parameter berasal dari kumpulan *field* yang sebelumnya telah terdefinisi di dalam dokumen

Microsoft Word. Kumpulan *field* yang nilainya diambil menjadi kumpulan parameter akan dilempar ulang dari fungsi *VBA* ke dalam *XML Web Service*, dan tugas penyimpanan selanjutnya terjadi di dalam *web server*.

Pada saat proses penyimpanan sukses dilalui, maka *XML Web Service* akan berusaha mengirimkan nilai balik berupa pesan sukses ke *VBA*. Pada titik ini, proses di *VBA* dianggap telah selesai dan secara keseluruhan proses *secure remote form field filling*

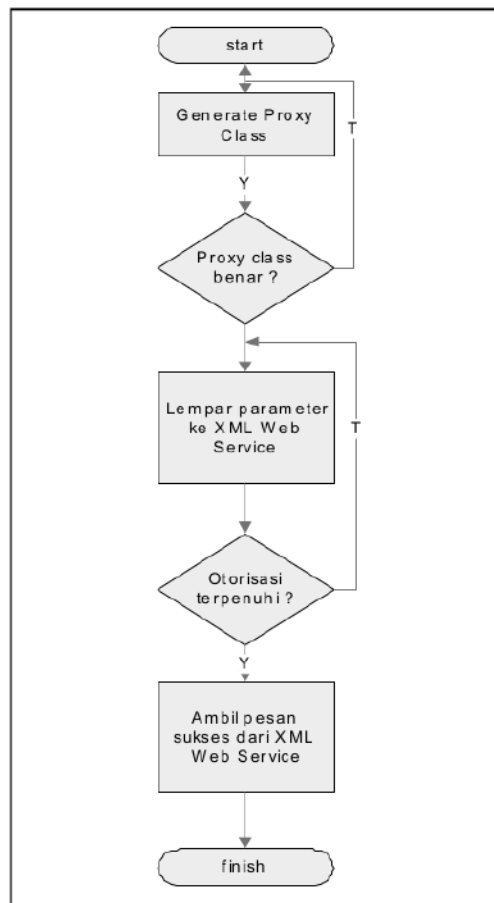
diasumsikan telah sukses. Untuk lebih jelas memahami proses di dalam *VBA*, dapat dilihat pada diagram alir Gambar 3.

HASIL DAN PEMBAHASAN

Dari perancangan yang telah dilakukan, maka dilakukan implementasi, dengan menggunakan sebuah situs *dummy* sebagai tempat penampungan sementara sekaligus bertindak sebagai *web server*. Selain sebagai *web server*, di dalamnya juga diasumsikan terdapat *server* basis data yang nantinya sebagai penampung data hasil isian formulir. Formulir yang asli di dalam situs, tidak ditampilkan, mengingat implementasi dapat dilakukan di dalam situs jenis apapun yang mengandung pengisian formulir.

Di dalam *web server* tersebut, terdapat *XML Web Service* yang bertindak sebagai jembatan penghubung dalam proses pengisian formulir. *Discovery XML Web Service* diasumsikan sama dengan alamat situs *dummy* yang diimplementasikan. Sedangkan, sekuritas *XML Web Service* dalam kasus ini dianggap menggunakan *setting default* atau diabaikan.

Sedangkan, dokumen *Microsoft Word* yang dijadikan sebagai bahan uji coba hanya terdiri dari tiga *field* untuk kepentingan kecepatan proses pada saat evaluasi dilakukan. Di dalam dokumen inilah terjadi juga implementasi konsep *disconnected*. Dokumen yang di dalamnya telah diinjeksi dengan *XML Web Service* di dalam *VBA*, hanya akan dieksekusi pada saat tombol *Submit Data* dilakukan. Apabila dokumen tersebut disimpan dengan menggunakan fasilitas yang ada di dalam *Microsoft Word* (baik penggunaan fitur *Save* ataupun *Save As*), tidak akan terjadi koneksi apapun ke situs yang dituju.



Gambar 3 Diagram Alir Proses dalam *VBA*

Bentuk dokumen yang di dalamnya telah diinjeksi merupakan bentuk dokumen isian dengan menggunakan fasilitas *field* yang terdapat dalam *VBA*, sehingga pengguna tidak lagi bisa melakukan *edit* terhadap dokumen, selain isian yang ada dalam *field*. Sedangkan, untuk pengisian *user id* dan *password* yang tersedia (sebagai bentuk pengamanan dalam penelitian ini), menggunakan fasilitas *form control* dalam *VBA* berupa *textbox*, dan juga bantuan *form control* berupa *button* untuk tombol proses *submit* data.

Pada saat proses pengisian dilakukan (jika memang bermaksud untuk melakukan *submit* data, dan proses pengisian telah dianggap selesai), tombol *submit* data akan langsung melakukan eksekusi *VBA* dengan injeksi *XML Web Service* untuk melakukan koneksi ke *web server*. Jika asumsi *user id* dan *password* telah benar dan sesuai dengan data di dalam *web server*, seluruh *field* yang ada akan dieksekusi untuk proses penyimpanan selanjutnya.

Di dalam proses tersebut, pengguna tidak akan pernah mengetahui proses yang terjadi, berkat adanya sifat enkapsulasi di dalam *VBA* saat melakukan eksekusi *XML Web Service*. Selain itu, pengguna hanya perlu melakukan koneksi ke *web server* hanya pada saat tombol *submit* dieksekusi. Sehingga, proses yang dilakukan dapat menghemat penggunaan

User ID	
Password	

Data Customer PT "NADIVA"

Nama :

Alamat :

Telp :

Gambar 4 Screenshot Dokumen Sebelum Pengisian

bandwidth dari pengguna. *Screenshot* dokumen sebelum pengisian dapat dilihat pada Gambar 4, sedangkan untuk *screenshot* dokumen setelah pengisian ditampilkan pada Gambar 5.

User ID	admin
Password	*****

Data Customer PT "NADIVA"

Nama : Alibaba

Alamat : Jl. Raya Kedung Baruk 908

Telp : 829739222

Gambar 5 Screenshot Dokumen Setelah Pengisian

SIMPULAN

Dalam penelitian ini, implementasi *secure remote form field fillin* diharapkan dapat memberikan solusi bagi *webmaster* yang memiliki formulir isian dengan jumlah yang banyak atau formulir isian yang membutuhkan waktu lama dalam proses pengisiannya. Dari sisi keilmuan, dengan penerapan *secure remote form field fillin* akan menjadi sebuah solusi baru dalam mengintegrasikan aplikasi *desktop* dan *web* melalui *XML Web Service*. Jika pada umumnya solusi yang diterapkan menggunakan aplikasi *desktop* normal (dengan pembuatan *executable program*), maka dalam penelitian diberikan solusi baru dengan menggunakan bantuan dokumen *Microsoft Word* yang diasumsikan telah jamak dimiliki oleh para pengguna (khususnya di Indonesia).

Di sisi lain, hasil dari penelitian ini masih memiliki beberapa kelemahan yang harus diperbaiki di penelitian lanjutan, yaitu keamanan dari dokumen *Microsoft Word* yang memerlukan pengaktifan

fasilitas *VBA* pada saat *setting* awal. Selain itu, sekuritas pengiriman data perlu diberikan metode enkripsi di dalamnya agar pada saat terjadi proses submit, data tidak bisa di-*sniffing* oleh pihak ketiga, mengingat sifat *XML* yang diumpankanbalikkan merupakan *plain text*.

Dalam pengembangannya, penelitian ini diharapkan bisa berlanjut menjadi sebuah dokumen *standard* berbentuk *template* yang dapat melakukan proses pembangkitan formulir isian dari sebuah aplikasi web yang membutuhkan solusi *secure remote form field*.

RUJUKAN

Cerami, E. 2005. *Web Service Essentials*. California: O'Reilly
Foggon, D etall. 2004. *Programming Microsoft .NETXML Web Services*. Washington: Microsoft Press

4

Hamilton, B & MacDonald, M. 2003. *ADO .NET in a Nutshell*. California: O'Reilly
Joubert, TJ & Ryan NP. 2003. *ADO .NET Programming*. Plano: Wordware
Rizky, S. 2008. *Sistem Informasi Terintegrasi dengan Menggunakan XML Web Services*. Surabaya: Cerdas Pustaka
Rizky, S. 2007. *Mudah dan Tepat Optimalisasi Microsoft Excel*. Surabaya: Prestasi Pustaka
Sceppa, D. 2002. *Microsoft ADO .NET*. Washington: Microsoft Press
Simon, J. 2002. *Excel Programming*. New York: Hungry Minds
Walkenbach, J. 2007. *Excel 2007 VBA Programming for Dummies*. Indianapolis: Wiley
Wicaksono, SR. 2007. *Secure Remote Form Field Fillin Dalam Dokumen Word Melalui XML Web Service (Call for Paper SITIA 2007)*. Surabaya: ITS

ORIGINALITY REPORT

4%

SIMILARITY INDEX

%

INTERNET SOURCES

2%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Universitas Jenderal Achmad Yani

Student Paper

1 %

2

Atulkumar, Tailor Manthan. "Assessment of Nutrient Dynamics and Physico - Chemical Status of Freshwater Reservoirs of Vadodara District, Gujarat, India.", Maharaja Sayajirao University of Baroda (India), 2020

Publication

1 %

3

Submitted to Universidad Ricardo Palma

Student Paper

1 %

4

V. S. Giridhar Akula. "chapter 40 Rule-Based Systems for Medical Diagnosis", IGI Global, 2017

Publication

<1 %

5

McKibbin, William J. "Isoquantal capital modulation: A harmonic modeling approach to understanding and managing the investment decision", Proquest, 20111109

Publication

<1 %

Chandra Kirana, Burham Isnanto. "Face Identification For Presence Applications Using ViolaJones and Eigenface Algorithm", Jurnal Sisfokom (Sistem Informasi dan Komputer), 2016

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12