

KONSOLIDASI DATABASE KEPENDUDUKAN NASIONAL

Riana Kuswandi¹, Husni Fahmi², Haret Faidah², Herald Setiadi³, Zainal Hasibuan³

¹ riana.kuswadi@sgu.ac.id. Swiss German University, Serpong, Indonesia

² {haret, fahmi} @inn.bppt.go.id. PTIK-BPPT Building II 21st Floor, Jl. MH Thamrin 8, Jakarta, Indonesia

³ heraldpana@yahoo.com, zhasibua@cs.ui.ac.id. Lab e-Gov Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia, Salemba, Jakarta, Indonesia

Abstrak

Berkembangnya teknologi telah membawa dampak yang sangat berarti terhadap cara manusia mempertukarkan data. Teknologi terbaru pada komunikasi data (*wireless*, satelit, *fiberoptic*, dan lain-lain) dan juga pada infrastruktur teknologi informasi seperti jaringan, *database*, *tools*, maupun *compiler-compiler software* telah banyak membawa beberapa keuntungan dalam hal efisiensi, keefektifan, dan keamanan data. Beberapa teknologi pertukaran data seperti *Web Services* yang berbasiskan pada struktur data standar yang berbasiskan teks seperti *XML* sekarang menjadi isu penting dalam meminimalisasi kompleksitas pertukaran data antar *database* yang berbeda dengan sistem operasi yang berbeda pula. Kaitannya dengan pertukaran data antara *database* yang berbeda dan sistem operasi yang berbeda, hal penting yang harus diperhatikan adalah keamanan data yang dipertukarkan. Aspek *Confidentiality*, *Integrity*, dan juga *Availability* dari sistem harus tetap terjaga dengan baik sebelum, selama dan sesudah pertukaran data. Makalah ini membahas pertukaran *database* dan sistem operasi yang berbeda antara aplikasi kependudukan dan data dapat dikonsolidasikan ke pusat dengan tetap menjaga keamanan dan konsistensinya.

Kata kunci: *Web services*, *XML*, *SOAP*, Enkripsi, Dekripsi

1. PENDAHULUAN

Latar belakang

Saat ini Departemen Dalam Negeri sedang mengembangkan suatu sistem informasi kependudukan nasional yang diberi nama Sistem Informasi Administrasi Kependudukan atau disingkat SIAK. SIAK ini dikembangkan bermula dari kebutuhan akan satu basis data yang tersentralisasi. Yang terjadi selama ini adalah propinsi-propinsi telah mengembangkan sistem kependudukannya masing-masing. Karena tidak ada standarisasi dan acuan dari Pusat, maka propinsi-propinsi ini menggunakan sistem operasi (*operating system*) dan sistem manajemen basis data (*database management system*) yang berbeda-beda. Salah satu tujuan SIAK adalah mengatasi adanya keragaman sistem operasi dan basis data di daerah, yaitu mengumpulkan salah satu sistem kependudukan nasional dan melakukan koordinasi dan pengelolaan yang berkesinambungan dari informasi kependudukan di berbagai daerah di Indonesia. SIAK jelas membutuhkan sistem basis data (*database*) terpusat untuk menyimpan informasi kependudukan ini. Basis data ini terdiri dari sistem basis data pusat yang ada di Jakarta dan sistem basis data daerah yang tersebar di banyak propinsi di seluruh wilayah Indonesia. Sistem basis data pusat ini mendapatkan data dari sistem basis data di propinsi-propinsi. Ada beberapa hal yang harus diperhatikan dalam penggabungan basis data ini. Pertama, setiap basis data menggunakan sistem operasi dan sistem manajemen basis data yang berbeda-beda. Kedua, basis

data ini tersebar di wilayah yang secara geografis saling berjauhan. Terakhir, tidak ada atau kurangnya sistem komunikasi terhubung yang berkesinambungan karena kurangnya infrastruktur komunikasi saat ini.

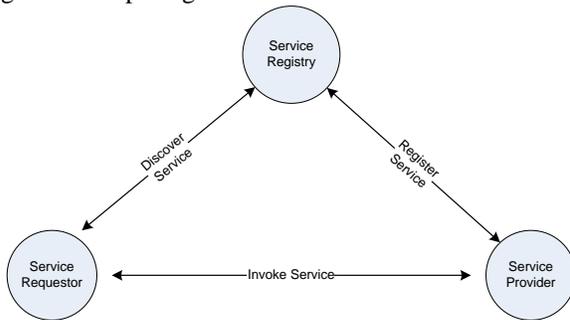
Untuk mengatasi kondisi ini, sistem baru yang dikembangkan menggunakan teknologi *Web Service*. *Web Service* adalah aplikasi Internet yang mempunyai sifat *interoperable* yaitu bisa diakses dan mengakses aplikasi lain dengan platform ataupun bahasa yang berbeda-beda sehingga sistem basis data yang berbeda-beda tidak menjadi kendala. Komponen-komponen *Web Service* adalah *XML*, *SOAP*, *WSDL* dan *UDDI*. Pada riset ini, hanya digunakan *XML*, *SOAP*, *HTTP* dan *SMTP*.

2. LANDASAN TEORI

Web Service

Web Service bisa dianggap sebagai perkembangan dan inovasi yang berkesinambungan yang terjadi pada aplikasi Internet. Karena penggunaan Internet yang sudah merasuk ke semua bidang termasuk dunia bisnis dan industri, maka tuntutan akan adanya aplikasi *business-to-business* (B2B), *application-to-application* (A2A) dan komunikasi antar proses aplikasi melahirkan teknologi *Service Oriented Architecture* (SOA). SOA memungkinkan suatu sumber daya komputasi atau suatu layanan komputasi pada jaringan atau Internet digunakan oleh aplikasi lain. Pada awalnya, aplikasi-aplikasi yang ada di Internet dikembangkan oleh perusahaan, kelompok atau orang yang berbeda-beda sehingga dikembangkan tidak dengan persyaratan kompatibilitas tertentu. Setiap

aplikasi di Internet bisa dikembangkan oleh *vendor* yang berbeda menggunakan bahasa yang berbeda dan berjalan pada *platform* yang berbeda. SOA berkembang untuk mengatasi perbedaan ini. SOA berkembang dengan sifat semua aplikasi harus bisa saling akses tidak tergantung pada bahasa, *platform* ataupun *vendor* pemiliknya. *Web Service* menggunakan teknologi SOA. *Web Service* mempublikasikan aplikasi dalam bentuk layanan (*services*) yang sekali dipublikasikan di Internet akan bisa ditemukan dan diakses oleh aplikasi lain di Internet tidak tergantung pada *platform* yang digunakan. Operasi suatu *Web Service* digambarkan sebagai hubungan antara tiga peran yang berbeda yaitu *Service Provider*, *Service Registry* dan *Service Requestor*. Hubungan ketiganya digambarkan pada gambar 1.



Gambar 1. Hubungan antara peran pada *Web Service*[1]

Service Provider adalah pihak yang memiliki layanan yang akan dipublikasikan di Internet. *Provider* ini akan mendaftarkan layanan yang dimilikinya kepada suatu *Service Registry*. *Service Registry* bertanggung jawab untuk mencatat layanan yang didaftarkan oleh *Web Provider* dan kelak menjawab permohonan *Service Requestor*. *Service Requestor* adalah pihak yang membutuhkan layanan tertentu di Internet. *Service Requestor* akan bertanya kepada *Service Registry* dan *Service Registry* akan memberitahu lokasi *Service Provider* yang memiliki layanan tersebut. Empat komponen penting pada *Web Service* adalah XML (*Extended Markup Language*), SOAP (*Simple Object Access Protocol*), WSDL (*Web Service Description Language*) dan UDDI (*Universal Description, Discovery and Integration*). XML adalah format data yang digunakan oleh *Web Service*. Pesan yang dikomunikasikan antar *Web Provider*, *Service Requestor* dan *Service Registry* adalah dalam bentuk pesan XML. SOAP memindahkan pesan XML dari satu titik ke titik lain dalam *Web Service*. SOAP juga menentukan protokol komunikasi yang digunakan. Protokol komunikasi bisa TCP, HTTP atau SMTP. WSDL bertugas untuk menerangkan *Web Service* yang tersedia pada suatu *Service Registry* dan UDDI bertugas untuk mencatat *Web Service* kemudian memberitahukan lokasinya bila kelak dibutuhkan. Sebagai suatu inovasi layanan Internet, seharusnya *Web Service* mempunyai banyak keunggulan dibandingkan dengan aplikasi pendahulunya tetapi sebagai fasilitas yang masih relatif berusia muda, masih

ada kekurangan-kekurangan pada *Web Service* yang masih harus diperbaiki.

Kelebihan *Web Service* adalah:

- *Web Service* mempunyai sifat *interoperability* sehingga bisa diakses oleh aplikasi yang berjalan pada *platform* yang berbeda-beda.
- *Web Service* menggunakan standar dan protokol terbuka pada Internet.
- Dengan menggunakan HTTP atau SMTP, *Web Service* bisa menembus pengamanan *firewall* suatu organisasi tanpa mengubah konfigurasi *firewall*.
- *Web Service* memungkinkan fungsi-fungsi pada banyak perangkat lunak di Internet untuk dipadukan menjadi satu *Web Service* baru.
- *Web Service* memungkinkan penggunaan ulang layanan dan komponen.
- *Web Service* bersifat *loosely-coupled* terhadap *client*.

Kekurangan *Web Service* adalah:

- *Web Service* termasuk layanan yang masih baru sehingga fitur standar seperti transaksi belum ada ataualaupun ada masih belum sebaik fitur yang sama pada *distributed computing open standard* yang sudah ada sebelumnya seperti CORBA.
- Unjuk kerja *Web Service* masih kurang baik dibandingkan dengan unjuk kerja *distributed computing* lain seperti RML, CORBA atau DCOM yang sudah ada lebih dahulu.

Web Service Interoperability Stack

Web Service bisa dianggap sebagai lapisan dari berbagai macam teknologi. Sebagai ilustrasi, suatu set lapisan teknologi pada *Web Service* digambarkan pada gambar 2.

Compositional	BPEL4WS, WS-Notification
Quality of Experience	WS-Security, WS-ReliableMessaging, WS-Transactions, WS-ResourceLifetime
Description	WSDL, WS-Policy, UDDI, WS-ResourceProperties
Messaging	XML, SOAP, WS-Addressing
Transports	HTTP, HTTPS, SMTP, Etc.

Gambar 2. Lapisan teknologi pada *Web Service*

Gambar 2 di atas memperlihatkan contoh hubungan antara teknologi pada tiap lapisan. Perpaduan dan lapisan teknologi yang digunakan sangat bervariasi dan tergantung pada kebutuhan aplikasi. Setiap aplikasi akan menggunakan *subset* tertentu dari teknologi di atas. Untuk riset yang dikembangkan, hanya digunakan lapisan

Transports dan *Messaging* saja. Lapisan *Transport* adalah lapisan terbawah pada *Web Service*. *Web service* bisa ditransportasikan menggunakan berbagai macam protokol seperti *Hypertext Transport Protocol (HTTP)* ataupun *Simple Mail Transfer Protocol (SMTP)*. Pada layanan *Messaging* digunakan *Web Service* seperti *Extended Markup Language (XML)* dan *Simple Object Access Protocol (SOAP)*. *Web Service* menggunakan XML sebagai format data yang ditransportasikan antara *requestor* dan *provider*. XML adalah bahasa pemrograman standar yang digunakan pada *Web Service*. SOAP menentukan cara mengirimkan data dalam format XML antara satu titik ke titik lain di Internet. Protokol yang digunakan bisa TCP, HTTP atau SMTP.

Keamanan *Web Service*

Web Service adalah perkembangan baru dari aplikasi di Internet dan memberi kontribusi yang cukup signifikan di era global di mana aplikasi di Internet tidak mengenal batas dan wilayah. Tetapi seperti pada aplikasi Internet lainnya, *Web Service* juga mempunyai resiko keamanan yang sama dengan aplikasi Internet pada umumnya. Ini berarti merancang dan mengembangkan suatu aplikasi *Web Service* berarti juga mempertimbangkan dan memikirkan banyak aspek keamanan di dalam suatu aplikasi Internet. Untuk suatu *Web Service*, ada beberapa aspek keamanan yang harus dipertimbangkan. Ada keamanan dari sisi Internetnya sendiri, keamanan pada program XML-nya atau keamanan pada aplikasi *Web Service*-nya sendiri.

Ada beberapa pendekatan pengamanan *Web Service* yang umum digunakan:

- Pengamanan pada proses pengiriman (*Transport-Level Security*) seperti pengamanan menggunakan *firewall*, VPN, *authentication*, *non-repudiation* dan *encryption*.
- Pengamanan pada pesan (*Message-Level Security*) seperti penggunaan *authentication* untuk melakukan validasi pemakai.
- Pengamanan pada data (*Data-level Security*) seperti proses enkripsi dan tanda tangan digital untuk menjamin tidak terjadi perubahan data.
- Pengamanan pada lingkungan (*Environment-level Security*) seperti prosedur pemakaian *Server*, penggunaan *password*, dan pengamanan secara fisik.

Riset yang dilakukan memfokuskan pada pengamanan pada pesan dan data yaitu *Message-Level Security* dan *Data-level Security*. Dibutuhkan waktu dan berbagai eksperimen untuk mendapatkan perpaduan teknologi yang tepat untuk bisa menentukan pengamanan seperti apa dan pengamanan terhadap ancaman apa. Suatu pengamanan program dan data yang baik harus bisa melindungi terhadap berbagai macam ancaman seperti penggunaan oleh yang tidak berhak, pengamanan data selama melalui saluran transmisi dan pengamanan terhadap berbagai macam kejahatan Internet yang sedang populer saat ini.

3. ANALISA DAN HASIL

Dari evaluasi sistem kependudukan yang sudah ada saat ini, maka telah diidentifikasi beberapa masalah yang dirangkum dalam Tabel 1. Tabel ini memperlihatkan hubungan antara masalah dan penyebabnya.

Tabel 1. Identifikasi Masalah

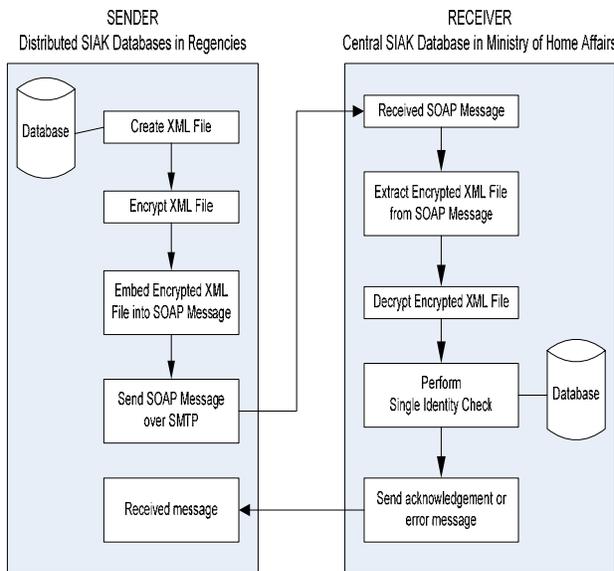
CAUSE AND EFFECT ANALYSIS	
Problem or Opportunity	Causes and Effects
1. Digunakannya beberapa sistem operasi dan <i>database</i> .	1. Beberapa daerah mengembangkan sendiri aplikasi dan <i>database</i> sesuai kebutuhan 2. Kesulitan mengelola dan menghubungkan data (kaitannya dengan SDM dan anggaran)
2. Secara geografis <i>database</i> tersebar didaerah-daerah	1. Ketersediaan infrastruktur yang berbeda pada daerah 2. Belum adanya kebijakan standar baku teknis pemakaian infrastruktur (<i>database</i> , jaringan, sistem operasi) didaerah-daerah

Tabel 2. Tujuan Sistem Baru

SYSTEM IMPROVEMENT OBJECTIVES	
System Objective	System Constraint
1. Sistem yang dapat mengintegrasikan data dari seluruh daerah ke pusat	1. Sistem baru mempunyai <i>platform</i> yang terbuka
2. Sistem yang aman dalam mengkonsolidasikan data ke pusat	2. Sistem yang baru ekonomis dalam biaya komunikasi dan perawatan
3. Sistem yang mempunyai tingkat integritas data yang tinggi	3. Sistem yang baru sesuai dengan kebijakan dan kebutuhan yang ada pada daerah

Disain yang Diusulkan

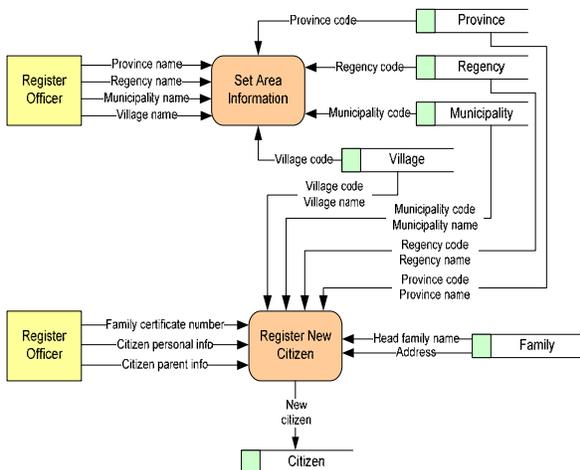
Sistem baru ini menggunakan *Web Service*. Pengirim adalah propinsi dan penerima adalah Depdagri di Jakarta. Karena data berasal dari berbagai propinsi yang menggunakan sistem basis data yang berbeda-beda maka sistem di pusat harus sistem basis data heterogen yang bisa menerima data berasal dari basis data yang berbeda-beda. Untuk menjaga agar konsistensi antara basis data di daerah-daerah dan basis data di pusat maka dilakukan proses konsolidasi. Proses konsolidasi diilustrasikan pada Gambar 3.



Gambar 3. Proses Konsolidasi antara basis data Pusat dan basis data Daerah

Data Flow Diagram

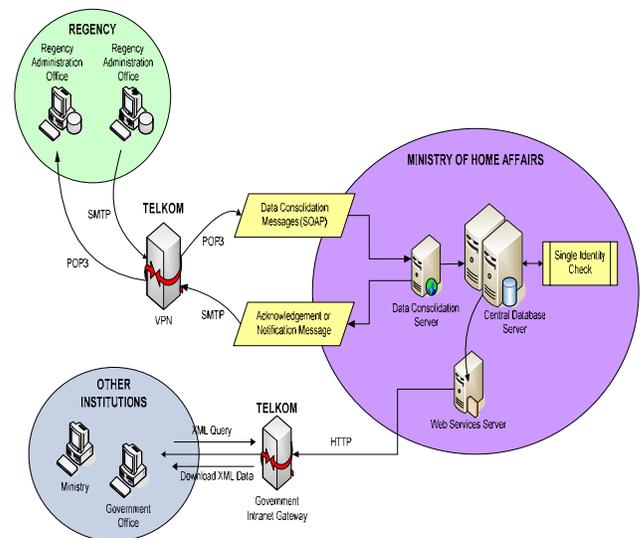
Data Flow Diagram (DFD) adalah suatu gambaran yang memperlihatkan aliran data dan proses kerja pada suatu aplikasi. DFD pada sistem ini digambarkan pada Gambar 4.



Gambar 4. Data Flow Diagram

Arsitektur Infrastruktur

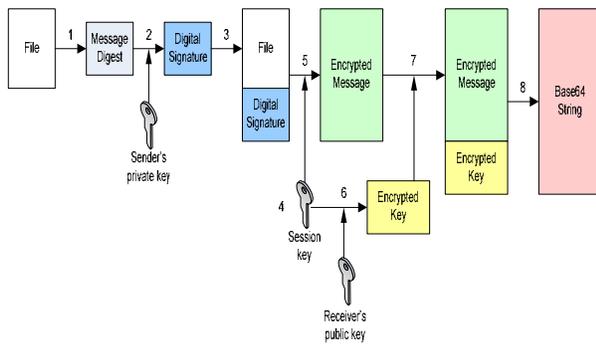
Setiap propinsi mempunyai suatu basis data yang menyimpan data penduduk di wilayahnya. Secara berkala atau sesuai kebutuhan, basis data pusat menerima kiriman data penduduk dari basis data di semua propinsi di daerah-daerah. Basis data pusat ini ada di Departemen Dalam Negeri di Jakarta. Komunikasi antara propinsi dengan pusat dilakukan menggunakan *Virtual Private Network (VPN)*. Untuk menjaga konsistensi, maka dilakukan proses konsolidasi. Pesan konsolidasi dikirimkan melalui SMTP dan diterima melalui POP3. Di samping itu, sistem di Depdagri juga bisa diakses dan digunakan oleh institusi dan departemen lain yang membutuhkannya. Komunikasi antara institusi lain dengan sistem di Depdagri dilakukan dengan menggunakan HTTP. Arsitektur dari hubungan ini diilustrasikan pada Gambar 5.



Gambar 5. Arsitektur Jaringan

Disain Keamanan

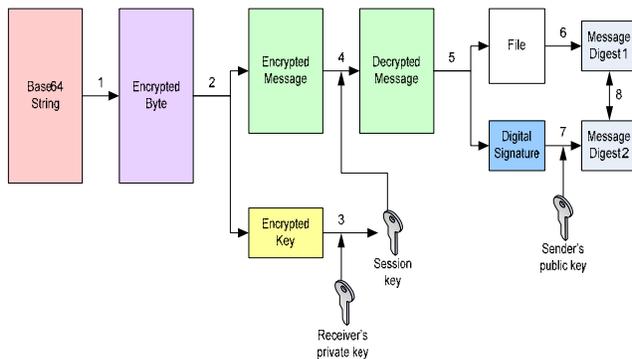
Pengamanan aplikasi dilakukan sebagai berikut: mula-mula dilakukan perpaduan dari kriptografi simetris dan asimetris untuk menjamin kerahasiaan data (*confidentiality*). Kemudian digunakan tanda tangan digital (*digital signature*) dengan menggunakan SHA-1 sebagai *hash code*. Tanda tangan digital dilakukan untuk menjaga integritas data dan otentikasi. Kemudian dilakukan pengamanan untuk *SOAP message format*. Proses enkripsi dan dekripsi diilustrasikan pada Gambar 6 dan Gambar 7.



ENCRYPTION PROCESS:

1. Create message digest using SHA1 hash function
2. Sign the message digest using sender's private key
3. Append the digital signature to the file
4. Generate a session key
5. Encrypt the file and digital signature using Rijndael algorithm with the session key
6. Encrypt the session key using receiver's public key
7. Append the encrypted key to the encrypted file
8. Convert the encrypted message and encrypted key to Base64 string

Gambar 6. Proses Enkripsi



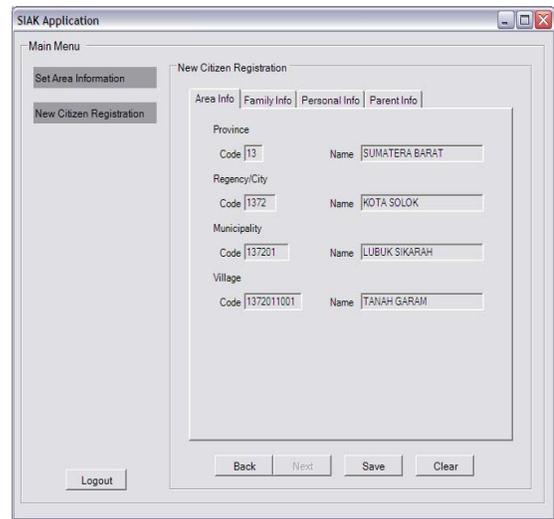
DECRYPTION PROCESS:

1. Convert Base64 string into byte array
2. Extract encrypted byte into encrypted message and encrypted key
3. Decrypt the encrypted key using receiver's private key
4. Decrypt the encrypted message using Rijndael algorithm with session key
5. Extract decrypted message into file and digital signature
6. Create message digest using SHA1 hash function
7. Verify digital signature using sender's public key
8. Compare both message digests. If the message digests are the same, then the message has been successfully received. Otherwise, the message might have been tampered with during transmission.

Gambar 7. Proses Dekripsi

Program di client

Data dimasukkan lengkap. Kemudian pemakai akan meng-klik tombol *Save*. Tombol *Save* ini akan memasukkan data penduduk baru ini ke basis data lokal di wilayahnya. Akhirnya, data penduduk baru ini dikirimkan ke basis data di Depdagri Pusat di Jakarta. Seperti terlihat pada gambar 8 yang berisi Nama Kepala Keluarga dan Alamat yang sudah ada di basis data.

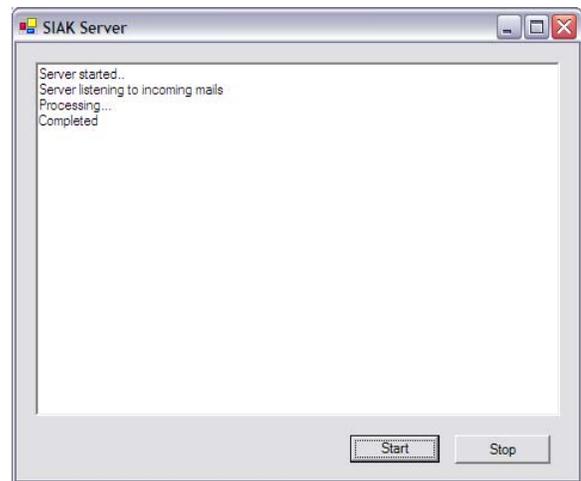


Gambar 8. Data Keluarga

Gambar 8 berisi Nama Kepala Keluarga dan Alamat yang sudah ada di basis data.

Program di Server

Layar pada sisi *server* adalah suatu *multi-line textbox* yang mempunyai dua buah tombol, yaitu tombol *Start* dan tombol *Stop*. Jika tombol *Start* diklik, maka *Server* akan mulai beroperasi dan jika *start* berhasil dengan baik maka akan muncul pesan seperti pada gambar 9.



Gambar 9. Menu untuk menjalankan *Server*

Pengecekan Identitas Tunggal

Dengan sistem ini, akan diperkenalkan Sistem Identitas Tunggal. Sistem Identitas Tunggal ini menjaga agar setiap penduduk hanya mempunyai satu Nomor Identitas Penduduk sehingga hanya mempunyai satu Kartu Tanda Penduduk dan tidak terjadi duplikasi. Proses pengecekan identitas tunggal ini dilakukan di basis data pusat pada saat data penduduk baru dimasukkan ke basis data pusat. Pengecekan dilakukan berdasarkan empat informasi yaitu, Nama, Tanggal Lahir, Jenis Kelamin dan Nama Ibu

Kandung. Prosedur akan melakukan pengecekan terhadap keempat informasi yang dimasukkan tersebut dan akan menghasilkan angka nol jika penduduk baru belum terdaftar dan akan menghasilkan angka lebih besar dari nol jika sudah terdaftar di basis data pusat. Contoh dilakukan terhadap dua buah data yaitu terhadap nama Riana Kuswadi, tanggal lahir 12 Juni 1984 (12 June 1984), jenis kelamin perempuan (*female*), dan nama ibu kandung Juliana Tedja. Prosedur pengecekan akan melihat apakah informasi tersebut sudah ada di basis data. Hasilnya adalah nilai satu, ini berarti informasi sudah ada di basis data pusat. Contoh kedua adalah Dax Permana, tanggal lahir 30 Juni 1984 (30 June 1984), jenis kelamin laki-laki (*male*) dan nama ibu kandung Ellen. Hasil prosedur pengecekan adalah nol, berarti data ini belum ada di basis data pusat. Jika hasilnya nol, maka akan ada berita konfirmasi ke daerah bahwa data tidak duplikat dan sudah dimasukkan ke basis data pusat, jika hasilnya tidak nol, maka akan ada pemberitahuan ke daerah bahwa mungkin ada duplikasi sehingga harus dilakukan pengecekan kembali.

4. KESIMPULAN

SIAK dikembangkan untuk mengatasi tiga masalah pada sistem kependudukan di Indonesia yaitu, basis data kependudukan di setiap daerah menggunakan sistem operasi dan sistem manajemen basis data yang berbeda-beda, basis data ini tersebar di wilayah yang secara geografis saling berjauhan dan kurangnya sistem komunikasi terhubung yang berkesinambungan karena kurangnya infrastruktur komunikasi. Untuk mengatasi masalah ini maka SIAK dikembangkan menggunakan aplikasi berdasarkan *Web Service*. *Web Service* mempunyai beberapa keunggulan diantaranya program yang dituliskan dalam bahasa dan *platform* yang berbeda bisa saling berkomunikasi dan *Web Service* menggunakan protokol Internet sehingga mudah dan sederhana. SIAK ini juga dirancang untuk menerapkan sistem Identitas Tunggal untuk menghindari duplikasi dan menghindarkan seorang penduduk mempunyai lebih dari satu Kartu Tanda Penduduk.

5. DAFTAR PUSTAKA

[1] Atkinson, Bob, Giovanni Della-Libera, Satoshi Hada, Maryann Hondo, Phillip Hallam-Baker, Johannes Klein, Brian LaMacchia, Paul Leach, John Manferdelli, Hiroshi Maruyama, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, John Shewchuk, and Dan Simon. "Specification: Web Services Security (WS-Security)," April 5, 2002, <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>, accessed June 2006.

[2] Bain, Tony, Denise Gosnell, and Jonathan A. Walsh. *Visual Basic .NET and SQL Server 2000: Building*

and Effective Data Layer. Birmingham, UK: Wrox Press Ltd., 2002.

[3] Bartel, Mark, John Boyer, Barb Fox, Brian LaMacchia, and Ed Simon. "XML-Signature Syntax and Processing," W3C Recommendation, February 12, 2002, <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>, accessed July 2006.

[4] Basiura, R., Mike Batongbacal, Brandon Bohling, Mike Clark, Andreas Eide, Robert Eisenberg, Kevin Hoffman, Brian Loesgen, Chris Miller, Matthew Reynolds, Bill Sempf, and Srinivasa Sivakumar. *Professional ASP .NET Web Services*. Birmingham, UK: Wrox Press Ltd., 2001.

[5] Bentley, Lonnie D., Dittmann, Kevin C. & Whitten, Jeffrey L. *Systems Analysis and Design Methods*. 6th ed. New York: McGraw Hill, 2004.

[6] Boulos, Tom. "Asynchronous Web Services Programming," *Builder.com*, July 14, 2002, <http://www.zdnetasia.com/builder/architect/web/0,39045497,39062860,00.htm>, accessed June 2006.

[7] Box, Don, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. "Simple Object Access Protocol (SOAP) 1.1," W3C Note, May 8, 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>, accessed June 2006.

[8] Cunnings, Robert, Simon Fell, and Paul Kulchenko. "SMTP Transport Binding for SOAP 1.1," November 29, 2001, http://www.pocketsoap.com/specs/smtplibbinding/smtplib_001.htm, accessed July 2006.

[9] Dillaway, Blair, Takeshi Imamura, and Ed Simon. "XML Encryption Syntax and Processing," W3C Recommendation, December 10, 2002, <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>, accessed July 2006.

[10] FIPS 180-1, "Secure Hash Algorithm -1 (SHA-1)," W3C, May 11, 1993, <http://www.itl.nist.gov/fipspub/fip180-1.htm>, accessed July 2006.

[11] Graham, S., Doug Davis, Simeon Simeonov, Glen Daniels, Peter Brittenham, Yuichi Nakamura, Paul Fremantle, Dieter König, and Claudia Zentner. *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*. 2nd ed. US: Sams Publishing, 2005.

[12] Hadiwinata, M. *Solusi Pemrograman XML Web Services dengan Visual Basic .NET*. Jakarta: PT. Elex Media Komputindo, 2003.

[13] Hageman, Keith, and Jeffrey Hasan. "Secure Web Services with WS-Security," *Apress*, July, 2004, <http://www.windowsitlibrary.com/Content/1219/06/1.html>, accessed July 2006.

[14] K., Komathy, Vivekanandan P. and Ramachandran V. "Component-based Security Model for XML Messaging Services," *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, 2002.

- [15] Marchal, Benoit. XML By Example. Indiana: Que, 2000. <http://safari.informit.com/0789725045>, accessed June 2006.
- [16] Nagappan, Ramesh, Robert Skoczylas, and Rima Patel Sriganesh. "Developing Web Services Using SOAP." In Developing Java Web Services. Indiana: Wiley Publishing Inc., 2003.
- [17] Nasution, Benny B., Elizabeth A. Kendall, and Asad I. Khan. "Algorithm Exchange of a Security Control System for Web Services Applications," Proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii, 2005.
- [18] NN. "Chapter 8 – Web Services Security," Nov 22, 2004, <http://searchwebservices.techtarget.com/searchWebServices/downloads/Newcomer.pdf>, accessed June 2006.
- [19] O'Neill, Mark, Phillip Hallam-Baker, Seán Mac Cann, Mike Shema, Ed Simon, Paul A. Watters, and Andrew White. Web Services Security. California: McGraw Hill, 2003.
- [20] Sempf, Bill, and Matthew Reynolds. Professional VB .NET. 2nd ed. Birmingham, UK: Wrox Press Ltd., 2002.
- [21] W3Schools, "XML Tutorial," <http://www.w3schools.com/xml/default.asp>, accessed June 2006.
- [22] W3Schools, "SOAP Tutorial," <http://www.w3schools.com/soap/default.asp>, accessed June 2006.
- [23] W3Schools, "Web Services Tutorial," <http://www.w3schools.com/webservices/default.asp>, accessed June 2006.
- [24] Wikipedia. <http://en.wikipedia.org/wiki/SOAP>, accessed June 2006.