

## Desain Arsitektur Enterprise Application Integration (EAI) Sebagai Middleware Untuk Aplikasi Pemerintah (E-Government)

Taufan Harry Prasetyo<sup>1</sup>, Fatchul Hijrih<sup>2</sup>

<sup>1,2</sup> Mahasiswa Magister Teknik Informatika STMIK AMIKOM Yogyakarta  
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281  
Email : [taufan\\_hrp@ugm.ac.id](mailto:taufan_hrp@ugm.ac.id) <sup>1</sup> [luhtaf04@gmail.com](mailto:luhtaf04@gmail.com) <sup>2</sup>

### Abstrak

Pemerintah telah memiliki banyak layanan E-government bagi masyarakatnya, namun layanan ini seolah berdiri sendiri, belum terintegrasi. Aplikasi E-KTP, SIM (Polri), Layanan kesehatan (BPJS) dan aplikasi layanan pemerintah lainnya seharusnya dapat terintegrasi dengan baik untuk meningkatkan kinerja pelayanan publik. Oleh karenanya diperlukan sebuah arsitektur Enterprise Application Integration (EAI) sebagai middleware untuk mengintegrasikan berbagai macam aplikasi yang sudah ada. Kolaborasi antar aplikasi dan layanan akan sangat beragam sehingga memerlukan middleware yang bisa beradaptasi dengan banyaknya perbedaan kebutuhan. Paper ini memberikan sebuah arsitektur dalam membangun sebuah middleware sebagai fungsi dari EAI. Performa, multi-agent, fleksibilitas layanan serta keamanan menjadi syarat mutlak dari middleware untuk pemerintahan. Untuk mendesain arsitektur middleware digunakan pendekatan interoperability. Dengan kemampuan interoperability yang baik, maka pertukaran dan pengintegrasian data dapat dilakukan. EAI/middleware untuk pemerintah memiliki beberapa point kritis yang perlu di perhatikan seperti faktor keamanan, interoperability, perubahan kebutuhan serta beragamnya layanan. Manajemen user, keamanan akses, manajemen layanan, multi platform web-service, pertukaran data, enkripsi XML/JSON merupakan komponen dari middleware yang penting dalam EAI untuk E-Government.

**Kata kunci:** Enterprise Application Integration, EAI, E-Government, middleware, web-service

### 1. Pendahuluan

Saat ini pemerintah masih mengalami kesulitan dalam hal interoperabilitas dan interkoneksi antar aplikasi dan layanan publik yang dimiliki pemerintah. Dapat diambil contoh interkoneksi aplikasi e-KTP dengan layanan ketenagakerjaan, BPJS, SIM-Polri dan lainnya dimana dalam proses tersebut memerlukan data kependudukan yang disediakan oleh E-KTP. Yang terjadi adalah saat tenaga kerja mengurus proses ketenagakerjaan masih dimintai fotokopi KTP dan surat keterangan lainnya, padahal belum tentu dokumen tersebut bisa divalidasi dengan baik. Contoh lainnya adalah proses imigrasi, saat mengurus passport maka pemohon diminta memenuhi dokumen seperti KTP, kartu keluarga, dan lain sebagainya yang jelas-jelas data tersebut telah tersedia pada aplikasi e-KTP. Masih banyak aplikasi layanan maupun informasi yang seharusnya saling terkoneksi satu dengan lainnya untuk dapat memberikan layanan e-Government yang memuaskan. Selain itu pemerintah bisa mendapatkan manfaat lain yaitu validitas data dan informasi, ketepatan dan kecepatan informasi serta data dan informasi yang lebih bernilai.

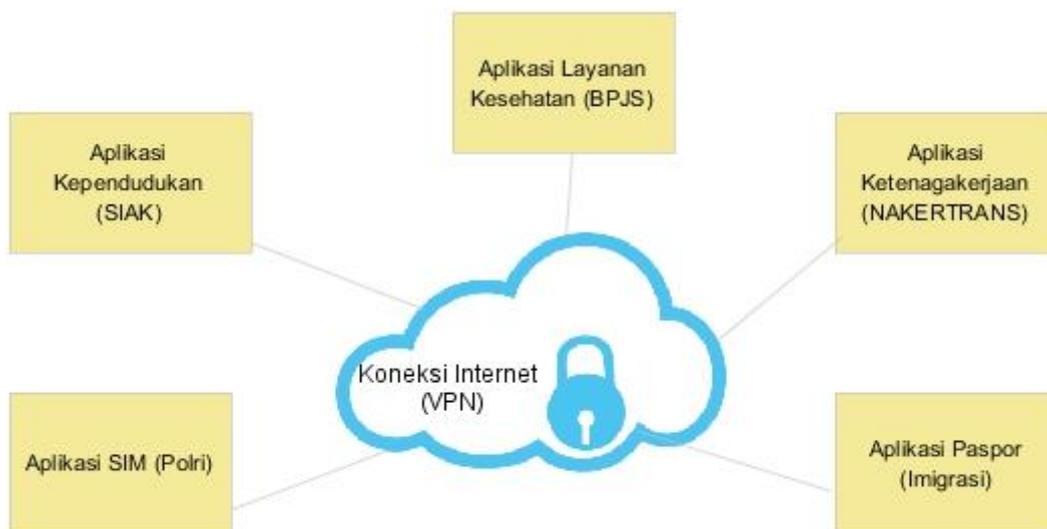
Enterprise Application Integration (EAI) adalah salah satu solusi yang umumnya digunakan untuk mengintegrasikan berbagai aplikasi dalam satu instansi atau perusahaan. [1] mendeskripsikan bahwa fungsi EAI adalah mengkoneksikan beberapa interface dari dua atau lebih aplikasi yang berbeda dalam sebuah institusi/perusahaan. [2,3] menyatakan bahwa perspektif dalam pengintegrasian ini terdiri dari pengintegrasian teknologi, layanan, data dan pengguna (*user*). Sedangkan [4] membagi EAI menjadi tiga layer yaitu layer proses, layer transformasi data, dan layer transportasi data. [5] mendefinisikan EAI sebagai penggunaan aplikasi atau software yang dapat mengintegrasikan berbagai aplikasi dalam sebuah institusi. Lebih lanjut lagi EAI fokus pada pengintegrasian dari point ke point (*point to point integration*) aplikasi serta teknologinya. Sedangkan [6] menjelaskan bahwa EAI dalam penerapannya merupakan sebuah *middleware*.

*Middleware* akan menjadi aplikasi yang memberikan layanan data atau biasa disebut *web-service* untuk aplikasi aplikasi lainnya yang ingin diintegrasikan. *Web-service* didesain untuk melaksanakan tugas tugas interoperasi dengan object/entitas lain yang compatible dan sesuai dengan standart platform tertentu [7]. Jiangjun dalam [8] menyatakan seiring dengan perkembangan proyek e-Government, platform e-Government generasi selanjutnya dapat mendukung manajemen bisnis, proses bisnis yang rasional, keputusan bisnis yang ilmiah, evaluasi kinerja kuantitatif dan sistem bisnis yang terintegrasi. *Web-service* sebagai jembatan data antar aplikasi harus memiliki kerahasiaan (*confidential*), otorisasi (*authorization*), integrasi data (*integration of data*), dan keaslian atas data (*proof of original*) [7]. Dikatakan juga bahwa web service (dalam bentuk xml) untuk aplikasi aplikasi pemerintah seharusnya di enkripsi, selain itu web service harus dapat menangani platform yang heterogen

Dari beberapa latar belakang yang telah disampaikan maka paper ini lebih fokus pada mendesain sebuah arsitektur *middleware* sebagai *web-service* untuk pengembangan e-Government. Adapun pertanyaan yang ingin dijawab dari paper ini adalah bagaimana membangun infrastruktur layanan *web-service* yang multi agent ke dalam sebuah layanan *data bus*. Selain itu bagaimana mendesain arsitektur *middleware* bisa menjembatani aplikasi peminta (*requester*) dengan aplikasi pemberi (*provider*) agar tidak terkoneksi secara langsung (*requester* tidak langsung mengakses database *provider*) guna meningkatkan keamanan.

## 2. Metodologi dan Desain

Setelah melakukan pembahasan perihal EAI pada bagian sebelumnya maka paper ini akan fokus pada bagaimana membangun model EAI untuk pemerintahan. [6] mendeskripsikan EAI secara konseptual sebagai sebuah *middleware* yang mengintegrasikan berbagai aplikasi kaitannya dalam berbagi data dan informasi baik dari database maupun proses. Sedangkan paper ini akan mendesain arsitektur EAI dengan perpektif *interoperability*. Dimana arsitektur dibangun untuk dapat melayani berbagai kebutuhan data dan informasi serta dapat saling bertukar informasi antara sistem yang satu dengan yang lainnya



Gambar 1 Koneksi Beberapa Aplikasi Layanan Pemerintah (e-government)

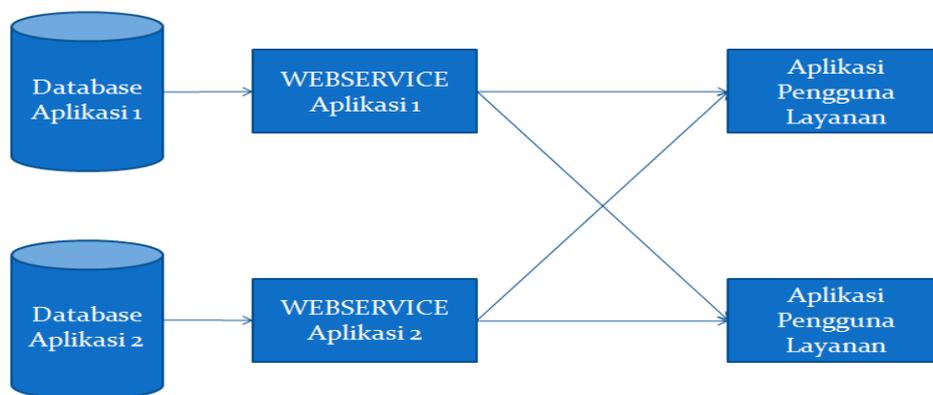
Gambar 1 menunjukan paling tidak dari lima aplikasi layanan e-government yaitu aplikasi layanan SIM (Polri), aplikasi kependudukan E-KTP/SIAK (Kemendagri), aplikasi layanan paspor (Dirjen Imigrasi), aplikasi ketenaga kerjaan (Kemenakertrans), dan aplikasi layanan kesehatan (BPJS) bisa saling terkoneksi data yang satu dengan yang lainnya. Mari ambil contoh aplikasi layanan Surat Ijin Mengemudi (SIM) dari Polri, saat kita membuat SIM kita akan diminta mengisi identitas diri dan menyertakan fotokopi KTP sebagai persyaratan membuat SIM. Lalu data akan diinput kedalam sistem dan akan diverifikasi kecocokan data tersebut. Verifikasi dilakukan dengan mencocokkan data yang diinputkan dengan fotokopi Kartu Tanda Penduduk (KTP) yang dipersyaratkan. Jika aplikasi SIM Polri bisa mengakses database aplikasi kependudukan seharusnya proses penginputan data tidak perlu dilakukan. Aplikasi SIM bisa mengambil informasi identitas pemohon SIM dari database kependudukan yang sudah

---

tersedia dalam aplikasi E-KTP/SIAK. Proses verifikasi data pun akan lebih valid karena selain data dan informasi dijamin keasliannya, data E-KTP/SIAK juga menyediakan foto/image, data sidik jari, serta retina mata warga bersangkutan. Selain itu efisiensi waktu bisa diperoleh dengan memanfaatkan koneksi antar aplikasi ini.

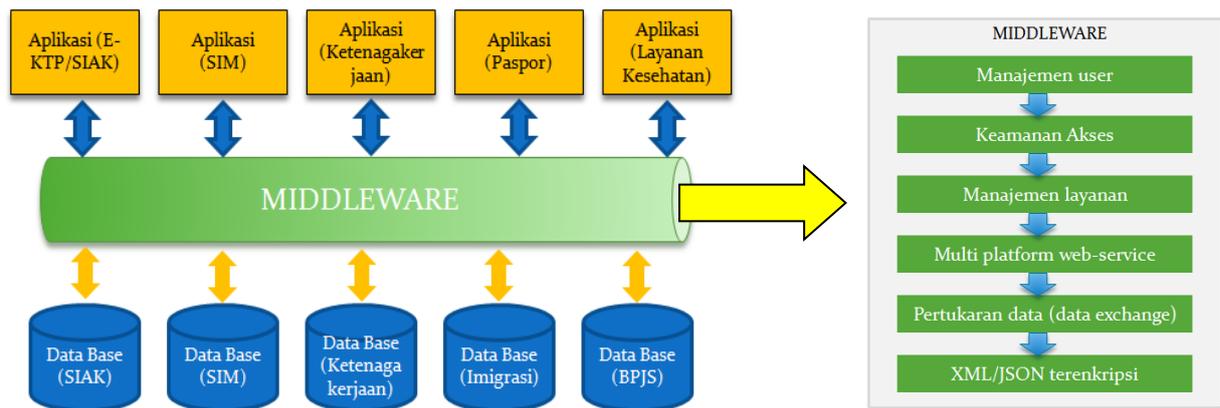
Namun aplikasi e-government ini sejak awal tidak dibangun untuk tersinkronisasi datanya dengan aplikasi lainnya. Sehingga dibutuhkan EAI sebagai *middleware* untuk dapat menghubungkan aplikasi aplikasi tersebut. *Middleware* ini umumnya dibangun sebagai sebuah layanan *services* yang dapat menghubungkan aplikasi yang satu dengan yang lainnya. Layanan tersebut kemudian disebut sebagai *web service*, layanan ini akan menyediakan informasi tanpa harus mengakses langsung dari database. Hal ini untuk bisa menjaga keamanan data dari pemilik aplikasi maupun menjaga tingkat kepercayaan (*trust*) dari masing masing pengguna layanan. Dalam membangun sebuah aplikasi *web service* untuk kepentingan pemerintah, keamanan data menjadi kritis dan harus memenuhi persyaratan sebagai berikut:[9]

1. Autentikasi (*authentication*) – Perlu adanya otentifikasi sebelum pengguna mendapatkan informasi dalam bentuk *web-service*. Hal ini untuk menjaga bahwa data/informasi yang di sediakan hanya diterima atau digunakan oleh penerima/pemohon(*requester*) yang tepat.
2. Otorisasi (*Authorization*) – Pemberian hak kepada pengguna tertentu untuk menjaga kerahasiaan data dan informasi
3. Kerahasiaan (*Confidentiality*) – Service sebaiknya tidak hanya berjalan pada komunikasi https SSL/TLS namun juga perlu dienkripsi
4. Keterlacakan (*Traceability*) – Semua informasi dan data yang mengalir melalui *web-service* harus memiliki catatan waktu dan tersimpan dalam logs untuk bisa dilacak.
5. (*Integrity & Non-Repudiation*) - Data yang dikirim melalui *web-service* harus memiliki keyakinan penuh bahwa data tersebut adalah milik entitas yang dimaksud



Gambar 2. Web-sevice konvensional atau tanpa middleware

Gambar 2 menunjukkan bahwa umumnya setiap aplikasi akan memberikan *web-service* nya masing masing, artinya setiap aplikasi harus menmbangun layanan *web-service* nya setiap ada permintaan layanan data dari aplikasi lain. Hal ini tentunya kurang efisien dalam hal waktu dan biaya. Disini peran EAI sebagai *middleware* untuk dapat mengintegrasikan aplikasi aplikasi yang beragam dengan efisien dan efektif. EAI harus dapat menintegrasikan beberapa aplikasi penyedia layanan *web-service* dengan aman, multiuser, dan mudah dalam melakukan kustomisasi layanan. Aman yang dimaksud adalah yang memenuhi kriteria authentication, authorization, confidentiality, traceability, integrity & non-repudiation. Sedangkan multiuser adalah bahwa *middleware* ini bisa digunakan oleh beberapa user ataupun aplikasi dalam menyediakan layanan *web-service* secara independen. Dan yang dimaksud dengan kustomisasi layanan bahwa pemilik aplikasi bisa melakukan bisa menentukan informasi apa yang akan dijadikan layanan serta bisa melakukan perubahan dengan mudah.



Gambar 3. Desain arsitektur EAI sebagai middleware dalam e-government

Dalam membangun arsitektur *middleware* untuk pengintegrasian layanan pemerintah, paper ini mencoba memberikan sebuah model seperti yang ditunjukkan pada Gambar 3. *Middleware* yang dibangun memiliki komponen diantaranya:

a) **Manajemen User**

Yang dimaksud dengan manajemen user adalah kemampuan untuk menangani dan mengelola lebih dari satu user/pengguna. *Middleware* dapat mengelola layanan ini untuk masing masing pengguna dimana pengguna ini nantinya akan mengelola *web-service* dari aplikasinya masing masing.

b) **Keamanan Akses**

Masing masing pengguna penyedia (*provider*) memiliki hak untuk memberikan layanan kepada pengguna lainnya (*requester*) yang memerlukan. Dan hanya pengguna (*requester*) yang berhak atau diberikan hak saja yang bisa menggunakan *web-service* dari pengguna penyedia (*provider*) tertentu. Implementasinya adalah dengan memberikan session id dan token pada setiap layanan *web-service* yang di broadcast/disediakan.

c) **Manajemen Layanan**

Manajemen layanan dimaksudkan agar para pengguna penyedia dapat mendesain dan mengelola layanan *web-service* yang diberikan, baik konten *web-service* maupun kepada siapa layanan ini akan diberikan.

d) **Multi platform web-service**

Layanan *web-service* memiliki output dengan beberapa platform seperti SOAP dengan xml, maupun dengan JSON yang belakangan ini marak digunakan. Hal ini penting agar *middleware* yang dibangun dapat mudahberadaptasi dengan banyak aplikasi yang sudah ada. Kedua teknologi ini dianggap dapat mengatasi isu kemampuan mensupport synchronous dan asynchronous

e) **Pertukaran data (data exchange)**

Layanan ini juga harus dapat menangani pertukaran data dua arah tidak hanya bisa membaca informasi namun bisa juga melakukan *record* pada *database* pengguna penyedia jika diperlukan dan diijinkan.

f) **Enkripsi XML/JSON**

Hasil output dari *web-service* informasi yang terkandung sebaiknya telah dienkrpsi untuk menjaga kerahasiaan dan keamanan data. Hal ini untuk mengedepankan prinsip kehati hatian dan meningkatkan kepercayaan pengguna layanan.

Dengan desain yang diberikan ini harapannya adalah *middleware* dapat mengintegrasikan beberapa aplikasi dalam layanan E-Government. Selain itu memiliki kemampuan yang cukup untuk melayani pengintegrasian aplikasi serta memiliki *interoperability* (kemudahan penggunaan) sehingga akan semakin banyak aplikasi yang dapat di integrasikan.

---

### 3. Simpulan

Paper ini telah memberikan pembahasan dan kajian tentang middleware sebagai EAI didalam mengintegrasikan aplikasi aplikasi pemerintahan. Selain itu paper ini juga telah memberikan arsitektur dasar untuk membangun sebuah *middleware* untuk mengintegrasikan layanan e-government. Dengan adanya middleware integrasi berbagai layanan pemerintah diharapkan dapat meningkat. Performa, kinerja dan kualitas layanan publik pemerintah dapat ditingkatkan. Komponen yang dibangun dalam middleware pada paper ini dirasa dapat memenuhi kebutuhan, kerahasiaan, dan fleksibilitas layanan. Beberapa komponen yang di perkenalkan dirasa cukup untuk dapat mengintegrasikan beragam sistem yang memiliki platform berbeda. Kelanjutan dari paper ini adalah bagaimana mendesai sebuah aplikasi middleware dari kerangka/arsitektur yang sudah ada.

### Daftar Pustaka

- [1] Kotinurmi, P. *Enterprise Application Integration. SoberIT Software Business and Engineering Institute. Helsinki University of Technology.* 2002. vol(no):132.
- [2] He, Wu, and Li Da Xu. *Integration of distributed enterprise applications: a survey. Industrial Informatics, IEEE Transactions on* 10.1 2014; vol: 35-42.
- [3] Thullner, Robert, Alexander Schatten, and Josef Schiefer. *Implementing enterprise integration patterns using open source frameworks.* ISIS, 2008; vol(no): 1-5.
- [4] Maamar, Zakaria, Youcef Baghdadi, and Wathiq Mansoor. *Towards an approach for Enterprise Application Integration based on specialized services and Business Objects. Information Reuse and Integration (IRI), 2010 IEEE International Conference on.* IEEE, 2010; vol(no): 400-402.
- [5] Cummins, Fred A. *Enterprise integration: an architecture for enterprise application and systems integration.* John Wiley & Sons, Inc., 2002; vol(no): 435-442.
- [6] Abdullah AL-Malaise AL-Ghamdi, Farrukh Saleem. *Enterprise Application Integration as a Middleware: Modification in Data & Process Layer. Science and Information Conference 2014 August 27-29, 2010; vol(no): 698-701.*
- [7] Rong-Xi, Zhang, and Zhang Yan-mei. *Design of XML Web Services Based E-government Heterogeneous Platform. Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on.* Vol. 1. IEEE, 2010; vol(no): 554 -557.
- [8] Wang, Zhidong, Zhi Liu, and Yun Yang. *Design and implementation of flexible E-Government platform based on XML Data-Bus and lightweight MVC execution framework ASSH. Computer Application and System Modeling (ICASM), 2010 International Conference on.* Vol. 5. IEEE, 2010; vol(no): V5-422 - V5-427.
- [9] Marques, Fábio, Gonçalo Paiva Dias, and André Zúquete. *A general interoperability architecture for e-government based on agents and web services. Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on.* IEEE, 2011;
- [10] Serain, Daniel. *Middleware and enterprise application integration: the architecture of e-business solutions.* Springer Science & Business Media, 2002;vol(no):248
- [11] Yan, Pei, and Jiao Guo. *Researching and Designing the Architecture of E-government Based on SOA. E-Business and E-Government (ICEE), 2010 International Conference on.* IEEE, 2010; Vol(no): 512-515.
- [12] Elfwing, Robert, Ulf Paulsson, and Lars Lundberg. *Performance of SOAP in Web Service environment compared to CORBA. Software Engineering Conference, 2002. Ninth Asia-Pacific.* IEEE, 2002; Vol(no): 84-93.
- [13] Gomes, Rui, and Jorge Ribeiro. *Use of Web Services in E-government Information Systems A Case Study. Information Management and Engineering, 2009. ICIME'09. International Conference on.* IEEE, 2009; Vol(no): 475-480.
- [14] Qiang Guo; Zheng Hai-Dong; Gui-ran Chang; Jie Li, *High Availability and Flexibility of Web Services in the E-government System, Software Engineering, 2009. WCSE '09. WRI World Congress. 2009;*, vol.1, (no): 372-376.
- [15] Qianfu Ni, Lu, Yarlagadda & Ming, *A collaborative engine for enterprise application integration*”, *Computers in Industry* 2006, Elsevier B.V.57 2006; Vol(no): 640–652.