

E-GOVERNMENT PERFORMANCE BASED MODEL

Frans Richard Kodong

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari no 2 Tambakbayan 55281 Yogyakarta Telp (0274)-485323
e-mail : frkodong@gmail.com

Abstrak

Pada negara-negara berkembang, seperti Indonesia, dampak globalisasi salah satunya di fokuskan pada perkembangan TIK (Teknologi Informasi dan Komunikasi), hal ini dapat dilihat dari maraknya perkembangan TIK diberbagai bidang, tidak ketinggalan dalam penyelenggaraan pemerintahan (*e-government*). Walaupun pemerintah sudah membuat kebijakan tentang *e-government*, namun dalam pelaksanaan pembangunannya masih menghadapi masalah, terutama menyangkut kinerja layanan *e-government*, yang merupakan indikator keberhasilan implementasi dari *e-government* tersebut. Beberapa model evaluasi kinerja dapat digunakan, namun dalam penelitian ini menggunakan metode *Early Performance Testing* dimana indikator kinerja dari layanan *e-government* dapat di tentukan seawal mungkin dalam tahap-tahap pembangunannya.

Keyword : *e-Government, Performance Indicator, Early Performance Testing.*

1. PENDAHULUAN

Pada negara berkembang seperti di Indonesia, dampak globalisasi akan di fokuskan pada *diversity, accountability, civil society, privatization, democratization, decentralization, reengineering* dan *empowering effect* dari perkembangan Teknologi Informasi dan Komunikasi (TIK), hal ini dapat dilihat dengan makin maraknya pembangunan TIK diberbagai bidang, tidak ketinggalan dalam penyelenggaraan pemerintahan, yang kemudian dikenal dengan istilah *e-government*. Di Indonesia terdapat lebih dari 400 pemerintah kabupaten dan kota, dimana masing-masing pemerintah daerah membangun *e-government* dengan caranya masing-masing walaupun pemerintah telah membuat kebijakan-kebijakan sebagai kerangka acuan pembangunan *e-government* yang cukup representative (INPRES No.3/2003, dll).

Beberapa pemerintah daerah telah membangun *e-government* dengan investasi yang cukup besar, namun dalam pelaksanaannya belum ada kriteria baik teknis maupun nonteknis yang menjadi *standard baseline* dalam pembangunan proyek tersebut, sehingga masih dipertanyakan apakah implementasi *e-government* benar-benar dapat meningkatkan kinerja penyelenggaraan pemerintahan.

Kinerja *e-government* dapat ditinjau dari beberapa faktor, dimana masing-masing faktor tersebut mempunyai parameter indikator kinerja. Salah satu faktor keberhasilan dari implementasi *e-government* adalah kinerja teknis yang mencakup infrastruktur jaringan, perangkat keras, maupun perangkat lunak yang akan dibangun.

Berdasarkan latar belakang tersebut diatas, maka perlu di rancang suatu model kinerja *e-government* dengan tujuan sebagai berikut : (1) Mengkaji metode-metode pengukuran kinerja sistem terdistribusi, (2) Mengkaji teknologi *middleware* sebagai *core* dari aplikasi layanan pemerintah, (3) Mengkaji dan merancang model pembangunan sistem aplikasi layanan pemerintahan.

2. TINJAUAN PUSTAKA

Penelitian tentang kinerja aplikasi terdistribusi telah banyak dilakukan tetapi dalam pembangunan *e-government* masih sangat kurang. Eric et.al. (1999) melakukan penelitian tentang penerapan teknologi *software interconnection* dan *middleware* seperti RMI, CORBA, ILU dan ActiveX dalam membangun aplikasiterdistribusi serta melakukan evaluasi terhadap *middleware*. Untuk pengujian kinerja secara empiris pernah dilakukan oleh Gorton dan Liu (2004) yang mengevaluasi kinerja dari produk-produk *middleware* sebagai *interconnection* aplikasi terdistribusi.

Giovani et.all (2004) melakukan peneitian tentang *early performance testing* pada aplikasi terdistribus, dimana pengujian dilakukan pada tahap awal pembangunan aplikasi tersebut. Penelitian dengan metode ini dilanjutkan oleh Nan Zhang (2004) dengan metode pengujian kinerja aplikasi terdistribusi berbasis *middleware*, dimana baik Giovani dan Nan Zhang melakukan pendekatan yang sama, diamana evaluasi kinerja dilakukan seawal mungkin. Berangkat dari ide dan pendekatan diatas, terutama metode *early performance testing*, maka akan di rancang suatu model pembanguna *e-government* berbasis kinerja.

2.1. e-Government.

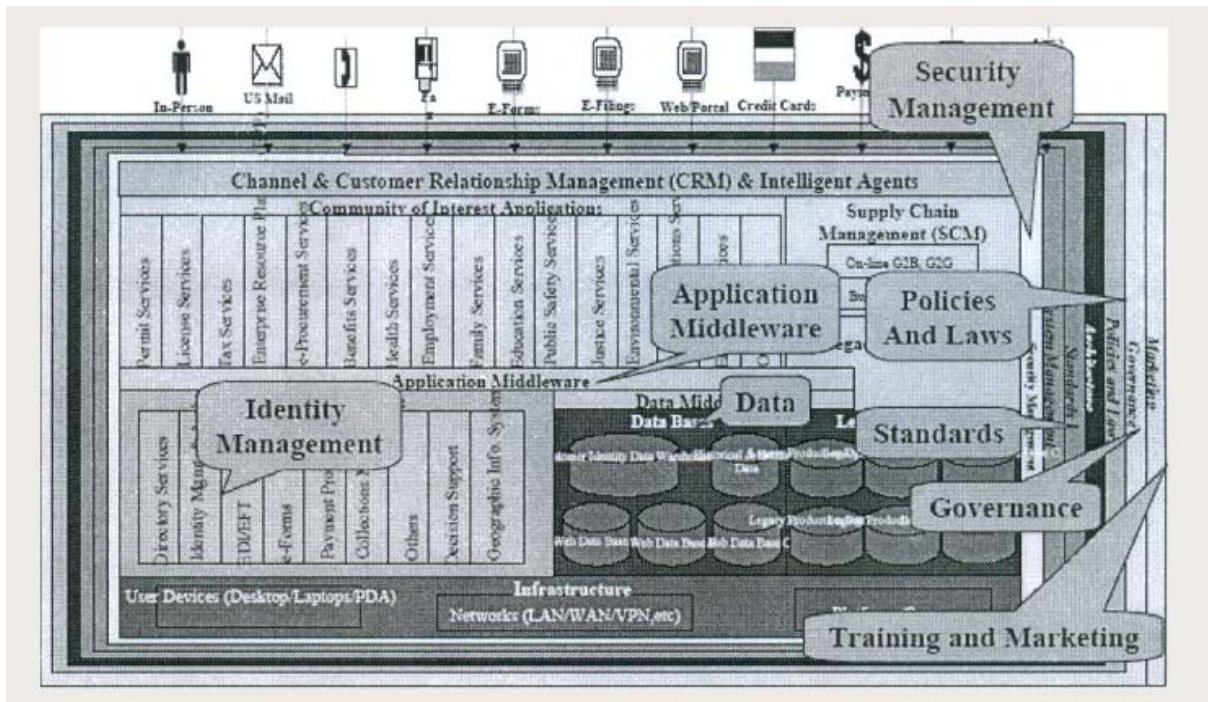
The World Bnk Group mendefinisikan *e-Government* sebagai berikut :

E-Government refers to the use by government agencies of information technologies (such as Wide Area Network, the Internet, and mobile computing) that have the ability to transform relation with citizens, businesses, and other arms of government.

Definisi lain dari Legislative Analyst's office :

Electronic government, or "e-government", is the process of transaction business between the public and government through the use of automated system and the Internet network, more commonly referred to as the Word Wide Web.

Pada intinya e-government adalah penggunaan TIK yang dapat meningkatkan hubungan antara pemerintah dan pihak lain, kemudian menghasilkan hubungan G2C (Government to Citizen), G2B (Government to Business Enterprises), dan G2G (inter government agency relationship).



Gambar 2.1. Middleware sebagai core dari E-Government Architecture

2.2. Kinerja Sistem Terdistribusi.

Saat ini sudah banyak teknologi yang dapat digunakan untuk membantu dalam pembangunan sistem terdistribusi. Beberapa aplikasi layanan pemerintahan tersebar di beberapa dinas, eksekutif, legislatif maupun masyarakat, dimana satu dan lainnya terkoneksi dengan infrastruktur jaringan yang terintegrasi pada pusat jaringan yang berada di sekretariat kabupaten (E-gov Selayar, 2003). Sebagai contoh Sistem Informasi kependudukan pada dinas Kependudukan, SI Pariwisata pada dinas Pariwisata, e-commerce pada dinas Penanaman Modal, e-Democracy pada DPRD, dll.

Karakteristik kinerja, seperti reponse time, throughput dan scalability merupakan faktor penting dari kualitas sistem terdistribusi. Dengan asumsi bahwa sistem atau aplikasi terdistribusi dibangun dengan teknologi middleware, seperti J2EE (Java 2 Enterprise Edition) atau CORBA (Common Object Request Broker Architecture).

Faktor-faktor lain yang juga mempengaruhi kinerja sistem terdistribusi adalah fungsionalitas dari teknologi middleware, seperti transactional dan services, remote communication primitives dan threading policy primitives. Pada tabel 2.1. dapat dilihat klasifikasi parameter-parameter utama yang berkaitan dengan pengujian kinerja dari sistem atau aplikasi terdistribusi (Giovani, 2004).

Hal yang menarik adalah, bagaimana model ini nantinya akan digunakan seawal mungkin dalam tahap-tahap pembangunan sistem dan menjadi jaminan yang dipakai bersama antara pemerintah dengan developer dan seluruh stakeholder yang terlibat dalam pembangunan e-government.

Tabel 2.1. Parameter kinerja sistem terdistribusi (Giovani, 2004)

Workload	Number of clients Client request frequency Client request arrival rate Duration of the test
Physical resources	Number and speed of CPU(s) Speed of disks Network bandwidth
Middleware configuration	Thread pool size Database connection pool size Application component cache size JVM heap size Message queue buffer size Message queue persistence
Application specific	Interactions with the middleware - use of transaction management - use of the security service - component replication - component migration Interactions among components - remote method calls - asynchronous message deliveries Interactions with persistent data - database accesses

2.4. CORBA.

CORBA adalah suatu standar untuk sistem objek oriented terdistribusi yang dikembangkan oleh OMG (*Object Management Group*), sebuah konsorsium yang terdiri lebih dari 800 perusahaan. CORBA memungkinkan kita menggunakan aplikasi tanpa adanya batasan platform, teknologi jaringan, bahasa pemrograman, maupun letak objek pemberi service yang dituju.

Tujuan CORBA adalah untuk pengembangan pemrograman objek terdistribusi. CORBA bukanlah bahasa pemrograman, melainkan merupakan suatu spesifikasi untuk mengembangkan objek-objek terdistribusi. Beberapa *software* yang mengimplementasikan CORBA misalnya ORBIX (oleh IONA *technologies*), VisiBroker (oleh Inspire), dan JavaIDL (oleh JavaSoft).

2.5. Pendekatan *Early Performance Testing pada Aplikasi Terdistribusi*.

Pendekatan ini terdiri dari beberapa langkah, sebagai berikut (Giovani, 2004) ;

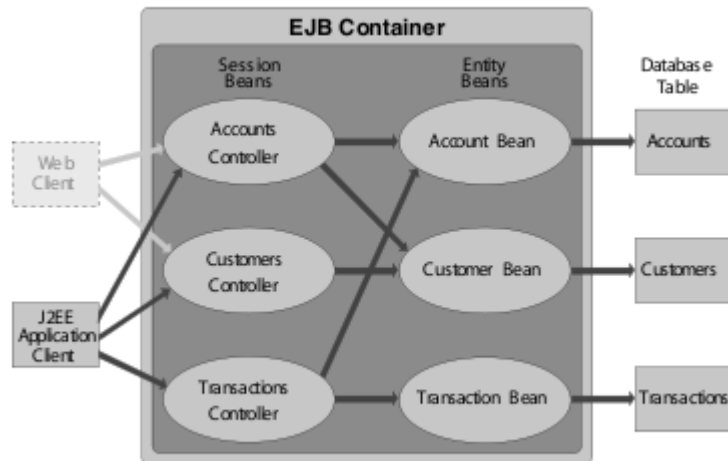
- Melakukan seleksi skenario *use-case* yang relevan terhadap kinerja, dan memberikan alternatif dari arsitektur perancangan.
- Pemetaan *use-case* yang sesuai terhadap penyebaran teknologi dan *platform*.
- Membuat *stubs* dari komponen-komponen yang tidak ada pada tahap awal pembangunan aplikasi, namun dibutuhkan untuk implementasi *use-case*.
- Eksekusi pengujian, yang meliputi ; penyebaran dari AUT (*application under test*), membuat *Workload generator*, inisiasi dari presistent data dan melaporkan hasil pengukuran kinerja.

Dalam pembangunan *e-government model* berbasis kinerja dapat menggunakan kombinasi dari metode GRAPPLE (*Guidlines for Rapid APPLication Engineering*) dan *Early Performance Testing*.

3. PENGUJIAN KINERJA APLIKASI TERDISTRIBUSI.

Evaluasi empiris dari kinerja dari aplikasi terdistribusi dilakukan dengan pengujian berdasarkan *middleware* dan atau komponen-komponen yang terdapat pada awal dari *software* proses. Evaluasi kinerja dilakukan pada kasus aplikasi perbankan (DBApp), yang merupakan aplikasi terdistribusi. Aplikasi ini dibangun dengan JAVA berisi 6000 LOC, dan merupakan salah satu fitur yang diberikan oleh *platform* J2EE, yang meliputi penggunaan transaksi dan keamanan (*security*). Selain itu DBApp di anggap aplikasi yang memadahi dari aplikasi terdistribusi berbasis komponen, dengan ukuran yang menengah.

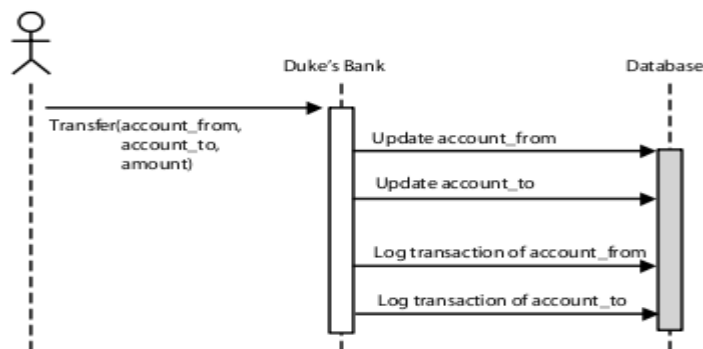
Organisasi dari DBApp dapat dilihat pada gambar 3.1. Aplikasi ini dapat di akses melalui Web dan aplikasi Client. ABApp terdiri dari enam komponen EJB (*Enterprise Java Bean*) yang menangani operasi yang dibutuhkan user pada hypothetic bank. Enam komponen tersebut dapat menyesuaikan dengan Class-class dari operasi yang berhubungan dengan bank account, customers and transaction, secara bergantian. Untuk setiap kelas operasi terdapat *session bean* dan *entity bean*. *Session bean* bertanggung jawab untuk *interface* pada user, dan *entity bean* menangani pemetaan komponen-komponen ke tabel database. Pola interaksi antar komponen di gambarkan menggunakan panah, lihat gambar 3.1.



Gambar 3.1. *The Duke's Bank Application*

EJBs yang merupakan komponen bisnis disebarakan dalam sebuah *container* dalam aplikasi server, merupakan bagian dari middleware. Dalam pengujian ini menggunakan Jboss application server dan MySQL database engine, dan dijalankan pada mesin yang sama.

Gambar 3.2. memperlihatkan *use-case* UML dari kasus diatas, dimana aplikasi Client menggunakan layanan Transfer yang tersedia pada DBApp. Layanan ini membutuh kan tiga parameter masukan, dua account dan jumlah uang yang akan ditrasfer. Komponen bisnis dari DBApp mendapatkan layanan menggunakan database untuk menyimpan data, database meminta empat kali, update balance dari dua account dan menyimpan rincian transaksi yang sesuai. Di asumsikan bahwa database engine adalah *software* yang sudah ada pada awal proses *software*.



Gambar 3.2. *Use-case for the Duke's Bank*

Pada pengujian digunakan database engine, struktur table dan SQL yang sama pada aplikasi yang sebenarnya. Perbedaan dari database, bisnis komponen dari DBApp di asumsikan tidak tersedia, sehingga dibuat dalam bentuk Stub-stub.

Implementasi Stub tersebut dengan memetakan abstract use case menggunakan *deployment* teknologi yang ada, yang dikenal adalah J2EE. Dalam penelitian ini belum memanfaatkan *software connector*, yang dilakukan adalah dengan menambahkan use-case dengan informasi yang dibutuhkan jika ; interaksi antara clients dan DBApp ditentukan bahwa layanan tranfer dipanggil sebagai suatu synchronous call dan memulai transaksi baru, kemudian

jika interaksi antara DBApp dan database ditentukan bawa empat pemanggilan pada sebuah transaksi tunggal yang berada pada sebuah SQL code, koneksi database telah di inisiasi setiap call. DBApp menggunakan *entity beans* dan mengelola *bean* untuk menangani interaksi terhadap tabel-tabel database.

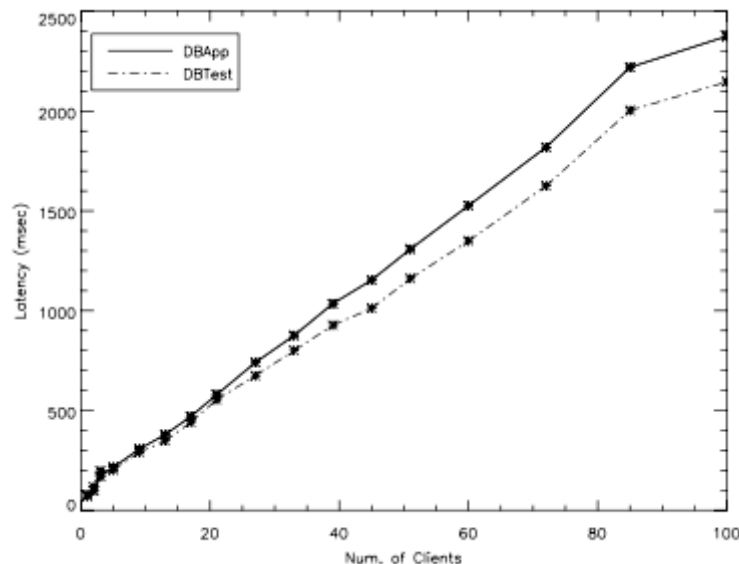
Berdasarkan informasi tersebut, di implementasikan Stub-stub yang diperlukan untuk mewujudkan interaksi pada *use-case* dan melakukan *deployment* (penyebaran) Test version dari DBApp di dalam Jboss *application server*.

Pada akhirnya, di implementasikan sebuah *workload generator* dan melakukan inisialisasi data-data pada database. *Workload generator* di aktifkan pada sejumlah clients bersamaan dengan melakukan pengukuran *response time* rata-rata. Untuk *persistence* data di inisialisasikan ketika setiap client melakukan transaksi penarikan uang dari *account* yang sesuai. Simulasi ini dilakukan pada ekelompok nasabah melalui internet.

Berdasarkan parameter-parameter kinerja yang terdapat pada tabel 2.1 didepan, dibuat sebuah *workload* dari kedua aplikasi baik DBApp maupun DBTest.

Simulasi dilakukan mulai satu clien sampai 100 clien meng-akses DBApp dan DBTest yang berada pada Jboss 3.0 *application server* dan di jalankan pada PC prntium III, CPU 1 Ghz, 512 MB RAM dan sistem operasi Lynux. Untuk membuat *workload* clien di jalankan pada Sun Fire 880 dengan 4 Sparc CPU 850 Mhz dan 8 GB RAM. Kedua mesin tersebut terkoneksi pada LAN dengan *bandwidth* 100 MBPS. Untuk Stub-stub digunakan jarak yang sama seperti komponen-komponen aplikasi sebenarnya.

Gambar 3.3 menggambarkan hasil eksperimen di atas, terlihat bahwa DBApp dan DBTest memberikan hasil yang mendekati sama, atau perbedaannya tidak signifikan.



Gambar 3.3. Hasil eksperimen DBApp dan DBTest

4. KESIMPULAN

Dari hasil pengujian menggunakan *Early Performance Testing* pada aplikasi terdistribusi DBApp yang merupakan fitur dari J2EE terhadap DBTest yang diantaranya terdiri dari komponen-komponen stub dapat disimpulkan bahwa pengujian dari kinerja aplikasi terdistribusi, termasuk aplikasi layanan pemerintahan (*e-government*) dapat dilakukan seawal mungkin pada tahap-tahap proses pembangunannya, sehingga dapat menciptakan *standard-baseline* kinerja yang perlu disepakati baik pemerintah, masyarakat dan seluruh stakeholder yang menggunakan layanan-layanan aplikasi pemerintahan tersebut, agar ada jaminan kinerja dari layanan tersebut nantinya saat di gunakan.

5. DAFTAR PUSTAKA

- Giovani Denaro et. All., 2004, *Early Performance Testing of Distributed Software Application*, WOSP 04, January 14-16-2004, Redwood City, CA. Copyright 2004 ACM 1-58113-673-0/04/0001
- Lukito, Edi Nugroho, 2004, *CORBA : Sistem terdistribusi*, Teknik Elektro UGM, www.komputasi.lipi.go.id
- Legislative Analyst's Office, 2001, E-Government in Clifornia, Providing Services to Citizens Through the Internet, www.lao.ca.gov/2001/012401_egovernment.html
- Nan Zheng, 2004, *Performance Testing of Middleware-Based Distributed Software Applications*, Seminar : Component Technologies, SS2004, www.informatik.tu.darmstadt.de/database/seminar/data/SCTseminar_final.pdf

Pemda Selayar, 2003, Kabupaten Maritim Selayar Menuju Era eGovernment, Disain, Arsitektur dan Implementasi e-Government.
Rahardjo, Budi, 2001, Membangun e-Government, PPAU Mikroelektronika ITB, Bandung
The Word Bank Group, 2001, E-Government Definition, www.worldbank.org/publicsector/definition.htm