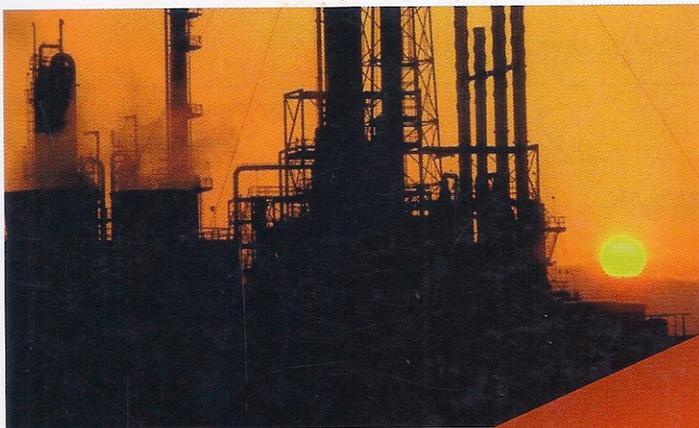




Seminar on Application and Research in Industrial Technology

SMART2008



***“Optimasi Ragam Aplikasi
Teknologi Industri dan
Pemanfaatan Energi untuk
Peningkatan Daya Saing Industri”***

Rabu, 27 Agustus 2008, Gedung Sekolah Pascasarjana UGM

A-8	PERENCANAAN ARSITEKTUR INFORMASI PERUSAHAAN STUDI KASUS UNIT RAWAT JALAN RUMAH SAKIT <i>Novie Theresia Br. Pasaribu</i>	062-069
A-9	ESTIMASI WAKTU PENYELESAIAN PERMINTAAN KONSUMEN MENGGUNAKAN SIMULASI PROSES PRODUKSI (STUDI KASUS: DIVISI CRM) <i>Ratna Ekawati dan Shanti Kirana A</i>	070-077
A-10	PENGEMBANGAN MODEL MATEMATIS ANTRIAN DENGAN KARAKTERISTIK <i>TIME-DEPENDENT</i> PADA JENIS ANTRIAN <i>SINGLE CHANNEL MULTI PHASE</i> <i>Sulis Liani dan Subagyo</i>	078-086
A-11	PERANCANGAN PERANGKAT PENILAIAN KINERJA UNTUK DOSEN BERDASARKAN KOMPETENSI DENGAN MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS (Studi Kasus di Universitas Kristen Maranatha) <i>Vivi Arisandhy, Melina Hermawan, dan Delina</i>	087-095
A-12	ANALISIS TINDAKAN PENANGANAN RISIKO PEMELIHARAAN MENARA TELEKOMUNIKASI (STUDI KASUS: PT X) <i>Yadrifil dan Martina Ratna Uli</i>	096-103
A-13	PENGEMBANGAN ALGORITMA INFERENSI DENGAN PENDEKATAN ITERATIF PADA ATURAN BERBASIS TERNARY GRID <i>Yuliadi Erdani dan Ismail Rokhim</i>	104-111
A-14	OWNERSHIP AND ECONOMIC PERFORMANCE OF THE INDONESIAN AUTOMOTIVE FIRMS: A QUAD SYSTEM CLUSTER DEVELOPMENT APPROACH <i>Abdusy Syakur Amin</i>	112-124
A-15	PENGUNAAN JAVAEE SIP SERVLET DALAM MENGGOMPOSISIKAN LAYANAN BERBASIS IMS <i>Audyati Gany</i>	125-131
A-16	USULAN STRATEGI OPERASI PRODUK SAMBUNGAN LANGSUNG JARAK JAUH (SLJJ) PT TELKOM <i>Gatot Yudoko dan Tedy Kristianto</i>	132-141
A-17	DECISION SUPPORT SYSTEM (DSS) UNTUK ANTISIPASI NON EXCUSABLE DELAY <i>Heri Suprpto</i>	142-147
A-18	POLA PERKEMBANGAN TEKNOLOGI DI SENTRA INDUSTRI PERAK (STUDI KASUS SENTRA INDUSTRI PERAK KOTAGEDE) <i>M. Arif Wibisono dan Rahmi Widayati</i>	148-157
A-19	STRATEGI PENGEMBANGAN KAWASAN INDUSTRI MASYARAKAT PERKEBUNAN (KIMBUN) KOMODITAS TEH DI KABUPATEN BANJARNEGARA <i>Poppy Arsil dan Masrukhi</i>	158-162
A-20	APLIKASI JARINGAN SYARAF TIRUAN UNTUK MENGENALI EKSPRESI WAJAH SECARA REAL TIME <i>Riyanto Sigit, Setiawardhana, dan Dadet Pramadihanto</i>	163-171

“Penggunaan JavaEE SIP Servlet dalam mengkomposisikan layanan berbasis IMS”

Audyati Gany

Jurusan Teknik Elektro, Universitas Kristen Maranatha
Jl.Prof.Drg. Suria Sumantri 65, Bandung 40164, Indonesia
Phone : +62+22 2012186, Fax : +62+22 2017622
Email : audyati.gany@eng.maranatha.edu
audy.gany@gmail.com

Abstrak

IP Multimedia Subsystem (IMS) memungkinkan para operator, penyedia layanan dan pengembang untuk mengkreasi suatu layanan baru dengan menggunakan standar routing SIP (*Session Initiation Protocol*). Untuk kelanjutan dan kelengkapan dalam mengkreasi layanan IMS, harus dibangun suatu susunan penggerak dengan menggunakan mekanisme inti IMS.

Sebagai contoh dalam mengkreasi layanan baru dengan menggunakan susunan penggerak beserta fungsi-fungsinya yaitu seperti pada aplikasi SIP dalam server aplikasi Java EE, SDP (*Service Delivery Platform*) atau standarisasi IMS *enabler*.

SIP *servlet container* didefinisikan sebagai suatu konsep versi JSR 289 yang berisi dan berfungsi untuk melaksanakan aplikasi-aplikasi SIP dan bersifat sebagai penyedia akses ke mekanisme SIP melalui Java API.

Java EE merupakan suatu platform dari *Java Enterprise Edition* yang digunakan pada domain telekomunikasi. Java EE SIP servlet merupakan suatu wadah dan susunan dari teknologi layanan.

Kata kunci : IMS, Java EE, SIP servlet

Pendahuluan

Kemajuan teknologi komunikasi telah berevolusi secara signifikan. Terutama dengan hadirnya fitur 3G. Fitur ini memungkinkan pengguna mengakses jaringan internet dengan cepat dan praktis, pengiriman data menjadi jauh lebih mudah dengan 3G, yang lebih dikenal dengan istilah *mobile broadband* internet. Kecepatan yang ada pada fitur ini-pun sekelas dengan *broadband* yang prosedur penggunaannya pun tidak jauh berbeda dengan alat komunikasi melalui internet lainnya.

Mobilitas konsumen yang selalu bergeser kelokasi-lokasi baru, juga menyebabkan para operator harus mempertimbangkan dalam peningkatan kemampuan layanan mereka. Membangun efisiensi layanan menimbulkan permasalahan pada faktor sistem integrasi yang terimplementasi pada penawaran ke konsumen.

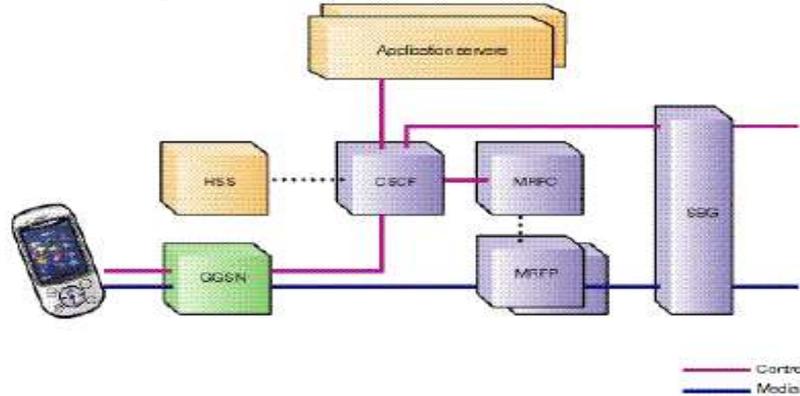
Komunitas JSR 289 sebagai standarisasi suatu antarmuka AR (*Application Router*) ke JavaEE SIP *servlet container* bertugas untuk mengendalikan perintah entitas SIP yang akan dihubungkan ke sesi SIP.

IMS (IP Multimedia Subsystem)

IMS berupa suatu *framework* yang memfasilitasi pengembangan beberapa tipe layanan multimedia, sehingga memudahkan pengguna untuk saling berkomunikasi dengan pengguna lain atau dengan PC (*Personal Computer*).

Arsitektur standard IMS (gambar 1), menawarkan VoIP (*Voice over IP*) dan jasa multimedia, IMS terintegrasi dengan baik pada jaringan data dan suara, IMS memungkinkan pendekatan layanan yang dikendalikan oleh lalu lintas *packet switched* dan SIP (*Session Initiation Protocol*) yang melakukan kriptografi untuk keamanan komunikasi, dengan memanfaatkan sirkuit dan teknologi *packet switched*. IMS adalah standar yang digunakan untuk

menyalurkan layanan-layanan yang mencakup multimedia yang juga menjembatani telekomunikasi dan teknologi internet.



Gambar 1. Arsitektur dari IMS

Arsitektur IMS ini terdiri atas beberapa lapisan (gambar 4), yaitu:

1. AS (*Application Server*) berfungsi sebagai tuan rumah dari berbagai layanan, terutama layanan *messaging* dan telepon.
2. HSS (*Home Subscriber Server*) HSS ini menyediakan informasi yang berhubungan dengan pelanggan.
3. CSCF (*Call/Session Control Function*) adalah node penting yang mempunyai tiga peranan pada fungsi pengendalian sesi panggilan, yaitu; S-CSCF (*Serving CSCF*), P-CSCF (*Proxy CSCF*), I-CSCF (*Interrogating CSCF*).
4. GGSN (*Gateway GPRS Support Node*) merupakan komponen utama jaringan selular yang menggunakan GPRS.
5. MRFC (*Media Resource Function Control*) dan MRFP (*Media Resource Function Payload*)
6. SBG (*Session Border Gateway*) melindungi sumber daya jaringan dan para pelanggan dengan cara: Memastikan pengguna adalah pelanggan yang mempunyai hak dan sah, mengendalikan karakteristik alur lalu lintas.

Java EE

Java adalah *OOP (Object Oriented Programming) language*, yang dikembangkan oleh Sun Microsystems. Saat ini, Java sudah diimplementasikan dalam empat keluarga teknologi yang berbeda, yang pertama adalah **Java Standard Edition (JavaSE)** untuk mengembangkan aplikasi desktop, **Java Enterprise Edition (JavaEE)** untuk mengembangkan aplikasi enterprise yang berorientasi pada arsitektur *server-client*, kemudian **Java Micro Edition (JavaME)** untuk mengembangkan aplikasi pada "*consumer devices*" seperti telepon genggam dan PDA, dan yang terakhir adalah **JavaCard** yang digunakan untuk implementasi pada *Smart Card*.

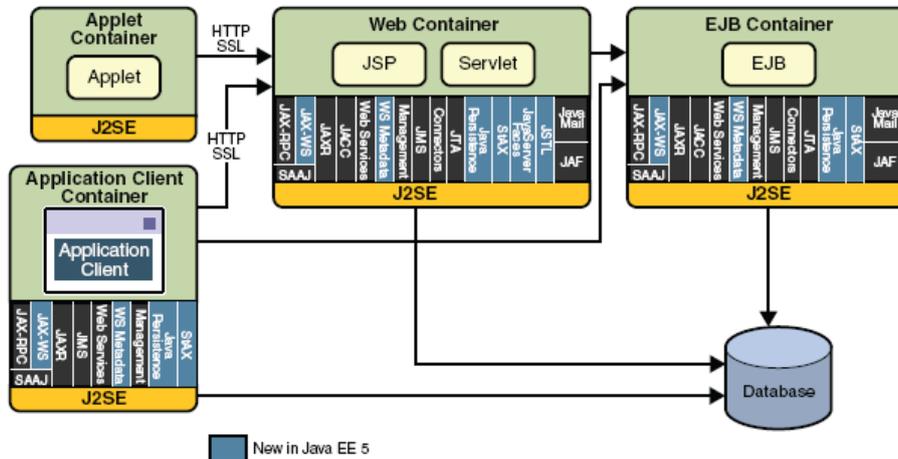
SunSPOT yang implementasinya juga menggunakan Java, memiliki karakter yang sedikit berbeda, yang bila diklasifikasikan, posisinya berada di antara "*consumer devices*" dan *Smart Card*, yang walaupun terdapat banyak *constraint* desain yang ketat, tetapi masih mempunyai fleksibilitas dalam pengembangan aplikasi. Tidak mengherankan bahwa beberapa konsep implementasinya didesain dalam pendekatan implementasi JavaME.

Ericsson, sebagai salah satu vendor terbesar perangkat telekomunikasi di dunia juga merupakan salah satu pendorong kemunculan teknologi IMS. Tidak hanya dari sisi perangkat, namun juga dari sisi pengembangan aplikasinya.

Pengembangan layanan komunikasi dengan JavaEE merupakan tantangan sendiri, terutama pada saat membangun layanan berbasis sinkronisasi HTTP, tetapi dengan SIP *servlet-container* dan SIP APIs, layanan komunikasi yang dibangun menjadi lebih untuk diimplementasikan, dan dengan JSR 289 SIP memungkinkan komunikasi application server.

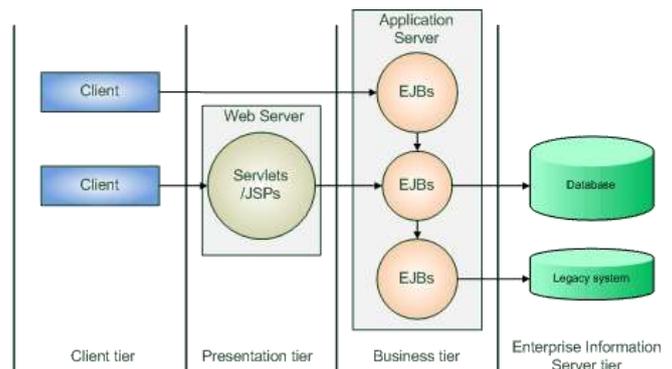
Arsitektur javaEE (gambar 2), meliputi : komponen-komponen UI (*User-Interface*) yang menggambarkan entitas html, APIs berfungsi untuk penataan dan validasi masukan, daftar tag untuk menempatkan JSF pada JSP. Fungsi tambahan dari kontainer adalah menyediakan konektivitas dengan sumber dari luar , implementasinya atau pengganti APIs,

dan sekuritas. JSF (*Java Server Faces*) adalah framework aplikasi web berbasis Java yang merupakan bagian dari JavaEE yang spesifik. Standar layanan, meliputi layanan komunikasi dengan HTTP, HTTPS, JMS (*Java Message Service*), konektivitas API untuk berelasi dengan database Java. Layanan sekuritas berupa JAAS (*Java Authentication and Authorization Service*) yang memungkinkan untuk membuktikan dan menjalankan akses kendali, mengimplementasikan Java sebagai standar PAM (*Pluggable Authentication Module*), dan JACC (*Java Authorization Service Provider Contract for Container*) yang menetapkan kontrak antara server aplikasi JavaEE dengan penyedia layanan otorisasi.



Gambar 2. Arsitektur JavaEE

Platform *Java Enterprise Edition* atau JavaEE adalah platform yang digunakan untuk membangun dan melaksanakan pendistribusian, aplikasi-aplikasi *multi-tier*. Aplikasi-aplikasi JavaEE didesain untuk melaksanakan infrastruktur perangkat lunak yang spesifik, sebagai contoh yaitu infrastruktur yang digambarkan pada diagram digambar 3.



Gambar 3. infrastruktur perangkat lunak

Pada diagram diatas, didapat aplikasi-aplikasi multi-tier yang digunakan atau terkoneksi ke database untuk menaruh dan menyajikan data. Web server yang berfungsi melayani permintaan dan melaksanakan fungsi-fungsi penayangan, dan disisi ujung dari klien akan bervariasi (sebagai contoh termasuk Apache Tomcat atau server IBM HTTP). Dengan catatan bahwa web server dapat disimpan pada Application Server untuk aplikasi-aplikasi yang lebih kecil, seperti yang biasa digunakan oleh bermacam klien (seperti : penjelajah internet, aplikasi-aplikasi desktop) untuk mengoperasikan aplikasi.

Ada dua hal penting yang perlu diperhatikan dalam merealisasikan infrastruktur ini, yaitu :

1. Aplikasi JavaEE berisi sejumlah modul-modul yang diperlukan untuk penyebaran kesetiap komponen infrastruktur
2. Beberapa vendor berbeda menyediakan komponen infrastruktur agar aplikasi dapat disebarkan.

JavaEE server adalah server aplikasi yang mengimplementasikan platform JavaEE APIs dan menyediakan layanan berstandar JavaEE, yang seringkali pula disebut Application Server (AS), karena digunakan untuk melayani aplikasi data ke klien, seperti halnya pada web server yang menyajikan halaman-halaman web ke penjelajah web.

JavaEE container merupakan antarmuka antar komponen dan fungsional tingkatan yang lebih rendah, yang disediakan oleh platform JavaEE untuk mendukung komponen tersebut. Fungsi dari container dipertegas oleh platform JavaEE, dan ini berbeda untuk tiap tipe komponen.

Session Initiation Protocol (SIP)

SIP (*Session Initiation Protocol*) adalah suatu protokol pengendali pada tingkat aplikasi yang awalnya dibangun oleh *IETF MMUSIC Working Group* dan sudah menjadi standar dalam mengatur sesi diantara satu atau lebih klien pada jaringan data IP, dan merupakan protokol pensinyalan yang baik untuk VoIP (*Voice Over IP*). SIP merupakan protokol berbasis *text* seperti pada HTTP (*Hypertext Transfer Protocol*) dan SMTP (*Simple Mail Transfer Protocol*) untuk berinteraktif pada sesi komunikasi awal di antara pengguna. Beberapa sesi dilengkapi pula dengan *voice, video, chat, game interaktif* dan *virtual reality*.

Tujuan dari SIP adalah memberikan fungsi-fungsi proses pemanggilan yang sebelumnya sudah ada dalam jaringan PSTN, seperti fitur operasi standar antara lain menghubungkan satu nomer tertentu, melakukan panggilan dan sebagainya.

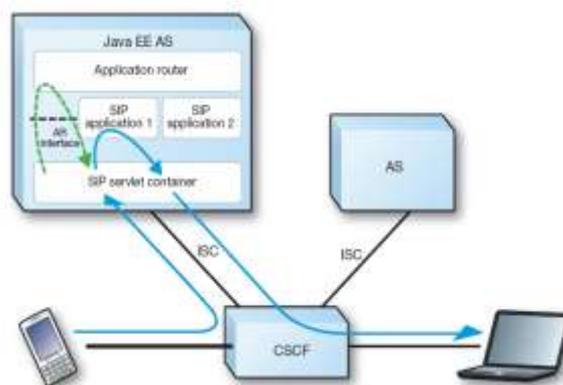
SIP juga bekerja untuk panggilan jarak jauh melalui sistem telephoni tradisional, SIP mengacu pada suatu protokol yang dilengkapi komputer untuk melakukan percakapan satu dengan lainnya tanpa harus melalui stasiun sentral. SIP ditawarkan dalam dua format yaitu berbasis komputer dan berbasis perangkat keras.

Penerapan JavaEE SIP Servlet dalam layanan berbasis IMS.

SIP servlet container yang terdefinisi pada konsep versi 289 memuat dan mengatur aplikasi-aplikasi SIP dan menyediakan akses ke mekanisme SIP (*Session Initiation Protocol*) melalui Java API.

Platform JavaEE sebanding dengan platform yang digunakan dalam domain telekomunikasi. *JavaEE AS (Application Server)* adalah platform yang mana SIP servlet container disebarkan. AS menyebarkan jaringan layanan melalui permintaan SIP yang kemudian direspon agar dapat dikirimkan dan diterima.

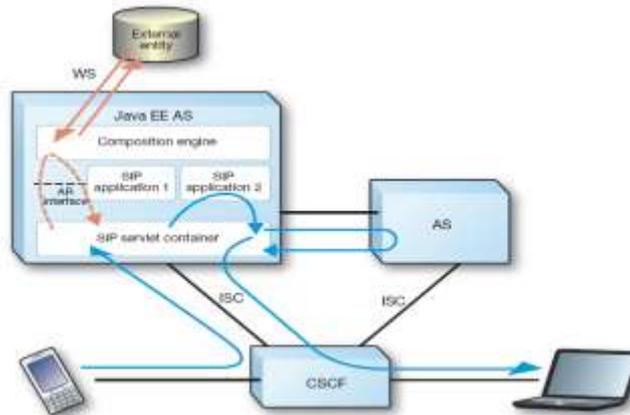
AS pada IMS terkoneksi ke CSCF (*Call Session Control Function*) melalui suatu antarmuka. ISC (*IMS server control*), mengajukan permintaan, yang diterima oleh AS dari CSCF dan kemudian melukukannya ke kontainer. Kontainer mengidentifikasi aplikasi SIP yang diminta untuk menetapkan suatu entitas yang disebut AR (*Application Router*). Kemudian kontainer mengirimkan permintaan tersebut ke aplikasi SIP yang dipilih. Penyediaan aplikasi SIP bukan berarti mengakhiri permintaan, kontainer akan meminta AR lagi untuk aplikasi SIP selanjutnya (Gambar 4.), dengan mendorong rute diatas rute SIP awal (seperti kebanyakan cara seperti rute CSCF pada suatu aplikasi server IMS).



Gambar 4 . SIP Servlet container

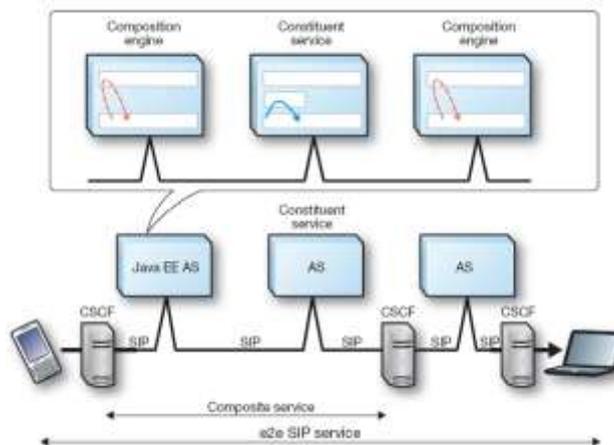
Susunan *engine* tidak membatasi layanan SIP, contohnya pada layanan web, dapat digunakan untuk permintaan dari entitas luar dalam memesan untuk membuat ketegasan jalur yang dapat digunakan. Pada gambar 5, digambarkan sebagai contoh hubungan ke aplikasi SIP 2 atau untuk melaksanakan proses bisnis SDP, seperti penetapan biaya atau untuk melakukan penagihan ke pengguna.

Lebih lanjut, aplikasi-aplikasi SIP yang membuka layanan SIP akan tersebar pada aplikasi-aplikasi layanan yang berbeda. Dalam hal ini, susunan *engine* dapat menggunakan antarmuka AR untuk mendorong rute SIP pada titik tersebut ke layanan aplikasi yang lain.



Gambar 5 . external data yang digunakan pada suatu route SIP request

Susunan layanan SIP merupakan teknologi yang dinamis, seperti yang digambarkan pada artikel ini yang menggambarkan framework aplikasi SIP, CSCF akan memperkenalkan sesi mana dari susunan engine yang dapat dioperasikan dan dihubungkan ke entitas SIP yang sesuai untuk iFC (*initial filter criteria*). Dengan menggunakan bagian dalam penyimpanan atau data yang diperoleh, susunan engine akan menggunakan mekanisme jalur SIP sejumlah entitas SIP yang dinamis (atau layanan pemilih), seperti ditunjukkan pada gambar 6, selama permintaan ruting SIP awal, susunan engine menjadi pelengkap CSCF.

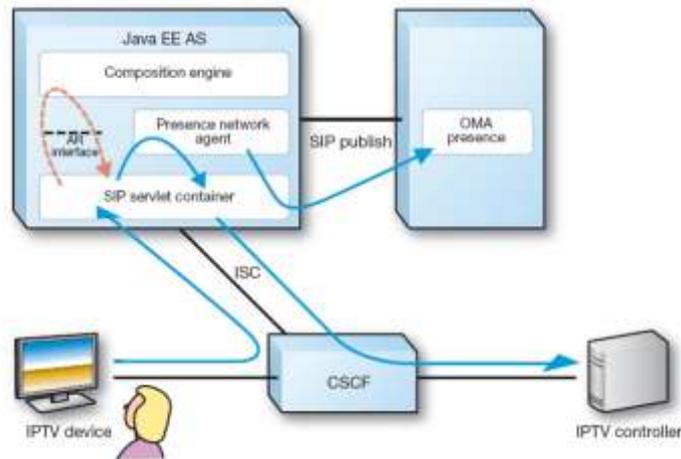


Gambar 6 . proses mengkreasi layanan komposisi dengan menggunakan routing SIP ke sejumlah pemilih layanan

Contoh implementasi pada IPTV

Pertimbangan hipotesis, misalkan pencampuran aplikasi IPTV berbasis IMS yang memungkinkan aplikasi digunakan untuk saling berinteraksi dalam percakapan (*chat*). Misalkan saat pengguna berada dirumah, dan menginginkan temannya melihat dengan menggunakan kanal IPTV seperti yang sedang dilihatnya, pendekatan tradisional untuk implementasi fungsi ini memerlukan memodifikasi perangkat IPTV untuk mengirimkan pesan SIP PUBLISH ke server yang kemudian akan memberitahu teman si pengguna. Upaya melibatkan dalam modifikasi perangkat IPTV akan terasa pengaruhnya pada pangsa pasar.

Selain itu perubahan pengendalian IPTV pada kanal penerima pilihan informasi akan mengalami penambahan kompleksitas (gambar 7).



Gambar 7. Susunan engine dalam jaringan agent berdialog IPTV SIP

Saat pengguna memilih kanal, susunan engine menangkap pesan SIP yang membawa informasi mengenai kanal yang dipilih, susunan engine menentukan (melakukan kesesuaian yang tepat) mengenai diketahui atau tidak termasuk kehadiran jaringan agent (PNA = *Presence Network Agent*) pada sesi IPTV SIP. Jika PNA sudah termasuk, susunan engine melaksanakan PNA, dan mengirimkan pesan SIP PUBLISH yang memberitahukan kanal terpilih untuk server yang tersedia (*presence server*).

Seperti diperlihatkan diatas, susunan engine dapat membuat rincian, yang keadaannya tergantung pada keputusan tentang keterlibatan layanan pemilih pada basis per-session. Selain itu diperlihatkan pula bahwa dimungkinkannya untuk digunakan sebagai logika bisnis, yang dapat dipakai atau diperluas dari suatu aplikasi IMS oleh susunan layanan tanpa perubahan produk sebenarnya serta kode awalnya.

Kebijakan lain dapat ditetapkan dengan datangnya informasi, yang mana pelanggan dapat mengubah kebijakan melalui suatu portal yang menyediakan akses untuk kebijakan database. Semua kebijakan diubah secara otomatis dengan memperhatikan laporan pada saat pengaturan untuk sesi selanjutnya.

Kesimpulan

Komunitas yang menjadi bagian JSR 289 harus melakukan menyesuaikan dengan standar AR (*Application Router*) sebagai antarmuka ke *JavaEE SIP Servlet Container*, yang mengendalikan perintah dalam entitas SIP yang terhubung ke sesi SIP.

Peningkatan susunan engine ditawarkan melalui cara yang fleksibel dalam mengimplementasikan layanan-layanan tambahan.

Implementasi susunan engine suatu algoritma *data-driven* menyediakan dinamika kendali sesi SIP sesuai dengan keadaan.

Sebagai bagian dari informasi pada pensinyalan SIP, susunan engine dapat digunakan sesuai dengan keadaan, sebagai pembatas, untuk data dari luar (misal layanan web) dan deskripsi resmi dari layanan SIP dalam membuat keputusan yang spesifik untuk permintaan ruting SIP.

Prinsip pengkomposisian layanan memperkenankan pemisahan mengenai dan aturan : perancangan aplikasi-aplikasi SIP dan pemosisian layanan SIP ke yang lainnya.

Faktor kritis untuk keberhasilan dikemudian hari adalah kontinuitas kerjasama dalam komunitas menuju framework bersama termasuk APIs, perluasan SIP dan format deskripsi layanan SIP.

Daftar Pustaka

- Ericsson Review No.3.2007
- JSR 289 Java Specification Requests SIP Servlet v1.1, <http://jcp.org/en/jsr/overview>
- Java Platform, Enterprise Edition (JavaEE), <http://java.sun.com/javaee>

- Nian Li-Ze & Mark S. Drew, “*Fundamentals of Multimedia*”, Pearson Prentice Hall, USA, 2004
- The Java EE5 Tutorial – September 2007
- The 3G IP Multimedia Subsystem, Merging the Internet and the Cellular Worlds, Gonzalo Camarillo & Miguel A. Garcia-Martin, John Wiley & sons Ltd. 2004
- www.ericsson.com

Penulis :

Audyati Gany, Staf pengajar Jurusan Teknik Elektro Universitas Kristen Maranatha, Bandung, Bidang konsentrasi Telekomunikasi, lahir di Jakarta