

IMPLEMENTASI PERSONAL PREFERENCE PADA SISTEM SHARE-IT BERBASIS PADA PLATFORM UNIX

Ary Mazharuddin S. - Agus Purwono – Yudhi Purwananto

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Email : shidiq@its-sby.edu

ABSTRAK

Arsitektur UPnP memungkinkan suatu aplikasi ataupun device untuk dapat terhubung ke dalam suatu jaringan dan berkomunikasi dengan aplikasi ataupun device lain di dalam jaringan tersebut tanpa perlu adanya konfigurasi (zero configuration) serta meniadakan batas-batas sistem operasi ataupun bahasa pemrograman yang digunakan untuk membuat aplikasi tersebut. Dengan sifatnya yang universal tersebut, UPnP semakin banyak dikembangkan serta dipergunakan. Membuka kemungkinan untuk terciptanya babak baru di dalam interaksi manusia dan komputer.

Sistem Share-It adalah aplikasi yang menerapkan arsitektur UPnP (Universal Plug and Play) untuk berbagi content dengan aplikasi lain yang sejenis. Content yang dimaksud di sini adalah file mp3. Sistem Share-It terdiri dari Server Device dan Client Device. Server Device berfungsi untuk menyediakan content, sedangkan Client Device mempunyai fungsi sebagai player yang dapat memainkan content file serta juga menyediakan content untuk dapat berbagi pakai dengan Client Device yang lain.

Dalam pengerjaan Tugas Akhir ini akan dikembangkan dan diimplementasikan Personal Preference pada sistem Share-It. Personal Preference itu sendiri mempunyai tujuan untuk memilihkan content file yang tersedia di jaringan sesuai dengan preference (pilihan) dari user. Pilihan (preference) tersebut berdasarkan dari kebiasaan user tergantung dari kondisi lingkungan sekitar device.

Kata kunci : UPnP, Share-It, Personal Preference.

1. PENDAHULUAN

Kemajuan yang begitu pesat di bidang TI (Teknologi Informasi) tidak bisa dipungkiri salah satu penyebabnya adalah fungsi awal dari TI yang diharapkan dapat semakin membantu atau mempermudah pekerjaan manusia. Semakin hari, manusia dengan beragam kompleksitas dalam kehidupannya semakin menuntut TI untuk lebih memudahkan dan memanjakan mereka.

UPnP (*Universal Plug and Play*) adalah arsitektur jaringan yang terdistribusi dan terbuka, didefinisikan oleh protokol yang digunakan, tidak bergantung pada sistem operasi manapun, bahasa pemrograman apapun, serta media perantara fisik. Sehingga UPnP memungkinkan vendor-vendor berbagai sistem operasi untuk membuat API (*Application Programming Interfaces*) yang sesuai dengan permintaan konsumen mereka.

Skenario *Share-It* mengilustrasikan penggunaan UPnP untuk automasi *sharing file audio* secara otomatis di tempat umum. Seseorang yang sedang bersantai atau menunggu seseorang di tempat-tempat umum seperti bandara, kafe, rumah makan atau tempat umum lainnya. Orang tersebut dapat menyalakan *device* atau peralatan MP3 player miliknya yang UPnP *enabled*, setelah menyala, peralatan tersebut terhubung pada jaringan yang berada di tempat itu. Peralatan tersebut kemudian mendeteksi sebuah aplikasi yang tertanam di tempat itu yang menyediakan berbagai macam file MP3

(aplikasi pusat) serta peralatan sejenis milik orang lain yang berada pada tempat yang sama (pada jaringan yang sama juga tentunya). Orang tersebut dengan peralatan miliknya dapat memainkan *file MP3* yang tersedia di aplikasi pusat atau pada peralatan sejenis lainnya sembari menunggu ataupun bersantai.

Dengan mengimplementasikan *Personal Preference* diharapkan sistem *Share-It* dapat lebih memanjakan *user*. *User* tidak perlu lagi memilah-milah daftar lagu yang tersedia. *Personal Preference* ini yang akan memilihkan lagu sesuai dengan kebiasaan *user*.

Tujuan dari pembuatan Tugas Akhir ini adalah untuk mengembangkan sistem *Share It* yang sudah ada, dengan mengimplementasikan fitur *personal preference* pada sistem ini dengan berbasis ada platform UNIX.

Permasalahan yang diangkat adalah :

- Bagaimana mengembangkan perangkat lunak *Share-It* dengan berbasis pada platform UNIX, yang dalam arsitektur UPnP dikenal dengan *device*, yang berfungsi sebagai *server side*, sedemikian sehingga mampu untuk diidentifikasi oleh peralatan lain yang baru ter-*attached* ke dalam jaringan dimana aplikasi tersebut berada, serta *broadcast* ke peralatan baru tersebut tentang servis yang tersedia yakni *share file MP3* yang dimilikinya sehingga peralatan lain dapat memainkan *file* tersebut dengan cara *streaming*.

- Bagaimana mengembangkan aplikasi *Share-It* yang berfungsi sebagai peralatan / *device* seperti dalam skenario diatas pada *platform* UNIX sebagai suatu perangkat lunak *client side* yang juga berfungsi sebagai *control point* yang dapat men-streaming file MP3 dari *control point* serta *device* lainnya dan memainkannya, men-share file MP3 yang dimiliki, serta memilih file MP3 yang tersedia di jaringan UPnP berdasarkan *personal preferences* yang telah ditentukan sebelumnya.
 - Bagaimana memodelkan *Personal Preference* dengan menggunakan RDF (*Resource Description Framework*) pada sistem *Share-It* sedemikian sehingga sistem tersebut mampu untuk memilih file mp3 yang tersedia di jaringan berdasarkan pada *preference* yang telah ditentukan sebelumnya seperti, album, penyanyi, genre lagu dan metadata lainnya yang terdapat pada file mp3.
 - Bagaimana memadukan *Personal Preference* dengan sistem *Share-It* sedemikian sehingga perpaduan sistem tersebut dapat memudahkan pengguna untuk memilih file mp3 yang terdapat di dalam jaringan.
 - Bagaimana mengatur hubungan antara masing-masing *control point* dengan *device* sehingga dapat mensimulasikan skenario tersebut dengan arsitektur UPnP.
- Dari permasalahan-permasalahan di atas, maka batasan masalah dalam Tugas Akhir ini adalah :
- Pengembangan perangkat lunak *Share It* dilakukan dengan menggunakan arsitektur UPnP dengan berbasis pada sistem operasi UNIX dan bahasa pemrograman Java.
 - Pembuatan salah satu komponen dari sistem *Personal Preference* yakni *user profile* yang seharusnya dihasilkan oleh sistem *Share-It* dan merupakan hasil dari pembelajaran sistem tersebut terhadap kebiasaan pengguna, adalah di luar dari lingkup Tugas Akhir ini.
 - Kondisi di lingkungan sekitar, seperti cuaca, kondisi pengguna, ruangan tempat pengguna berada, yang dipergunakan sebagai salah satu komponen dalam sistem *Personal Preference* adalah variabel yang ditentukan oleh penulis. Bukan hasil keluaran dari sensor pada aplikasi atau *device* sistem *Share-It* ini.

2. ARSITEKTUR TCP/IP

Jaringan dengan protokol TCP/IP (Transmission Control Protocol / Internet Protocol) adalah fondasi teknis dari internet dan LAN(Local Area Network) paling modern. [3] Jaringan ini memberikan mekanisme untuk komunikasi antar komputer dan untuk memindahkan informasi. TCP/IP adalah sekumpulan standar tentang mekanisme kerja jaringan, sehingga software dan hardware dari

berbagai vendor yang berbeda dapat saling berkomunikasi.

TCP/IP terdiri dari banyak protokol yang berbeda-beda. Setiap protokol berkaitan dengan aspek tertentu dari jaringan dan berbeda dengan aspek lainnya. Protokol-protokol tersebut tersusun sebagai sebuah layer dalam stack, di mana:

- Setiap layer memberikan fasilitas yang dapat digunakan oleh layer di atasnya.
- Setiap layer menggunakan fasilitas yang disediakan oleh layer protokol di bawahnya.

Konsep layer ini membuat layer pada lapisan atas tidak perlu mengetahui kerumitan mekanisme yang terjadi pada layer di bawahnya. Misalnya layer IP tidak perlu mengetahui mekanisme bagaimana paket-paket IP ditransmisikan pada jaringan fisik oleh layer fisik (physical layer).

Adapun protokol- protokol yang menyusun jaringan TCP / IP adalah sebagai berikut :

Layer Fisik / Network Interface

Layer ini bertanggungjawab untuk mengirim dan menerima data pada media fisik. Contohnya adalah ethernet, SLIP (Serial Line Interface Protocol) dan PPP (Point to Point Protocol).

Agar dapat bekerja, TCP/IP harus menggunakan beberapa sistem hardware jaringan yang mendasar. Sejauh ini Ethernet adalah sistem yang paling umum untuk LAN. Namun TCP /IP juga banyak berjalan pada tipe hardware yang berbeda-beda seperti token ring, jalur telepon dial-up melalui modem (implementasi protokol SLIP dan PPP).

TCP/IP berinteraksi dengan Ethernet (layer fisik jaringan) untuk menggunakan kapabilitasnya untuk menggerakkan paket di sepanjang kabel. Ethernet menentukan standar untuk interface elektrik, tipe kabel, konfigurasi konektor dan sebagainya, serta bagaimana hardware mengirim sebuah paket ethernet dari satu mesin ke mesin lain pada kabel yang sama.

Layer Internet / Internet Protocol (IP)

Semua layer di atas layer internet harus melewati layer ini untuk dikirimkan menuju alamat yang tepat. Pengiriman paket melalui IP bersifat unreliable, connectionless, datagram delivery service yang artinya tidak menjamin datagram yang dikirim pasti sampai ke tempat tujuan dan tidak ada mekanisme handshaking antara pihak pengirim dan penerima (connectionless). Paket data yang dikirim bersifat independen terhadap yang lain (datagram delivery service) sehingga tiap paket tidak selalu melalui rute yang sama walaupun tujuan akhirnya sama.

Layer Transport / Host to Host

Layer Transport merupakan layer komunikasi data yang mengatur aliran data antara dua host (host to host), untuk keperluan aplikasi di atasnya. Ada dua buah protokol pada layer ini yaitu TCP dan UDP. Dua protokol inilah yang nantinya terlibat langsung dalam penyusunan arsitektur UPnP (selain HTTP).

TCP

TCP / Transmission Control Protocol menyediakan service yang dikenal sebagai connection oriented, reliable, byte stream service.

Connection Oriented berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan (handshake) terlebih dahulu. Reliable berarti TCP menerapkan proses deteksi kesalahan dan retransmisi. Byte Stream Service berarti paket dikirimkan ke tujuan secara berurutan.

UDP

UDP / User Datagram Protocol merupakan protokol tranposrt yang connectionless. Tidak ada pengurutan kembali paket yang datang , acknowledgment terhadap paket data yang datang, atau retransmisi jika ada paket yang bermasalah.

Kemiripan UDP dengan TCP adalah penggunaan port number. Port number ini membedakan pengiriman datagram ke beberapa aplikasi berbeda yang terletak pada komputer yang sama.

UDP bersifat unicast, multicast atau broadcast. Sifat multicast dan broadcast yang bertujuan mengirim ke banyak client akan efisien jika menggunakan metode connectionless. Kemampuan UDP utamanya dalam unicast (TCP juga bersifat unicast namun connection oriented) dan multicast inilah yang dimanfaatkan UPnP dalam tahap Discovery.

Layer Application

HTTP

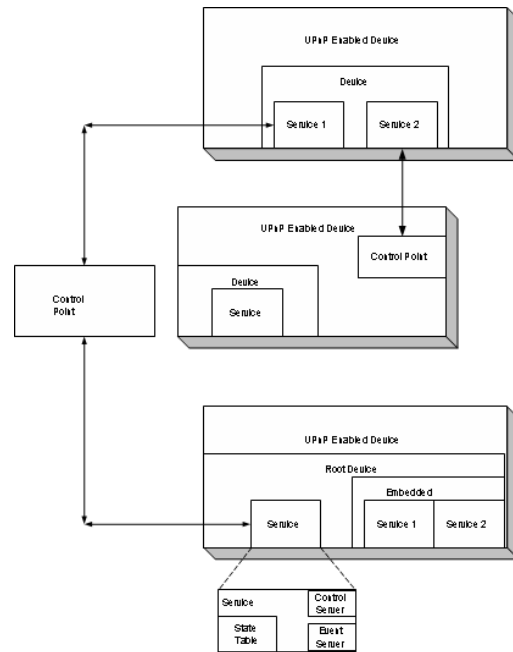
HTTP / Hyper Transfer Text Protocol merupakan protokol layer application yang biasa digunakan untuk la WWW (World Wide Web / web). HTTP digunakan untuk mengirimkan informasi dalam berbagai format, bahasa, dan kumpulan karakter. Sintaks pesan HTTP berdasarkan pada MIME-Multipurpose Internet Mail Extensions

Protokol HTTP bersifat request-response, yaitu dalam protocol ini client menyampaikan pesan request ke server dan server kemudian memberikan response yang sesuai dengan request tersebut. Sifat request dan response ini dimanfaatkan oleh UPnP untuk komunikasi data pada layer application dengan menggunakan protokol – protokol tambahan (SSDP, GENA, dan SOAP) yang di-extend dalam pesan HTTP.

3. ARSITEKTUR UPNP

Arsitektur UPnP merupakan arsitektur yang dibangun atas dasar protokol TCP/IP dengan memanfaatkan protokol HTTP yang di-extended dengan protokol lain pada Layer Aplikasinya.

Susunan paling dasar dari sebuah jaringan UPnP adalah terdiri dari *devices*, *control points*, dan servis. Berikut ini adalah diagram jaringan UPnP dengan komponen-komponennya.



Gambar 1. Jaringan UPnP dengan Komponennya

Sebuah UPnP *device* adalah sebuah wadah atau tempat dari berbagai macam servis dan *nested device* yang tertanam di dalamnya.

Sebuah UPnP servis adalah unit atau komponen yang paling kecil pada sebuah jaringan UPnP. Servis menyatakan tentang aksi-aksi yang bisa dilakukan serta memodelkan status dari aksi-aksi tersebut dengan menggunakan *state variable*.

Sebuah *control point* pada jaringan UPnP adalah sebuah pengontrol yang mampu untuk menemukan dan mengendalikan *device* yang lain.

Tahapan dalam UPnP terdiri dari :

1. *Addressing*
Addressing merupakan langkah awal (ke-0) dalam skenario jaringan UPnP. Pada tahap ini *device* UPnP mendapatkan alamat IP atau nama *host*. Alamat tersebut bisa didapatkan secara dinamis melalui DHCP ataupun melalui konfigurasi IP statis.
2. *Discovery*
Discovery adalah langkah pertama yang harus dilakukan pada jaringan UPnP. Ketika sebuah *device* pertama kali terhubung ke dalam sebuah jaringan, protokol *discovery* dari UPnP memungkinkan *device* tersebut untuk mengumumkan servis-servis yang dimilikinya kepada *control point-control point* yang terdapat di jaringan. Begitu juga ketika sebuah *control point* terhubung pada sebuah jaringan, protokol *discovery* dari UPnP (*SSDP, Simple Discovery*) memungkinkan *control point* tersebut untuk mencari *device* dengan tipe yang diinginkan pada jaringan tersebut
3. *Description*
 Pada tahap *discovery*, *Control point* masih sedikit mengetahui tentang informasi sebuah

Device (hanya informasi yang dibawa oleh *Discovery messages* yaitu tipe *device* atau servis UUID, dan URL lokasi UPnP *device description* berada. Supaya *Control point* mengetahui lebih detail informasi tentang sebuah *device* maka *Control point* harus menerima deskripsi dari *device* tersebut dan servis yang disediakan *device* tersebut.

4. *Control*
Control merupakan langkah ketiga dalam skenario jaringan UPnP. Dalam langkah ini, *control point* bisa melakukan *invoke action* pada *device* dan *query control* terhadap *state variable* untuk mendapatkan *actual value*.
5. *Eventing*
Eventing merupakan langkah keempat dalam skenario jaringan UPnP. Melalui *eventing control point* memantau perubahan nilai *state variable* pada *device*.
6. *Presentation*
Presentation merupakan langkah kelima dalam skenario jaringan UPnP. *Presentation* ditampilkan dalam model *HTML* untuk *controlling* atau melihat status *device*.

4. RESOURCE DESCRIPTION FRAMEWORK (RDF)

Pada level yang paling sederhana, *Resource Description Framework* (RDF) adalah sebuah bahasa yang berbasis XML yang dipergunakan untuk mendeskripsikan sebuah *resource*. Sedangkan definisi dari *resource* itu sendiri cukup luas. Dimulai dengan pengertian umum dari *resource* yaitu sebagai *file* elektronik yang tersedia melalui web. *Resource* seperti itu dapat diakses melalui *Uniform Resource Locator* (URL). Jika dokumen-dokumen XML menambahkan metadata sebagai bagian dari dokumen tersebut, salah satu kegunaan dari RDF adalah untuk menciptakan metadata sebagai bagian yang berdiri sendiri. Jadi, daripada menandai atau menjelaskan bagian dalam (internal) dari sebuah dokumen, RDF mengambil metadata tentang bagian luar (eksternal) dari dokumen tersebut, seperti pengarang, tanggal pembuatan ataupun tipe dari dokumen tersebut.

Fungsi utama dari RDF adalah mendeskripsikan sebuah *resource*. Sehingga dasar dari model RDF adalah sebuah *resource*. Semua hal bisa dikatakan sebagai *resource* atau literal. Segala sesuatunya yang dapat diidentifikasi melalui URI adalah sebuah *resource*. Sebuah *resource* yang menggambarkan tentang *attribute*, karakteristik, atau hubungan antar *resource*, dinamakan *property*. Elemen dasar lainnya dari model RDF adalah *statement*, yaitu sebuah *resource* (subjek) yang terhubung dengan *resource* yang lain atau dengan sebuah literal (objek) melalui *resource* ketiga yang disebut dengan predikat. Sebuah *statement* dapat didefinisikan sebagai

<subjek> memiliki properti <predikat> dengan nilainya adalah <objek>.

5. BAHASA PEMROGRAMAN JAVA

Bahasa pemrograman Java merupakan bahasa dengan konsep berorientasi obyek yang sifatnya *free* dan *open source*. Karena sifatnya tersebut, banyak dukungan terhadap berbagai API (*Application Programming Interface*) untuk berbagai keperluan, misalnya dalam hal ini *library protocol* UPnP.

Mekanisme dasar pemrograman seperti *Threading*, pemrograman jaringan TCP/IP juga telah di-*support* dalam pemrograman Java.

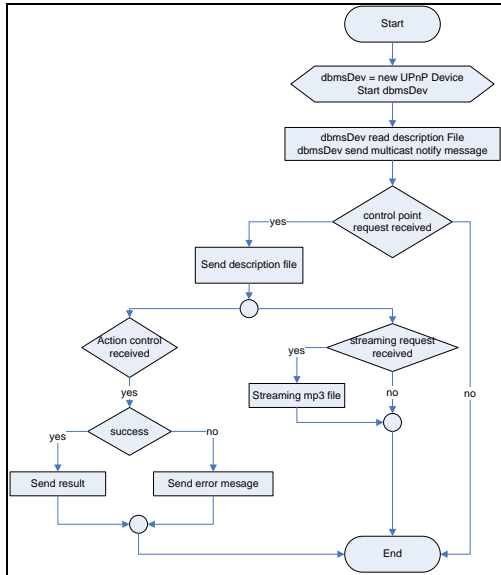
Keunggulan bahasa Java yang membuatnya lebih *powerfull* dikenal dengan istilah “*Write Once Run Anywhere*” atau biasa disebut *multiplatform*. Bahasa Java bisa berjalan pada berbagai *platform* sistem operasi seperti UNIX, Linux, Windows, Macintosh, Symbian dan lainnya tanpa harus mengubah kode program yang telah ditulis. Hal ini dikarenakan bahasa Java berjalan di atas sebuah *Java Virtual Machine*(JVM). [6].

6. DESAIN DAN IMPLEMENTASI SISTEM

Sistem *Share-It* dengan *Personal Preference* terdiri dari dua bagian subsistem yakni *Server Device* dan *Client Device*. Kedua subsistem tersebut mendukung UPnP. Sehingga dalam konteks jaringan UPnP, *Server Device* berperan sebagai UPnP *device* sedangkan *Client Device* selain berperan sebagai UPnP *control point*, juga berperan sebagai UPnP *device*. Implementasi *Personal Preference* dilakukan di dalam subsistem *client device*.

Server Device

Merupakan bagian / subsistem dari sistem *Share-It* yang berperan sebagai UPnP *device*. *Server Device* memiliki kemampuan untuk menyediakan *content file* mp3, dan berlaku sebagai *web server* (membuka dan mendengarkan *port* http untuk melayani http *request*) sehingga dapat menstreaming *file* mp3 ke *control point* yang menghendaknya.



Gambar 2. Flowchart Server Device

Implementasi Server Device terdiri dari class – class sebagai berikut :

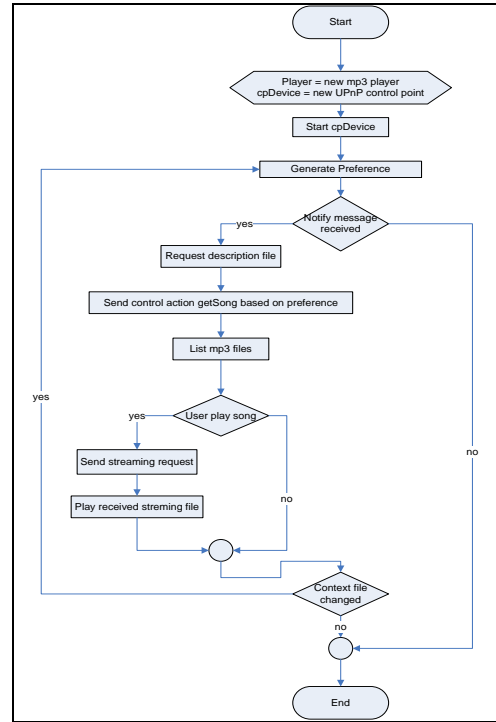
- Class DBMSFrame, Dipergunakan sebagai GUI (Graphical User Interface) atau antarmuka untuk UPnP device.
- Class DBMSDevice, berperan sebagai UPnP device.
- Class viewCollection, tampilan untuk melihat daftar koleksi file mp3.
- Class MySQLDBMS, Digunakan untuk berhubungan dengan MySQL.
- Class ContentInfo, digunakan sebagai node untuk struktur data daftar lagu berbentuk tree.

Client Device

Merupakan bagian / subsistem dari sistem Share-It yang berperan sebagai UPnP device. Server Device memiliki kemampuan untuk menyediakan content file mp3, dan berlaku sebagai web server (membuka dan mendengarkan port http untuk melayani http request) sehingga dapat menstreaming file mp3 ke control point yang menghendaknya.

Implementasi Client Device terdiri dari class – class sebagai berikut :

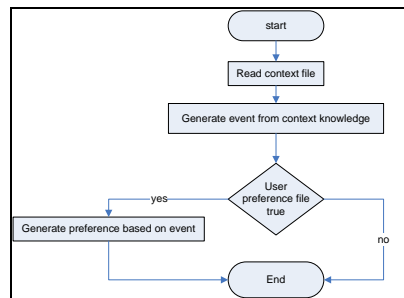
- Class ControlPointInterface, antarmuka untuk UPnP control point..
- Class DCPDevice, berperan sebagai control point pada arsitektur UPnP.
- Class profileChooserFrame, antarmuka untuk menu memilih preference yang akan digunakan.
- Class Preference, class utama untuk mengatur preference dari aplikasi Client Device.



Gambar 3. Flowchart Client Device

Personal Preference

Personal Preference diimplementasikan di dalam Client Device. Sehingga Client Device dapat menampilkan content file mp3 yang tersedia di jaringan UPnP berdasarkan preference dari user. Preference dari user ditentukan oleh kebiasaan (habbit) dari user ketika memainkan lagu pada kondisi-kondisi tertentu (cuaca, lokasi, waktu).



Gambar 4. Flowchart Personal Preference

7. UJI COBA

Uji coba dibagi menjadi dua studi kasus, yang dilakukan dengan tujuan untuk menguji fungsionalitas dan ketegaran Sistem Pendistribusian Alert dengan Personal Preference.

1. Uji Fungsionalitas

Uji coba ini bertujuan untuk mengetahui apakah masing-masing proses pada Sistem Share-It dengan Personal Preference berjalan sesuai dengan deskripsi sistem dan fungsi proses yang telah diuraikan pada Bab III.

Skenario uji coba ini adalah dengan menjalankan Server Device dan Client Device. Client Device akan mendeteksi Server Device dan kemudian

menampilkan daftar *file* mp3 yang dimiliki oleh *Server Device* sesuai dengan *Personal Preference*-nya masing-masing. *Client Device* lainnya akan dijalankan secara berturut-turut kemudian. Setelah dijalankan selama lima menit, salah satu *Client Device* akan dimatikan. Setelah sepuluh menit pertama, *variable* kondisi lingkungan sekitar pada *file context* pada *Client Device 2* akan diubah. Kedua hal ini dilakukan untuk menguji proses perubahan daftar *file* mp3 jika terdapat perubahan *device* dan kondisi lingkungan sekitar.

Uji Ketahanan

Uji coba ini bertujuan untuk mengetahui apakah sistem *Share-It* dengan *Personal Preference* mampu untuk menangani aliran data (*traffic*) dalam jumlah yang besar. *Traffic* yang dimaksud dalam uji coba ini adalah aliran data ketika terjadi proses mp3 *streaming* ataupun ketika terdapat lebih dari satu *Client Device* pada jaringan yang berada pada satu subnet yang sama.

Uji coba ini dilakukan dengan skenario membandingkan *bandwith stream* ketika yang terhubung ke jaringan berturut-turut dua, tiga dan empat buah *Client Device*. Perbandingan *bandwith stream* juga dilakukan ketika terjadi aliran data mp3 *streaming* dengan jumlah *Client Device* yang sama.

Hasil uji coba dapat dilihat pada tabel di bawah

Tabel 1. Rata-rata Bandwith stream ketika Server & Client idle

	Jumlah Client Device		
	1 (satu)	2 (dua)	3 (tiga)
Server Device	0.0138	0.033	0.0385
Client Device	0.0054	0.0648	0.0825

Tabel 2. Rata-rata Bandwith stream ketika streaming mp3

	Jumlah Client Device		
	1 (satu)	2 (dua)	3 (tiga)
Server Device	0.231	0.583	0.6985
Client Device	0.3915	0.4428	0.459

Dari tabel di atas, bisa dilihat bahwa *traffic bandwith* tertinggi terjadi ketika proses *streaming* berlangsung. Namun demikian, pada saat *idle*, di jaringan tetap terdapat *traffic*, hal ini dikarenakan meski tidak ada proses *streaming file* mp3, *client* dan *server* tetap berkomunikasi dengan *client* mengirimkan pesan berulang-ulang selama selang waktu tertentu.

8. SIMPULAN DAN SARAN

Simpulan yang dapat diambil dari selama implementasi hingga uji coba adalah sebagai berikut

- Arsitektur UPnP dapat diterapkan di berbagai macam *platform (multiplatform)*. Penulis

menggunakan windows XP, Windows Server 2003, serta Linux Debian 2.6.8 dalam uji coba serta pengembangan sistem *Share-It* dengan *Personal Preference*.

- *Personal Preference* dengan menggunakan permodelan data RDF telah dapat diterapkan pada aplikasi *Share-It*. Sehingga sistem *Share-It* dengan *Personal Preference* mampu menampilkan *content file* mp3 yang disediakan oleh UPnP *device* pada satu subnet jaringan sesuai dengan *preference* dari *user*.
- Sistem *Share-It* dengan *Personal Preference* belum dapat diterapkan pada kondisi lingkungan yang riil. Mengingat pemantauan kondisi lingkungan sekitar untuk menghasilkan *preference* masih dilakukan melalui simulasi.
- Penggunaan simulasi untuk pemantauan kondisi lingkungan sekitar dapat dikembangkan dengan menggunakan alat atau detektor yang dapat benar-benar memantau kondisi lingkungan sekitar. Sehingga sistem *Share-It* dengan *Personal Preference* dapat diterapkan di dalam kondisi lingkungan yang riil.
- Salah satu komponen dari *Personal Preference* yakni *profile file* yang mana pada Tugas Akhir kali ini merupakan sebuah *file profile* yang sudah jadi, dapat dikembangkan sehingga sistem *Share-It* dapat menghasilkan *file profile* tersebut dengan belajar dari kebiasaan *user* ketika memutar *file* mp3.

Saran yang dapat penulis berikan adalah sebagai berikut

- Penggunaan simulasi untuk pemantauan kondisi lingkungan sekitar dapat dikembangkan dengan menggunakan alat atau detektor yang dapat benar-benar memantau kondisi lingkungan sekitar. Sehingga sistem *Share-It* dengan *Personal Preference* dapat diterapkan di dalam kondisi lingkungan yang riil.
- Salah satu komponen dari *Personal Preference* yakni *profile file* yang mana pada Tugas Akhir kali ini merupakan sebuah *file profile* yang sudah jadi, dapat dikembangkan sehingga sistem *Share-It* dapat menghasilkan *file profile* tersebut dengan belajar dari kebiasaan *user* ketika memutar *file* mp3.

9. DAFTAR PUSTAKA

1. Contributing Members of the UPnP™ Forum, UPnP™ Device Architecture, <http://www.UPnP.org/specs/architecture/>, 8 Juni 2000.
2. Satoshi Kono, Cyberlink for Java Programmer Guide, CyberGarage, 2002.
3. Nial Mansfield, Practical TCP / IP, Andi Offset, 2004.
4. Gabriel Bouvigne, MPEG Audio Layer I/II/III frame header, http://www.mp3-tech.org/programmer/frame_header.html, 1999.

5. Javazoom team, MP3 decoder/player/converter library for Java™ platform, <http://www.javazoom.net/javalayer/javalayer.html>, 2004.
6. Patrick Naughton, Java Handbook, Andi Offset, 2002.
7. Intel® Tools for UPnP Technologies, <http://www.intel.com/technology/UPnP/tech.htm#tools>, 2005.
8. Tim Berners-Lee. 1998. What the Semantic Web can represent. W3C. <http://www.w3.org/DesignIssues/RDFnot.html>.
9. Sergey Melnik. 2001. RDF Resource. <http://www-db.stanford.edu/~melnik/rdf/index.html>.
10. Resource Description Framework. W3C. 2002. <http://www.w3.org/RDF>
11. Resource Description Framework Vocabulary Description Language version 1.0: RDF Schema. W3C. 2002. <http://www.w3.org/TR/rdf-schema>.