Media Informatika, Vol. 2, No. 1, Juni 2004, 55-61
ISSN: 0854-4743

APLIKASI PENCARI IDL DAN OBJEK PADA SISTEM TERDISTRIBUSI BERBASIS CORBA

Edhot Purwoko[1], Wawan Indarto[2]
Jurusan Teknik Informatika, Fakultas Teknologi Industri,Universitas Islam Indonesia
Jl. Kaliurang Km. 14 Yogyakarta 55501
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148
E-mail: [1]sikoes_edot@yahoo.com, [2]wawan@fti.uii.ac.id

ABSTRAK

Common Object Request Broker Architecture (CORBA) didesain agar komponen cerdas diizinkan untuk dapat saling menjelajah satu dengan yang lain dan saling berinteraksi melalui suatu objek bus yang bernama ORB (Object Request Broker). Melalui ORB inilah objek – objek yang terletak di local atau di remote pada sistem terdistribusi berbasis CORBA saling berkomunikasi. Untuk mendefinisikan sistem berbasis CORBA terlebih dahulu kita mendefinisikan IDL (Interface Definition Language). IDL dikembangkan untuk menyediakan fungsi khusus, yaitu agar dapat digunakan untuk mendefinisikan suatu antarmuka (Interface) layanan yang tersedia dalam suatu sistem terdistribusi. Interface Repository (IR) adalah sebuah database yang menyimpan informasi antarmuka objek CORBA, IR menyimpan data deskripsi antarmuka (tipe). Informasi yang berada di Interface Repository sama dengan informasi yang berada di file IDL, tetapi informasi yang berada di IR dipresentasikan agar client lebih mudah menggunakan. Location Service adalah suatu ekstensi untuk spesifikasi CORBA yang menyediakan fasilitas untuk keperluan umum yang mengalokasikan instance objek. Location Service berkomunikasi langsung dengan Smart Agent yang memelihara sebuah katalog. Katalog tersebut mengandung daftar dari instance yang diketahuinya. Interface Repository dan Location Service dapat dimanfaatkan Untuk mempermudah pengembang aplikasi berbasis CORBA. Dengan adanya Interface Repository dan Location Service pengembang aplikasi berbasis CORBA dapat memperoleh informasi mengenai IDL dan Objek yang sudah pernah dibuat oleh pengembang aplikasi berbasis CORBA sebelumnya. Sehingga dengan menggunakan IDL dan Objek yang sudah pernah dibuat tentunya mengurangi kompleksitas dan mahalnya biaya pengembangan aplikasi dengan objek terdistribusi. Karena tujuan tersebut diatas maka dibuatlah sebuah Pencari IDL Dan Objek Pada Sistem Terdistribusi Berbasis CORBA.

Kata kunci: CORBA, Interface Repository, Location Service, Smart Agent, Objek, IDL, pencarian, Interface, ORB.

1. PENDAHULUAN

1.1 Latar Belakang

CORBA (*Common Object Request Broker Architecture*) adalah sebuah middleware yang dihasilkan oleh sebuah konsorsium yang bernama OMG (*Object*

Management Group). CORBA didesain agar komponen cerdas diizinkan untuk dapat saling menjelajah satu dengan yang lain dan saling berinteraksi melalui suatu objek bus yang bernama ORB (*Object Request Broker*). Melalui ORB inilah objek–objek yang terletak di local atau di remote pada sistem terdistribusi berbasis CORBA saling berkomunikasi. Untuk mendefinisikan sistem berbasis CORBA terlebih dahulu kita harus mendefinisikan IDL (*Interface Definition Language*). CORBA dikatakan sebagai sebuah spesifikasi middleware yang ideal yang mendukung suatu sistem terdistribusi yang menerapkan teknologi berbasis objek, selain itu CORBA juga dikatakan sebagai suatu sistem yang terbuka, artinya bahwa teknologi CORBA merupakan sebuah standarisasi yang terbuka bagi siapa saja yang ingin menerapkan. Dengan adanya dua hal tersebut, pengembang dapat membangun sebuah aplikasi yang dapat berkomunikasi walaupun satu dengan yang lainnya menggunakan bahasa pemrograman yang berbeda.

IDL dikembangkan untuk menyediakan fungsi khusus, yaitu agar dapat digunakan untuk mendefinisikan suatu antarmuka (*Interface*) layanan yang tersedia dalam suatu sistem terdistribusi. Untuk menggunakan IDL dan Objek yang terdapat pada sistem berbasis CORBA pengguna harus mengetahui minimal nama interface objek yang akan digunakan. Jika informasi objek telah diketahui sebelumnya maka hal tersebut tidaklah menjadi masalah, tetapi jika pengguna objek tidak mengetahui informasi mengenai objek yang akan digunakan maka hal tersebut akan menyulitkan. Salah satu cara untuk mendapatkan informasi mengenai objek yang akan dipakai adalah dengan menanyakan kepada pemilik objek tersebut, bagaimana jika pemilik objek tersebut berada di tempat yang jauh dan pengguna tidak mengenal siapa pemilik objek tersebut, hal ini tentunya akan menyulitkan pengguna objek dalam mencari informasi objek yang akan digunakannya.

Spesifikasi CORBA menyebutkan suatu cara yang memungkinkan pengguna CORBA untuk bisa mendapatkan informasi mengenai IDL yakni dengan melakukan pengaksesan *Interface Repository*, tetapi ini hanya sekedar spesifikasi saja sedangkan perangkat lunak yang mengimplementasikan hal tersebut belumlah ada.

Dalam pencarian objek yang ada di sistem dapat digunakan *Location Service* yang terdapat pada Visibroker. Dengan melakukan pengaksesan ke *Location Service* diharapkan pengguna dapat mengetahui Objek aktif yang terdapat pada sistem berbasis CORBA. Dengan adanya aplikasi yang mampu menjelajah sistem terdistribusi berbasis CORBA diharapkan pengguna dan pengembang bisa mendapatkan informasi mengenai objek aktif dan IDL yang akan digunakan.

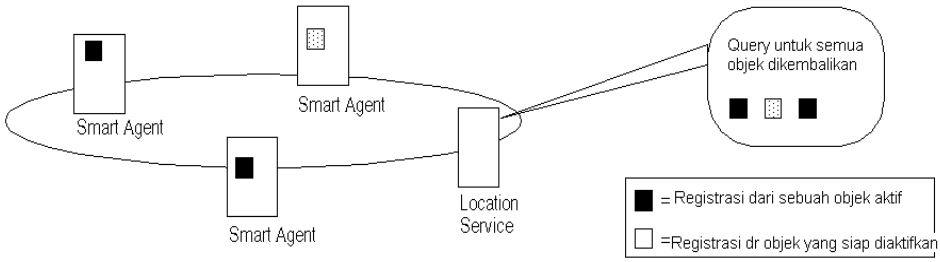
1.2 Tujuan

Menjelaskan mekanisme dari Visibroker Dan CORBA, kemudian menerapkannya ke dalam Aplikasi Pencari IDL dan Objek Pada Sistem Terdistribusi Berbasis CORBA yang memiliki kemampuan untuk menjelajah sistem terdistribusi berbasis CORBA.

2. PEMBANGUNAN SISTEM

2.1 Deskripsi Sistem

Location Service merupakan suatu ekstensi untuk spesifikasi CORBA yang menyediakan fasilitas untuk keperluan umum yang mengalokasikan instance objek. Location Service berkomunikasi langsung dengan *Smart Agent* yang memelihara sebuah catalog. Catalog tersebut mengandung daftar dari instance yang diketahuinya. Ketika query dilakukan oleh *Location Service*, sebuah *Smart Agent* akan meneruskan query tersebut ke *Smart Agent* yang lain dan menjumlah pengembaliannya dalam suatu hasil dan mengembalikan ke *Location Service*. Berikut adalah gambar yang menunjukkan penggunaan *Smart Agent* untuk menemukan instance dari suatu objek.



Gambar 1. Penggunaan Smart Agent dalam menemukan instance dari objek

Seperti terlihat pada Gambar 1, bagian-bagian penting dalam sistem ini adalah:

a. Location Service

Location Service adalah sekumpulan metode yang membuat aplikasi mampu menjelajah objek yang terdapat dalam jaringan dengan menggunakan *Smart Agent* dari Visibroker. Aplikasi dapat melakukan query berdasarkan Interface Repository ID atau kombinasi antara Interface Repository ID dengan nama instance. Hasil dari query dapat dikembalikan sebagai referensi objek yang lain atau deskripsi instance yang lebih lengkap. Sebuah referensi objek biasanya menangani sebuah instance khusus dari sebuah objek yang dialokasikan oleh *Smart Agent*. Deskripsi instance mengandung referensi objek, seperti instance nama interface, nama instance, nama host, nomor port yang digunakan dan informasi mengenai ketetapan.

Location Service mengetahui semua tentang instance objek yang telah didaftarkan ke dalam sebuah POA (Portable Object Adapter) dengan menggunakan kebijakan *BY_INSTANCE* dan objek yang didaftarkan menetap dalam BOA (Basic Object Adapter). Location Service dapat membuat digunakannya informasi smart agent yang menjaga setiap instance objek. untuk tiap objek instance, Location Service merawat informasi terenkapsulasi dalam struktur ObjLocation::Desc seperti dibawah ini.

```
// IDL untuk struktur Desc
Struct {
Object ref;
```

```
:: IIOP:: ProfileBodyValue iiop_locator;
String repository_id;
String instance_name;
Boolean activable;
String agent_hostname;
};
Typedef sequence<Desc> DescSeq;
```

Location Service sangatlah berguna untuk tujuan seperti load balancing dan pengawasan. Dengan menggunakan kemampuan yang ada di *location Service* inilah Aplikasi Pencari IDL dan Objek Pada Sistem Terdistribusi Berbasis CORBA akan melakukan pengaksesan dan mendapatkan informasi mengenai objek aktif yang berada di sistem terdistribusi berbasis CORBA.

b. Interface Repository

Interface Repository (IR) adalah sebuah database yang menyimpan informasi antarmuka objek CORBA, berbeda dengan Visibroker *Location Service* yang menahan data pendeskripsian instance objek, sedangkan IR menyimpan data deskripsi antarmuka (tipe). Informasi yang berada di *Interface Repository* sama dengan informasi yang berada di file IDL, tetapi informasi yang berada di IR dipresentasikan agar client lebih mudah menggunakan.

Client yang menggunakan *Interface Repository* dapat juga memanggil dengan menggunakan *Dynamic Invocation Interface* (DII). Biasanya client mengakses *Interface Repository* untuk mempelajari mengenai objek yang tidak diketahui dan Dynamic Invocation Interface digunakan untuk memanggil metode yang terdapat dalam objek. Bagaimanapun juga antara DII dengan IR tidak terdapat kebutuhan untuk saling berkomunikasi.

c. Smart Agent

Dalam Aplikasi Pencari IDL dan Objek Pada Sistem Terdistribusi Berbasis CORBA dibutuhkan suatu kemampuan yang mampu menjelajah sistem terdistribusi untuk mendapatkan informasi IDL dan objek aktif yang bersifat public (umum), kemampuan ini dimiliki oleh *Smart Agent* milik Visibroker.

Visibroker *Smart Agent* (osagent) adalah dinamis, layanan petunjuk terdistribusi yang menyediakan fasilitas yang digunakan program client dan implementasi objek. Sebuah Smart Agent dapat berjalan minimal pada satu host termasuk dengan jaringan lokal. Ketika program client memanggil bind{} pada suatu objek, Smart Agent secara otomatis akan mencari informasi. *Smart Agent* mengalokasikan implementasi khusus sehingga sebuah koneksi dapat dibentuk antara client dengan implementasi. Komunikasi dengan Smart Agent sangatlah transparan bagi program client.

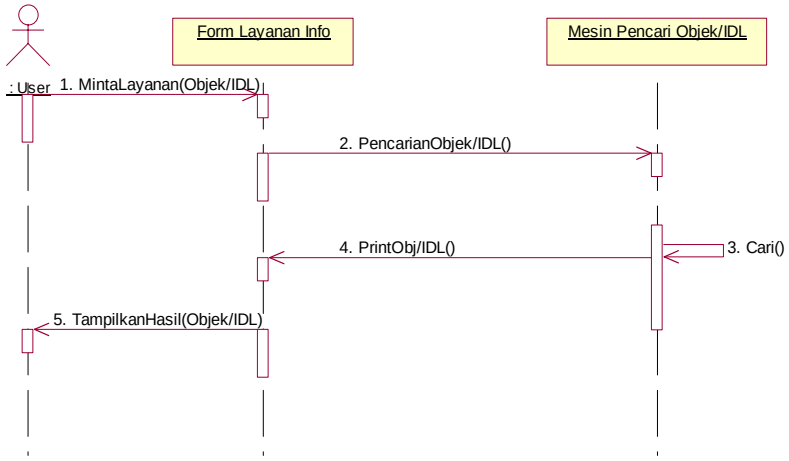
Visibroker mengalokasikan *Smart Agent* agar dapat digunakan oleh program client atau implementasi objek menggunakan sebuah pesan broadcast. Pertama kali *Smart Agent* direspon bahwa sedang digunakan, setelah *Smart Agent* dialokasikan, sebuah koneksi point to point yang menggunakan protokol UDP digunakan untuk mengirim registrasi dan melihat permintaan untuk Smart Agent. Protokol UDP digunakan karena kebanyakan orang menggunakan protokol TCP.

Semua registrasi dan pengalokasian permintaan adalah dinamis sehingga tidak membutuhkan konfigurasi file atau pemetaan untuk perawatan.

Smart Agent dalam jumlah yang banyak dalam jaringan akan bekerjasama untuk menyediakan load balancing dan layanan untuk client agar dapat mengakses objek server. Visibroker dapat menetapkan koneksi jika hubungan antara program client dengan objek server terputus karena suatu masalah yang tidak dikehendaki. Jika kegagalan terdeteksi maka akan dicoba secara otomatis untuk mencari objek server yang berada di host lain.

2.2 Cara Kerja Sistem

Sequence diagram berikut menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari objek-objek yang ada, sebagai gambaran terhadap cara kerja sistem yang dikembangkan secara sederhana seperti ditunjukkan pada Gambar 2.



Gambar 2. Sequence Ddiagram pencarian objek dan IDL

Pencarian ini dibatasi pada penyedia layanan CORBA yang menggunakan Visibroker tetapi tidak menutup kemungkinan objek yang tidak menggunakan visibroker dapat ikut terlacak asalkan *interface* dari objek tersebut telah didaftarkan dahulu.

3. KEBUTUHAN HARDWARE DAN SOFTWARE

3.1 Hardware

Hardware yang digunakan dalam mengembangkan aplikasi ini adalah:

a. Prosesor Intel Pentium 2 Ghz atau yang kompatibel dengannya.
b. RAM minimal 128 MB
c. *Harddisk* dengan kapasitas 2 GB atau lebih
d. VGA dan monitor dengan resolusi minimal 800 x 600 pixel
e. CD ROM *drive*
f. *Mouse*

g. Keyboard
h. LAN Card

3.2 Software

Perangkat lunak yang digunakan dalam pengembangan sistem adalah :

a. Desain Antar Muka :
   1. Microsoft Windows XP
   2. NetBeans IDE 3.6 sebagai *tool* untuk mendesain antarmuka
b. *Scripting* dan *tool* pengembangan
   1. Microsoft Windows XP dan Linux Mandrake 9.2
   2. EditPlus sebagai editor untuk penulisan *script* java
   3. JAVA, sebagai bahasa scripting yang digunakan untuk pembuatan dan pengembangan aplikasi
   4. Visibroker 4.5 *for* Java digunakan sebagai compiler dan interpreter serta berfungsi sebagai penyedia layanan CORBA
   5. J2SDK 1.3 sebagai penyedia Java Virtual Machine yang dibutuhkan untuk menjalankan aplikasi dan *tool* pengembangan dapat dijalankan dengan normal
   6. Rational Rose 2000 sebagai alat bantu pemodelan UML dalam proses analisis dan desain sistem

4. ASPEK PENGEMBANGAN PRODUK

4.1 Umum

Spesifikasi CORBA telah diadopsikan oleh Object Management Group (OMG) dialamatkan untuk kompleksitas dan mahalnya biaya pengembangan aplikasi dengan objek terdistribusi. CORBA menggunakan pendekatan berorientasi objek untuk pembuatan komponen yang dapat digunakan ulang dan dapat berbagi dengan aplikasi yang lain. Masing–masing objek mengkapsulkan detail pekerjaan dan memberikan sebuah definisi interface yang baik, yang mengurangi rumitnya aplikasi. Biaya yang dikeluarkan akan berkurang karena objek yang telah dibuat dan ditest dapat digunakan lagi berkali–kali.

4.2 Proyeksi Pengembangan Sistem

4.2.1 Aspek Perangkat Lunak

Visibroker untuk Java adalah sebuah CORBA 2.0 ORB yang telah lengkap dan mendukung lingkungan pengembangan untuk membangun, menyebarkan, dan mengatur aplikasi dengan objek terdistribusi yang beroperasi lintas platform. Objek yang dibuat menggunakan Visibroker untuk Java akan mudah diakses oleh aplikasi berbasis web yang berkomunikasi menggunakan IIOP (Internet Inter ORB Protocol) yaitu standar komunikasi antara objek yang berjalan di internet, intranet dan dalam lingkungan komputasi perusahaan. Visibroker memiliki sebuah implementasi asli dari IIOP, menjamin performa yang tinggi, interaksi aplikasi terdistribusi.

CORBA dan Java adalah dua infrastruktur yang saling melengkapi satu sama lain dengan baik. CORBA memiliki transparansi jaringan, sedangkan Java

memiliki transparansi pada implementasi dan CORBA mencoba mencari hubungan ketika lingkungan aplikasi Java portable dengan dunia objek intergalactic kehilangan hubungan.

4.2.2 Aspek Konten Sistem

Layanan yang dapat diberikan berupa informasi mengenai deskripsi antarmuka yang terkandung dalam *Interface Definition Language*(IDL) yang telah didaftarkan kedalam *Interface Repository* dan data pendeskripsian instance objek yang telah didaftarkan ke dalam *Location Service*. Lengkap tidaknya informasi yang didapat tergantung dari berapa banyak Agent yang dapat dikenali oleh Agent yang berada di localhost.

Pengembangan kedepan dapat digambarkan sebagai berikut, Aplikasi Pencari IDL Dan Objek Pada Sistem Terdistribusi Berbasis CORBA ini masih dapat dikembangkan dalam pencarian IDL yaitu saat ini pencarian IDL hanya dapat melakukan pencarian pada alamat *Repository ID* "IDL:omg.org/CORBA/Repository:1.0", Pengembang berikutnya dari aplikasi yang sudah ada ini mungkin dapat menambahkan pencarian untuk semua Repository ID yang dapat diketemukan oleh pencarian *Repository ID*.

5. KESIMPULAN

Setelah dilakukan serangkaian pengujian maka dapat disimpulkan bahwa Aplikasi Pencari IDL Dan Objek Pada Sistem Terdistribusi Berbasis CORBA ini sudah dapat berjalan dengan baik dan memberikan manfaat antara lain :

1. Mempermudah pencarian objek dan IDL yang sudah didaftarkan ke *Location Service* dan *Repository ID*.
2. Memudahkan programmer dalam mengembangkan aplikasi berbasis CORBA karena informasi mengenai Objek dan IDL dapat lebih mudah didapatkan.
3. Dengan adanya aplikasi ini diharapkan total waktu yang digunakan untuk mengembangkan aplikasi berbasis CORBA akan terkurangi.
4. Waktu yang digunakan untuk mencari Objek dan IDL secara manual dapat digantikan dengan pengembangan aplikasi sehingga hasil akhir dari aplikasi yang dikerjakan akan lebih maksimal.
5. Kompleksitas dan mahalnya biaya pengembangan aplikasi dengan objek terdistribusi dapat dikurangi.

PUSTAKA

Arnold, K., Gosling, J., dan Holmes, D. (2000) *The JAVA Programming*. New Jersey: Addison Wesley.

Bennet, S., McRobb, S., Farmer, R. (1995). *Object–Oriented System Analysis and Design Using UML*. New York: McGraw-Hill.

Edward, J. (1997). *3-Tier Client/Server at Work* . New York: John Wiley & Sons.

Inprise Corporation (2000). *Visibroker for Java Programmer's Guide*, Inprise Corporation.

Mahmoud, Q. H. (1999).*Distributed Programming with Java*. Manning.

Orfali, R., Harkey, D. (1998). *Client/Server Programming with JAVA and CORBA*. New York: John Wiley & Sons.

Pedrick, D., Weedon, J., Goldberg, J., dan Bleifield, E. (2000). *Programming with VisiBroker*. New York: John Wiley & Sons.