

SISTEM TERDISTRIBUSI DISTRIBUTED OBJECT CORBA

Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

-
- CORBA (Common Object Request Broker Architecture) memungkinkan kita menggunakan aplikasi **tanpa adanya batasan platform**, teknologi jaringan, bahasa pemrograman, maupun letak objek pemberi service yang dituju.



Arsitektur Corba

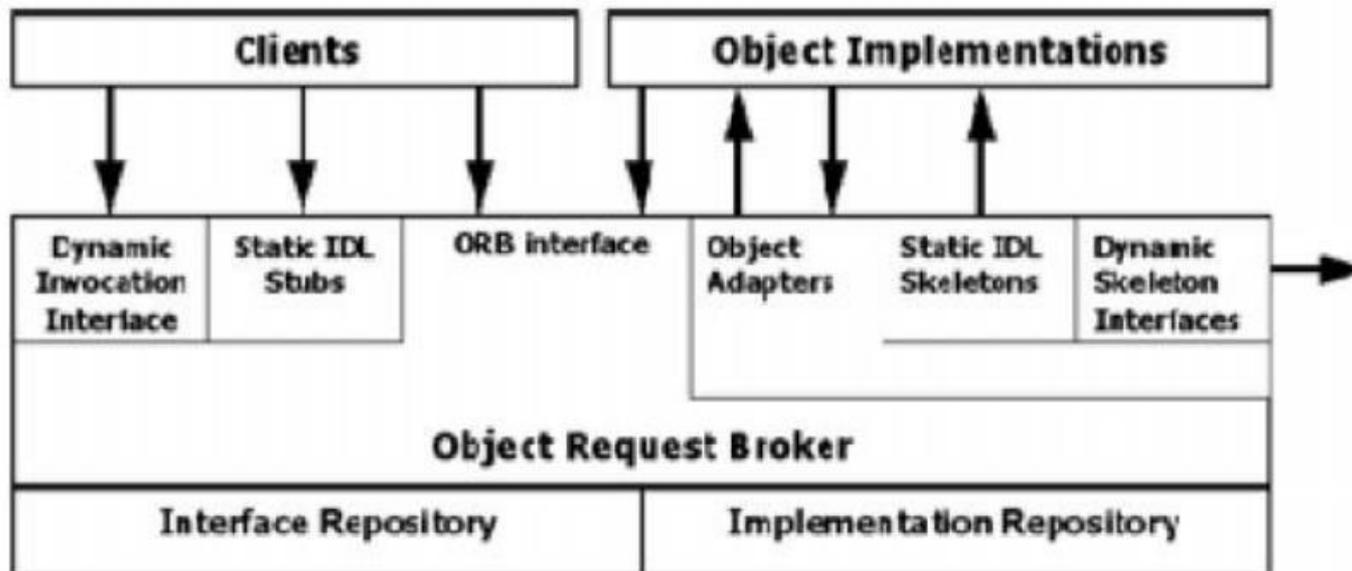
- **CORBA** merupakan suatu spesifikasi yang dikembangkan oleh OMG (Object Management Group), sebuah konsorsium yang terdiri lebih dari 800 perusahaan.
- **Tujuan CORBA** adalah untuk pengembangan pemrograman obyek terdistribusi. CORBA bukanlah Bahasa pemrograman, tapi merupakan spesifikasi untuk mengembangkan objek-objek terdistribusi. Beberapa software yang mengimplementasikan CORBA misalnya ORBIX (oleh IONA Technologies), VisiBroker (oleh Inspire), dan JavaIDL (oleh JavaSoft).
- CORBA memiliki **arsitektur yang berbasiskan model objek**. Model ini diturunkan dari abstrak Core Object Model yang didefinisikan OMG di dalam OMA (Object management Architecture).
- Model ini merupakan **gambaran abstrak** yang tidak dapat diimplementasikan tanpa menggunakan teknologi tertentu. Dengan model tersebut, suatu aplikasi dibangun dengan standard yang telah ditentukan.

Arsitektur Corba

- Sistem CORBA terdiri dari **objek-objek yang mengisolasi suatu client** dari suatu server dengan menggunakan interface enkapsulasi yang didefinisikan secara ketat.
- **Objek-objek CORBA** dapat berjalan diatas berbagai platform, dapat terletak dimana saja dalam suatu network, dan dapat dikodekan dengan bahasa pemrograman apapun asal memiliki IDL mapping.
- **Object Management Architecture (OMA)** mendefinisikan berbagai fasilitas highlevel yang diperlukan untuk komputasi berorientasi objek.
- Bagian utama dari OMA adalah **Object Request Broker (ORB)**.
- ORB merupakan suatu mekanisme yang memberikan transparansi lokasi, komunikasi, dan aktivasi. ORB adalah semacam software bus untuk objek-objek dalam CORBA. Berdasarkan OMA, spesifikasi CORBA harus dipatuhi ORB.

Komponen Utama CORBA

- CORBA disusun oleh **komponen-komponen utama**:
 1. ORB (Object Request Broker)
 2. IDL (Interface Definition Language)
 3. DII (Dynamic Invocation Interface)
 4. IR (Interface Repositories)
 5. OA (Object Adapter)



Komponen CORBA

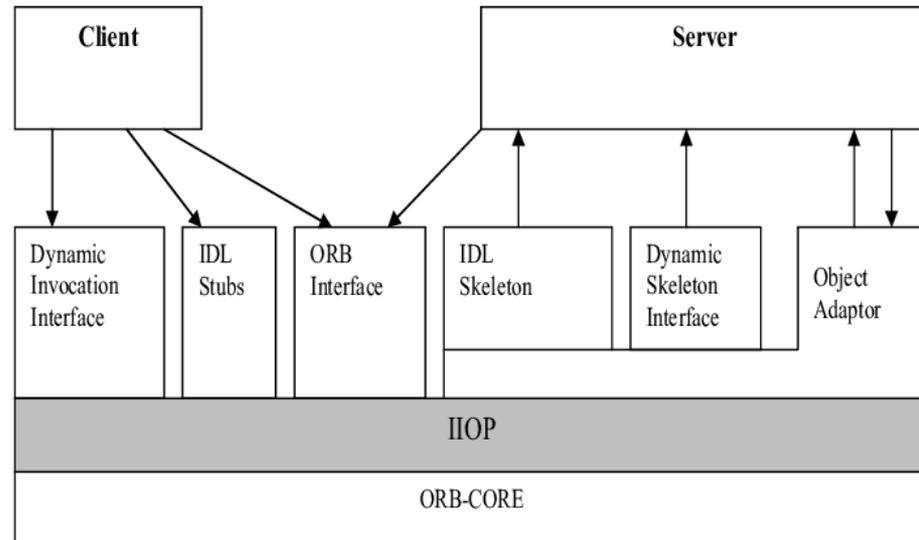
Komponen CORBA pada sisi *Client*:

1. *Client Application*
2. *Client IDL Stubs*
3. *Dynamic Invocation Interface*
4. *Interface Repository*
5. *Client Side ORB Interface*
6. *ORB Core*



Komponen CORBA yang terletak di sisi *Server*

1. *Server Side ORB Interface*
2. *Static IDL Skeleton*
3. *Dynamic Skeleton Interface*



[Internet Inter-ORB Protocol \(IIOP\)](#)

Komponen Utama Penyusun CORBA

1. Object Request Broker (ORB)

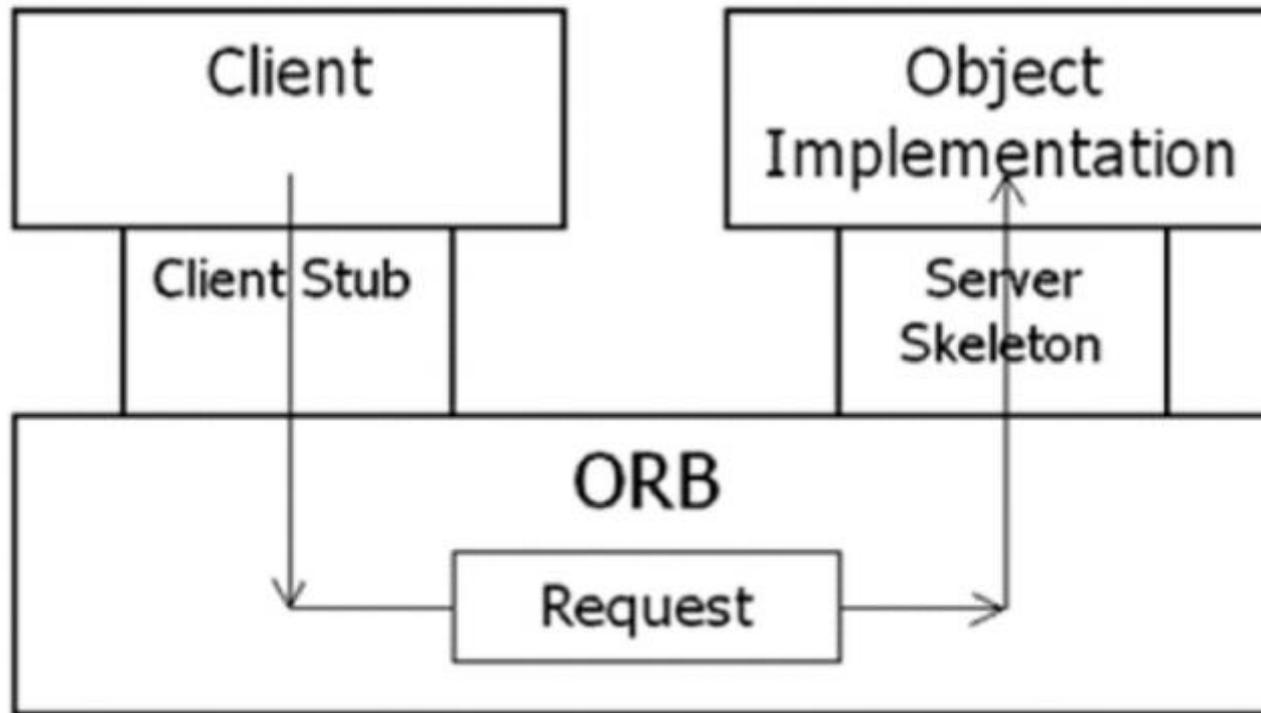
ORB merupakan inti dari CORBA dan bertanggung jawab untuk menjalankan semua mekanisme yang dibutuhkan, yaitu:

- a. Menemukan implementasi objek untuk memenuhi suatu request
- b. Menyiapkan implementasi objek untuk menerima suatu request
- c. Melakukan komunikasi data untuk memenuhi suatu request

Sebuah permintaan (request) yang dikirimkan suatu client ke suatu object implementation akan melewati ORB. Dengan **ORB**, yang terdiri dari **interface**, suatu client dapat berkomunikasi dengan object implementation tanpa adanya batasan platform, teknologi jaringan, bahasa pemrograman, dan letak objek.

Komponen Utama Penyusun CORBA

1. Object Request Broker (ORB)



Gambar Request

Komponen Utama Penyusun CORBA

2.Interface

Interface suatu objek dapat didefinisikan dengan cara statis, yaitu menggunakan IDL (Interface Definition Language). IDL mendefinisikan tipe suatu objek berdasarkan operasi-operasi (yang mungkin dijalankan pada objek tersebut) dan parameter operasi tersebut.

Interface dapat pula ditambahkan ke dalam suatu IRS (Interface Repository Service) yang menggambarkan komponen-komponen dari interface suatu objek. Client dapat mengakses komponen-komponen ini saat runtime.

Komponen Utama Penyusun CORBA

3.Object Reference (OR)

Object Reference (OR) merupakan informasi yang dibutuhkan untuk menentukan sebuah objek dalam ORB. Client dan Object Implementation (OI) memiliki bagian yang tertutup dari OR dengan language mapping, yang kemudian disekat dari representasi aktualnya. Dua implementasi ORB dapat memiliki representasi OR yang berbeda.

Representasi OR pada sisi client hanya valid selama masa hidup client tersebut.

Semua ORB harus menyediakan **language mapping** yang sama untuk sebuah OR (umumnya disebut objek) untuk sebuah bahasa pemrograman tertentu. Hal ini memungkinkan sebuah program ditulis dalam bahasa apapun untuk mengakses OR secara independent terhadap ORB tertentu.

Komponen Utama Penyusun CORBA

4. Interface Definition Language (IDL)

Objek-objek CORBA dispesifikasikan menggunakan **interface**, yang merupakan penghubung antara client dan server.

IDL digunakan untuk mendefinisikan interface tersebut.

IDL menentukan tipe-tipe suatu objek dengan mendefinisikan interface-interface objek tersebut. Sebuah interface terdiri dari kumpulan operasi dan parameter operasi tersebut. **IDL hanya mendeskripsikan interface**, tidak mengimplementasikannya.

Meskipun sintaks yang dimiliki oleh IDL menyerupai sintaks bahasa pemrograman C++ dan Java, namun IDL bukan bahasa pemrograman.

Komponen Utama Penyusun CORBA

5.Object Adapter

Object Adapter (OA) merupakan cara utama bagi sebuah Object Implementation (OI) untuk mengakses service yang disediakan oleh ORB.

Tugas utamanya adalah melakukan masking (menutupi) perbedaan dalam implementasi objek untuk memperoleh portability yang lebih tinggi.

Komponen Utama Penyusun CORBA

6. ORB Interface

ORB Interface merupakan interface yang berhubungan langsung dengan ORB yang sama untuk semua ORB dan tidak tergantung pada interface suatu objek atau Object Adapter (OA).

Karena banyak fungsionalitas ORB yang disediakan melalui OA, stub, skeleton, maupun dynamic invocation; maka ada sedikit operasi yang umum bagi semua objek.

Komponen Utama Penyusun CORBA

7. Interface Repository (IR)

IR merupakan online database yang berisi tentang **meta informasi** tentang tipe dari objek ORB. **Meta informasi** yang disimpan meliputi informasi tentang modul, interface, operasi, atribut, dan eksepsi dari objek.

IR menyediakan cara lain untuk **menentukan interface** ke suatu objek. Interface ini dapat ditambahkan ke layanan IR. Dengan menggunakan **IR**, sebuah client akan **mencari objek yang tidak diketahui pada saat kompilasi**, menemukan informasi tentang interface objek tersebut dan implementasi suatu aktivasi dan deaktivasi.

Komponen Utama Penyusun CORBA

8. Implementation Repository

Implementation Repository terdiri dari informasi yang memperbolehkan ORB untuk mencari dan mengaktivasi implementasi suatu objek.

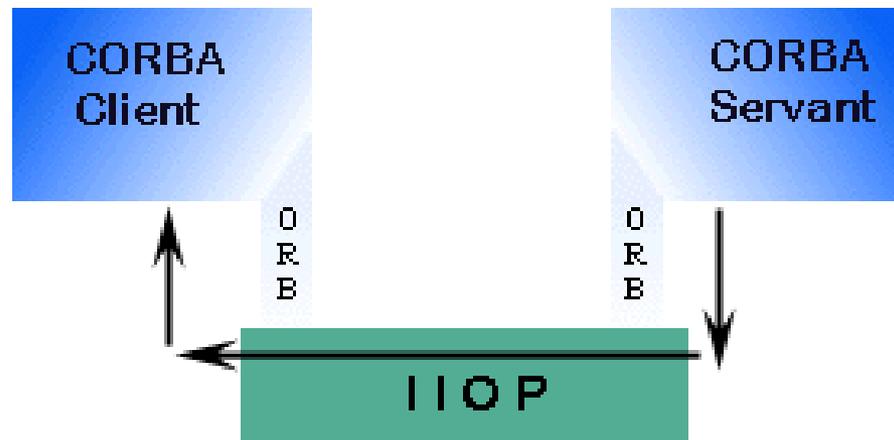
Meskipun untuk suatu ORB atau lingkungan operasi, IR merupakan tempat yang konvensional untuk **menyimpan suatu informasi**.

Komponen Utama Penyusun CORBA

9. Internet Inter-ORB Protocol (IIOP)

CORBA mendefinisikan IIOP untuk mengatur bagaimana objek berkomunikasi melalui jaringan.

IIOP merupakan open protocol yang berjalan diatas TCP/IP.



RMI (Remote Method Invocation)

- Sistem komputasi terdistribusi yang bekerja di banyak tempat mengharuskan beberapa komputer untuk bisa berkomunikasi satu sama lain.
- Untuk komunikasi, Bahasa Java mendukung pemakaian socket yang sifatnya fleksibel dan mencukupi untuk keperluan komunikasi umum. Tapi di sisi lain, untuk membuat socket, client dan server harus terhubung melalui protokol pada application level untuk meng-encode dan men-decode data-data yang akan dikirimkan.

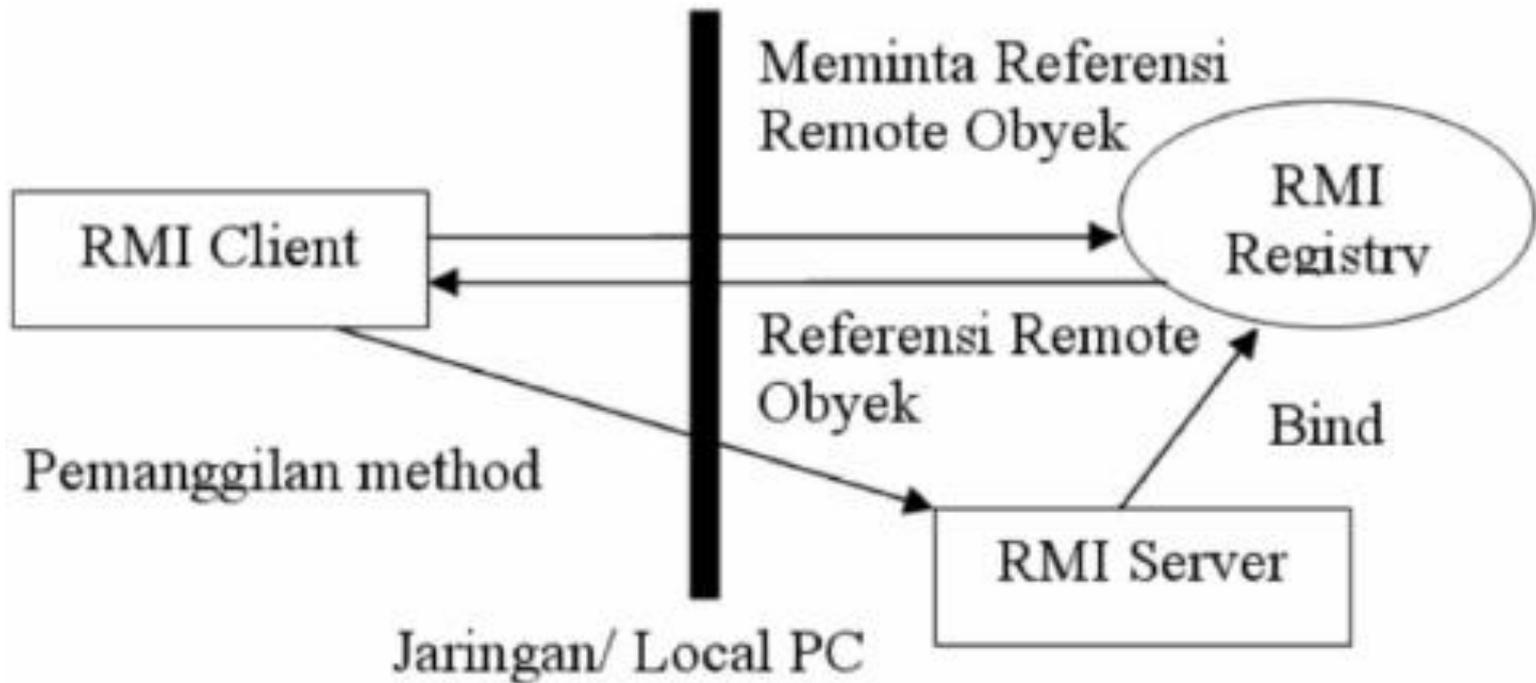
RMI (Remote Method Invocation)

- RMI adalah salah satu bagian dari J2SE yang digunakan untuk membangun aplikasi terdistribusi menggunakan Bahasa Java.
- RMI adalah kumpulan kelas dalam Java yang digunakan untuk menangani pemanggilan (invocation) method secara jarak jauh (remote) dalam suatu jaringan internet. Dimana idenya adalah memisahkan obyek-obyek secara terdistribusi dalam mesin-mesin yang berbeda.
- RMI menggunakan prinsip pemrograman berorientasi obyek dimana obyek satu dapat saling berkomunikasi dengan obyek lainnya dan untuk membangunnya dibutuhkan interface.
- RMI terdiri dari RMI Client dan RMI Server.

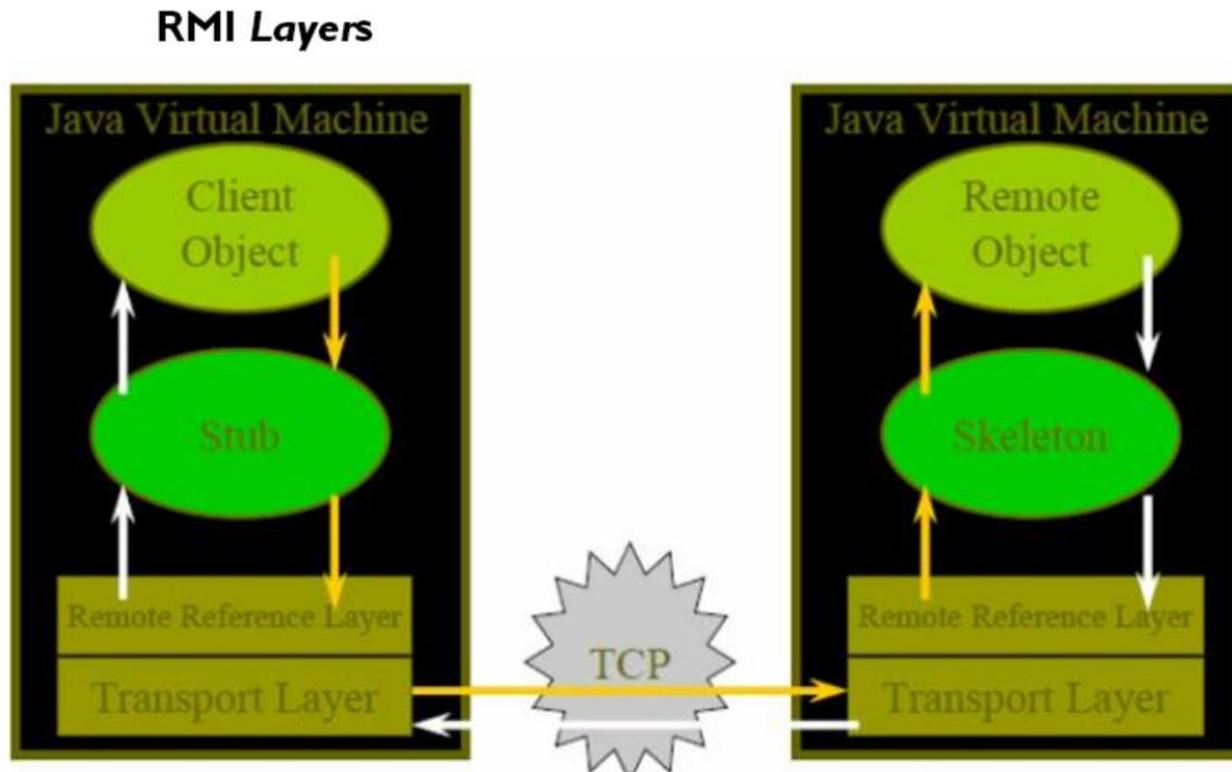
Arsitektur RMI

- RMI Server akan mendaftarkan remote obyeknya ke RMI Registry melalui bind dengan nama unik.
- RMI Client yang akan melakukan suatu pemanggilan method dari remote obyek, harus meminta referensi obyek ke RMI Registry berdasarkan nama kelas obyek tersebut.
- Dalam RMI harus ada pendefinisian interface (behaviour) dan implementasi interface (berupa kelas).
- RMI hanya dimiliki oleh Bahasa Java saja.

Arsitektur RMI



RMI Layer



- Stub dan skeleton merupakan interface antara aplikasi dan RMI System. Stub bertindak sebagai client side proxy, sedangkan skeleton sebagai server side proxy.

Stub dan Skeleton

Fungsi Stub:

- Meminta lokasi remote server obyek pada remote reference layer
- Memberitahu remote reference layer bahwa semua data parameter telah terkirim sehingga pemanggilan method sesungguhnya dapat dilakukan oleh server.
- Memberitahu remote reference layer bahwa pemanggilan telah lengkap

Fungsi skeleton:

- Mengirimkan panggilan method pada server obyek sesungguhnya.

Cara kerja RMI

