

# Pertemuan 12

## Sistem Berbasis Objek Terdistribusi

# Pokok Bahasan

- Paradigma sistem berbasis object terdistribusi
- Arsitektur
- Object terdistribusi
- Organisasi object
- Enterprise java bean
- Object server
- Object adapter

# Paradigma sistem berbasis object terdistribusi

Paradigma pertama terdiri dari objek terdistribusi. Dalam sistem berbasis objek terdistribusi, gagasan tentang objek memainkan peran kunci dalam membangun transparansi distribusi. Pada prinsipnya, segala sesuatu diperlakukan sebagai objek dan klien ditawarkan layanan dan sumber daya dalam bentuk objek yang dapat mereka panggil. Objek terdistribusi membentuk paradigma penting karena relatif mudah menyembunyikan aspek distribusi di balik antarmuka objek. Lebih jauh lagi, karena suatu objek dapat berupa apa saja, itu juga merupakan paradigma yang kuat untuk membangun sistem. Dalam pembahasan ini, akan melihat bagaimana prinsip-prinsip sistem terdistribusi diterapkan ke sejumlah sistem berbasis objek yang terkenal.

# Arsitektur

Orientasi objek membentuk paradigma penting dalam pengembangan perangkat lunak. Sejak diperkenalkan, ia telah menikmati popularitas besar. Popularitas ini berasal dari kemampuan alami untuk membangun perangkat lunak menjadi komponen yang terdefinisi dengan baik dan lebih atau kurang independen. Pengembang dapat berkonsentrasi pada penerapan fungsi spesifik yang independen dari pengembang lain.

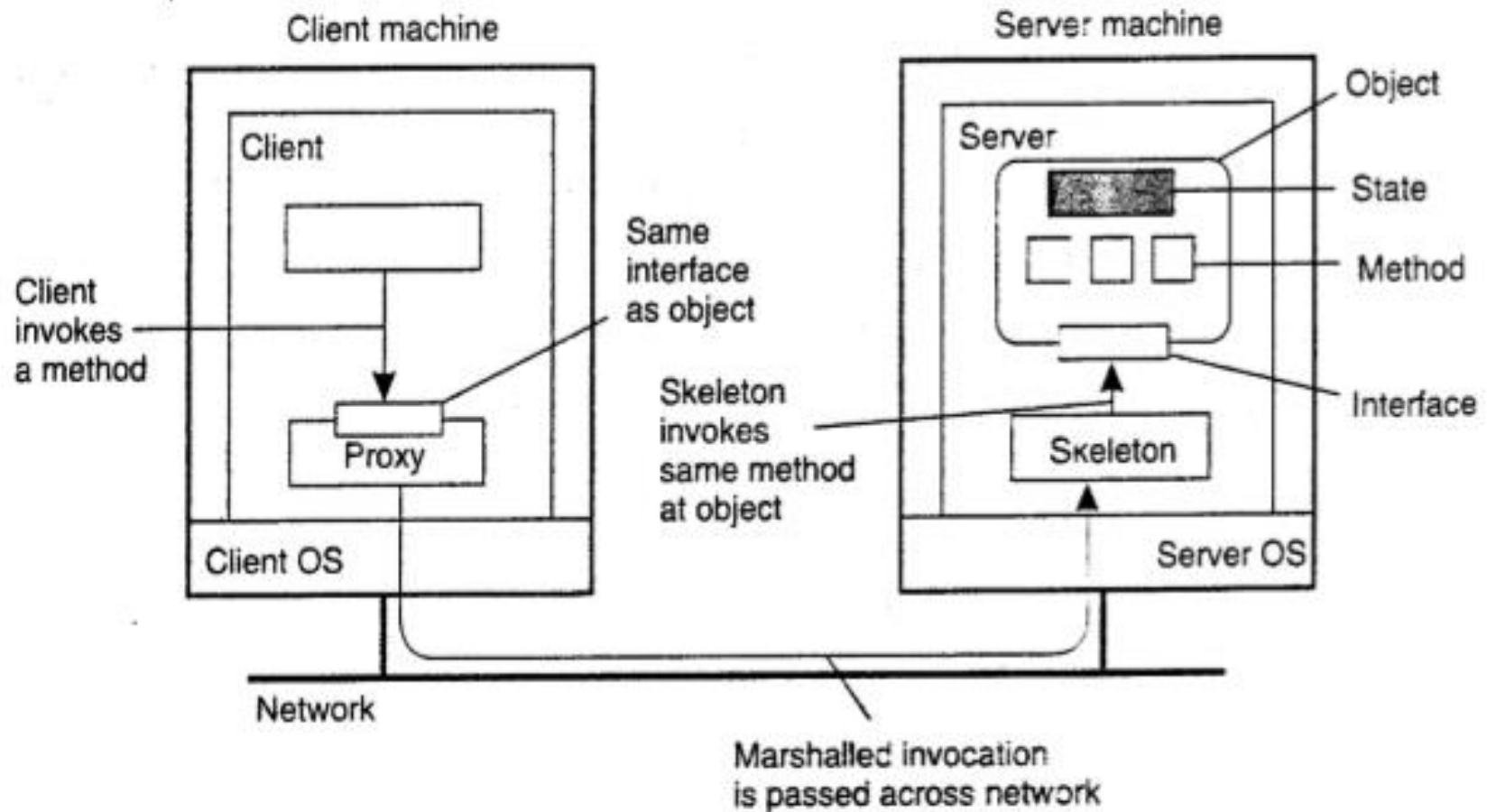
Orientasi objek mulai digunakan untuk mengembangkan sistem terdistribusi pada tahun 1980. Lagi pula, gagasan tentang objek independen yang di-host oleh server jarak jauh sementara mencapai tingkat transparansi distribusi yang tinggi membentuk dasar yang kuat untuk mengembangkan generasi baru dari sistem terdistribusi. Pada bagian ini, pertama-tama akan melihat lebih dalam arsitektur umum sistem terdistribusi berbasis objek, setelah itu dapat dilihat bagaimana prinsip-prinsip spesifik telah digunakan dalam sistem ini.

# Objek Terdistribusi

Fitur utama dari suatu objek adalah bahwa ia merangkum data. disebut negara, dan operasi pada data tersebut, disebut metode. Metode tersedia melalui antarmuka. Penting untuk dipahami bahwa tidak ada cara "legal" suatu proses untuk dapat mengakses atau memanipulasi keadaan suatu objek selain dengan menggunakan metode yang disediakan untuknya melalui antarmuka objek. Objek dapat mengimplementasikan banyak antarmuka. Demikian juga, mengingat definisi antarmuka, mungkin ada beberapa objek yang menawarkan implementasi untuk itu.

Pemisahan antara antarmuka dan objek yang mengimplementasikan antarmuka ini sangat penting untuk sistem terdistribusi. Pemisahan yang ketat memungkinkan untuk menempatkan antarmuka di satu mesin, sedangkan objek itu sendiri berada di komputer lain. Organisasi ini, yang ditunjukkan pada Gambar 1, umumnya disebut sebagai objek terdistribusi.

# Organisasi Objek



Gambar 1. Organisasi umum objek remote dengan proxy sisi klien.



Ketika klien mengikat ke objek yang didistribusikan. implementasi antarmuka objek, yang disebut proxy, kemudian dimuat ke ruang alamat klien. Proxy adalah analog dengan rintisan klien dalam sistem RPC. Satu-satunya hal yang dilakukannya adalah metode marshal ke dalam pesan dan pesan balasan unmarshal untuk mengembalikan hasil metode kepada klien. Objek aktual berada di mesin server, di mana ia menawarkan antarmuka yang sama seperti pada mesin klien.

Permintaan permohonan masuk pertama kali dilewatkan ke sebuah rintisan server, yang membatalkan perintahnya untuk membuat pemanggilan metode pada antarmuka objek di server. Stub server juga bertanggung jawab untuk mengatur balasan dan meneruskan pesan balasan ke proksi sisi klien.

Pada sisi server sering disebut sebagai kerangka karena menyediakan sarana kosong untuk membiarkan server middleware mengakses objek yang ditentukan pengguna. Dalam praktiknya, sering berisi kode tidak lengkap dalam bentuk kelas khusus bahasa yang perlu dikhususkan lebih lanjut oleh pengembang.

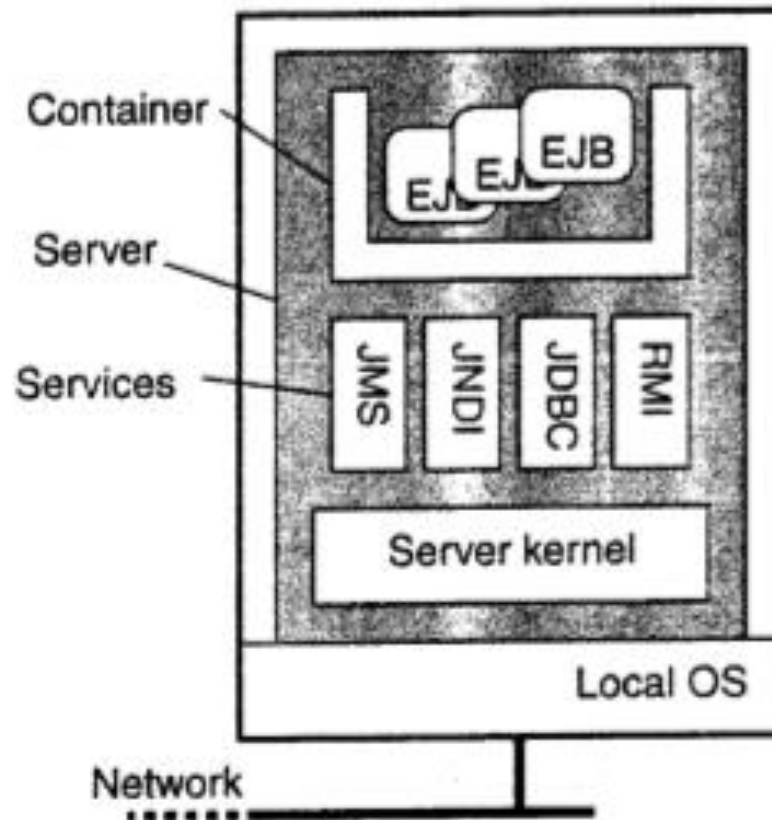
Ciri khas, tetapi agak berlawanan dengan intuisi dari sebagian besar objek yang terdistribusi adalah keadaannya tidak terdistribusi: ia berada pada satu mesin. Hanya antarmuka yang diimplementasikan oleh objek yang tersedia di mesin lain. Objek semacam itu juga disebut sebagai objek jarak jauh. Dalam objek terdistribusi umum, keadaan itu sendiri dapat didistribusikan secara fisik di beberapa mesin, tetapi distribusi ini juga tersembunyi dari klien di belakang antarmuka objek.

# Enterprise Java Beans (EJB)

EJB pada dasarnya adalah objek Java yang di-host oleh server khusus yang menawarkan berbagai cara bagi klien jarak jauh untuk memanggil objek itu. Yang penting adalah server ini menyediakan dukungan untuk memisahkan fungsionalitas aplikasi dari fungsionalitas berorientasi sistem. Yang terakhir mencakup fungsi untuk mencari objek, menyimpan objek, membiarkan objek menjadi bagian dari transaksi, dan sebagainya. Bagaimana pemisahan ini dapat direalisasikan dibahas di bawah ini ketika berkonsentrasi pada server objek.

Dengan pemisahan ini dalam pikiran, EJBs dapat digambarkan seperti yang ditunjukkan pada Gambar 2. Masalah penting adalah bahwa EJB tertanam di dalam wadah yang secara efektif menyediakan antarmuka ke layanan dasar yang diimplementasikan oleh server aplikasi. Wadah tersebut dapat secara otomatis mengikat EJB ke layanan ini, yang berarti bahwa referensi yang tepat sudah tersedia untuk seorang programmer. Layanan umum termasuk layanan untuk pemanggilan metode jarak jauh (RMI), akses basis data (JDBC), penamaan (JNDI), dan pengiriman pesan (JMS).

# EJB Server



Gambar 3. General architecture of an EJB server.

Memanfaatkan layanan ini lebih atau kurang otomatis, tetapi memang mengharuskan programmer membuat perbedaan antara empat jenis EJB:

1. Stateless session beans
2. Stateful session beans
3. Entity beans
4. Message-driven beans

# Object Servers

Server objek adalah server yang dirancang untuk mendukung objek terdistribusi. Perbedaan penting antara server objek umum dan server (lebih tradisional) lainnya adalah bahwa server objek dengan sendirinya tidak menyediakan layanan tertentu. Layanan spesifik diimplementasikan oleh objek yang berada di server. Pada dasarnya, server hanya menyediakan sarana untuk memanggil objek lokal, berdasarkan permintaan dari klien jarak jauh. Sebagai akibatnya, relatif mudah untuk mengubah layanan hanya dengan menambahkan dan menghapus objek.

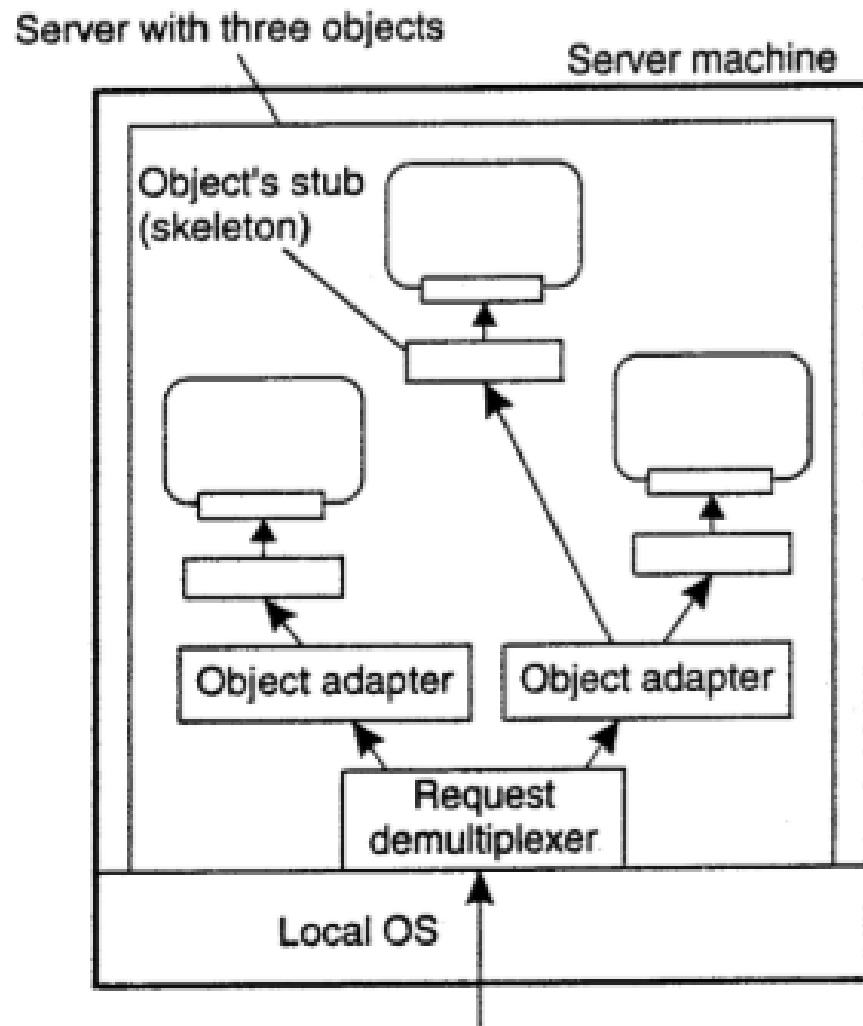


Server objek dengan demikian bertindak sebagai tempat di mana objek tinggal. Objek terdiri dari dua bagian: data yang mewakili statusnya dan kode untuk mengeksekusi metodenya. Apakah bagian-bagian ini dipisahkan atau tidak, atau apakah implementasi metode dibagi oleh beberapa objek, tergantung pada server objek. Juga, ada perbedaan dalam server objek wayan memanggil objeknya.

# Object Adapter

Keputusan tentang cara memohon objek biasanya disebut sebagai kebijakan aktivasi, untuk menekankan bahwa dalam banyak kasus objek itu sendiri harus terlebih dahulu dibawa ke ruang alamat server (mis., Diaktifkan) sebelum benar-benar dapat dipanggil. Yang diperlukan kemudian adalah mekanisme untuk mengelompokkan objek per kebijakan. Mekanisme seperti itu kadang-kadang disebut adaptor objek, atau sebagai alternatif pembungkus objek

Adaptor objek memiliki satu atau lebih objek di bawah kendalinya. Karena server harus mampu mendukung objek secara bersamaan yang memerlukan kebijakan aktivasi yang berbeda, beberapa adapter objek dapat berada di server yang sama secara bersamaan. Ketika permintaan dikirim ke server, permintaan itu pertama kali dikirim ke adaptor objek yang sesuai, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Object Adapter