

Implemented using Web Services

# **SERVICE ORIENTED ARCHITECTURE (SOA)**

*Oleh: Ahmad Syauqi Ahsan*

# TUJUAN

---

- ✘ Mengerti konsep dasar dari Service Oriented Architecture (SOA).
- ✘ Memahami manfaat SOA.
- ✘ Mengerti kapan perlu menggunakan SOA.

# TANTANGAN DI DUNIA IT

---

- ✘ Harus menangani kompleksitas ketika membangun aplikasi baru, mengganti aplikasi yang lama, dan tetap melakukan proses maintenance dan permintaan untuk peningkatan fungsi.
- ✘ Kompleksitas:
  - + Sistem IT yang berbeda (OS, Database Server, Aplikasi).
  - + Organisasi yang terpisah
  - + Kebijakan yang berbeda pada masing2 organisasi

# STUDI KASUS

---

- ✘ Sebuah Pemerintahan kota (pemkot) terdiri dari 4 organisasi besar yang masing2 memiliki sistem IT sendiri:
  - + Dinas Pendapatan Daerah (DB: Oracle, Programming Tool: Oracle Developer, OS: Sun Solaris).
  - + Dinas Pekerjaan Umum (DB: MySQL, Programming Tool : PHP, OS: Linux).
  - + Dinas Kesehatan (DB: SQL Server, Programming Tool: ASP.net, OS: Windows Server 2003)
  - + Dinas Kependudukan (DB: Akses, Programming Tool: VB, OS: Windows XP)

# STUDI KASUS (2)

---

## ✘ Tantangan:

- + Tiap2 organisasi dapat menggunakan data dari organisasi yang lain.
- + Menyediakan layanan kepada publik yang mengintegrasikan ke 4 dinas tersebut → pengguna tidak perlu mengakses sistem dari masing2 dinas.

# SERVICE ORIENTED ARCHITECTURE

- ✘ SOA merupakan suatu paradigma untuk mengelola dan memanfaatkan beberapa informasi (atau kemampuan) yang terdistribusi yang mungkin dikelola dan dimiliki oleh beberapa pihak dengan kebijakan yang berbeda.
- ✘ SOA terdiri dari beberapa services yang saling berkomunikasi satu sama lain.
- ✘ SOA diimplementasikan menggunakan infrastructure, programming model, dan teknologi (web services) yang standart.
- ✘ Prinsip2 dasar SOA: encapsulation, loose coupling, contract, abstraction, reusability, composability, autonomy, discoverability

# SOA ITU...

---

- ✘ Bukan solusi untuk segalanya
- ✘ Hanyalah sebuah “alat” diantara kumpulan “alat” yg lain.
- ✘ Bukan suatu teknologi khusus
- ✘ Bukan nama baru untuk EAI (Enterprise Application Integration)
- ✘ Bukan metode baru untuk melakukan RPC (Remote Procedure Call).

# APA ITU SOA ?

---

## ✘ Wilayah tujuan:

- + Sistem berskala besar (enterprise).
- + Layanan yang mencakup skala internet.
- + Mengurangi biaya dalam berbisnis.

## ✘ Manfaat:

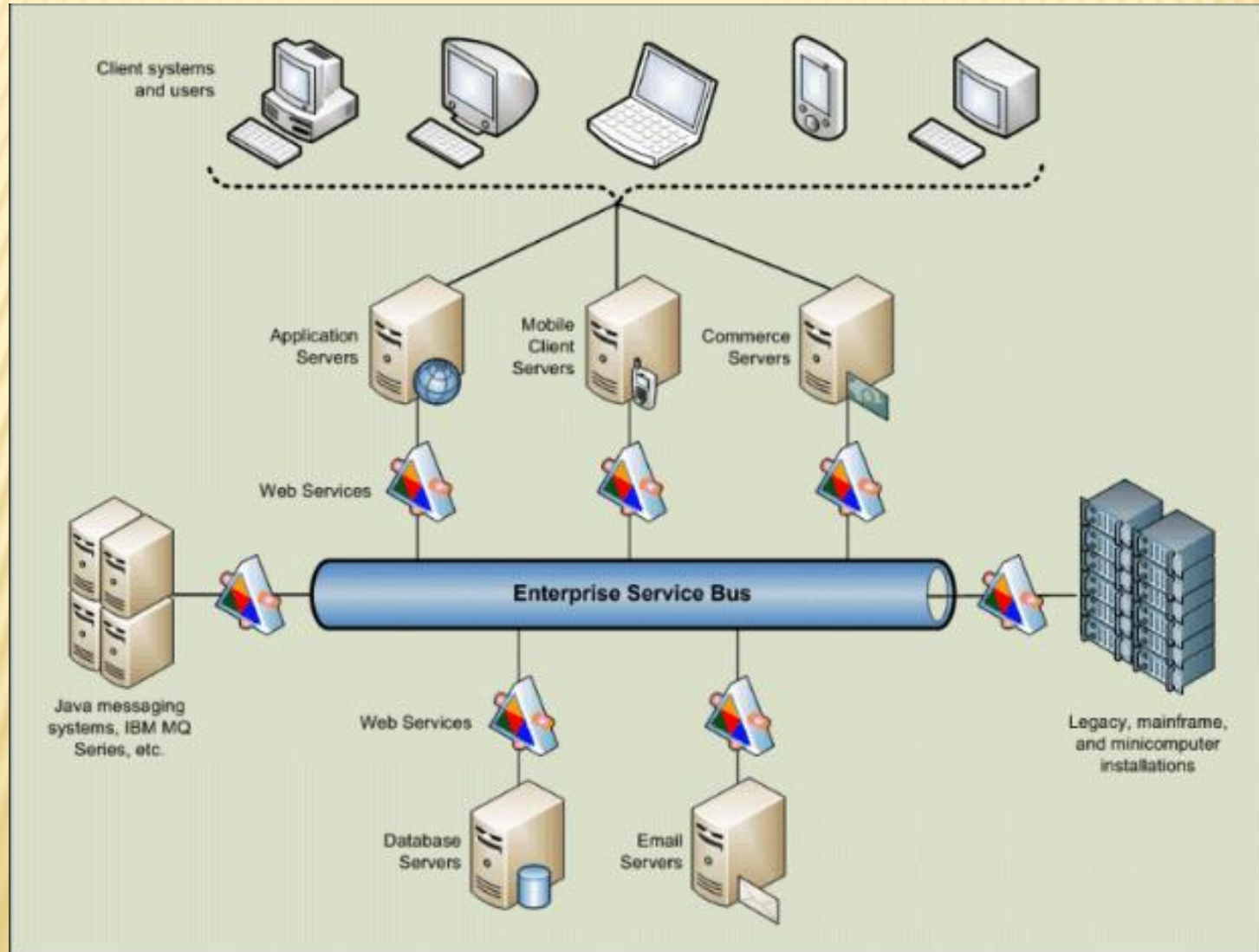
- + Membangun sistem yang “scalable” (dapat dikembangkan).
- + Mengelola sistem yang kompleks.
- + Memungkinkan untuk menggunakan fungsi2 yang sudah ada (tidak perlu membuat fungsi2 baru).



# KENAPA SOA BERBEDA ?

- ✘ SOA merefleksikan batas2 kepemilikan.
  - + CORBA, RMI, COM, DCOM, dll. Semua berusaha mengimplementasikan sistem terdistribusi yang transparan.
  - + **Kepemilikan** adalah esensi dari SOA.
- ✘ SOA berorientasi pada tugas (task oriented)
  - + Service diejawantahkan dalam bentuk fungsi-fungsi.
- ✘ SOA terinspirasi oleh bentuk **organisasi** di dunia nyata.
  - + Organisasi berfungsi bagi kita (manusia), maka harus berfungsi juga pada mesin.

# CONTOH DIAGRAM SOA



# APA ITU SERVICES ?

---

- ✘ Adalah mekanisme untuk mengakses satu atau beberapa fungsi.
- ✘ Service merupakan komponen yang dapat digunakan sebagai “bahan” untuk membentuk suatu aplikasi yang mempunyai fungsi lebih kompleks.
- ✘ Service bisa sederhana “berikan alamat seseorang” atau serumit “lakukan proses pembayaran tagihan”.
- ✘ Service merupakan satu fungsi yang berbeda satu sama lain yang berbasiskan pada data.
- ✘ Bagaimana suatu service diimplementasikan dan bagaimana pengguna dapat mengakses service, dibatasi oleh infrastructure dari SOA.

# KARAKTERISTIK DARI SERVICE

- ✘ Seharusnya berbasiskan pada open standart sehingga memudahkan dalam pengintegrasian.
- ✘ Loose Coupling: pengguna service hanya perlu menyediakan data yang telah didefinisikan dan akan mendapatkan hasil yang telah didefinisikan pula. Service yang akan melakukan semua prosesnya.

# KARAKTERISTIK DARI SERVICE (2)

- ✘ Location agnostic: Pengguna dari service tidak perlu mengetahui bagaimana suatu service diimplementasikan. Infrastructure dari SOA akan memberikan mekanisme pengaksesan dengan Service-Level Agreement (SLA).
- ✘ Stateless: suatu service tidak mempunyai tahapan. Service menerima parameter yang diberikan pengguna, melakukan fungsi2 yang telah didefinisikan, dan memberikan hasil yang diharapkan, dalam satu kesatuan proses. Jika suatu service melibatkan suatu transaksi, maka transaksi tersebut akan langsung di-commit dan perubahan data disimpan dalam database.

# SOA TIMELINE

---

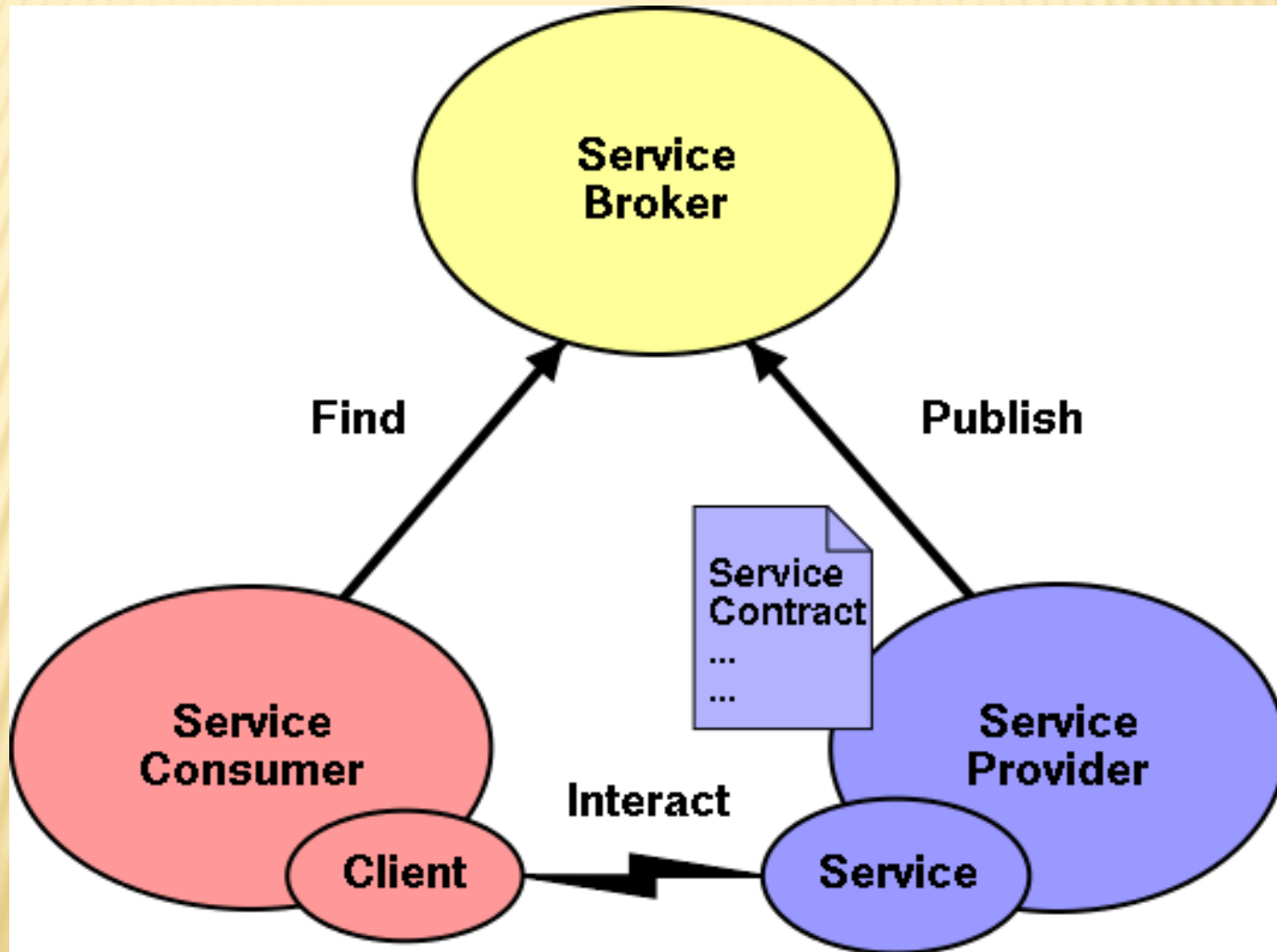
- ✘ Extensible Markup Language (XML)
  - + Dibuat menggunakan Standard Generalized Markup Language pada 60an.
- ✘ Web Services
  - + Spesifikasi dari Simple Object Access Protocol (SOAP) pada 2000 mentrigger penggunaan Web Services.
- ✘ Awal SOA
  - + Model awal dari SOA terdiri dari 3 komponen: service requester, service provider and service registry.
  - + WSDL - Web Services Definition Language
  - + UDDI - Universal Description, Discovery, and Integration
- ✘ SOA
  - + Generasi kedua dari SOA: pengembangan beberapa fungsi.

# SOA DAN WEB SERVICES

---

- ✘ SOA merupakan suatu arsitektur (bukan teknologi).
- ✘ Dalam implementasinya, SOA menggunakan teknologi Web Services.
- ✘ Web Services:
  - + Teknologi standart untuk bertukar pesan (informasi) melalui/menggunakan jaringan web.
  - + Tidak mempunyai ketergantungan (independent) terhadap OS maupun aplikasi.
  - + Dapat ditingkatkan untuk memenuhi level “enterprise” dalam hal keamanan, reliability, dan transaksi.

# WEB SERVICES SIMPLE ARCHITECTURE





# WEB SERVICES DAN XML

- ✘ Web Services adalah teknologi yang berbasis XML.
- ✘ XML (eXtensible Markup Language):
  - + Mempunyai tipe data dan struktur yang standart, tidak tergantung dengan bahasa pemrograman, sistem operasi, maupun sistem software.
  - + Teknologi yang dapat digunakan untuk pembuatan dan pertukaran dokumen2 bisnis, juga termasuk bahasa standart untuk berbagai macam industri.
  - + Terdapat banyak sekali software yang dapat mengelola XML. Termasuk software untuk melakukan proses parser, query, dan transformasi pada dokumen XML.

# CONTOH XML SEDERHANA

```
<?xml version="1.0"?>
<note>
  <to>Giono</to>
  <from>Joni</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

# CORBA VS WEB SERVICES UNTUK SOA

- ✘ CORBA (Common Object Request Broker Architecture) merupakan suatu standart yang didefinisikan oleh OMG (Object Management Group) yang memungkinkan suatu komponen dari satu aplikasi ditulis dalam berbagai bahasa pemrograman dan dijalankan pada beberapa komputer yang saling bekerja-sama.
- ✘ Beberapa orang yang familiar dengan CORBA menganggap bahwa Web Services adalah CORBA yang menggunakan XML.
- ✘ Beberapa pengaplikasian CORBA merupakan SOA.
- ✘ Kesalahan dari CORBA adalah tidak mendefinisikan standart untuk interoperability. Web Services menggunakan SOAP untuk standarisasi.

# CORBA VS WEB SERVICES UNTUK SOA (2)

- ✘ Ditinjau dari perspektif teknis, kita dapat menggunakan CORBA untuk semua hal yang bisa menggunakan Web Services. Dan untuk beberapa aplikasi, CORBA masih merupakan pilihan yang lebih bagus.
- ✘ Akan tetapi, ditinjau dari perspektif manusia, jika seseorang tidak familiar dengan CORBA atau dia baru mengenal distributed computing, Web Services jauh lebih mudah untuk dipelajari dan digunakan. Dan, fitur2 yang tidak terdapat pada Web Services tidaklah terlalu penting jika dibandingkan dengan interoperability yang didapat.

# KEAMANAN PADA WEB SERVICES

- ✘ Pengamanan pada teknologi Web Services dapat diimplementasikan pada berbagai level:
  - + Transport level: dapat menggunakan Firewall, Virtual Private Network (VPN), basic authentication, atau encryption.
  - + Message-level: dapat menggunakan “authentication token” untuk melakukan validasi identitas penerima pesan.
  - + Data-level: dapat menggunakan encryption atau digital signature untuk menjaga agar data tidak dapat dibuka.

# Q & A

---