



**PROGRAM STUDI  
TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH  
*REKAYASA PERANGKAT LUNAK LANJUT*

# Rekayasa Perangkat Lunak Lanjut

*Pengenalan Web App + Req. Web App*



**Disusun Oleh:**

***Tim Pengampu RPLL***

# Silabus Mata Kuliah

1. Pendahuluan
2. *Overview: Analisis Terstruktur*
3. Overview: Perancangan Terstruktur – Arsitektur, Interface, Data
4. Analisis Berorientasi Objek
5. Perancangan Berorientasi Objek
- 6. Pengenalan Web App. + Requirement Web App.**
7. Konsep Web Engineering

# Aplikasi Web

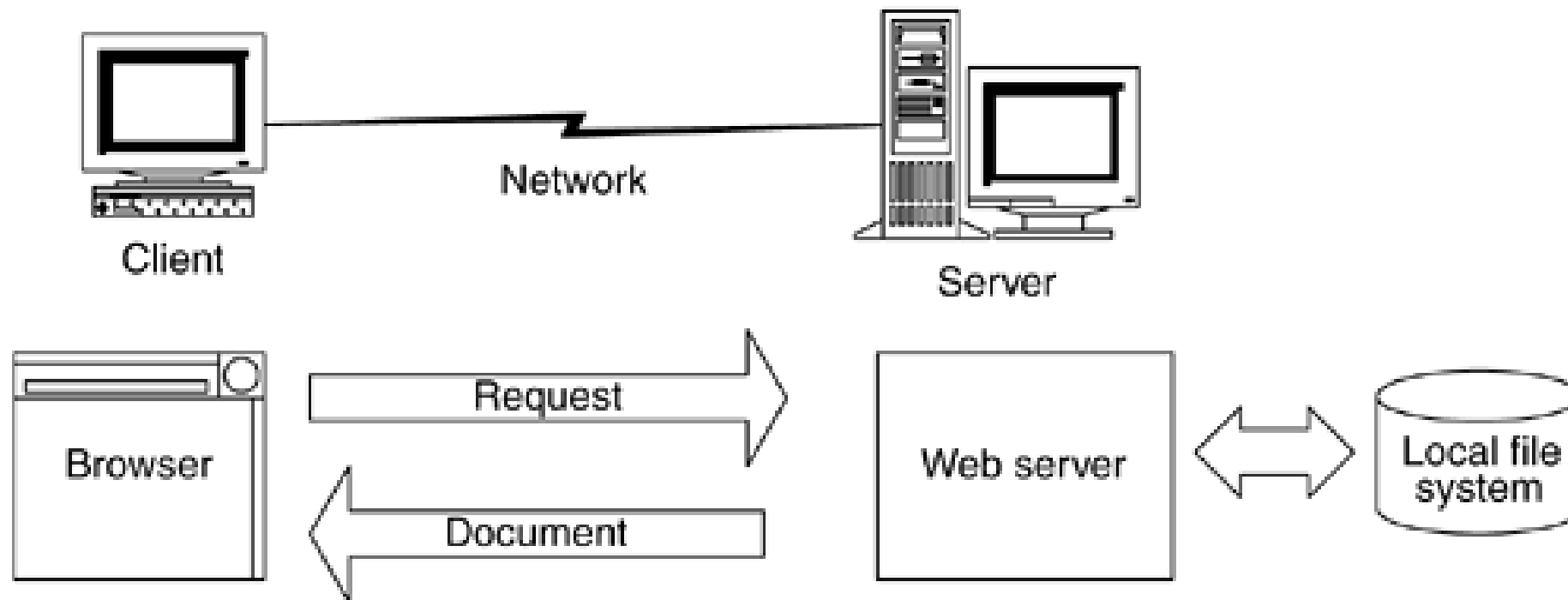


# Aplikasi Web

**Aplikasi Web** adalah sistem Web yang memungkinkan pengguna untuk mengeksekusi logika bisnis dengan Web browser.

**Aplikasi Web** menggunakan teknologi yang memungkinkan untuk membuat konten mereka dinamis dan untuk memungkinkan pengguna sistem untuk mempengaruhi logika bisnis pada server.

# Dasar Sistem Web



# Arsitektur aplikasi Web

- ✓ Ini berisi komponen utama yang sama dari sebuah situs Web:
  - Web server,
  - koneksi jaringan, dan
  - browser klien.
- ✓ Aplikasi Web juga mencakup server aplikasi.
- ✓ Penambahan server aplikasi memungkinkan sistem untuk mengelola logika bisnis.

# Client State Management

- ✓ Salah satu tantangan umum aplikasi Web adalah mengelola klien pada server.
  - Karena sifat connectionless klien dan server komunikasi, server tidak memiliki cara mudah untuk melacak setiap permintaan klien dan untuk mengasosiasikannya dengan permintaan sebelumnya, karena setiap permintaan halaman Web kemudian menetapkan dan istirahat lalu koneksi lagi.
- ✓ W3C telah mengusulkan mekanisme pengelolaan state HTTP, lebih dikenal sebagai "cookies".

# Cookies

**Cookies** adalah informasi yang berupa file kecil yang ditanam pada KOMPUTER CLIENT. Cookies biasanya digunakan oleh sebuah website untuk mengidentifikasi data pengguna. Data yang disimpan pada Cookies dikirim bersamaan dengan request yang dilakukan oleh browser terhadap suatu halaman website.



# Cookies

- ✓ Cookie adalah serangkaian teks yang disimpan pada komputer Anda oleh situs web yang Anda kunjungi. Pada umumnya cookie menyimpan pengaturan atau preferensi Anda untuk suatu situs web tertentu, misalnya bahasa yang dipilih, atau lokasi (negara) Anda.
- ✓ Biasanya, ukuran data yang kecil, antara 100 dan 1K byte; Namun, batas resmi sekitar 4K.

# Sessions

- ✓ **Session** adalah cara untuk menjaga suatu variabel tetap ada selama sesi kunjungan user.
- ✓ Session memungkinkan pelacakan akses pemakai, pengaturan pemakaian aplikasi oleh pemakai dan meningkatkan layanan situs web.
- ✓ Setiap pengunjung akan diberi sebuah id yang unik, yang disebut dengan id session (**session\_id**). ID ini dapat disimpan dalam suatu cookie pada sisi user atau disertakan dalam URL.

## Enabling Technologies

- ✓ *Enabling technologies* adalah mekanisme halaman web menjadi dinamis dan menanggapi input pengguna.
- ✓ Hal yang terlibat adalah eksekusi modul yang terpisah oleh server web.
  - Browser akan meminta modul (Server web diartikan sebagai permintaan untuk memuat dan menjalankan modul).
  - Output modul biasanya halaman HTML yang diformat dengan benar bisa berupa gambar, audio, video maupun data lainnya.

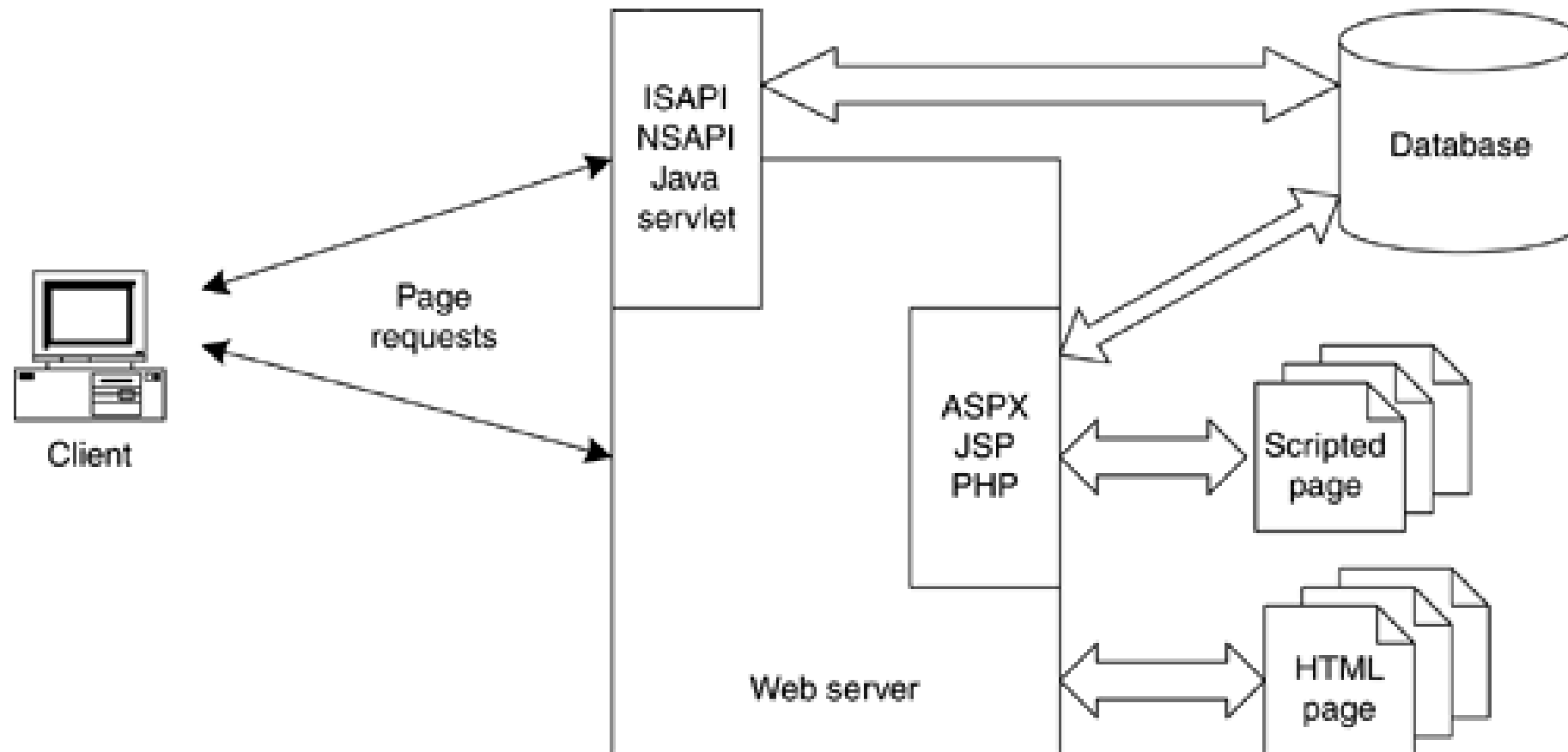
## Enabling Technologies

- ✓ Mekanisme asli untuk memproses input pengguna dalam sistem Web adalah *Common Gateway Interface* (CGI),
  - cara standar untuk memungkinkan pengguna Web untuk menjalankan aplikasi di server.
  - dapat ditulis dalam bahasa apapun
  - bahasa yang paling umum untuk modul CGI skala kecil adalah Perl (ekstraksi dan pelaporan bahasa praktis), yang ditafsirkan setiap kali dijalankan.

## Enabling Technologies

- ✓ Dua **masalah** terbesar dengan CGI:
  - tidak secara otomatis menyediakan jasa manajemen sesi
  - setiap pelaksanaan modul CGI membutuhkan proses baru dan terpisah pada aplikasi / server Web
- ✓ **Solusi** yang tersedia:
  - dengan menambahkan plug-in ke server Web.
  - plug-in memungkinkan server Web untuk berkonsentrasi pada melayani permintaan HTTP standar dan menunda halaman dieksekusi untuk proses lain yang berjalan

# Teknologi Web Server



## Dynamic Client

- ✓ Dalam web dinamis, interaksi yang terjadi antara pengguna dan server sangat kompleks.
- ✓ Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan browser.
- ✓ *Request* (permintaan) dari pengguna dapat diproses oleh server yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya.
- ✓ Halaman-halaman web tersebut memiliki database.
- ✓ Web dinamis, memiliki data dan informasi yang berbeda-beda tergantung input apa yang disampaikan *client*.
- ✓ Dokumen yang sampai di *client* akan berbeda dengan dokumen yang ada di web server.

# Dynamic Client Technology: Scripting

- ✓ **JavaScript** adalah teknologi scripting yang paling banyak digunakan dalam browser
- ✓ Salah satu bagian dari revolusi teknologi Java; sebuah implementasi dari bahasa scripting yang berakar pada bahasa pemrograman Java
- ✓ JavaScript tertanam pada halaman HTML, jadi berdampingan dengan dokumen
- ✓ The `<script>` and `</script>` tags, digunakan untuk menentukan daerah JavaScript.
- ✓ Contoh:  

```
<SCRIPT LANGUAGE="JavaScript"> alert('Hello World.') </SCRIPT>
```
- ✓ Tidak semua browser kompatibel dengan javascript



# Dynamic Client Technology: Java Applet

- ✓ Penggunaan Java pada klien biasanya dalam bentuk applet
- ✓ Applet adalah:
  - Kontrol antarmuka pengguna yang ditempatkan di halaman Web
  - Terdiri dari kedua sistem dan kelas kustom
- ✓ Applet direferensikan oleh halaman Web
  - perlu diidentifikasi oleh tag: `<object>` untuk mengidentifikasi:
    - jenis objek (Java),
    - nama file kelas untuk memuat dan menjalankan, dan,
    - opsional, lokasi, atau URL, di mana file atau file dapat ditemukan pada jaringan.
- ✓ Contoh:
  - `<OBJECTcodetype="application/java"classid="java:Bubbles.class"codebase="http://www.wae-uml.org/javaclasses/">`  
Java applet that draws animated bubbles. `</OBJECT>`

## Dynamic Client Technology: ActiveX/ COM

- ✓ ActiveX digunakan untuk memperluas fungsi client-side dalam banyak cara yang sama seperti dilakukan Java applet
- ✓ ActiveX dibangun oleh Microsoft Component Object Model (COM)
- ✓ COM adalah infrastruktur dimana pengembang dapat membangun komponen, masing-masing dalam bahasa pilihan nya, dan berbagi benda-benda ini untuk membangun yang lebih besar, sistem yang lebih kompleks
- ✓ Menginstal sebuah objek ActiveX biasanya berarti menyalin DLL (dynamic link library), EXE, atau file yang OCX ke hard drive lokal dan menambahkan beberapa entri ke registry
- ✓ Karena objek ActiveX sering memiliki dependensi pada objek ActiveX lain atau DLL, ini juga perlu diinstal pada computer
- ✓ Kontrol ActiveX ditempatkan di halaman HTML dengan tag <object>.

## Distributed Object

- ✓ Menggunakan distributed objects dalam aplikasi Web dapat memecahkan banyak masalah fungsi dan kinerja dalam pengembangan aplikasi Web
- ✓ Kuncinya adalah untuk menggabungkan sistem objek terdistribusi tanpa kehilangan manfaat utama dari arsitektur Web
- ✓ Manfaat yang paling penting dari Web adalah kemudahan deployment.
- ✓ Terdapat dua pola infrastruktur yakni **Java's RMI** dan **Microsoft's DCOM**.
- ✓ Tujuan dari keduanya adalah untuk menyembunyikan rincian komunikasi yang didistribusikan dan tanggung jawab infrastruktur
- ✓ Keduanya bekerja pada prinsip transparansi lokasi, yang menyatakan bahwa objek desainer / pelaksana tidak harus perlu mengetahui lokasi dari contoh objek tertentu

# XML

- ✓ **Extensible Markup Language** (XML) adalah sebuah markup language yang mempunyai himpunan aturan untuk pengkodean dokumen dalam format yang dimengerti oleh manusia dan mesin.
- ✓ Tujuan perancangan dari XML menekankan pada kesederhanaan, keumuman, dan kegunaan melalui Internet.
- ✓ **XML** adalah format data tekstual dengan dukungan yang kuat melalui Unicode untuk bahasa-bahasa di seluruh dunia.
- ✓ Meskipun desain dari XML memfokuskan pada dokumen, yang digunakan secara luas untuk representasi struktur data yang bebas, sebagai contoh pada web services.

The logo consists of the letters 'XML' in a bold, yellow, sans-serif font, enclosed within a thin brown rectangular border.

## EXAMPLE

<address>

    <street>123                    Pine                    Rd.</street>

    <city>Lexington</city>

    <state>SC</state>

    <zip>19072</zip>

</address>

# Web Services

- ✓ **Web service** adalah sebuah service yang tersedia dalam Internet yang menggunakan sistem pesan XML terstandarisasi.
- ✓ Web service tidak terikat pada sistem operasi maupun bahasa pemrograman. Ada beberapa alternatif dalam pertukaran pesan XML.
- ✓ Contohnya, XML Remote Procedure Calls (XML-RPC) atau SOAP dapat digunakan dalam pertukaran pesan.
- ✓ Alternatif lain adalah dengan hanya menggunakan HTTP GET/POST untuk mengirimkan pesan XML.
- ✓ Standard Web Service:
  - SOAP
  - UDDI
  - WSDL

# SOAP

- ✓ SOAP: Simple Object Access Protocol
- ✓ Dibangun di atas XML
  - pesan SOAP hanyalah sebuah dokumen berformat XML yang memvalidasi terhadap DTD atau skema tertentu.
- ✓ Ketika digunakan dengan HTTP, pesan SOAP dokumen dapat dikirim ke server Web, yang memanggil fungsi tertentu yang ditetapkan oleh layanan Web.
- ✓ SOAP dapat dilihat di <http://www.w3.org/TR/SOAP>.
- ✓ Pesan SOAP terdiri dari:
  - Top elemen dari XML
  - SOAP Header
  - SOAP Body

# UDDI

UDDI: [Universal Description, Discovery, and Integration](#)

- ✓ Sebuah mekanisme untuk penerbitan dan menggambarkan layanan Web untuk klien potensial
- ✓ UDDI Adalah seperangkat pendaftar yang direplikasi mengenai informasi tentang layanan Web pada jaringan.
- ✓ Pendaftaran layanan melibatkan empat inti jenis struktur data: informasi bisnis, layanan informasi, informasi yang mengikat, dan informasi spesifikasi

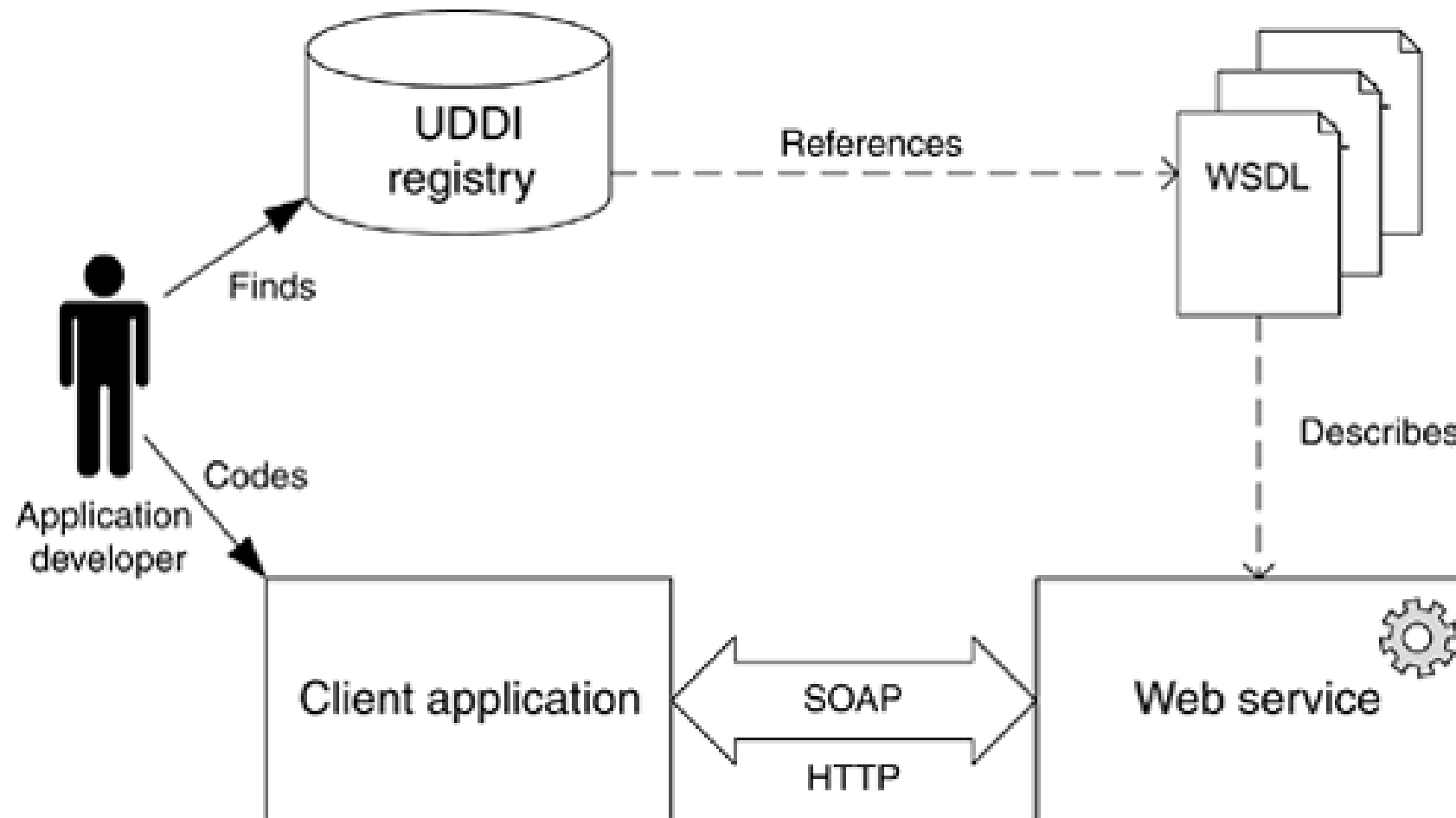


# WSDL

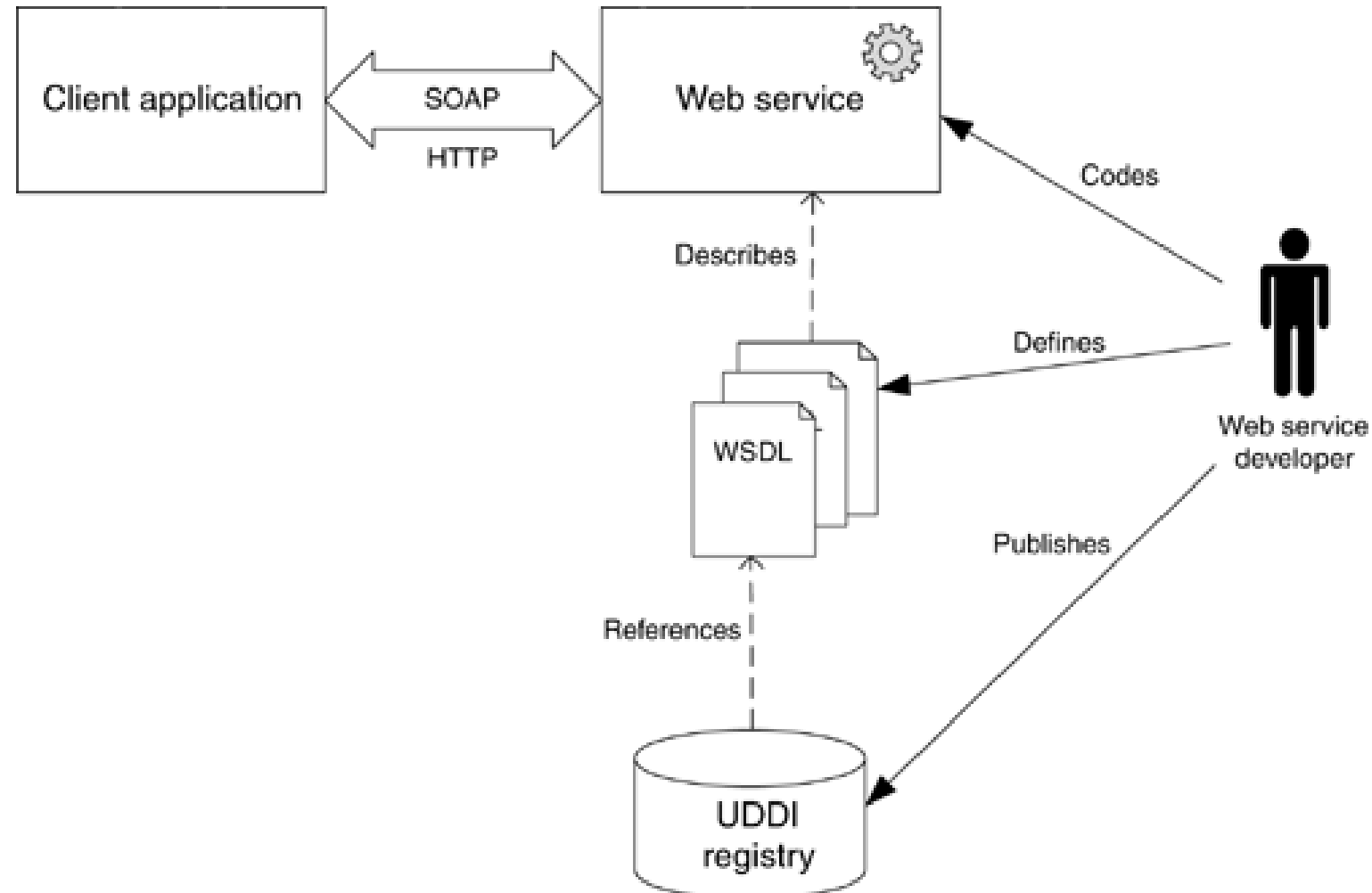
WSDL: [Web Services Description Language](#)

- ✓ Sebuah spesifikasi lebih rinci dari antarmuka SOAP
- ✓ Spesifikasi WSDL menjelaskan secara rinci bagaimana untuk memohon layanan Web dan apa yang diharapkan ketika merespon
- ✓ Sebuah deskripsi WSDL mendefinisikan layanan Web sebagai kumpulan titik akhir jaringan, atau port
  - Setiap port mendefinisikan koleksi operasi yang dapat dipanggil.
  - Setiap operasi mencakup seperangkat input dan output pesan: parameter.
  - Sebuah operasi memetakan sebuah jenis port untuk protokol tertentu, HTTP, dan format data SOAP.
  - Sebuah port memberi contoh jenis port dan mengikat pada alamat jaringan tertentu.

# WS in the development and run-time environment



# Development of Web Services



# WSDL

WSDL: [Web Services Description Language](#)

- ✓ Sebuah spesifikasi lebih rinci dari antarmuka SOAP
- ✓ Spesifikasi WSDL menjelaskan secara rinci bagaimana untuk memohon layanan Web dan apa yang diharapkan ketika merespon
- ✓ Sebuah deskripsi WSDL mendefinisikan layanan Web sebagai kumpulan titik akhir jaringan, atau port
  - Setiap port mendefinisikan koleksi operasi yang dapat dipanggil.
  - Setiap operasi mencakup seperangkat input dan output pesan: parameter.
  - Sebuah operasi memetakan sebuah jenis port untuk protokol tertentu, HTTP, dan format data SOAP.
  - Sebuah port memberi contoh jenis port dan mengikat pada alamat jaringan tertentu.

Ada Pertanyaan



## Tugas (Individu)

Jelaskan Perbedaan infrastruktur JAVA RMI dengan Microsoft DCOM

1. Jelaskan perbedaan dari cara kerjanya
2. Jelaskan perbedaan infrastrukturnya
3. Berikan masing-masing contohnya

Tambahan Penjelasan: SOAP, UDDI dan WSDL

# Tugas 2 (Kelompok 3 Orang)

1. Silahkan pilih Anggota Kelompok Anda (Bebas), **maksimal** 3 orang
2. Penentuan Kelompok **didasarkan pada yang tercepat menulis di grup WA**, Tulis: Kelompok 1 dst, beserta Nama-Nama Anggotanya
3. Tugas ada di slide berikutnya
4. Silahkan diskusikan jawabannya, kemudian dibuat laporan (diktik) dengan menyertakan Referensi (dari Buku/ E-Book, dan atau Jurnal)
5. **Jawaban di Upload Kulino** pada kolom **Tugas Pertemuan-4 (Perwakilan saja)**
6. Beri nama file: **Tugas2\_NIM1\_NIM2\_NIM3** (*NIM tuliskan 5 digit terakhir saja*)

## Tugas Kelompok (Max 3. orang)

Carilah Penjelasan, dan Contoh mengenai hal-hal dibawah ini:

### Kelompok 1

- Apa itu Web Engineering
- Jelaskan Konsep Web Engineering
- Karakteristik Web App

### Kelompok 2

- Kriteria Web berdasarkan Fungsional/ kategori
- Kategori Web App
- Document Centric Web Site

### Kelompok 3

- Interactive Web App
- Transactional Web App
- Workflow-based Web App

### Kelompok 4

- **Collaborative Web App**
- **Social Web**
- **Portal-Oriented Web App**

### Kelompok 5

- Ubiquitous Web App
- Semantic Web
- Popular Site





***THANKS***

ANY QUESTIONS?