Sistem Terdistribusi

TIK-604

Husni.trunojoyo.ac.id

Web Service: SOA dan REST

Topik Khusus (Belajar Mandiri)

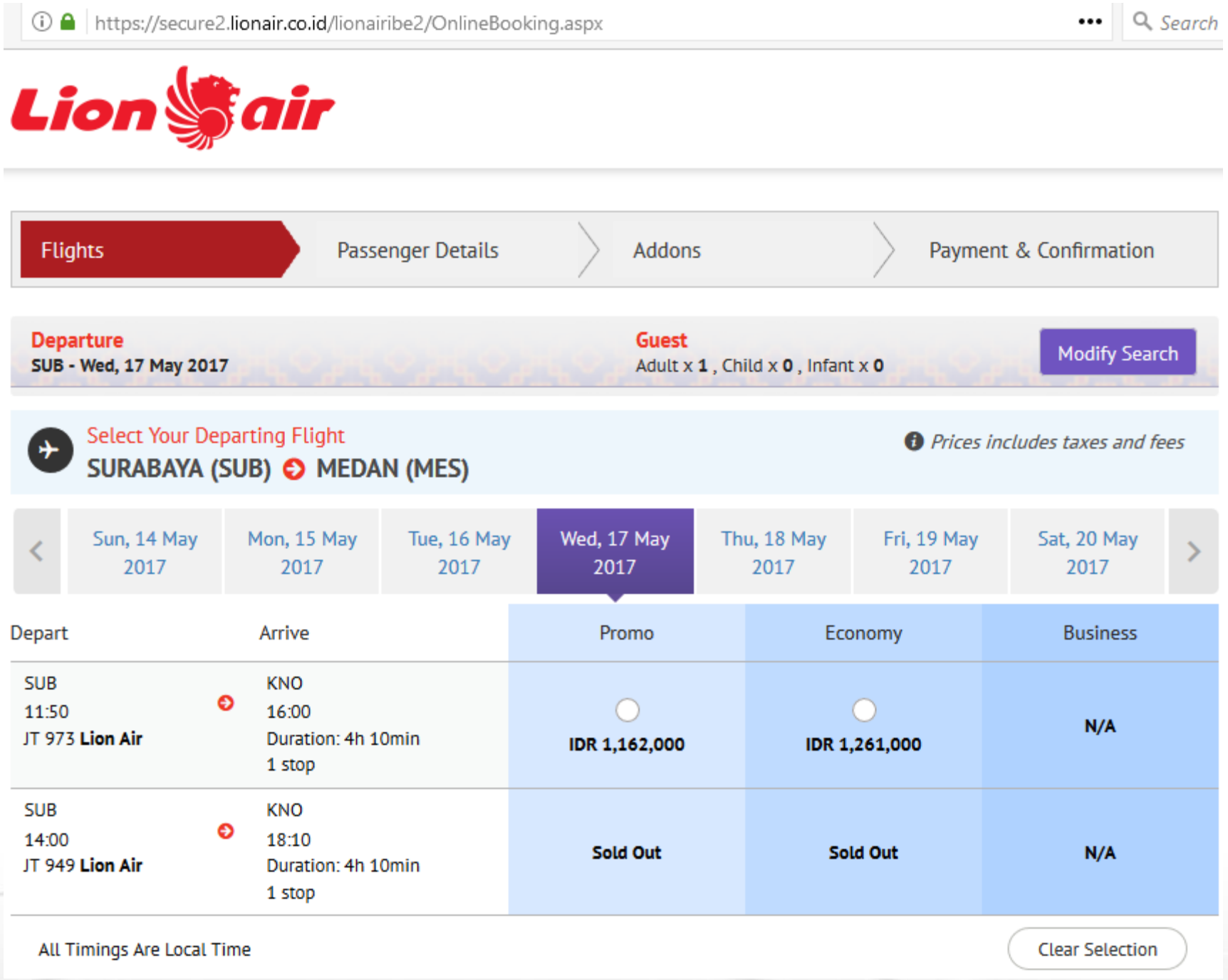
Husni

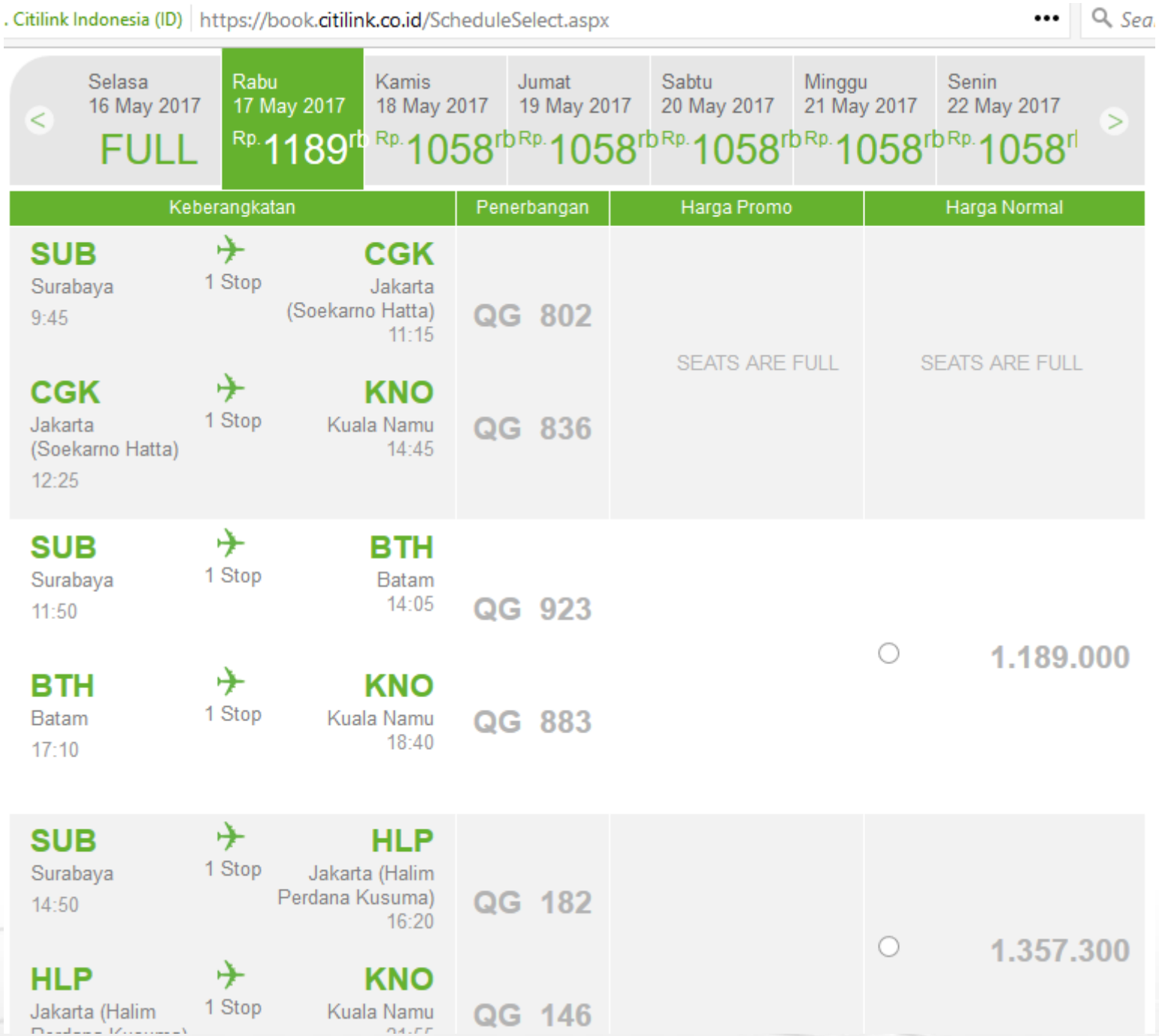
husni@trunojoyo.ac.id

Garis Besar Bahasan

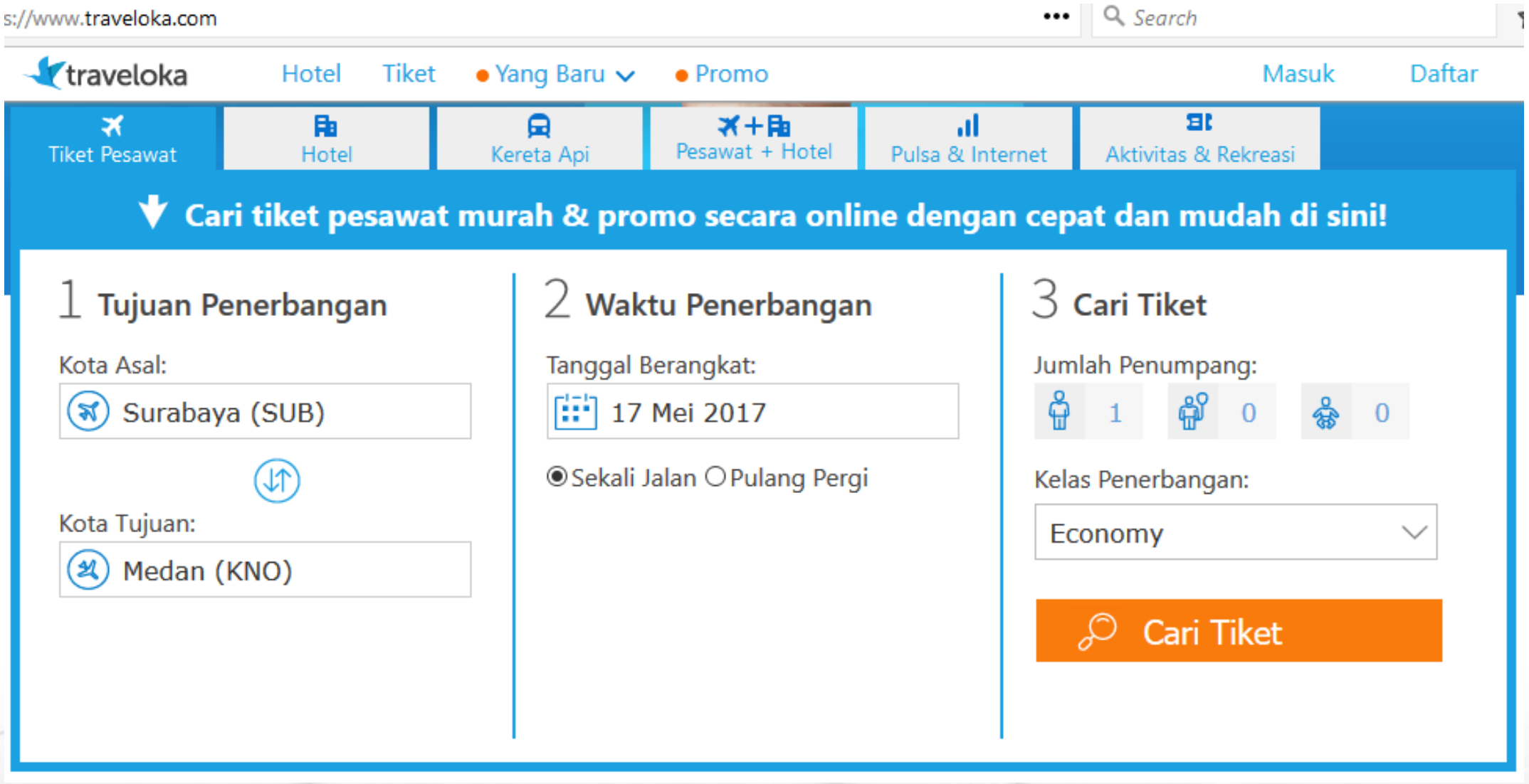
1. Aplikasi terdistribusi, Web Services & Service-Oriented Architecture (SOA)
2. Infrastruktur Web Service perusahaan, Standard dan Protocolnya
   - SOAP, WSDL, HTTP, XML, WS-*, …
3. Protokol HTTP
4. RESTful Web Services
   - Representational State Transfer (REST)
   - Operasi CRUD dan metode HTTP
   - Postman: Client REST.

Aplikasi Terdistribusi, Web Services dan SOA


https://secure2.lionair.co.id/lionairibe2/OnlineBooking.aspx
Search
Lion air
Flights
Passenger Details
Addons
Payment & Confirmation
Departure
SUB - Wed, 17 May 2017
Guest
Adult x 1 , Child x 0 , Infant x 0
Modify Search
Select Your Departing Flight
SURABAYA (SUB) MEDAN (MES)
Prices includes taxes and fees
Sun, 14 May 2017
Mon, 15 May 2017
Tue, 16 May 2017
Wed, 17 May 2017
Thu, 18 May 2017
Fri, 19 May 2017
Sat, 20 May 2017
Depart
Arrive
Promo
Economy
Business
SUB
11:50
JT 973 Lion Air
KNO
16:00
Duration: 4h 10min
1 stop
IDR 1,162,000
IDR 1,261,000
N/A
SUB
14:00
JT 949 Lion Air
KNO
18:10
Duration: 4h 10min
1 stop
Sold Out
Sold Out
N/A
All Timings Are Local Time
Clear Selection


. Citilink Indonesia (ID) | https://book.citilink.co.id/ScheduleSelect.aspx
Selasa 16 May 2017 FULL
Rabu 17 May 2017 Rp. 1189rb
Kamis 18 May 2017 Rp. 1058rb
Jumat 19 May 2017 Rp. 1058rb
Sabtu 20 May 2017 Rp. 1058rb
Minggu 21 May 2017 Rp. 1058rb
Senin 22 May 2017 Rp. 1058rb
Keberangkatan
Penerbangan
Harga Promo
Harga Normal
SUB Surabaya 9:45
1 Stop
CGK Jakarta (Soekarno Hatta) 11:15
QG 802
SEATS ARE FULL
SEATS ARE FULL
CGK Jakarta (Soekarno Hatta) 12:25
1 Stop
KNO Kuala Namu 14:45
QG 836
SUB Surabaya 11:50
1 Stop
BTH Batam 14:05
QG 923
BTH Batam 17:10
1 Stop
KNO Kuala Namu 18:40
QG 883
1.189.000
SUB Surabaya 14:50
1 Stop
HLP Jakarta (Halim Perdana Kusuma) 16:20
QG 182
HLP Jakarta (Halim
1 Stop
KNO Kuala Namu
QG 146
1.357.300

Teritegrasi di Traveloka.com



Surabaya (SUB)-Juanda → Medan (KNO)-Kuala Namu

Rabu, 17 Mei 2017 | 1 Dewasa | Economy

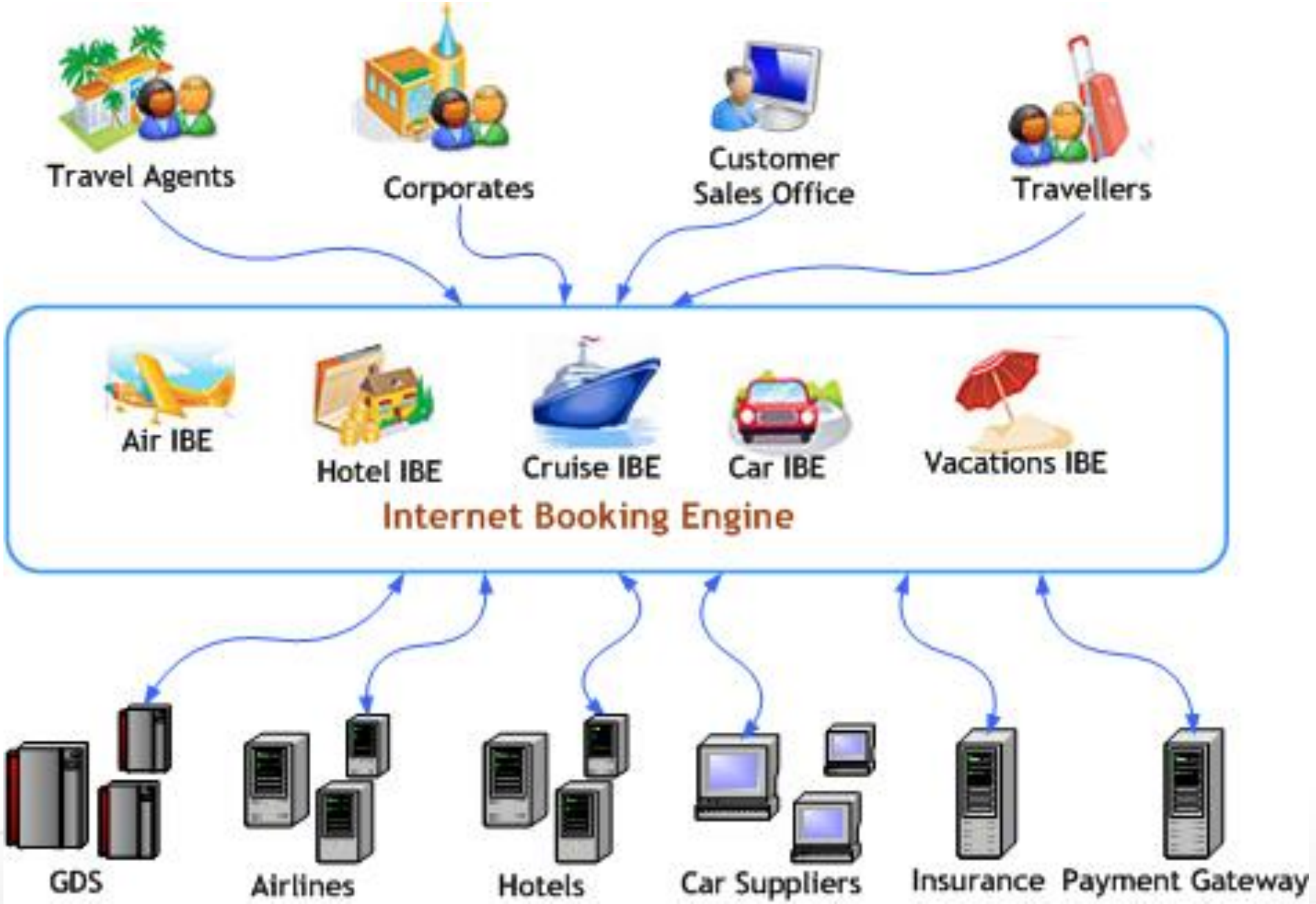
Ganti pencarian

Filter: Transit Waktu Maskapai Filter lain

Lihat tanggal lain

Maskapai	Berangkat	Tiba	Durasi	Fasilitas	Harga per orang
Sriwijaya	16:30 Surabaya (SUB)	08:10 (+1h) Medan (KNO)	15j 40m 1 transit	20	~~Rp 1.102.099~~ Rp 1.093.800 Pilih
Detail Penerbangan Rincian Harga					
Sriwijaya	18:25 Surabaya (SUB)	08:10 (+1h) Medan (KNO)	13j 45m 1 transit	20	~~Rp 1.102.099~~ Rp 1.093.800 Pilih
Detail Penerbangan Rincian Harga					
Citilink Multi-maskapai	18:45 Surabaya (SUB)	08:10 (+1h) Medan (KNO)	13j 25m 1 transit	20	~~Rp 1.126.800~~ Rp 1.119.966 Pilih
Detail Penerbangan Rincian Harga					

Citilink Multi-maskapai	19:10 Surabaya (SUB)		07:20 (+1h) Medan (KNO)	12j 10m 1 transit	20	~~Rp 1.127.900~~ Rp 1.122.195 Pilih
Detail Penerbangan	Rincian Harga					
Citilink Multi-maskapai	19:10 Surabaya (SUB)		08:20 (+1h) Medan (KNO)	13j 10m 1 transit	20	~~Rp 1.127.900~~ Rp 1.122.195 Pilih
Detail Penerbangan	Rincian Harga					
Citilink Multi-maskapai	20:00 Surabaya (SUB)		07:20 (+1h) Medan (KNO)	11j 20m 1 transit	20	~~Rp 1.127.900~~ Rp 1.122.195 Pilih
Detail Penerbangan	Rincian Harga					
Citilink Multi-maskapai	20:00 Surabaya (SUB)		08:20 (+1h) Medan (KNO)	12j 20m 1 transit	20	~~Rp 1.127.900~~ Rp 1.122.195 Pilih
Detail Penerbangan	Rincian Harga					
Citilink Multi-maskapai	20:00 Surabaya (SUB)		09:20 (+1h) Medan (KNO)	13j 20m 1 transit	20	~~Rp 1.127.900~~ Rp 1.122.195 Pilih


Travel Agents
Corporates
Customer Sales Office
Travellers
Air IBE
Hotel IBE
Cruise IBE
Car IBE
Vacations IBE
Internet Booking Engine
GDS
Airlines
Hotels
Car Suppliers
Insurance
Payment Gateway

Aplikasi Terdistribusi

- Sebagian besar aplikasi modern sifatnya terdistribusi.
  - Beberapa komponen saling berinteraksi
- Model aplikasi terdistribusi
  - "Client-Server“: koneksi socket / WebSocket *persistent*
  - "Distributed Objects“: client adalah obyek dari server
    - DCOM, CORBA, Java RMI, .NET Remoting, …
    - **Kadaluarsa**, tidak digunakan lagi dalam aplikasi modern
  - "Web Services" / "RESTful Web Services" (trend terkini)
    - RESTful (HTTP, REST, JSON) dan layanan berat (SOAP, WSDL, XML)

Services: Dunia Nyata dan Perangkat Lunak

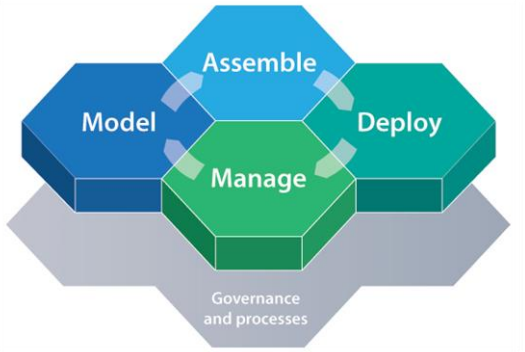
- Dalam dunia nyata "**service**" berarti:
  - Suatu bagian kerja yang dilaksanakan oleh *service provider*
  - Menerima beberapa input dan memproduksi hasil yang diinginkan
  - Mis. Suatu supermarket: bayar uang dan terima buah-buahan
  - Ada karakteristik kualitas (harga, waktu eksekusi, pembatas, dll.)
- Dalam dunia software "**service**" bermaksud:
  - Menerima beberapa input, melaksanakan kerja, memproduksi keluaran
  - Model request-response: client meminta, server menanggapinya.

Web Services dan Clients

- Suatu "**web service**" adalah
  - Service perangkat lunak yang berkomunikasi di atas protokol web standard
  - Service klasik (kelas berat) menggunakan SOAP, WSDL, XML, WS-*
  - Service kelas ringan (RESTful) menggunakan HTTP, REST dan JSON.
- Suatu "**client**" (consumer, konsumen) menggunakan services
  - Meminta (r**equest**) sesuatu untuk dikerjakan
  - Mendapatkan **hasil** yang diharapkan
  - Atau memperoleh suatu **error** (kesalahan).

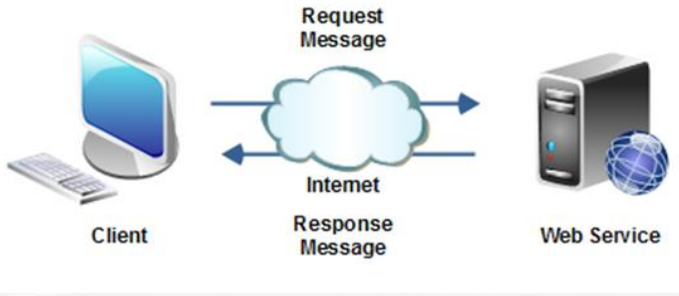
Service-Oriented Architecture (SOA)

- **SOA** (**S**ervice-**O**riented **A**rchitecture): suatu konsep arsitektural untuk pengembangan sistem perangkat lunak
  - Memanfaatkan blok bangunan (komponen) yang dapat digunakan ulang (*reusable*) bernama "services"
  - SOA == membagi-bagi (*decouple*) perangkat lunak monolitik menjadi layanan-layanan dapat digunakan kembali
- Services dalam SOA adalah
  - Otonom, fungsi bisnis ***stateless***
  - Menerima permintaan dan mengembalikan tanggapan
  - Menggunakan antarmuka standard & terdefinisi bagus (protokol standard)



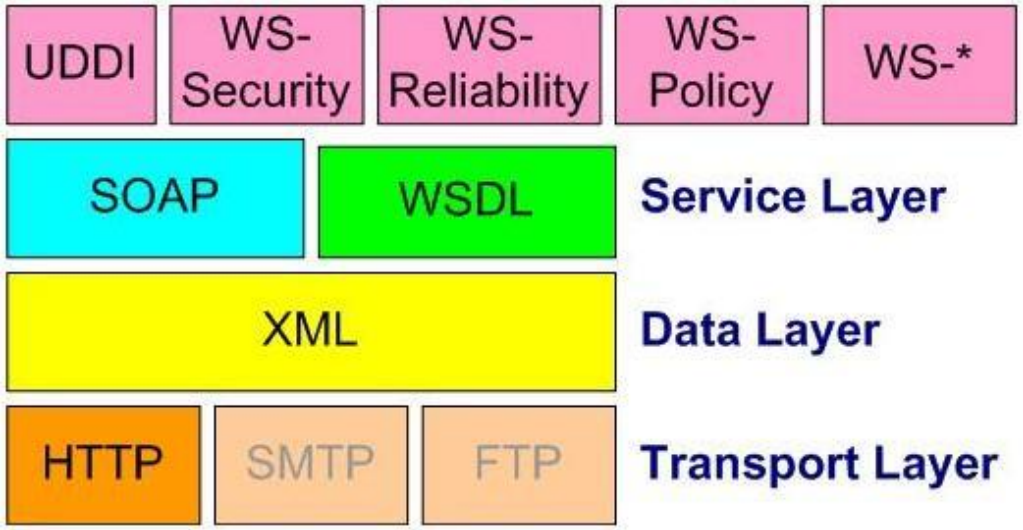
Service di SOA

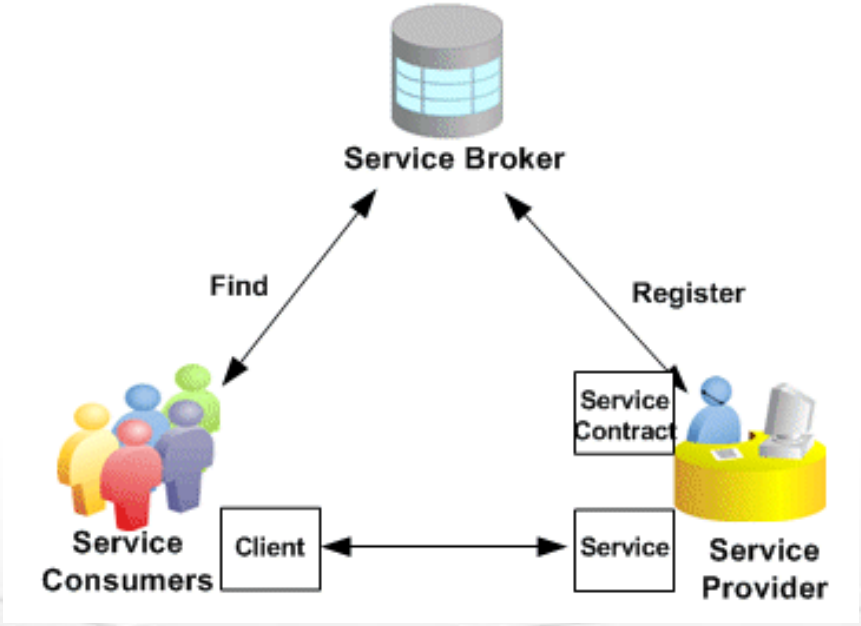
- Otonom
  - Setiap service beroperasi secara otonom
  - Tanpa pengetahuan adanya service yang lain
- Stateless
  - Tidak mengingat suatu status tahan lama antar requests
    - Dapat menyimpan status dalam database & merujuknya berdasarkan ID
  - Mudah diskalakan → cukup tambahkan node berikutnya
- Model request-response
  - Client bertanya, server mengembalikan jawaban
  - Server tidak pernah mengirimkan request kepada client.



Service di SOA (2)

- Komunikasi melalui protokol standard
  - HTTP, FTP, SMTP, RPC, MSMQ, ...
  - JSON, XML, SOAP, RSS, WS-*, ...
- Tidak tergantung Platform
  - Bebas OS, platform,
    bahasa, framework, …
- Discoverable
  - Registrasi dan perantara (broker) service



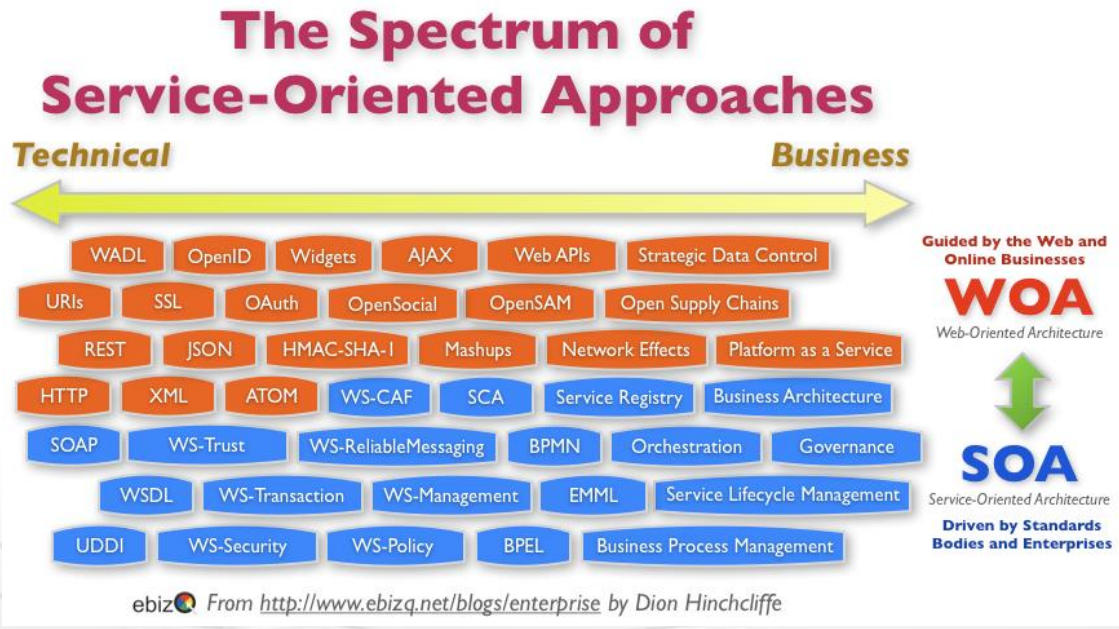


SOA Kelas Ringan (SOA di Internet)

- Perusahaan Internet mengimplementasikan *lightweight* SOA di Internet
  - Juga disebut WOA (Web-Oriented Architecture)
  - Contoh: Google, Amazon, Facebook, Twitter, Parse.com, ...
  - Berbasis pada standard web ringan (lightweight):
    - AJAX dan Rich Internet Applications (RIA)
    - REST, JSON, RSS, XML, API berbayar
- **RESTful Web services == lightweight Web services**
  - Menggunakan request HTTP simpel & respon JSON simpel

Heavyweight SOA (SOA di Perusahaan)

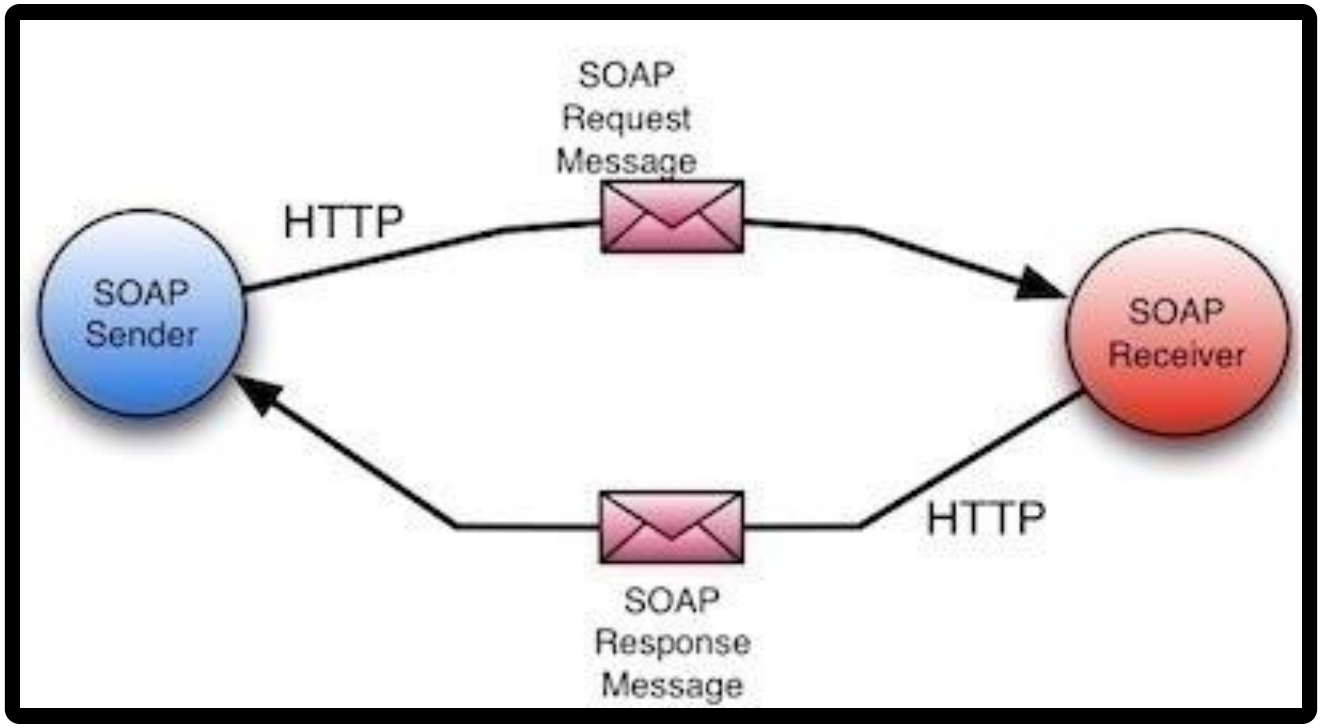
- **Stack SOA kelas berat**
  - Digerakkan oleh proses bisnis: BPM, BPMN, BPEL, ...
  - Enterprise application integration (EAI)
  - Integrasi B2B dan portal berbasis SOA
  - Framework seragam: SCA dan WCF
  - Enterprise Service Bus (ESB)
  - Pemerintah (kontrol) SOA
  - Banyak standard publik seperti **WS-***



Standard Web Service: WS-*

- Service Discovery Standards
  - UDDI, RDDL, XRI, XRDS
- Service Messaging Standards
  - SOAP, SOAP over JMS, MTOM, WS-Addressing
- Service Meta-Data Standards
  - WSBPEL, WSDL, WADL, WSFL, WS-Policy, WS-PolicyAssertions, WS-PolicyAttachment, WS-MetadataExchange (WS-MEX)
- Web Service Security Standards
  - XML-Signature, WS-SecurityPolicy, WS-Security, WS-Trust, WS-SecureConversation
- Quality of Service Standards
  - WS-ReliableMessaging (WS-RM), WS-Coordination, WS-AtomicTransactions, WS-TX

https://www.oasis-open.org/standards



Infrastruktur Web Service Perusahaan

SOAP / WSDL / XML / HTTP

Infrastruktur Web Services Kelas Berat

- Komponen infrastruktur Web Service kelas berat (klasik) :
  - Deskripsi
    - WSDL (Web Service Definition Language)
  - Metadata
    - WS-MetadataExchange (WS-MEX), DISCO
  - Format pengikat
    - SOAP, XML, XSD
    - HTTP

WSDL Service Description (WSDL)

- **WSDL** (**W**eb **S**ervices **D**escription **L**anguage)
  - Mendiskripsikan yang dapat dilakukan Web service
    - Nama dari metode (messages) yang tersedia
    - Parameter Input dan output, nilai yang dikembalikan
    - Tipe data yang digunakan untuk parameter atau hasil
    - Endpoints: ports dan bindings
  - WSDL adalah standard terbuka berbasis XML dari W3C.

WSDL: Contoh

```
<?xml version="1.0" encoding="utf-8"?>
<definitions
 xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:s="http://www.w3.org/2001/XMLSchema"
 xmlns:s0="http://www.devbg.org/ws/MathService"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
 xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
 targetNamespace="http://www.devbg.org/ws/MathService"
 xmlns="http://schemas.xmlsoap.org/wsdl/">

 <types> … </types>
 <message name="AddSoapIn"> … </message>
 <portType name="MathServiceSoap"> … </portType>
 <binding name="MathServiceSoap" … > … </binding>
 <service name="MathService"> … </service>
</definitions>
```

Penemuan Web Service

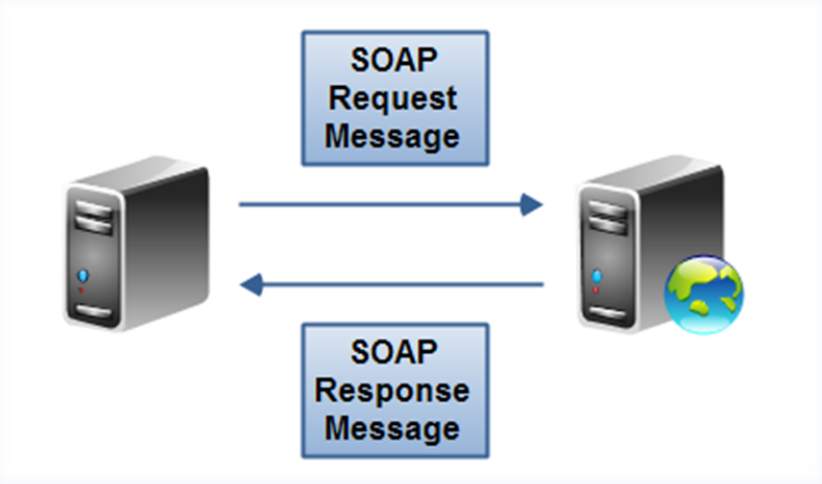
- Proses mendapatkan metadata dari service (deskripsi)
- Biasanya suatu **URL** ditanyai untuk me-retrieve metadata
- Dua protokol untuk interogasi
  - **WS-MetadataExchange** (WS-MEX)
    - Protokol berstandar yang dikembangkan oleh Microsoft, Sun, SAP, …
  - **DISCO**
    - Protokol Microsoft lama untuk digunakan dengan pendaftaran UDDI

SOAP: Format Request / Hasil

- **SOAP** (**S**imple **O**bject **A**ccess **P**rotocol)
  - Format berbasis XML terbuka untuk pengiriman pesan
  - Standar terbuka dari W3C
- Suatu pesan **SOAP** terdiri dari:
  - **SOAP header** – mendeskripsikan parameter pesan (metadata)
  - **SOAP body** – data pesan (badan *request* atau *response*)
- Biasanya pesan **SOAP** dikirim di atas **HTTP**
  - TCP Opsional / antrian pesan (*message queue*) / jalur lain dapat digunakan

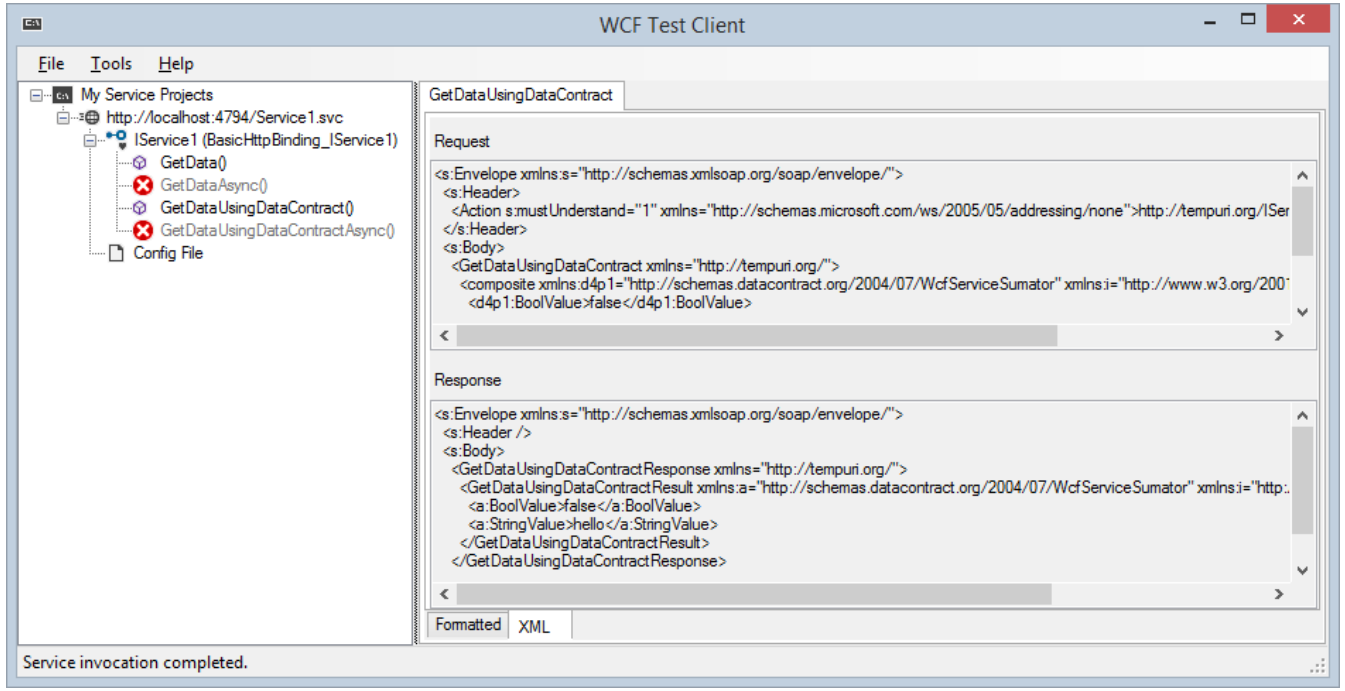
SOAP Request: Contoh

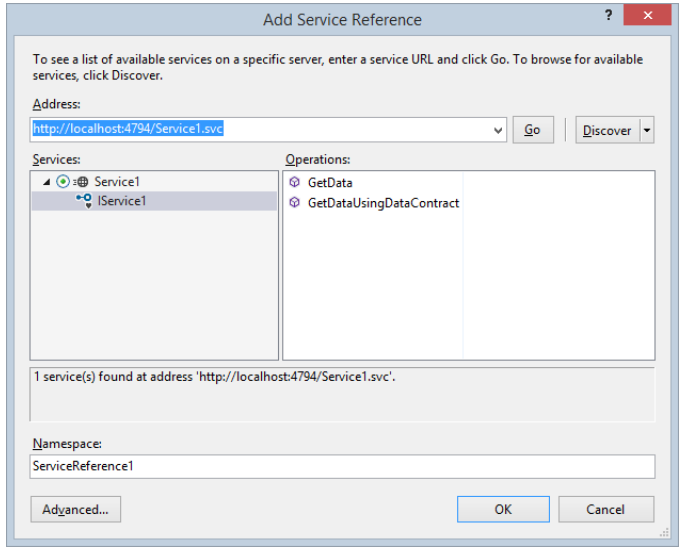
```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CalcDistance xmlns="http://www.devbg.org/Calc">
      <startPoint>
        <x>4</x> <y>5</y>
      </startPoint>
      <endPoint>
        <x>7</x> <y>-3</y>
      </endPoint>
    </CalcDistance>
  </soap:Body>
</soap:Envelope>
```



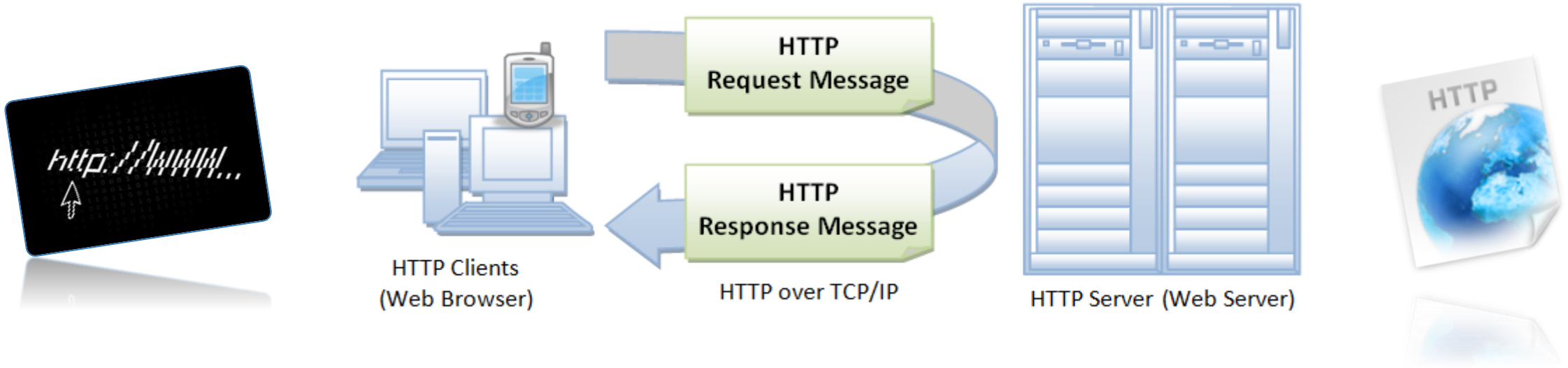
SOAP Response: Contoh

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CalcDistanceResponse
     xmlns="http://www.devbg.org/Calc/">
      <CalcDistanceResult>8.54400374531753</CalcDistanceResult>
    </CalcDistanceResponse>
  </soap:Body>
</soap:Envelope>
```





Web Services Kelas Berat (Berbasis pada SOAP & WSDL)



Protokol HTTP

Bagaimana HTTP Bekerja?

HTTP

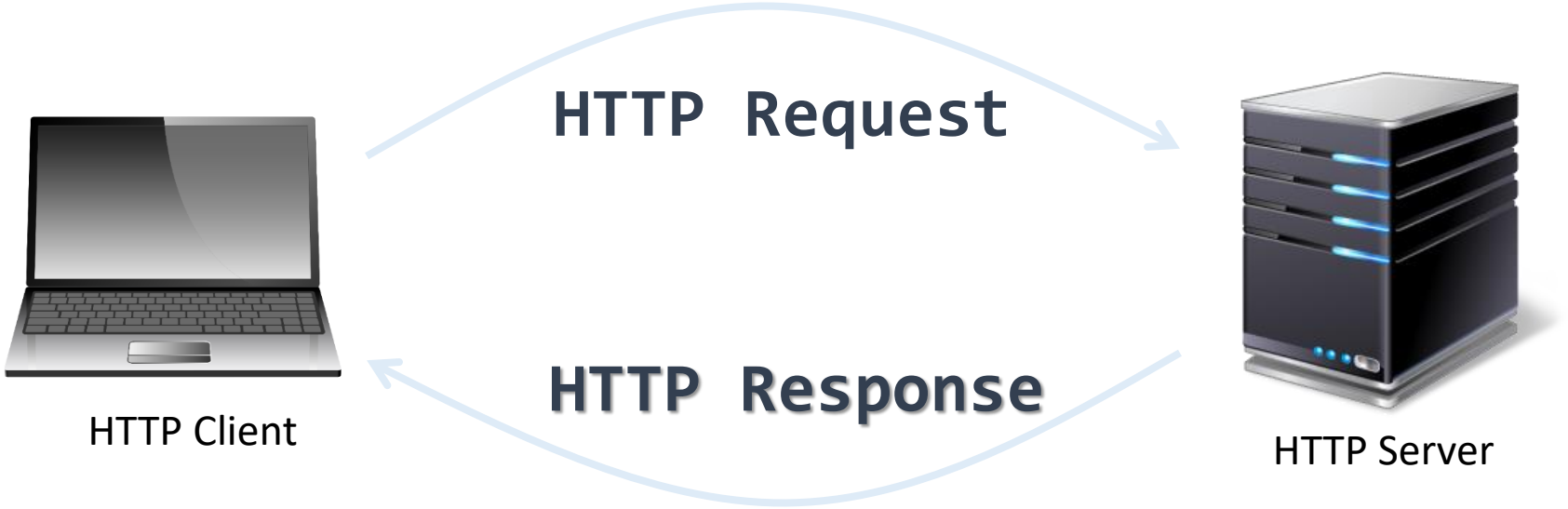
- HTTP == Hyper Text Transfer Protocol
  - Protokol client-server untuk pen-transfer-an sumber daya web (file HTML, images, styles, scripts, data, dll.)
  - Protokol tersebar luas untuk komunikasi Internet saat ini
  - Model request-response (client me-request, server menjawab)
  - Format berbasis teks (*human readable*)
  - Bersandar pada sumber daya unik URL
  - Menyediakan metadata sumber daya (mis. *encoding*)
  - Stateless (***cookies*** dan penyimpanan Web dapat mengatasi ini)

HTTP: Protokol *Request-Response*

- Program Client
  - Berjalan pada *end host*
  - Meminta (*request*) sumber daya

Program Server

- Berjalan pada server
- Menyediakan sumber daya



Percakapan HTTP: Contoh

- HTTP request:

```
GET /courses/javascript HTTP/1.1
Host: www.softuni.bg
User-Agent: Mozilla/5.0
<CRLF>
```

Baris kosong menunjukkan akhir dari request header

- HTTP response:

```
HTTP/1.1 200 OK
Date: Mon, 5 Jul 2010 13:09:03 GMT
Server: Microsoft-HTTPAPI/2.0
Last-Modified: Mon, 12 Jul 2014 15:33:23 GMT
Content-Length: 54
<CRLF>
<html><title>Hello</title>
Welcome to our site</html>
```

Baris kosong menunjukkan akhir dari response header

Metode HTTP Request

- HTTP mendefinisikan metode-metode *request*
  - Menentukan aksi untuk dikerjakan pada sumber daya yang diidentifikasi

Metode	Deskripsi
GET	Me-retrieve sumber daya (eksekusi query)
POST	Membuat sumber daya
PUT	Mengubah sumber daya
DELETE	Menghapus (delete) sumber daya
HEAD	Me-retrieve header sumber daya
OPTIONS	Me-requests opsi komunikasi

Pesan Request HTTP

Pesan Request HTTP

- Pesan request yang dikirimkan oleh client berisi:
  - Baris request HTTP
    - Metode request (GET / POST / PUT / DELETE / …)
    - URI sumber daya (URL)
    - Versi protokol
  - Header request HTTP
    - Parameter tambahan
  - Badan (*body*) HTTP: data opsional, mis. *posted form fields*

```
<method> <resource> HTTP/<version>
<headers>
(empty line)
<body>
```

Request HTTP GET: Contoh

- Contoh request HTTP **GET**:

```
GET /courses/javascript HTTP/1.1
Host: www.softuni.bg
Accept: */*
Accept-Language: bg
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0(compatible;MSIE 6.0;Windows NT 5.0)
Connection: Keep-Alive
Cache-Control: no-cache
<CRLF>
```

Baris request HTTP

Request header HTTP

Request body kosong

Request HTTP POST: Contoh

- Contoh request HTTP **POST**:

```
POST /webmail/login.phtml HTTP/1.1
Host: www.abv.bg
Accept: */*
Accept-Language: bg
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0(compatible;MSIE 6.0; Windows NT 5.0)
Connection: Keep-Alive
Cache-Control: no-cache
Content-Length: 59
<CRLF>
username=mente&password=top*secret!
<CRLF>
```

Baris request HTTP

Request header HTTP

Request body memegang data form yang di-submit-kan

Get Kondisional HTTP: Contoh

- Contoh request GET kondisional HTTP:

```
GET /apply HTTP/1.1
Host: www.softuni.bg
User-Agent: Gecko/20100115 Firefox/3.6
If-Modified-Since: Tue, 9 Mar 2015 11:12:23 GMT
<CRLF>
```

- Mengambil sumber daya hanya jika telah berubah di server
  - Server membalas dengan "**304 Not Modified**" jika sumber daya belum berubah
  - Atau sebaliknya "200 OK" dengan versi terakhir.

Pesan Respon HTTP

Pesan Respon HTTP

- Pesan respon yang dikirimkan oleh server HTTP mengandung:
  - Baris status respon HTTP
    - Versi protokol
    - Kode status
    - Frase status
  - Header dari respon
    - Menyediakan meta data mengenai sumber daya yang dikembalikan
  - Badan (*body*) dari respon
    - Isi dari respon HTTP (data)

```
HTTP/<version> <status code> <status text>
<headers>
<CRLF>
<response body – the requested resource>
```

Respon HTTP: Contoh

- Contoh respon HTTP dari Web server:

```
HTTP/1.1 200 OK
Date: Fri, 17 Jul 2010 16:09:18 GMT+2
Server: Apache/2.2.14 (Linux)
Accept-Ranges: bytes
Content-Length: 84
Content-Type: text/html
<CRLF>
<html>
  <head><title>Test</title></head>
  <body>Test HTML page.</body>
</html>
```

Baris status respon HTTP

Response Header HTTP

Response body HTTP

Respon HTTP: Contoh

- Contoh respon HTTP dengan hasil error :

```
HTTP/1.1 404 Not Found
Date: Fri, 17 Nov 2014 16:09:18 GMT+2
Server: Apache/2.2.14 (Linux)
Connection: close
Content-Type: text/html
<CRLF>
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /img/logo.gif was not found on this server.<P>
<HR><ADDRESS>Apache/2.2.14 Server at Port 80</ADDRESS>
</BODY></HTML>
```

Baris status respon HTTP

Response Header HTTP

Response Body HTTP

Kode Respon HTTP

- Kelas kode respon HTTP
  - 1xx: **informational** (mis., "100 Continue")
  - 2xx: **successful** (mis., "200 OK", "201 Created")
  - 3xx: **redirection** (mis., "304 Not Modified", "301 Moved Permanently", "302 Found")
  - 4xx: **client error** (mis., "400 Bad Request", "404 Not Found", "401 Unauthorized", "409 Conflict")
  - 5xx: **server error** (mis., "500 Internal Server Error", "503 Service Unavailable")

Content-Type dan *Content-Disposition*

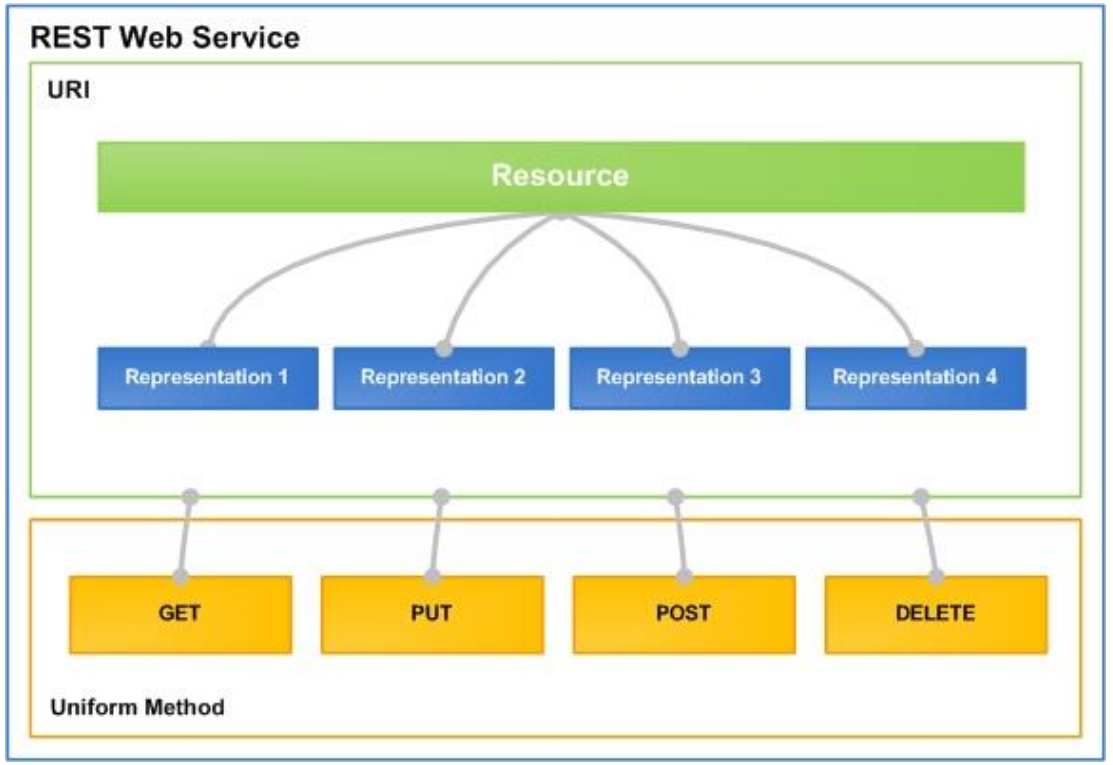
- Header respon **`Content-Type`** digunakan oleh server untuk menetapkan bagaimana output akan diproses.
- Contoh:

Halaman HTML ber-encode UTF-8; akan ditampilkan dalam browser

```
Content-Type: text/html; charset=utf-8
```

```
Content-Type: application/pdf
Content-Disposition: attachment; filename="Report-April-2015.pdf"
```

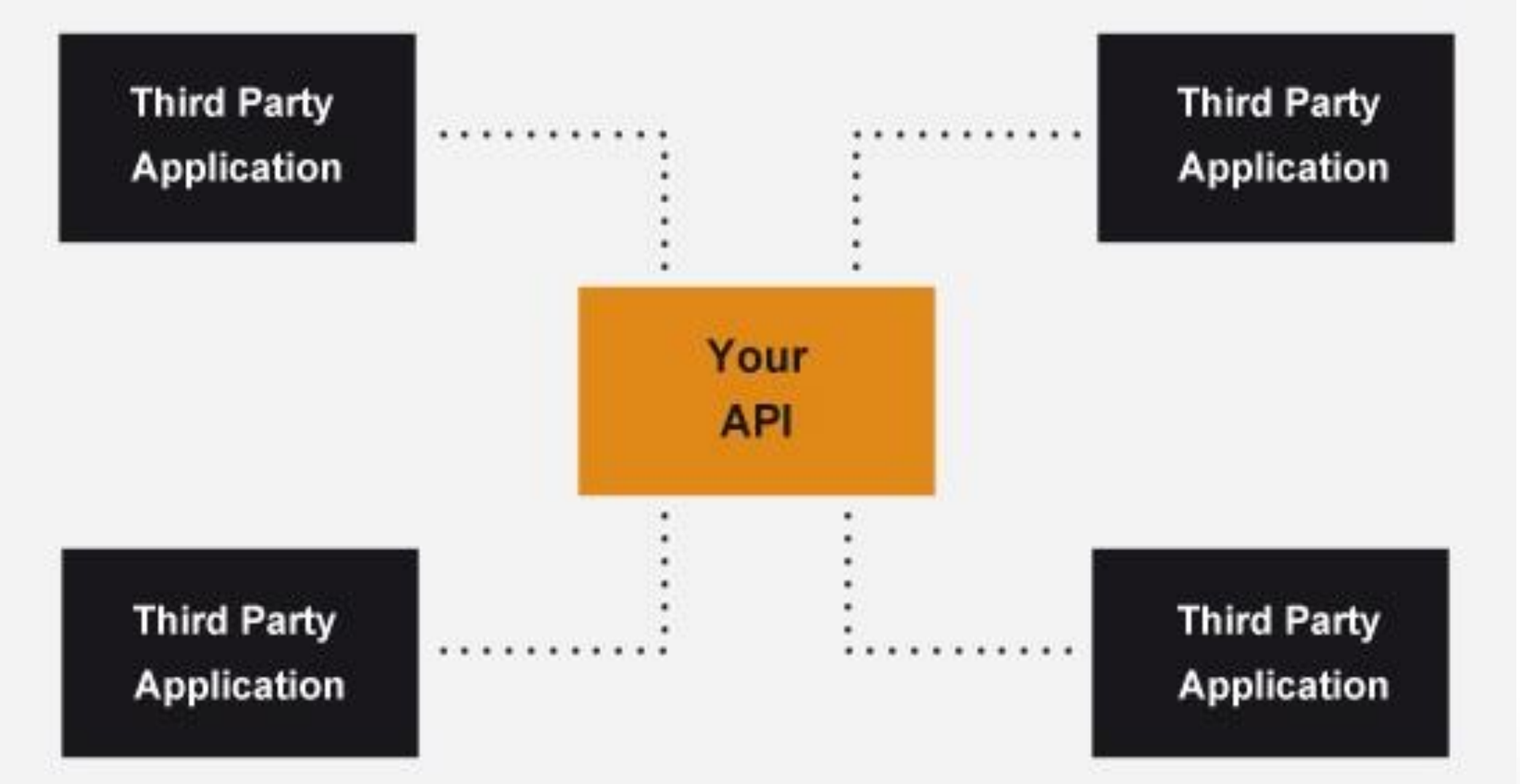
Ini akan mendownload file PDF bernama **`Financial-Report-April-2015.pdf`**



Layanan Web RESTful

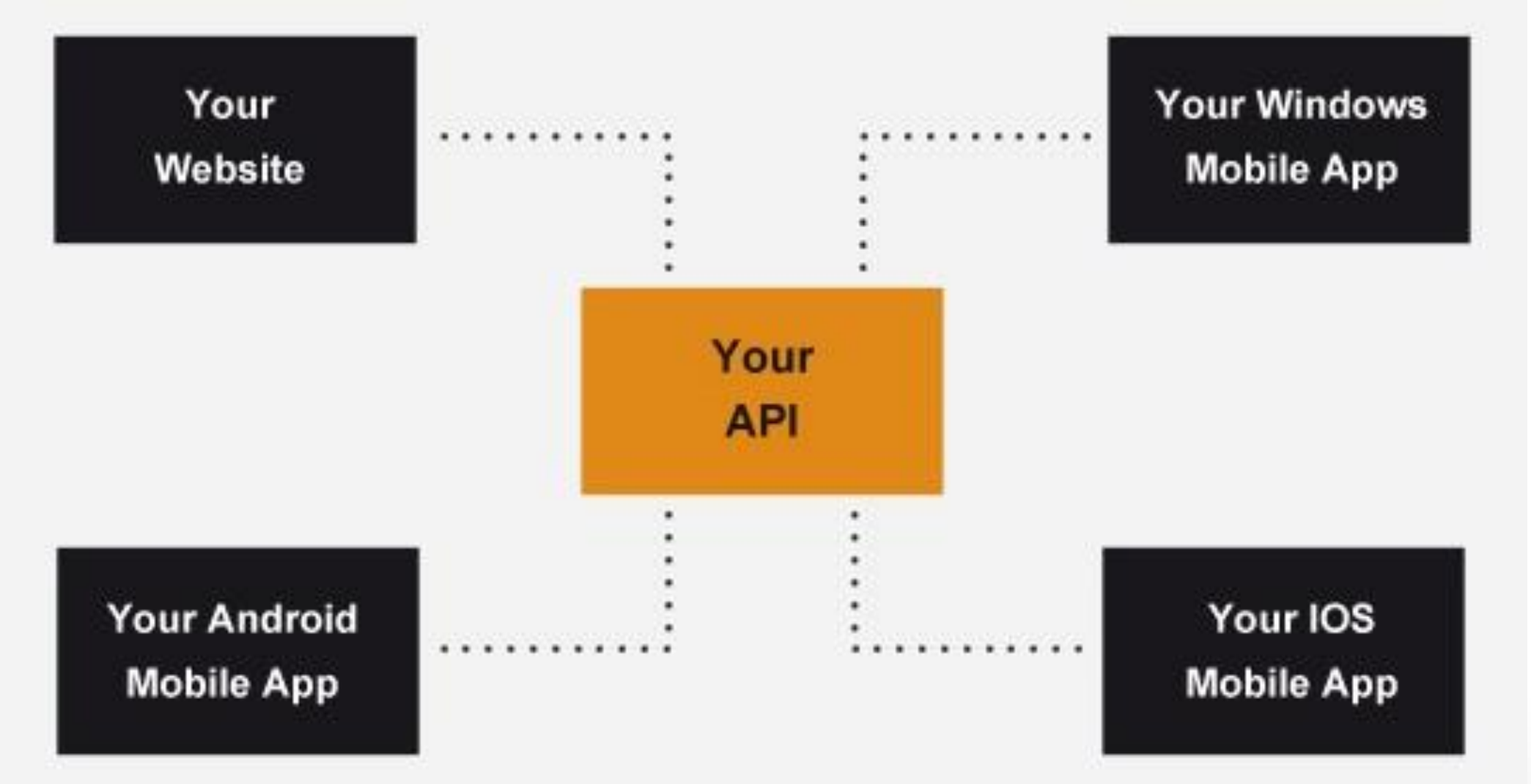
Arsitektur Kelas Ringin bagi Web Services

API Mengintegrasikan Aplikasi Pihak Lain



API Mengumpulkan Hasil dari API Pihak Lain

API Mengintegrasikan Semua Aplikasi Kita



Apa itu REST?

"*Representational State Transfer* (REST) merupakan *software architecture style* berisi panduan dan praktek terbaik bagi pembuatan *web service* yang dapat diskalakan."

http://en.wikipedia.org/wiki/Representational_State_Transfer

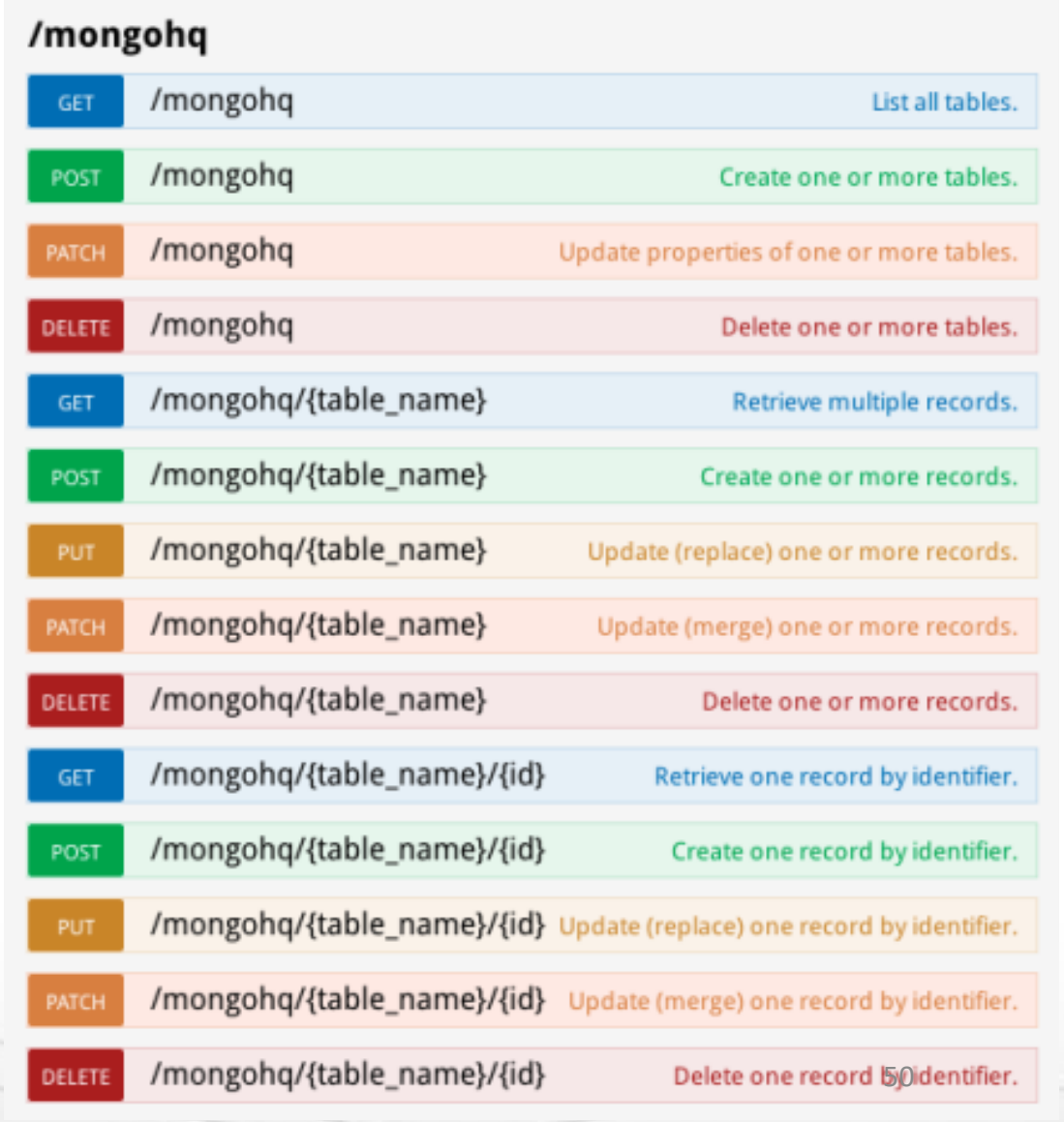
- Status dan fungsi dari aplikasi termasuk sumber daya (*resources*)
  - Setiap sumber daya dikaitkan dengan URI yang unik
  - Setiap sumber daya mendukung operasi-operasi standard (CRUD)
- Secara natif memetakan ke protokol HTTP
  - Metode HTTP: **GET**, **POST**, **PUT**, **DELETE**, **PATCH**, **OPTIONS**, …

Operasi CRUD dalam REST API

URL	HTTP Verb	POST Body	Result
http://yourdomain.com/api/entries	GET	empty	Returns all entries
http://yourdomain.com/api/entries	POST	JSON String	New entry Created
http://yourdomain.com/api/entries/:id	GET	empty	Returns single entry
http://yourdomain.com/api/entries/:id	PUT	JSON string	Updates an existing entry
http://yourdomain.com/api/entries/:id	DELETE	empty	Deletes existing entry

RESTful Web Services & Metode HTTP

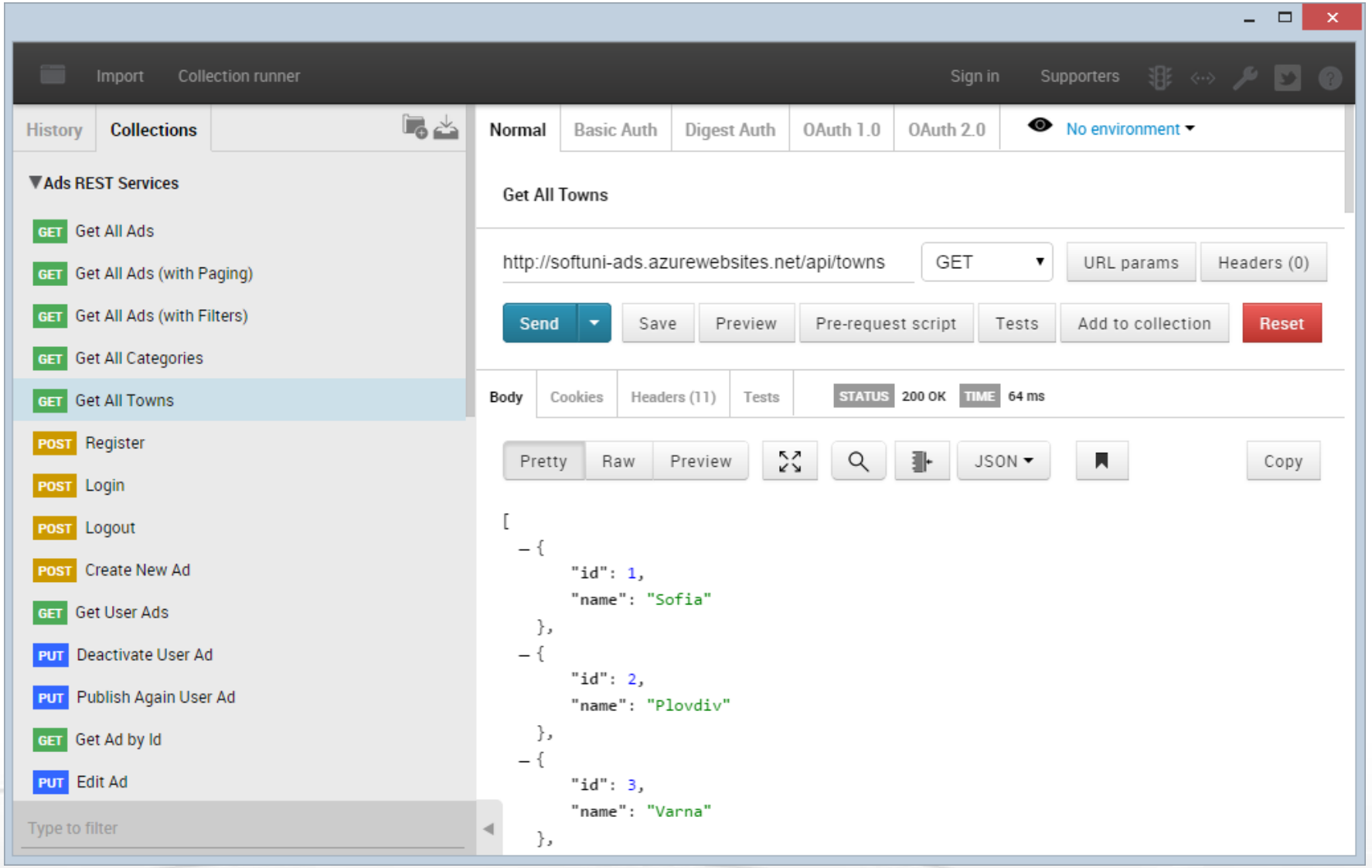
- Satu URI per sumber daya
  - Banyak operasi per URI
- Dapatkan semua/satu sumber daya berdasarkan ID-nya
  - GET http://myservice.com/api/Books
  - GET http://myservice.com/api/Books/3
- Tambahkan sumber daya baru
  - POST http://myservice.com/api/Books
- Modifikasi (update) sumber daya
  - PUT http://myservice.com/api/Books/3

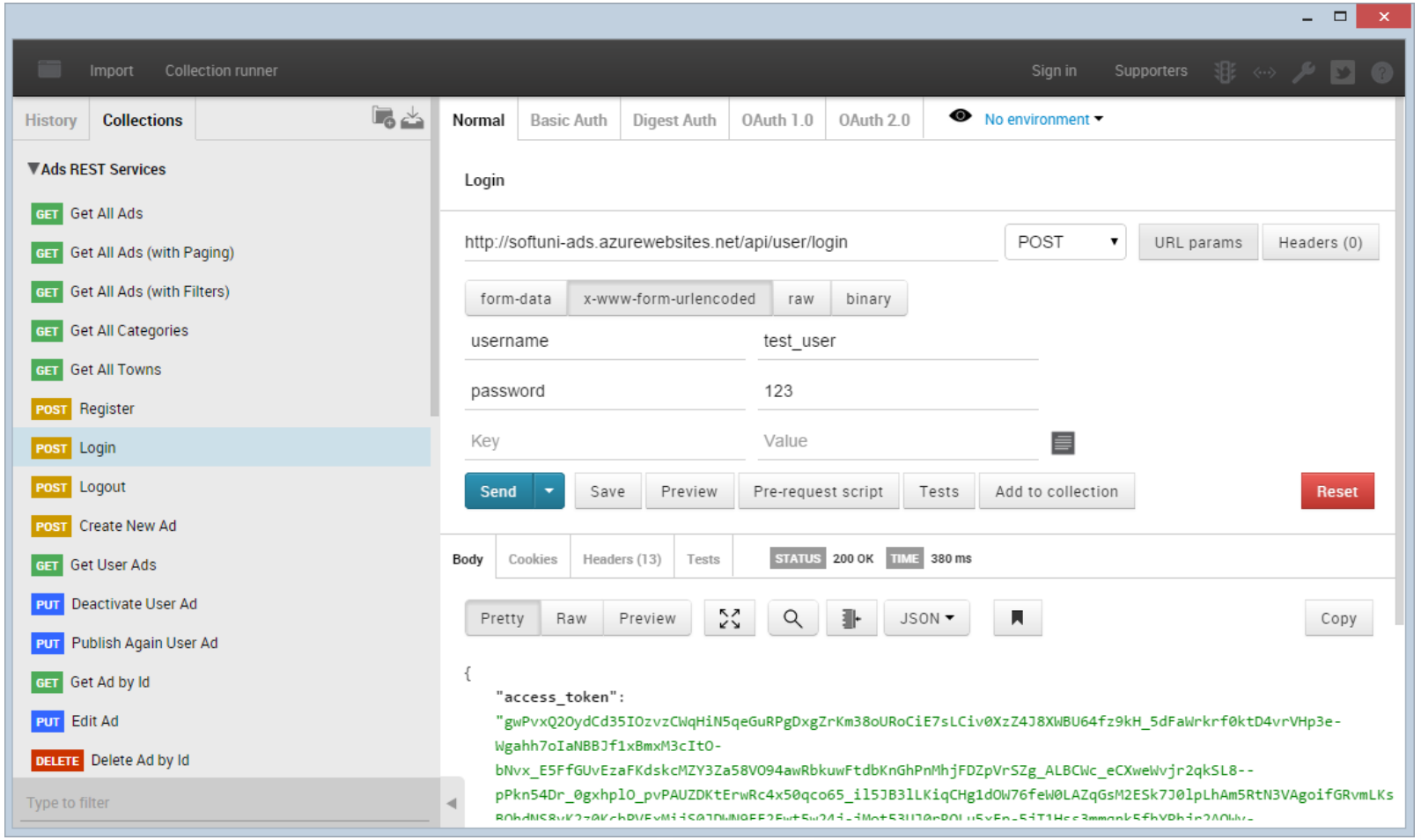


RESTful Web Services & Metode HTTP (2)

- Hapus (delete) sumber daya
  - DELETE http://myservice.com/api/Books/3
- Update bagian dari sumber daya (update parsial)
  - PATCH http://myservice.com/api/Books/3
- Me-retrieve metadata sumber daya
  - HEAD http://myservice.com/api/Books/3
- Memeriksa sumber daya (biasa digunakan dalam AJAX untuk meminta perijinan)
  - OPTIONS http://myservice.com/api/Books/3

Postman: Client REST di Web Browser

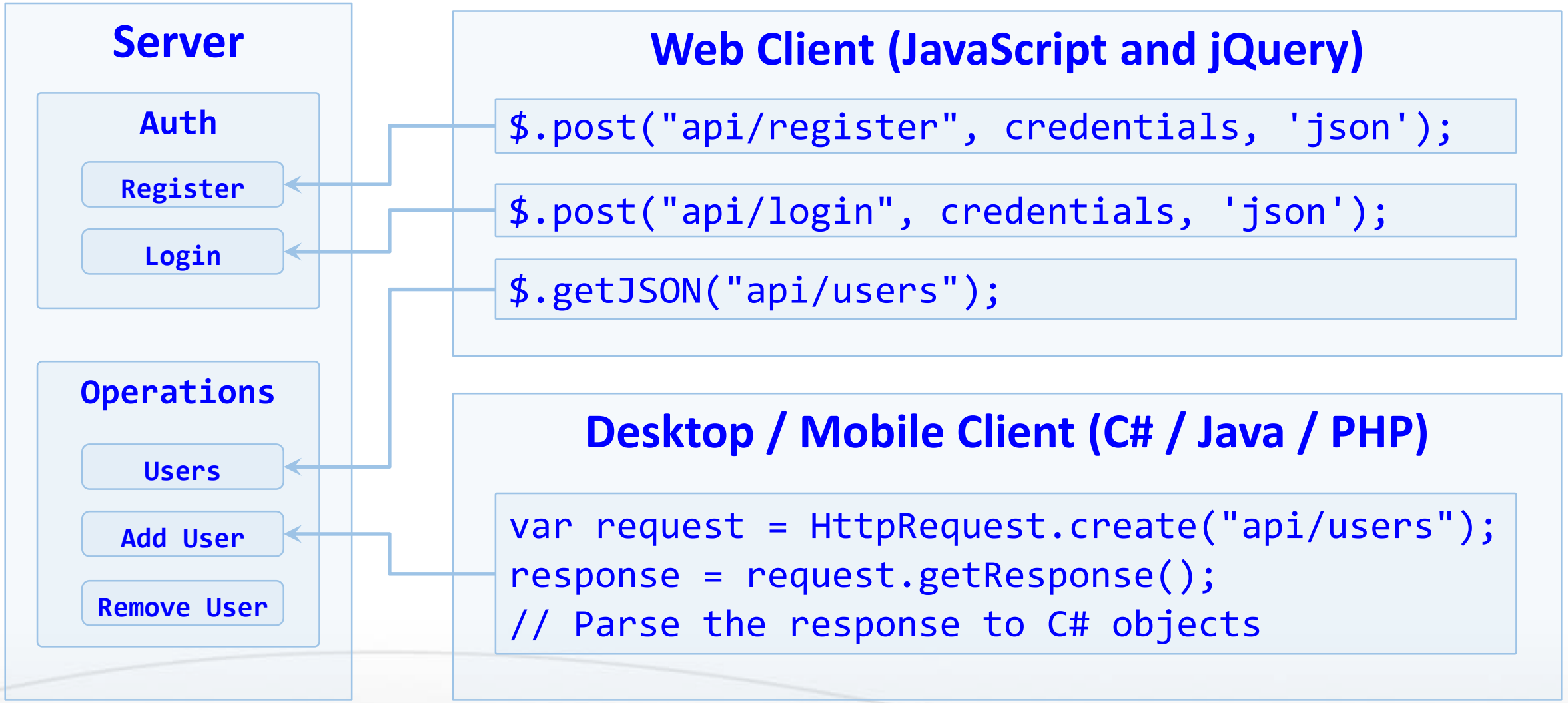




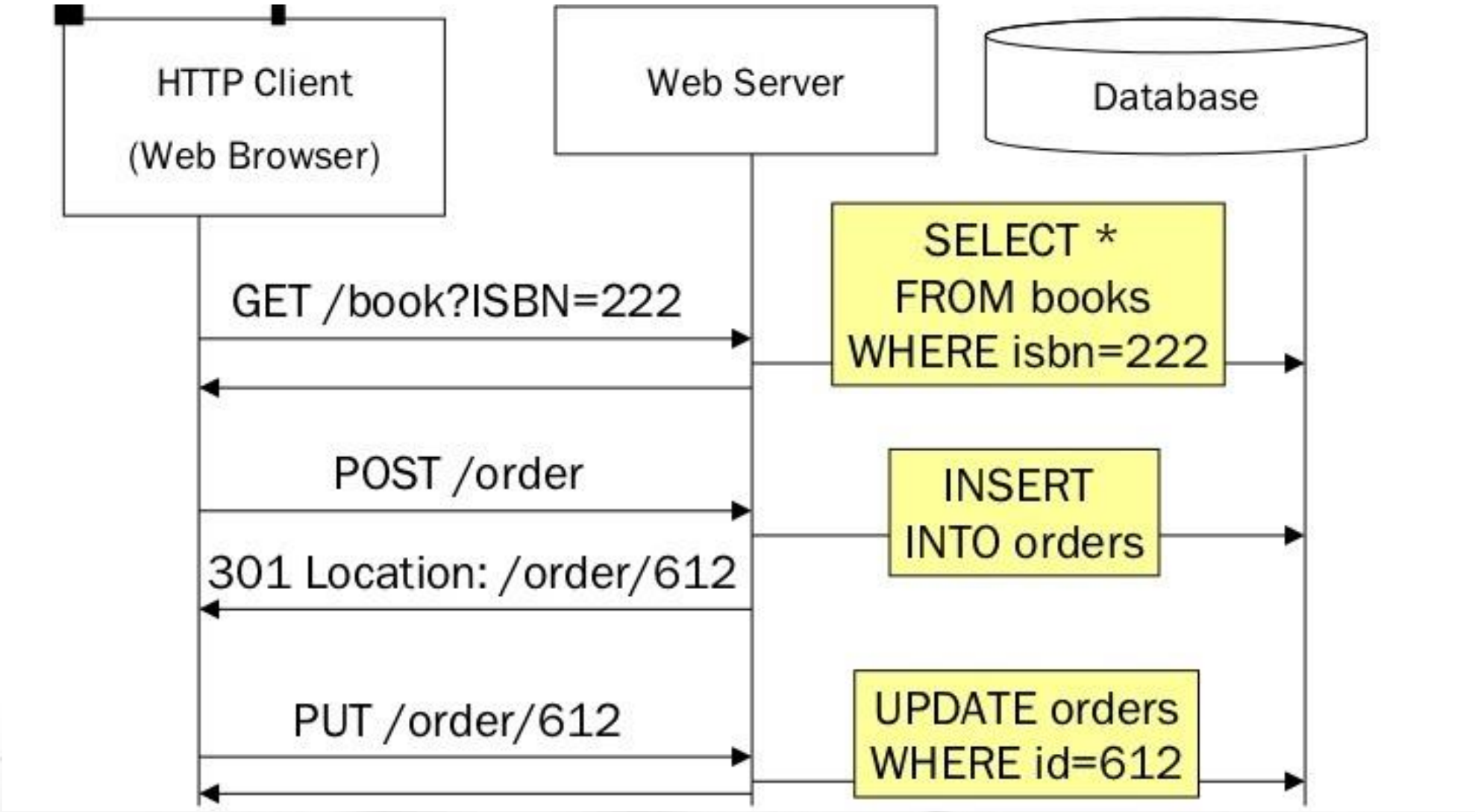
Postman

Live Demo

RESTful API: Contoh



Kemiripan REST API dan Query SQL



XML, JSON, RSS, Atom

Perbandingan Format Data Service Umum

XML

- XML adalah bahasa *markup* untuk representasi data
  - Digunakan untuk encoding dokumen dalam bentuk *machine-readable*
  - Format berbasis teks, terdiri dari *tags*, *attributes* dan *content*
  - Menyediakan data dan metadata dalam waktu yang sama

```
<?xml version="1.0"?>
<library>
  <book><title>HTML 5</title><author>Bay Ivan</author></book>
  <book><title>WPF 4</title><author>Microsoft</author></book>
  <book><title>WCF 4</title><author>Kaka Mara</author></book>
  <book><title>UML 2.0</title><author>Bay Ali</author></book>
</library>
```

JSON

- JSON (**J**ava**S**cript **O**bject **N**otation)
  - Standard untuk representasi struktur data dan larik asosiatif
  - Standard terbuka berbasis teks kelas ringan
  - Diturunkan dari bahasa JavaScript

```
{
  "firstName": "John", "lastName": "Smith", "age": 25,
  "address": { "streetAddress": "17 Tintyava Str.",
     "city": "Sofia", "postalCode": "1113" },
  "phoneNumber": [{ "type": "home", "number": "212 555-1234"},
    { "type": "fax", "number": "646 555-4567" }]
},
{ "firstName": "Bay", "lastName": "Ivan", "age": 79 }
```

RSS / Atom

- RSS (**R**eally **S**imple **S**yndication)
  - Famili dari format feed Web untuk mengakses publikasi situs
    - Mis. Entri di blog, headline berita, video, dll.
  - Berbasis pada XML, dengan skema XSD terstandard
- Dokumen RSS (*feeds*) merupakan daftar item (*list of items*)
  - Masing-masing mengandung title, author, publish date, summarized text dan metadata
- Protokol Atom bertujuan untuk meningkatkan RSS & memungkinkan publikasi.

RSS: Contoh

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0">
<channel>
  <title>W3Schools Home Page</title>
  <link>http://www.w3schools.com</link>
  <description>Free web building tutorials</description>
  <item>
    <title>RSS Tutorial</title>
    <link>http://www.w3schools.com/rss</link>
    <description>New RSS tutorial on W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>http://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
</channel>
</rss>
```

Bahasan Selanjutnya

- Konkurensi dan Sinkronisasi
  - Menjelaskan perlunya sinkronisasi
  - Menganalisa bagaimana komputer mensinkronkan jam dan mengakses sumber daya
    - Algoritma sinkronisasi Jam (clock synch)
    - Algoritma Ekslusi Mutual (*mutual exclusion*)