

Pemrograman Jaringan 5

anton@ukdw.ac.id

HTTP

- Hypertext Transport Protocol (RFC 1945)
- Language of the Web
 - Protocol yang digunakan untuk komunikasi antara web browsers dan web servers
 - Since 1990
- TCP port 80
- Penyempurnaan HTTP 1.0 menjadi versi 1.1 dispesifikasikan oleh IETF dengan RFC 2616
- HTTP merupakan implementasi dari protokol TCP
- Bersifat “Stateless”
 - Tidak ada informasi yang disimpan

HTTP (2)

- HTTP bersifat request – response:
 - HTTP client (user agent misalnya) mengirimkan permintaan (request) ke HTTP server dan server meresponse sesuai request tersebut
- User agent: Mozilla, Netscape, Microsoft Internet Explorer atau browser berbasis teks, Lynx atau links
- Perbedaan mendasar antara HTTP/1.1 dengan HTTP/1.0 adalah penggunaan hubungan persistent.
- HTTP/1.0 membuka satu koneksi untuk tiap permintaan URI
 - header = Connection: close
- HTTP/1.1 dapat menggunakan sebuah koneksi TCP untuk beberapa permintaan URI (persistent)
 - header = Connection: Keep-Alive
 - kecuali jika client menyatakan tidak hendak menggunakan hubungan persistent (header = Connection: close).

HTTP - methods

- Methods
 - GET
 - retrieve a URL from the server
 - simple page request, limited
 - run a CGI program
 - run a CGI with arguments attached to the URL
 - POST
 - preferred method for forms processing
 - run a CGI program
 - parameterized data in sysin
 - more secure and private

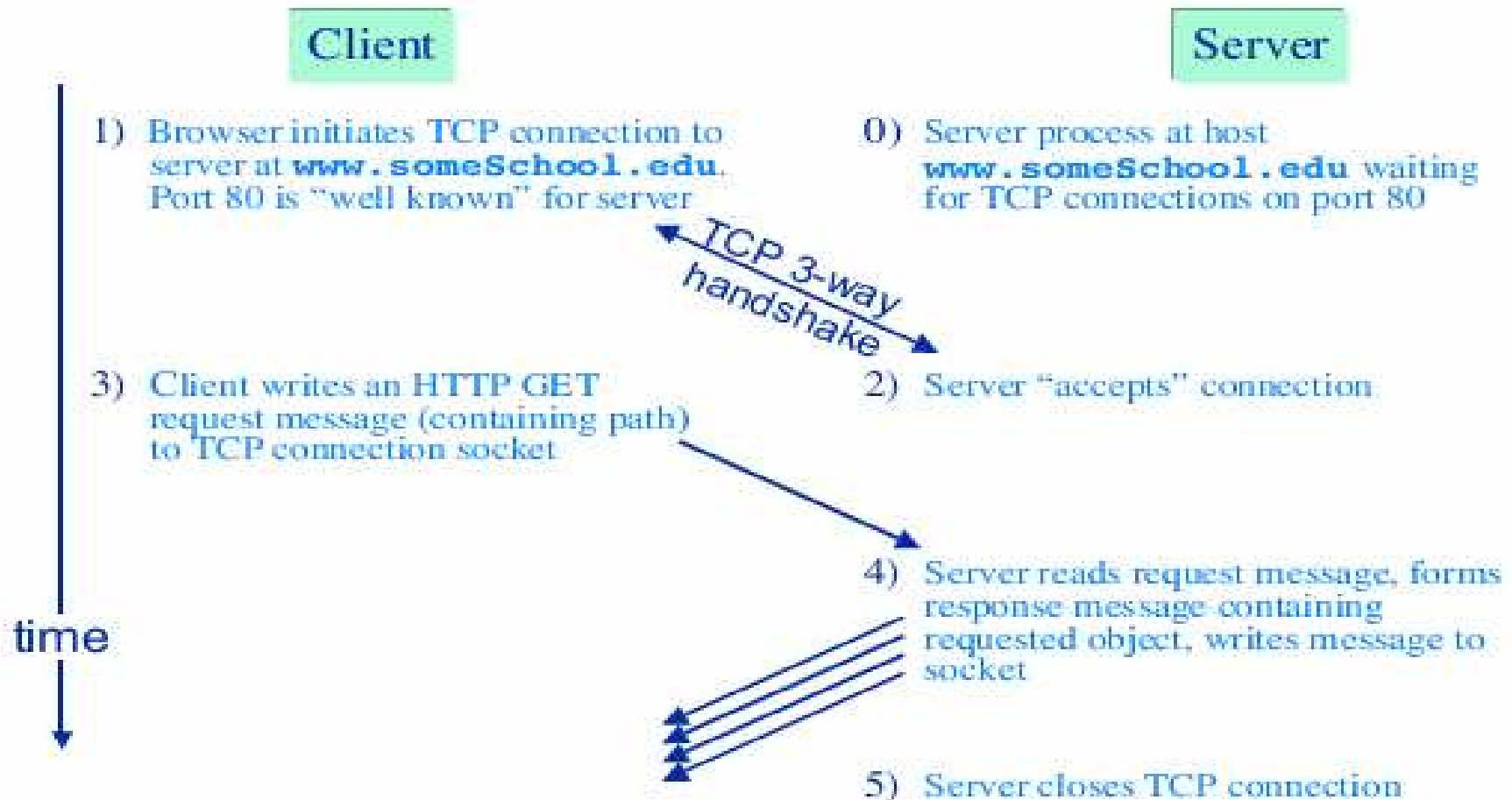
HTTP - methods

- Methods (cont.)

- HEAD

- requests URLs status header only
 - used for conditional URL handling for performance enhancement schemes
 - retrieve URL only if not in local cache or date is more recent than cached copy

URL `www.someschool.edu/someDept/home.index`



HTTP method syntax

- ◆ Two types of HTTP message formats: *request*, *response* messages
 - » ASCII (human-readable format)
- ◆ HTTP request message:

» Request line

```
method <SP> path <SP> version <CR><LF>
```

```
header field name ":" value <CR><LF>
```

» Optional header lines

```
⋮
```

```
header field name ":" value <CR><LF>
```

» Present only for some methods (e.g., POST)

```
<CR><LF>
```

```
entity body
```

Method: Head

- **Gunakan Telnet**

```
telnet> open localhost 80
```

Maka hasilnya adalah:

```
telnet> open localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escaped character is '^]'.
HEAD / HTTP/1.1 <enter>
Host: localhost <enter>
Connection: close <enter>
<enter>
<enter>
```

```
HEAD / HTTP/1.1 <enter>
Host: localhost <enter>
Connection: Keep-Alive <enter>
<enter>
<enter>
```


Hasil

```
[antonie@localhost ~]$ telnet
telnet> open 192.168.1.2 80
Trying 192.168.1.2...
Connected to 192.168.1.2 (192.168.1.2).
Escape character is '^]'.

```

```
HTTP/1.1 200 OK
Date: Mon, 12 Sep 2005 14:18:29 GMT
Server: Apache/2.0.54 (Fedora)
Accept-Ranges: bytes
Content-Length: 3931
Connection: close
Content-Type: text/html; charset=UTF-8

```

```
Connection closed by foreign host

```

Method: GET

- Buat file cobaget.php

```
<HTML>
```

```
<HEAD><TITLE>Coba GET</HEAD>
```

```
</HTML>
```

```
<BODY>
```

```
Percobaan nilai NIM adalah = <? echo $nim; ?>
```

```
Percobaan nilai nama adalah = <? echo $nama; ?>
```

```
</BODY>
```

GET

- **Ketik perintah berikut:**

```
GET cobaget.php?nim=22002529&nama=anton HTTP/1.1 <enter>
```

```
Host: localhost <enter>
```

```
Connection: close <enter>
```

```
<enter>
```

```
<enter>
```

Hasil GET

```
[antonie@localhost ~]$ telnet 192.168.1.2 80
Trying 192.168.1.2...
Connected to 192.168.1.2 (192.168.1.2).
Escape character is '^]'.
GET /cobaget.php?nim=22002529&nama=anton HTTP/1.1
Host: 192.168.1.2
Connection: close

HTTP/1.1 200 OK
Date: Mon, 12 Sep 2005 14:28:57 GMT
Server: Apache/2.0.54 (Fedora)
X-Powered-By: PHP/5.0.4
Content-Length: 120
Connection: close
Content-Type: text/html; charset=UTF-8

<HTML>
<HEAD><TITLE>Coba GET</HEAD>
</HTML>
<BODY>
Percobaan nilai NIM adalah = 2202529Percobaan nilai nama adalah =
anton</BODY>
Connection closed by foreign host.
```

Method : POST

```
POST cobapost.php HTTP/1.1 <enter>
```

```
Host: localhost<enter>
```

```
Content-Type: application/x-www-form-urlencoded <enter>
```

```
Content-Length: 16 <enter>
```

```
<enter>
```

```
nim=22002529&nama=anton <enter>
```

```
<enter>
```

```
<enter>
```

Hasil POST

```
[antonie@localhost ~]$ telnet 192.168.1.2 80
Trying 192.168.1.2...
Connected to 192.168.1.2 (192.168.1.2).
Escape character is '^]'.
POST /cobapost.php HTTP/1.1
Host: 192.168.1.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 16
```

```
nim=22002529&nama=anton
```

```
HTTP/1.1 200 OK
Date: Mon, 12 Sep 2005 14:30:28 GMT
Server: Apache/2.0.54 (Fedora)
X-Powered-By: PHP/5.0.4
Content-Length: 124
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
<HTML>
<HEAD><TITLE>Coba POST</HEAD>
</HTML>
<BODY>
Percobaan nilai NIM adalah = 22002529Percobaan nilai nama adalah =
anton

</BODY>
Connection closed by foreign host.
```

HTTP Status Codes

- 200 OK
- 201 created
- 202 accepted
- 204 no content
- 301 moved perm.
- 302 moved temp
- 304 not modified
- 400 bad request
- 401 unauthorized
- 403 forbidden
- 404 not found
- 500 int. server error
- 501 not impl.
- 502 bad gateway
- 503 svc not avail

HTTP - URLs

- URL

- Uniform Resource Locator

- protocol (http, ftp, news)
 - host name (name.domain name)
 - port (usually 80)
 - directory path to the resource
 - resource name

- <http://www.myplace.com/www/saya%20makan.html>

- <http://www.myplace.com:80/cgi-bin/t.exe?a=1&b=2>

Java URL class

- java.net.URL
- public final class URL extends Object implements Serializable
- Konstruktor
 - URL(String spec)
 - URL(["http://localhost:8081/coba.asp?a=1&b=2"](http://localhost:8081/coba.asp?a=1&b=2));
 - URL(String protocol, String host, int port, String file)
 - URL("http", "localhost", 8081, "coba.php");
 - URL(String protocol, String host, String file)
 - URL("http", "localhost", "coba.php");

URL class method

<code>String getAuthority()</code>	mengambil bagian authority dari URL
<code>int getDefaultPort()</code>	mengambil port default dari protokol URL
<code>String getFile()</code>	mengambil nama file dari URL
<code>String getHost()</code>	mengambil nama host komputer dari URL
<code>String getPath()</code>	mengambil nama path resource dari URL
<code>int getPort()</code>	mengambil port dari URL
<code>String getProtocol()</code>	mengambil protokol dari URL
<code>String getQuery()</code>	mengambil query string dari URL
<code>String getRef()</code>	mengambil referensi atau anchor dari URL
<code>String toExternalForm</code>	mengembalikan URL lengkap

`Object getContent()`, mengambil konten dari sebuah object

boolean `sameFile(URL other)`, membandingkan dua buah URL

Exception

- All these constructors throw a **MalformedURLException** if you try to create a URL for an unsupported protocol
- May throw a **MalformedURLException** if the URL is syntactically incorrect.

```
try {
    URL u = new URL("http://www.audubon.org/");
}
catch (MalformedURLException ex) {
    System.err.println(ex);
}

try {
    URL u = new URL("http", "fourier.dur.ac.uk", 8000, "~/dma3mjh/jsci/");
}
catch (MalformedURLException ex) {
    System.err.println(ex);
}

try {
    URL u1 = new URL("http://www.ibiblio.org/javafaq/index.html");
    URL u2 = new URL (u1, "mailinglists.html");
}
catch (MalformedURLException ex) {
    System.err.println(ex);
}
```

Contoh

```
import java.net.*;
```

```
public class myURL
```

```
{
```

```
    public static void main (String args[])  
        throws Exception
```

```
{
```

```
    URL hp = new  
    URL ("http://localhost/index.php#arc?a=1&b=2");
```

```
    System.out.println ("Protocol: " + hp.getProtocol());  
    System.out.println ("Host :" + hp.getHost());  
    System.out.println ("Default Port :" + hp.getDefaultPort());  
    System.out.println ("Port :" + hp.getPort());  
    System.out.println ("File :" + hp.getFile());  
    System.out.println ("Query :" + hp.getQuery());  
    System.out.println ("Ref :" + hp.getRef());  
    System.out.println ("Full :" + hp.toExternalForm());
```

```
}
```

```
}
```

```
java myURL  
java myURL  
Protocol: http  
Host :localhost  
Default Port :80  
Port :-1  
File :/index.php  
Query :null  
Ref :arc?a=1&b=2  
Full :http://localhost/index.php#arc?a=1&b=2
```

Contoh-contoh

- Lihat contoh ProtocolTester.java
- Lihat contoh URLSplitter.java
- Lihat contoh SourceViewer.java
- Lihat contoh ContentGetter.java
- Lihat contoh URISplitter.java

Class URLEncoder

Untuk mengubah string URL menjadi bentuk format URL standard

```
import java.net.*;
public class myEncURL
{
    public static void main (String args[])
        throws Exception
    {
        String prmHari = URLEncoder.encode (
            "senin jam 07:30" );
        try {
            URL hp = new
                URL ("http", "localhost",
                    "/daftar.php?hari=" + prmHari);
            System.out.println ("File :" + hp.getFile() );
            System.out.println ("Query :" + hp.getQuery() );
            System.out.println ("Full :" +
                hp.toExternalForm());
        }
        catch (MalformedURLException mURLe) {
            System.out.println("Error: " + mURLe);
        }
    }
}
```

Hasil:
File :/daftar.php?hari=senin+jam+07%3A30
Query :hari=senin+jam+07%3A30
Full :http://localhost/daftar.php?hari=senin+jam+07%3A30

URLDecoder

```
import java.net.*;

public class URLDec
{
    public static void main (String args[])
        throws Exception
    {
        String prmHari = URLDecoder.decode (
            "senin+jam+07%3A30" );
        System.out.println("Setelah didecode = "+prmHari);
    }
}
```

Hasil Output:

Setelah didecode = senin jam 07:30

Contoh

```
import java.net.*;
import java.io.*;

public class CobaURL {
    public static void main(String[] args)
        throws Exception {
        String nim = URLEncoder.encode("22 00 2529");
        String nama =
            URLEncoder.encode("Antonius Rachmat C");
        URL urlLokal = new URL("http", "localhost",
            "/cobaget.php?nim=" + nim + "&nama=" + nama);
        System.out.println("URL yang diminta : \n" +
            urlLokal.toExternalForm());
        BufferedReader in = new BufferedReader(
            new InputStreamReader(
                urlLokal.openStream()));

        String inputLine;

        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);

        in.close();
    }
}
```

Hasil

URL yang diminta :

```
http://localhost/cobaget.php?nim=22+00+2529&nama=Antonius+Rachmat+C
```

```
<HTML>
```

```
<HEAD><TITLE>Coba GET</TITLE></HEAD>
```

```
<BODY>
```

```
Percobaan nilai NIM adalah = 22 00 2529<br>
```

```
Percobaan nilai nama adalah = Antonius Rachmat C</BODY>
```

```
</HTML>
```

Dari contoh, terlihat bahwa untuk dapat memperlakukan data output dari `URLopenStream()` sebagai deretan karakter, maka kita harus membuat obyek dari `BufferedReader` sehingga kita dapat mengambil data baris demi baris.

URLConnection

- Kelas ini merupakan kelas yang dapat melakukan koneksi secara langsung ke alamat URL yang diinginkan
- Kelas URLConnection adalah kelas abstrak, jadi kita harus membuat kelas turunan dari kelas URLConnection, kemudian memanggil method `openConnection()`

```
try{
    URL URLanda = new URL("urlanda");
    URLConnection obyekKoneksi = URLanda.openConnection();
catch (MalformedURLException e) {
    ...
}
catch (IOException e) {
    ...
}
```

URLConnection Method

getContentEncoding ()

Returns the value of the content-encoding header field

getContentLength ()

Returns the value of the content-length header field.

getContentType ()

Returns the value of the content-type header field.

getDate ()

Returns the value of the date header field.

getLastModified ()

Returns the value of the last-modified header field.

getExpiration ()

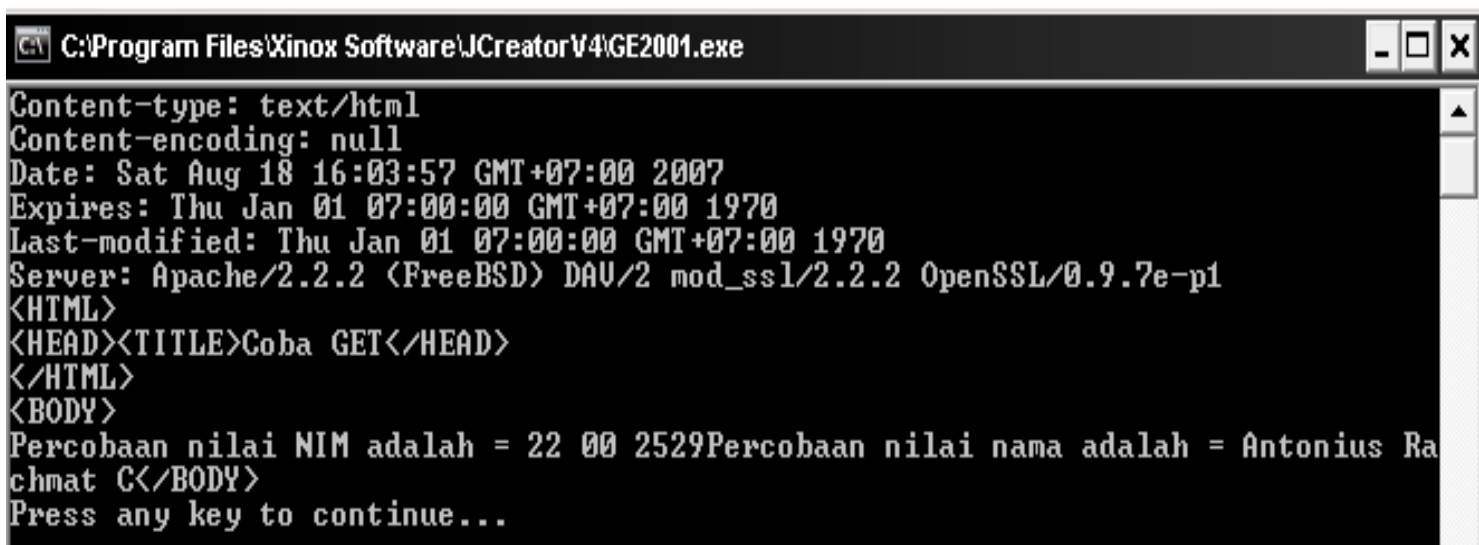
Returns the value of the expires header field.

Request GET

- Contoh request GET

- URLConnGET

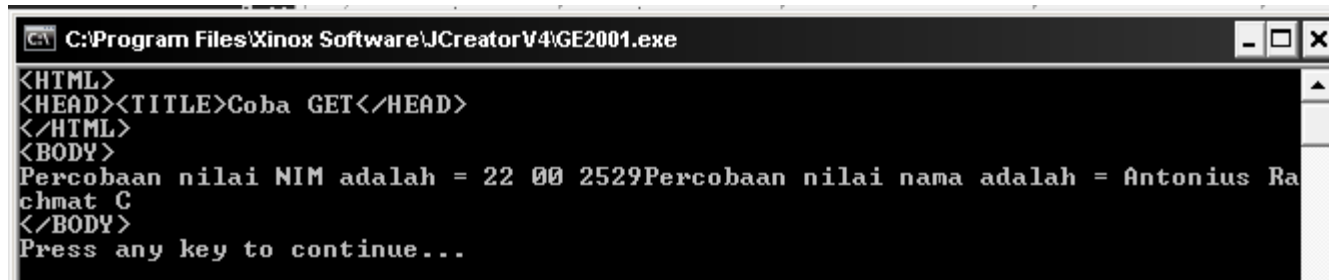
Buka koneksi dengan menggunakan `openConnection()` dan pada saat akan menerima Stream dari server gunakan `getInputStream()`.



```
C:\Program Files\Xinox Software\JCreatorV4\GE2001.exe
Content-type: text/html
Content-encoding: null
Date: Sat Aug 18 16:03:57 GMT+07:00 2007
Expires: Thu Jan 01 07:00:00 GMT+07:00 1970
Last-modified: Thu Jan 01 07:00:00 GMT+07:00 1970
Server: Apache/2.2.2 (FreeBSD) DAV/2 mod_ssl/2.2.2 OpenSSL/0.9.7e-p1
<HTML>
<HEAD><TITLE>Coba GET</HEAD>
</HTML>
<BODY>
Percobaan nilai NIM adalah = 22 00 2529Percobaan nilai nama adalah = Antonius Ra
chmat C</BODY>
Press any key to continue...
```

Request POST

- Buat URL yang diinginkan
- Buka koneksi URL yang diinginkan tersebut
- setDoOutput() true
- Ambil OutputStream dari koneksi, agar dapat mengirimkan data ke server
- Tulis data ke OutputStream
- Tutup OutputStream
- Lihat Contoh URLConnPost



```
<HTML>
<HEAD><TITLE>Coba GET</HEAD>
</HTML>
<BODY>
Percobaan nilai NIM adalah = 22 00 2529Percobaan nilai nama adalah = Antonius Ra
chmat C
</BODY>
Press any key to continue...
```

HTTPURLConnection

- HttpURLConnection yang merupakan kelas turunan dari kelas URLConnection
- Method GET
 - Buat URLConnection
 - Buat koneksi dari URLConnection yang ada dan dicasting menjadi HttpURLConnection, panggil method openConnection()
 - setRequestMethod() menjadi GET
 - Periksa getResponseCode(), jika error maka Exit
 - Jika tidak maka baca baris demi baris keluaran dari server, dengan menggunakan getInputStream()

HTTPURLConnection

- Method POST
 - Buat URLConnection
 - Buka koneksi dari URLConnection yang dibuat dengan HttpURLConnection
 - setRequestMethod() POST
 - setDoOutput() TRUE karena kita akan menulis ke Server 5. Buat DataOutputStream() dari getOutputStream()
 - Tulis ke Server dengan menggunakan method writeBytes()
 - Tutup DataOutputStream
 - Baca output dari Server baris demi baris melalui getInputStream()