

WORKSHOP PEMROGRAMAN JARINGAN

MODUL 11

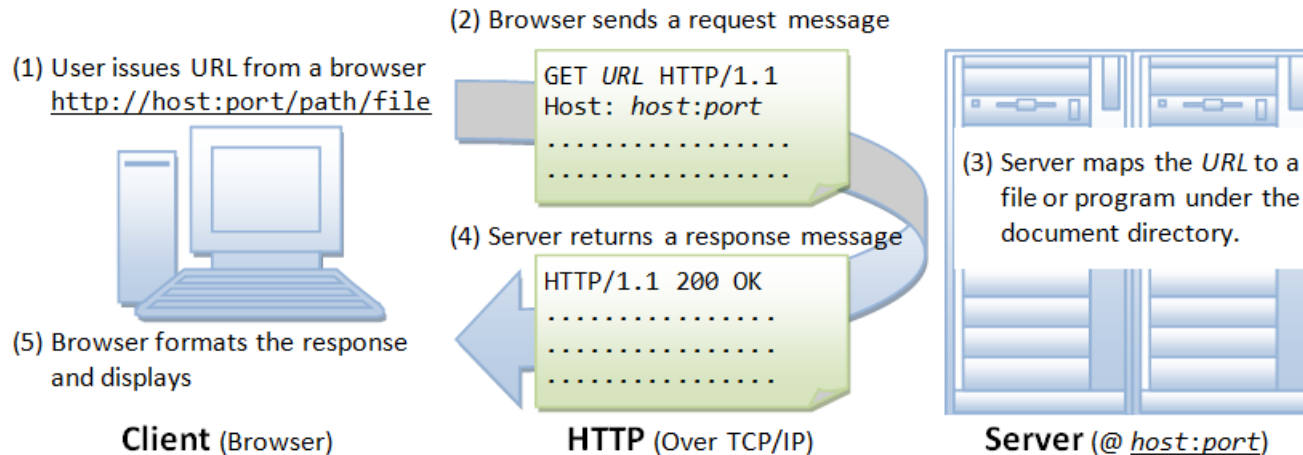
(PROGRAMMING HTTP)

Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

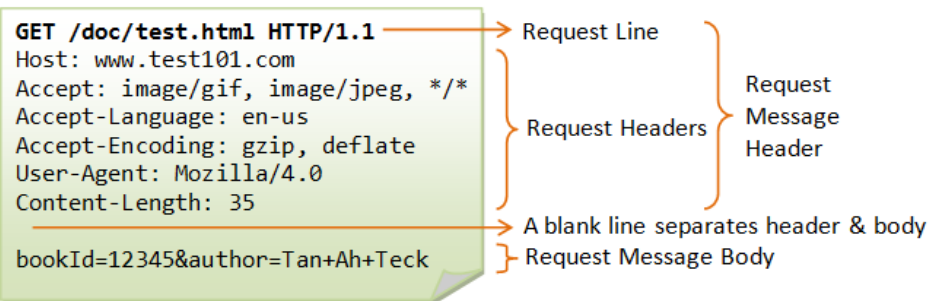
TOPIK PEMBAHASAN

- Download data dari HTTP Server
- Melayani permintaan akses HTTP
- Submit web form

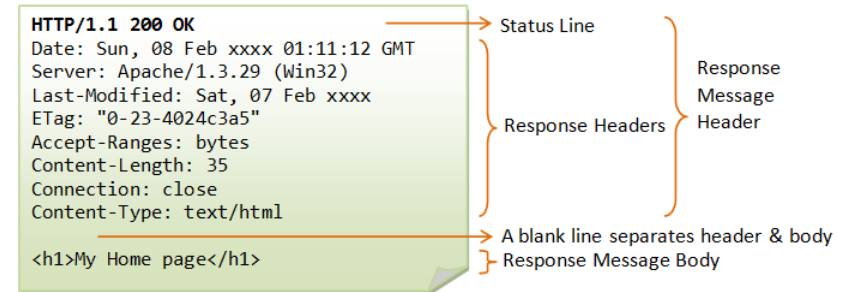
HTTP Server



HTTP Request Message



HTTP Response Message



Download data dari HTTP Server

- Menggunakan HTTP protokol: request dan response
- Instalasi paket:
 - # pip2 install urllib atau pip3 install urllib3
- Hasil listing 4.1

```
$ python 4_1_download_data.py --host=http://www.python.org
<!doctype html>
<!--[if lt IE 7]> <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9">
<![endif]-->
<!--[if IE 7]> <html class="no-js ie7 lt-ie8 lt-ie9">
<![endif]-->
<!--[if IE 8]> <html class="no-js ie8 lt-ie9">
<![endif]-->
<!--[if gt IE 8]><!--><html class="no-js" lang="en" dir="ltr"> <!--
<![endif]-->

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/
    jquery/1.8.2/jquery.min.js">

  <meta name="application-name" content="Python.org">
```

Program

```
import urllib.request
class HTTPClient:

    def __init__(self, host):
        self.host = host

    def fetch(self):
        response = urllib.request.urlopen(self.host)
        # Comment out the above line and uncomment the below for
        # Python 2.7.x.
        #response = urllib2.urlopen(self.host)

        data = response.read()
        text = data.decode('utf-8')
        return text

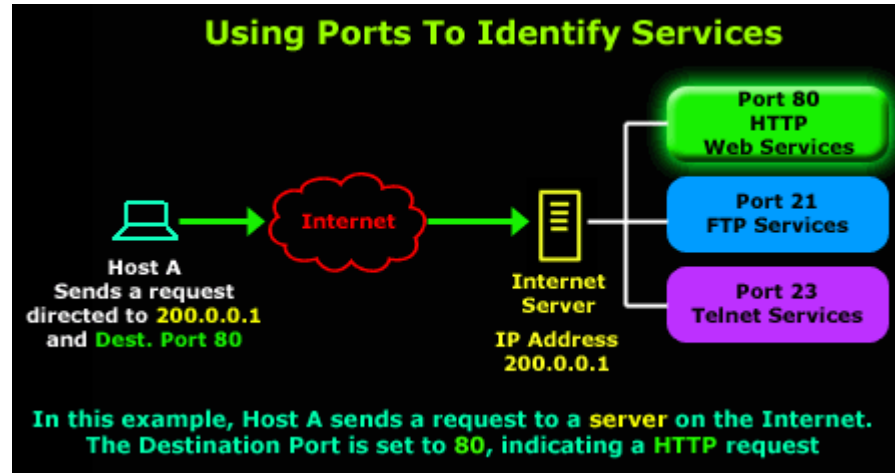
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='HTTP Client Example')
    parser.add_argument('--host', action="store",
                        dest="host", default=REMOTE_SERVER_HOST)

    given_args = parser.parse_args()
    host = given_args.host
    client = HTTPClient(host)
    print (client.fetch())
```

HTTP Request message

HTTP Response message

Membuat layanan HTTP Server



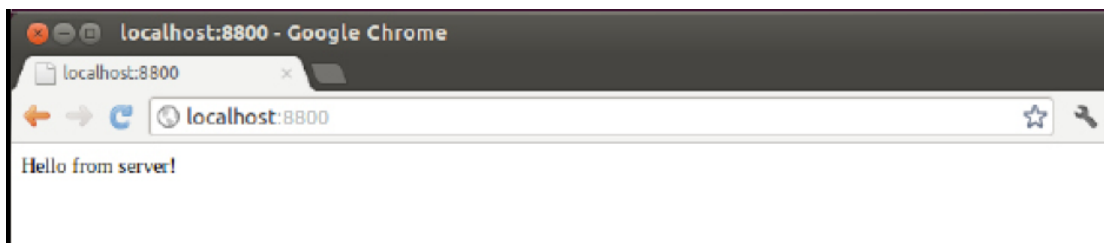
index.html

- Sistem komunikasi client-server pada http
- Menggunakan **CustomHTTPServer** class utk aplikasi server
- Hasil listing 4.2

Server

```
$ python 4_2_simple_http_server.py --port=8800
Custom HTTP server started on port: 8800
localhost - - [18/Apr/2013 13:39:33] "GET / HTTP/1.1" 200 -
localhost - - [18/Apr/2013 13:39:33] "GET /favicon.ico HTTP/1.1" 200
```

Client



Program

```
class RequestHandler(BaseHTTPRequestHandler):
```

```
    """ Custom request handler """
```

```
    def do_GET(self):
```

```
        """ Handler for the GET requests """
```

```
        self.send_response(200)
```

```
        self.send_header('Content-type', 'text/html')
```

```
        self.end_headers()
```

```
        # Send the message to browser
```

```
        self.wfile.write("Hello from server!")
```

```
        return
```

```
class CustomHTTPServer(HTTPServer):
```

```
    "A custom HTTP server"
```

```
    def __init__(self, host, port):
```

```
        server_address = (host, port)
```

```
        HTTPServer.__init__(self, server_address, RequestHandler)
```

```
def run_server(port):
```

```
    try:
```

```
        server= CustomHTTPServer(DEFAULT_HOST, port)
```

```
        print ("Custom HTTP server started on port: %s" % port)
```

```
        server.serve_forever()
```

```
    except Exception as err:
```

```
        print ("Error:%s" %err)
```

```
    except KeyboardInterrupt:
```

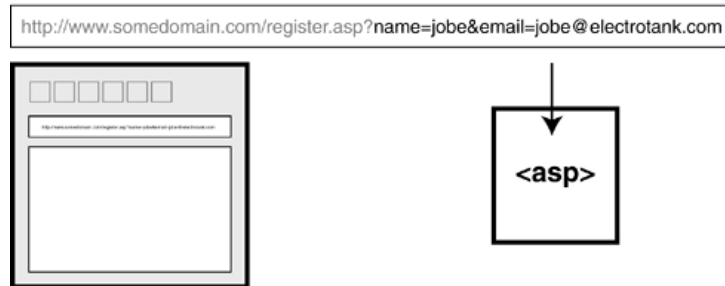
```
        print ("Server interrupted and is shutting down...")
```

```
        server.socket.close()
```

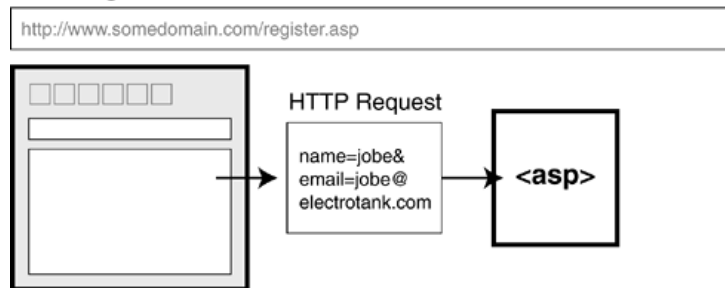
Error, harus di-encode dulu:
`self.wfile.write("Hello from server!".encode())`

Submit web forms

Using GET



Using POST



- Untuk mengirim data ke sisi server
- Menggunakan: GET dan POST
- Silahkan dicoba listing 4.4

Program

```
ID_USERNAME = 'signup-user-name'
ID_EMAIL = 'signup-user-email'
ID_PASSWORD = 'signup-user-password'
USERNAME = 'username'
EMAIL = 'you@email.com'
PASSWORD = 'yourpassword'

SIGNUP_URL = 'https://twitter.com/account/create'
```

```
def submit_form():
    """Submit a form"""
    payload = {ID_USERNAME : USERNAME,
               ID_EMAIL    : EMAIL,
               ID_PASSWORD : PASSWORD,}

    # make a get request
    resp = requests.get(SIGNUP_URL)
    print ("Response to GET request: %s" %resp.content)
    # send POST request
    resp = requests.post(SIGNUP_URL, payload)
    print ("Headers from a POST request response: %s" %resp.headers)

if __name__ == '__main__':
    submit_form()
```

Hasil

```
$ python 4_4_submit_web_form.py
Response to GET request: <?xml version="1.0" encoding="UTF-8"?>
<hash>
  <error>This method requires a POST.</error>
  <request>/account/create</request>
</hash>
Headers from a POST request response: {'status': '200 OK', 'content-
length': '21064', 'set-cookie': '_twitter_sess=BAh7CD--
d2865d40d1365eeb2175559dc5e6b99f64ea39ff; domain=.twitter.com;
path=/; HttpOnly', 'expires': 'Tue, 31 Mar 1981 05:00:00 GMT',
'vary': 'Accept-Encoding', 'last-modified': 'Sun, 05 May 2013
15:59:27 GMT', 'pragma': 'no-cache', 'date': 'Sun, 05 May 2013
```

Percobaan

- Cobalah listing 4.1 untuk download data dari server
- Cobalah listing 4.2 untuk membangun http server
- Cobalah listing 4.4 untuk mengirim data ke server
- Buatlah analisa, kesimpulan dari semua percobaan dan tugas