

 <p>UNIVERSITAS INDONESIA</p> <p><i>Veritas, Probitas, Justitia</i></p> <hr/> <p>FACULTY OF COMPUTER SCIENCE</p>	<p>Perancangan dan Pemrograman Web (CSF2600201)</p> <p>Semester Gasal 2016/2017</p> <p>Fakultas Ilmu Komputer – Universitas Indonesia</p> <p>Ujian Tengah Semester</p>
---	--

Tujuan	Menguji pemahaman konsep <i>client side web programming</i>
Deadline	24 Oktober 2016 pukul 20.00 waktu Scele
Pengumpulan dan Penamaan Berkas	Kumpulkan UTS Anda sebelum <i>deadline</i> pada slot pengumpulan di Scele dengan format file UTS_[NPM]_[Nama].zip/rar Contoh: UTS_12345_Hisyam.zip atau UTS_12345_Hisyam.rar

Soal

Anda diminta membuat aplikasi *web* sederhana tentang profil diri mahasiswa seperti CV *online*. Anda diberikan kebebasan berkreasi terkait *layout* dan tampilan aplikasi *web* tersebut. Pada aplikasi *web* tersebut, Anda diminta untuk membuat beberapa fitur: (1) *login & logout*, (2) Mengubah profil pengguna, dan (3) Menambah teman ke dalam koneksi pengguna.

Langkah pertama yang perlu dilakukan adalah mengunduh file *uts-ppw.zip* di Scele yang berisi *skeleton* dari web CV online dan pindahkan dalam *public directory* dari *web server* kalian (misal di dalam *htdocs*). Dalam *file* zip tersebut kalian akan mendapatkan sekumpulan file seperti berikut:

```

/--- uts-ppw
    /--- css
        mycv.css
        normalize.css
    /--- images
        photo.png
    /--- js
        jquery-3.1.0.min.js
    /--- services
        cv.php
        teman.txt
        user.txt
    home.html
    index.html

```

Jika tidak menggunakan css framework, bisa menggunakan struktur yang sudah ada dan terdapat beberapa file yang tidak perlu diubah, antara lain ***normalize.css***, ***jquery-3.1.0.min.js***, ***cv.php***, ***teman.txt***, dan ***user.txt***. Jika menggunakan css framework seperti *bootstrap* dan *materialize*, silahkan bisa disesuaikan struktur dan kebutuhan file pendukungnya (css, jquery, dsb); namun tetap tidak diperbolehkan mengubah file dalam folder ***services*** yaitu file ***cv.php***, ***teman.txt***, ***user.txt***. Boleh memindahkan file tersebut di direktori lain (misal root) jika diperlukan.

Tampilan dan *code* dari *home.html*, *index.html*, *mycv.css* hanya sebagai contoh dan ilustrasi. Silahkan diubah sesuai dengan kreativitas masing-masing. Kerjakan UTS PPW dengan menggunakan web server. Silahkan boleh menggunakan server kawung atau server localhost (dengan xampp). Kemudian lengkapi fitur dari web tersebut seperti ketentuan berikut.

1. Login & Logout

Pada fitur ini, pengguna (mahasiswa) memasukkan NPM (sebagai username) dan *password* (password sama dengan NPM) untuk bisa masuk ke web. Jika data sesuai dengan salah satu data yang ada di *server*, maka akan dibuatkan *session* sesuai pengguna tersebut dan pengguna bisa mengakses halaman utama yang berisi profil mahasiswa tersebut. Dengan menyimpan *session* menggunakan *web storage*, maka *user* tetap dalam status login walaupun halaman *web* berganti atau di-*refresh*. Jika gagal *login*, maka akan menampilkan pesan *error* "*username* tidak ditemukan atau *password* tidak sesuai". Tambahkan juga menu untuk *logout* ketika *user* sudah *login*. Ketika pengguna memilih *logout* atau menutup *browser*, maka *session* pengguna akan otomatis dihapus. Berikut adalah contoh tampilan login dan halaman utama.

Login

<input type="text"/>	
<input type="password"/>	<input type="button" value="Login"/>

Gambar 1. Contoh menu *login*


Fulan Fulanah
Faculty of Computer Science, Universitas Indonesia Building C Room 3307 UI Depok Campus, Indonesia, 16424 +62 21 786 3419 fulanfulanah@cs.ui.ac.id
Education
* 2012 - 2013 : Master of Computer Science at Universitas Indonesia (UI) Depok * 2007 - 2011 : Bachelor of Informatics at Institut Teknologi Sepuluh Nopember (ITS), Surabaya

Gambar 2 Contoh halaman utama

untuk melakukan login, gunakan AJAX untuk mengirimkan request ke server. Detil informasi mengenai pengiriman request adalah sebagai berikut:

- Alamat url: “../services/cv.php”
Dengan asumsi pengiriman *request* dilakukan dari *file* javascript yang berada dalam *folder* js. silahkan disesuaikan url-nya sesuai lokasi file pengirim *request*.
- Data yang dikirimkan: perintah=login, user=npm, dan password=password
variable perintah dengan *value* “login”.

variable user dengan *value* npm dari *input* yang dimasukkan.

variable password dengan *value* password dari *input* yang dimasukkan. Untuk *login* yang sukses, nilai *password* sama dengan *npm*.

- Kembalian
Data yang dikembalikan adalah *text* dengan struktur JSON. Jika login berhasil maka respon yang dikembalikan berupa *text* berikut:

```
{"status": "ok", "user": "npm", "nama": "namaMahasiswa"}
```

dan jika gagal maka server akan mengembalikan respon {"status": "gagal"}.

Contoh:

Contoh pengiriman *request* melalui *url*:

```
../services/cv.php?perintah=login&user=[NPM]&password=[password]
```

Contoh pengiriman *request* menggunakan jQuery:

URL: “../services/cv.php”

data: {"perintah": "login", "user": "NPM", "password": "password"}

method: “POST”

Parameter [NPM] dan [password] diisi dengan NPM masing-masing pengguna

Misal: ../services/cv.php?perintah=login&user=1234567&password=1234567

2. Mengubah profil pengguna

Pengguna yang berhasil *login* dapat memilih menu *update* untuk mengubah profil yang akan menampilkan *form* untuk menambahkan beberapa data diri pengguna. Pengguna mengisi beberapa data diri, diantaranya:

Nama (*required*), Alamat (*required*), Keahlian (*optional*, boleh lebih dari satu)

Berikan juga validasi pada *form* tersebut menggunakan Javascript/jQuery (sesuai dengan keterangan dalam tanda kurung di atas). Data profil pengguna yang diubah tidak perlu disimpan. Fitur ini tidak perlu mengirimkan request apa pun ke server.

3. Menambah teman

Pengguna dapat mencari profil pengguna lain dengan memasukkan nama atau NPM sebagai *keyword*. Kemudian sistem akan menampilkan mahasiswa yang sesuai dengan *keyword* yang diberikan. Dari daftar mahasiswa yang berhasil dikembalikan, pengguna bisa menggunakan tombol **“Connect”** untuk menambahkan mahasiswa tersebut sebagai temannya.

Untuk mencari teman, gunakan AJAX untuk mengirimkan request ke server. Detil informasi mengenai pengiriman request adalah sebagai berikut:

- Alamat url: **“../services/cv.php”**
Dengan asumsi pengiriman *request* dilakukan dari *file* javascript yang berada dalam *folder* js. silahkan disesuaikan url-nya sesuai lokasi file pengirim *request*.
- Data yang dikirimkan: perintah=cari dan nama=namaAtauNPMMahasiswaDicari
variable perintah dengan *value* “cari”

variable nama dengan *value* nama atau npm mahasiswa dari input yang dimasukkan
- Kembalian
Data yang dikembalikan adalah *text* dengan struktur JSON. Jika pengiriman *request* berhasil maka bentuk respon yang dikembalikan adalah seperti berikut.

```
{"users": [  
    {"user": "npmTeman", "nama": "namaTeman"},  
    {"user": "npmTeman", "nama": "namaTeman"},  
    {"user": "npmTeman", "nama": "namaTeman"},  
    ...  
    {"user": "npmTeman", "nama": "namaTeman"}  
]}
```

dan jika tidak ada data teman, maka *server* akan mengembalikan respon `{"users": []}`

Contoh:

Contoh pengiriman *request* melalui *url*:

```
../services/cv.php?perintah=cari&nama=[Nama]
```

Contoh pengiriman *request* menggunakan jQuery:

URL: **“../services/cv.php”**

data: **{“perintah”:“cari”, “nama”: [Nama] }**

method: **“POST”**

Parameter [Nama] diisi nama atau npm pengguna yang ingin dicari

Misal: `../services/cv.php?perintah=cari&nama=Andi`

Untuk menambahkan teman, gunakan AJAX untuk mengirimkan *request* ke server. Detil informasi mengenai pengiriman *request* adalah sebagai berikut:

- Alamat url: “../services/cv.php”
Dengan asumsi pengiriman *request* dilakukan dari *file* javascript yang berada dalam *folder* js. silahkan disesuaikan url-nya sesuai lokasi file pengirim *request*.
- Data yang dikirimkan: perintah=tambah, user=npmlUserLogin, dan userteman=npmlCalonTeman
variable perintah dengan *value* “tambah”

variable user dengan *value* npml user yang sedang login

variable userteman dengan *value* npml user yang akan ditambahkan sebagai teman
- Kembalian
Data yang dikembalikan adalah *text* dengan struktur JSON. Jika pengiriman *request* berhasil maka bentuk respon yang dikembalikan adalah seperti berikut.

`{"status": "ok"}`

dan jika tidak berhasil, maka *server* akan mengembalikan respon `{"status": "gagal"}`

Contoh:

Contoh pengiriman *request* melalui *url*:

`../services/cv.php?perintah=tambah&user=[NPM]&userteman=[NPMteman]`

Contoh pengiriman *request* menggunakan jQuery:

URL: “../services/cv.php”

data: {“perintah”: “tambah”, “user”: [NPM], “userteman”: [NPMteman] }

method: “POST”

Parameter [NPM] diisi dengan NPM user yang sedang login dan [NPMteman] diisi NPM user yang akan ditambah

Misal: `../services/cv.php?perintah=tambah&user=1234567&userteman=987654`

Output: status = ok/gagal

4. Melihat teman

Dengan fitur ini, pengguna dapat melihat daftar teman yang dimiliki. Untuk mendapatkan data teman yang dimiliki, gunakan AJAX untuk mengirimkan *request* ke *server*. Detil informasi mengenai pengiriman *request* adalah sebagai berikut:

- Alamat url: “../services/cv.php”

Dengan asumsi pengiriman *request* dilakukan dari *file* javascript yang berada dalam *folder* js. silahkan disesuaikan url-nya sesuai lokasi file pengirim *request*.

- Data yang dikirimkan: perintah=teman dan user=npm
variable perintah dengan value "teman"

variable user dengan value npm dari user yang sedang login

- Kembalian
Data yang dikembalikan adalah *text* dengan struktur JSON. Jika pengiriman *request* berhasil maka bentuk respon yang dikembalikan adalah seperti berikut

```
{"users": [  
    {"user": "npmTeman", "nama": "namaTeman"},  
    {"user": "npmTeman", "nama": "namaTeman"},  
    {"user": "npmTeman", "nama": "namaTeman"},  
    ...  
    {"user": "npmTeman", "nama": "namaTeman"}  
]}
```

dan jika tidak ada data teman, maka *server* akan mengembalikan respon `{"users": []}`

Contoh:

Contoh pengiriman *request* melalui *string url*:

```
../services/cv.php?perintah=teman&user=[NPM]
```

Contoh pengiriman request menggunakan jQuery:

URL: `"../services/cv.php"`

data: `{"perintah": "teman", "user": [NPM]}`

Parameter [NPM] diisi NPM user yang sedang login

Misal: `../services/cv.php?perintah=teman&nama=1234567`

Penilaian

1. *Login & logout* **[20 poin]**
2. Menampilkan halaman utama/profil **[10 poin]**
3. Mengubah profil pengguna **[15 poin]**
4. Mencari teman **[15 poin]**
5. Menambahkan teman **[15 poin]**
6. Menampilkan daftar teman **[15 poin]**
7. Tampilan (CSS, efek, animasi) yang menarik **[10 poin]**

Dalam mengerjakan UTS ini gunakan *script* pada *client-side* saja, sesuai materi di perkuliahan, seperti: HTML, CSS, jQuery, AJAX, dan Web Storage.