

Python CGI

1. Catatan	2
2. Header HTTP	3
3. CGI	4
4. Hello World	5
Hello World	5
Redireksi	5
Traceback manager: modul cgi	6
5. Mendapatkan input	7
Dukungan CGI: modul cgi	7
Input single value	7
Input multiple value	8
Upload file	9
Request method	10
6. Cookies	12
Mengatur cookies	12
Mendeteksi cookies	14
7. Session	16

1. Catatan

- a) Dianjurkan untuk membaca materi Python Dasar terlebih dahulu. Aturan penulisan kode dapat dibaca pada materi Python Dasar.
- b) Peserta dianjurkan untuk menguasai dasar-dasar HTML. Dasar-dasar HTML tidak akan dibahas di materi training.
- c) HTTP server yang digunakan adalah Apache HTTP server versi 2.2:
 - 1. Peserta dianjurkan untuk membaca dokumentasi Apache HTTP Server 2.2. Konfigurasi Apache HTTP Server tidak akan dibahas di materi ini.
 - 2. Direktori userdir setiap user (~/.public_html) telah diset dengan hak untuk menjalankan script CGI (+ExecCGI) . Ekstensi .py dan .cgi akan dianggap sebagai program CGI.
- d) Pengembangan tetap mengacu pada multiplatform (setidaknya Windows dan Linux), kecuali disebutkan berbeda.
- e) Dalam melakukan pemrograman CGI dengan Python, training akan menggunakan standard library.
- f) Semua contoh dituliskan dalam satu script Python. Di dunia nyata, umumnya, dipisahkan.
- g) Instruktur akan memaparkan perbandingan singkat Python+CGI dengan solusi lain Python untuk web, seperti CherryPy, Django, Web2py atau web.py, dan solusi populer lain, seperti PHP.

2. Header HTTP

- a) Aplikasi web bekerja dalam HyperText Transfer Protocol (HTTP).
- b) Dalam bekerja dengan HTTP, melibatkan penggunaan header-header HTTP.
- c) Contoh header adalah:
 - 1. Content-Type: menentukan MIME type konten yang dikirim.
 - 2. Set-Cookie: mengirimkan cookie
 - 3. Location: redirection ke lokasi tertentu
 - 4. Selengkapnya: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>
- d) Programmer yang bekerja dengan CGI perlu mengetahui header-header tersebut.
- e) Header-header dicetak ke stdout.
- f) Setelah HTTP header diberikan, kirimkan baris kosong 2 kali. Setelah itu, programmer bekerja dengan HTML.

3. CGI

a) CGI: Common Gateway Interface:

1. CGI dapat dibangun dengan berbagai bahasa pemrograman dan digunakan pada berbagai sistem. CGI tidak mengharuskan penggunaan bahasa tertentu di sistem tertentu.
2. CGI berfungsi sebagai pintu gerbang ke sistem-sistem lain.
3. Protokol untuk menjalankan software eksternal dari web server, untuk menghadirkan dynamic content .

b) Pengembangan aplikasi CGI dengan Python:

1. Pengguna Python hanya membutuhkan web server dan Python terinstall di sistem.
2. Standard library Python dapat mempermudah pemrograman CGI.
3. Pastikan setiap script (di Linux) telah diberikan hak akses executable (`chmod +x <script.py>`) sebelum diakses.

c) Komunikasi dengan web server melibatkan stdin, stdout dan environment variable:

1. Dalam aplikasi umum, stdin umumnya keyboard dan stdout umumnya console/layar.
2. Dalam CGI, data yang dikirim oleh user (diantaranya lewat web browser) akan dikirim lewat stdin. Output program CGI lewat stdout akan dikirimkan ke user oleh web server.
3. Beberapa environment variabel akan diset dan dapat digunakan oleh developer.

4. Hello World

Hello World

Script `hello.py` berisikan contoh Hello World CGI.

```
hello.py:
#!/usr/bin/env python

print 'Content-type: text/html'
print

print 'Hello World'
```

Output:

- Ketika script diakses dari web browser, tulisan Hello World akan ditampilkan.

Penjelasan:

- Output akan menggunakan HTML, dengan MIME text/html. Header Content-Type digunakan.

Redireksi

Script `location.py` berisikan contoh redireksi ke URL lain dengan header Location.

```
location.py:

#!/usr/bin/env python

print 'Content-type: text/html'
print 'Location: test.py'
print
```

Output:

- Ketika script diakses dari web browser, halaman aktif akan berpindah ke `test.py`.

Penjelasan:

- Untuk redireksi, kita bisa mempergunakan header Location.

Traceback manager: modul cgitb

Modul cgitb dapat digunakan selama proses pengembangan untuk melaporkan exception yang terjadi. Penggunaan modul ini sangat membantu dalam pengembangan aplikasi CGI dengan Python.

Script cgierror.py berisikan contoh penggunaan modul cgitb.

cgierror.py:

```
#!/usr/bin/env python
```

```
import cgitb
cgitb.enable()
```

```
print 'Content-type: text/html'
print
```

```
print 1/0
```

Output:

- Ketika script diakses dari web browser, pesan kesalahan akibat pembagian dengan 0 akan ditampilkan, terformat rapi.

Penjelasan:

- Setelah import modul dilakukan, gunakan fungsi enable().

Catatan:

- Sebaiknya, jangan gunakan modul ini ketika aplikasi digunakan di lingkungan produktif. Informasi yang dilaporkan bisa mengekspos beberapa detail sistem.

5. Mendapatkan input

Dukungan CGI: modul cgi

Pengembangan aplikasi CGI dapat dilakukan dengan relatif lebih mudah dengan bantuan modul cgi. Dengan mempergunakan modul ini, kita bisa mendapatkan input dari user, termasuk upload file, dengan cepat, mudah, dan reliable.

Input single value

Script `singlevar.py` berisikan contoh mendapatkan input dari user.

```
singlevar.py:
#!/usr/bin/env python

import cgi
import os

print 'Content-type: text/html'
print

form = cgi.FieldStorage()

html_form = '''
<form action='singlevar.py'>
name: <input type='text' name='name'>
<input type='submit'>
</form>
'''

if not form.has_key('name'):
    print html_form
else:
    name = form.getfirst('name', '')
    name = cgi.escape(name)
    print 'Hi, %s' %(name)
```

Output:

- Ketika script diakses dari web browser, sebuah form akan ditampilkan. User dapat mengisi nama pada field name dan klik pada tombol Submit untuk mengirim nama yang diisikan.
- Setelah itu, server akan menampilkan pesan Hi, <nama>.

Penjelasan:

- Kita menggunakan pendekatan satu file untuk form dan prosesornya. Dengan demikian, untuk parameter action form, kita dapat mengisikan dengan nama file script kita.
- Pertama-tama, kita akan membuat objek form, yang merupakan instance dari

class FieldStorage modul cgi. Objek form nantinya akan berisikan input dari user.

```
form = cgi.FieldStorage()
```

- Apabila form tidak memiliki key 'name', maka form input (variabel html_form) akan ditampilkan.
- Apabila form memiliki key 'name', maka name bisa didapatkan dengan method getfirst(). Dalam contoh ini, getfirst() akan berusaha mendapatkan variabel name, dan apabila tidak ditemukan, akan mengembalikan string kosong.

```
name = form.getfirst('name', '')
```

- Input dari user tidak bisa selalu dipercaya. Oleh karena itu, escaping dilakukan dengan fungsi escape() modul cgi. Proses escaping akan mengkonversi karakter "&", "<" dan ">" menjadi HTML sequence.
- Sebuah form kita siapkan dalam variabel html_form. Di dalam form tersebut, kita mendefinisikan input:
 - text, variabel: name (<input type='text' name='name'>)
 - submit

Input multiple value

Script multivar.py berisikan contoh mendapatkan input dari user, dalam bentuk satu variabel, dengan multiple nilai yang diasosiasikan. Hal ini umumnya ditemukan dalam kasus dimana user bisa memilih lebih dari satu nilai, untuk satu field.

multivar.py:

```
#!/usr/bin/env python
```

```
import cgi
import os
```

```
print 'Content-type: text/html'
print
```

```
form = cgi.FieldStorage()
```

```
html_form = '''
<form action='multivar.py'>
Python version:<br>
<br><input type='checkbox' name='version' value='2.4'>2.4
<br><input type='checkbox' name='version' value='2.5'>2.5
<br><input type='checkbox' name='version' value='2.6'>2.6
<br><input type='submit'>
</form>
'''
```



```
if not form.has_key('version'):
    print html_form
else:
    version = form.getlist('version')
    print 'You are using Python version: '
    for i in version:
        print i
```

Output:

- Ketika script diakses dari web browser, sebuah form akan ditampilkan. Di form tersebut, user bisa memilih versi Python yang digunakan. Sebagai catatan, user bisa memilih lebih dari satu versi.
- Setelah user memilih dan klik tombol Submit, maka server akan menampilkan versi Python yang dipilih oleh user.

Penjelasan:

- Bacalah penjelasan sebelumnya, apabila diperlukan.
- Variabel version diasosiasikan pada tiga checkbox, dengan nilai (value) yang berbeda.
- Untuk mendapatkan multiple value untuk sebuah variable, gunakan method `getlist()`.

Upload file

Script `upload.py` berisikan contoh upload file.

upload.py:

```
#!/usr/bin/env python

import cgi
import os

#windows check
try:
    import msvcrt
    msvcrt.setmode (0, os.O_BINARY) #stdin
    msvcrt.setmode (1, os.O_BINARY) #stdout
except ImportError:
    pass
#end of windows check

print 'Content-type: text/html'
print

form = cgi.FieldStorage()

html_form = '''
<form action='upload.py' method='post' enctype='multipart/form-data'>
```

```
Select file: <input type='file' name='myfile'>
<input type='submit'>
</form>
'''

if not form.has_key('myfile'):
    print html_form
else:
    myfile = form['myfile']
    if myfile.filename:
        fn = os.path.basename(myfile.filename)
        open('/tmp/' + fn, 'wb').write(myfile.file.read())
        print 'File %s uploaded successfully.' %(fn)
```

Output:

- Ketika script diakses dari web browser, sebuah form akan ditampilkan. Di form tersebut, user bisa memilih file yang akan diupload.
- Setelah user memilih dan klik tombol Submit, file akan diupload. Apabila sukses, maka file hasil upload akan disimpan dan pesan akan ditampilkan.

Penjelasan:

- Bacalah penjelasan sebelumnya, apabila diperlukan.
- Windows membutuhkan stdin dan stdout diset ke modus binary. Oleh karena itu, kita akan melakukan import msvcrt. Apabila berhasil, maka kita set modus untuk handle file bersangkutan.

```
#windows check
try:
    import msvcrt
    msvcrt.setmode (0, os.O_BINARY) #stdin
    msvcrt.setmode (1, os.O_BINARY) #stdout
except ImportError:
    pass
#end of windows check
```

- Form untuk file upload memiliki enctype dengan nilai multipart/form-data. Untuk form method, kita menggunakan post. Untuk upload file, kita menggunakan input dengan tipe file.
- Untuk mendapatkan file hasil upload, akseslah form dengan key adalah nama variable input file, yang merupakan instance FieldStorage.
 - Atribut filename adalah nama file yang dikirim user. Gunakan os.path.basename() untuk mendapatkan basename nama file tersebut.
 - Atribut file adalah objek file hasil upload. Simpan file hasil upload dengan membuka file baru (modus wb) dan menuliskan semua yang berhasil dibaca dari objek file upload.

Request method

Untuk mengetahui request method yang digunakan, kita bisa membaca

environment variable REQUEST_METHOD, seperti dicontohkan dalam script reqmethod.py.

```
reqmethod.py:
#!/usr/bin/env python

import cgi
import os

print 'Content-type: text/html'
print

form = cgi.FieldStorage()

html_form = '''
<form action='reqmethod.py' method='post'>
name: <input type='text' name='name'>
<input type='submit'>
</form>
'''

if not form.has_key('name'):
    print html_form
else:
    name = form.getfirst('name', '')
    name = cgi.escape(name)
    print 'Hi, %s' %(name)
    print 'Request method: %s.' %(os.environ['REQUEST_METHOD'])
```

Output:

- Ketika script diakses dari web browser, sebuah form akan ditampilkan. User dapat mengisi nama pada field name dan klik pada tombol Submit untuk mengirim nama yang diisikan.
- Setelah itu, server akan menampilkan pesan Hi, <nama> dan diikuti pesan Request method: <method>.
- Akseslah os.environ untuk mendapatkan environment variable.

Catatan:

- Cobalah untuk mengakses reqmethod.py dengan query string name=test dan lihatlah hasilnya.

6. Cookies

HTTP adalah protokol stateless. Setelah user me-request dan server merespon, maka komunikasi selesai. Apabila developer ingin membangun kondisi dimana server mengetahui user sedang berada dalam sesi tertentu (misal: sedang login sebagai user tertentu, dalam proses transaksi pembelian barang, dan lain-lain), maka harus dikerjakan tersendiri, seperti dengan melibatkan penggunaan HTTP cookies.

HTTP Cookies, atau umum disebut cookies saja, adalah data yang dikirim oleh server, diterima dan disimpan oleh client (umumnya web browser) dan dikirim balik ke server ketika client mengakses server tersebut. Cookies dapat memiliki atribut tertentu, dan salah satunya adalah kapan suatu cookie expired.

Web browser modern umumnya mendukung cookies, dan menyediakan fasilitas untuk mendisablenya. Walau, secara default, web browser umumnya diset untuk menerima cookies.

Menggunakan Python, bekerja dengan cookies sangat dipermudah dengan modul Cookie.

Mengatur cookies

Contoh bekerja dengan cookies dan mengatur kapan suatu cookie berakhir dapat ditemukan pada script cookie1.py berikut.

```
cookie1.py:
#!/usr/bin/env python

import os, Cookie

cookie = Cookie.SimpleCookie()
cookie['user'] = 'test'
cookie['user']['expires'] = 10

print cookie
print 'Content-type: text/html'
print

cookie_str = os.environ.get ('HTTP_COOKIE')
if not cookie_str:
    print '<br>First time user (or Cookies disabled?)'
else:
    cookie.load (cookie_str)
    user = cookie['user'].value
    print '<br>User = %s' %(user)
```

Output:

- Ketika pertama kali cookiel.py diakses dari web browser, pesan 'First time user (or Cookies disabled?)' akan ditampilkan.
- Ketika kedua kali diakses, dalam jeda waktu di bawah 10 detik, maka pesan 'User = test' akan ditampilkan.
- Ketika akses berikutnya dilakukan, namun setelah 10 detik dari request sebelumnya, maka pesan 'First time user (or Cookies disabled?)' akan kembali ditampilkan. Hal ini disebabkan karena cookies yang kita set telah berakhir (expired).

Penjelasan:

- Untuk mengatur dan mengirimkan cookies:
 - Import modul Cookie
 - Instansiasi SimpleCookie (dari modul Cookie)
 - Setelah itu, kita bisa membuat cookie baru dengan memberikan key baru untuk objek SimpleCookie.
 - Setiap key baru diberikan, sebuah instance Morsel akan dibuat, yang mewakili setiap cookie baru yang dibuat. Untuk mendapatkan nilai cookie yang diset, atribut value dapat digunakan. Setiap cookie dapat memiliki key-key berikut:
 - comment: komentar
 - domain: domain dimana cookie valid. Awali dengan sebuah titik.
 - secure
 - expires: waktu dimana cookie berakhir setelah dikirimkan, dalam satuan detik. Secara default, berakhir setelah browser ditutup.
 - max-age
 - version
 - path: path dimana cookie valid. Secara default, berlaku pada path script tersebut. Set ke / agar valid untuk keseluruhan domain.
 - Baca juga: <http://www.w3.org/Protocols/rfc2109/rfc2109>
 - Cetak objek SimpleCookie

```
cookie = Cookie.SimpleCookie()
cookie['user'] = 'test'
cookie['user']['expires'] = 10
```

```
print cookie
```

- Untuk mendapatkan cookie (yang dikirim balik):
 - Baca environment variable (os.environ) HTTP_COOKIE
 - Bangun struktur data dengan method load() objek SimpleCookie
 - Akses atribut value cookie untuk mendapatkan nilai cookie

```
cookie_str = os.environ.get ('HTTP_COOKIE')
if not cookie_str:
    print '<br>First time user (or Cookies disabled?)'
else:
    cookie.load (cookie_str)
    user = cookie['user'].value
    print '<br>User = %s' %(user)
```

Mendeteksi cookies

Cookies bisa didisable oleh user, lewat pengaturan preferensi web browser. Untuk mendeteksi apakah cookies diaktifkan atau tidak, lakukanlah langkah-langkah berikut:

1. kirimkan cookies
2. redireksi/refresh halaman
3. periksa apakah cookies tersedia

Langkah-langkah ini diperlukan karena: pada pertama kali request, server hanya mengirimkan cookies, namun user tidak mengirimkan balik (sehingga tidak ada cookies). Pada saat redireksi dilakukan, user melakukan request (dan oleh karenanya, mengirimkan balik cookies). Pada request kedua kali ini, keberadaan cookies bisa terdeteksi.

Script detectcookie.py berisikan contoh pendeteksian cookies.

detectcookie.py:

```
#!/usr/bin/env python

import Cookie
import os
import cgi

cookie = Cookie.SimpleCookie()
cookie['user'] = 'test'
cookie['user']['expires'] = 10

use_cookie = False
start_check = False

form = cgi.FieldStorage()
checkcookie = form.getfirst ('checkcookie')
if checkcookie:
    cookie_str = os.environ.get ('HTTP_COOKIE')
    if cookie_str:
        use_cookie = True
else:
    start_check = True

if start_check == True:
    print 'Location: detectcookie.py?checkcookie=1'

print cookie
print 'Content-type: text/html'
print

if use_cookie == True:
    print 'Cookie is ENABLED.'
```

```
else:  
    print 'Cookie is DISABLED.'
```

Output:

- Ketika script diakses dari web browser, pesan 'Cookie is ENABLED' atau 'Cookie is DISABLED' akan ditampilkan, sesuai dengan pengaturan cookies di web browser.

Penjelasan:

- Pada akhirnya, variabel `use_cookie` akan bernilai `True` atau `False`. `True` berarti cookies diaktifkan. Pertama-tama, nilai `use_cookies` diset `False`.
- Variabel `start_check` akan menentukan apakah redireksi akan dilakukan atau tidak. Apabila bernilai `True`, maka redireksi dilakukan (dengan header `Location`). Pertama-tama, nilai `start_check` diset `False`.
- Script akan berusaha mendapatkan variabel `checkcookie` dari query string. Apabila didapatkan, maka cookies akan dicek. Apabila tidak didapatkan, maka redireksi dilakukan.

7. Session

Session dapat diartikan sebagai cookies, namun yang disimpan di server. Data yang dikirimkan antara server dan client umumnya berupa Session ID. Di server, session ID tersebut umumnya menjadi bagian dari nama file, yang digunakan untuk menyimpan informasi session.

Session dapat diimplementasikan dengan cookies. Apabila cookies tidak diaktifkan, session bisa pula diimplementasikan dengan Query string ataupun dengan field hidden. Salah satu kelemahan implementasi dengan query string dan field hidden adalah session ID harus selalu dimaintain, baik di query string, ataupun di field hidden. Sementara, salah satu kelemahan session dengan cookies adalah tidak dapat bekerja ketika cookies di-disable. Sebagian developer menggunakan pendekatan untuk menggunakan salah satu dari query string atau field hidden apabila cookies di-disable.

Training ini hanya memfokuskan pada session yang diimplementasikan dengan Cookies.

Terdapat beberapa hal yang harus dilakukan ketika mengimplementasikan session:

- session ID: definisikan nama variabel session ID dan nilai yang mungkin. Nama variabel bebas. Sebagai contoh adalah SID, atau SESSIONID. Sementara, nilai yang mungkin, haruslah unik. Modul md5 dan sha bisa dipergunakan untuk hash time. Apabila diperlukan, kombinasikan dengan modul random, ataupun dengan teknik lainnya.
- Format file data session. Data session (cookies dan lainnya) perlu disimpan di filesistem server. Developer bisa mempergunakan format yang diinginkan.
- Mekanisme penggunaan session.