

Hands-on Workshop

“Python untuk berbagai kebutuhan”

Catatan

- Mengetahui bahasa pemrograman Python
- Dilengkapi contoh-contoh program pendek (< 30 baris kode)
- Menggunakan pustaka standar apabila memungkinkan
- Kompatibel dengan Python 2 dan Python 3
- Menggunakan CPython
- Bukan tutorial atau training lengkap!
- Selesai dalam 4 jam!

Instalasi

- Mungkin sudah terinstall di sistem Anda!
- Atau, download dari <https://www.python.org>
- Instal juga editor teks untuk pemrograman

Prompt Python

- Biasakanlah dengan penggunaan prompt Python untuk berbagai kebutuhan (kalkulator, help, pemrograman sederhana)
- Cobalah beberapa tipe data mendasar seperti int, str, list, dict, bool



Bagian ini membutuhkan contoh penggunaan

Memulai program sederhana

- Ketiklah di editor teks
- Pastikan tab dan spasi yang konsisten. Apabila repot, gantilah penekanan tab menjadi sejumlah karakter spasi (misal: 4). Editor teks untuk pemrograman umumnya mendukung.
- Ingatlah bahwa kode Python case-sensitive

Contoh main()

```
def main():  
    pass  
  
if __name__ == '__main__':  
    main()
```

Dasar-dasar bahasa

- Kondisi
- Perulangan
- Fungsi
- Eksepsi
- Apabila diperlukan: [id-python.pdf](#) ([noprianto.com](#))



Bagian ini membutuhkan contoh pembahasan

Bekerja dengan file

- Mari membuat program untuk mengopi file
- Pembahasan:
 - Command-line argument
 - Eksepsi
 - Membuat dan menggunakan modul dan fungsi
 - Menampilkan teks ke stdout
 - Baca dan tulis file

Bekerja dengan file (copy_file.py)

```
import os
import sys

def copy_file(src, dest, size=-1):
    ret = 0
    length = float(os.path.getsize(src))
    with open(src, 'rb') as fi, open(dest, 'wb') as fo:
        while True:
            read = fi.read(size)
            if not read:
                break
            ret += len(read)
            fo.write(read)
            percent = ret * 100 / length
            sys.stdout.write('%.3f%% ' %(percent))
    return ret
```

Bekerja dengan file

```
import sys
import copy_file

def get_size(s):
    try:
        return int(s)
    except ValueError:
        return -1

def main(src, dest, size=-1):
    try:
        print(copy_file.copy_file(src, dest, size))
    except IOError:
        sys.exit('error')

if __name__ == '__main__':
    if not len(sys.argv) > 2:
        sys.exit('usage: %s <source> <destination> [size]' % (sys.argv[0]))
    size = -1
    if len(sys.argv) > 3:
        size = get_size(sys.argv[3])
    main(sys.argv[1], sys.argv[2], size)
```

Database

- Mari membuat program untuk bekerja dengan database SQLite3
- Pembahasan:
 - Membuat tabel
 - Insert
 - Select

Database

```
import sqlite3

def main():
    with sqlite3.connect(':memory:') as conn:
        cur = conn.cursor()
        query = 'create table test(id integer primary key, name text)'
        cur.execute(query)
        #
        for i in range(10):
            name = 'hello %s' %(i)
            query = 'insert into test(name) values(?)'
            cur.execute(query, [name])
        #
        query = 'select * from test order by id'
        cur.execute(query)
        data = cur.fetchall()
        print(data)

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

Tanggal dan waktu

- Mari membuat program untuk menghitung selisih waktu antara dua tanggal
- Pembahasan:
 - Menampilkan tanggal sesuai format
 - Parsing tanggal sesuai format

Tanggal dan waktu (datetimeutil.py)

```
import time

FORMAT_DATE_TIME = '%Y-%m-%d %H:%M:%S'
DAYS_IN_YEAR = 365.2425

def is_valid_date_time(s):
    try:
        s = s.strip()
        p = time.strptime(s, FORMAT_DATE_TIME)
        return True
    except:
        pass
    return False

def date_time_diff(s1, s2):
    if not is_valid_date_time(s1) or not is_valid_date_time(s2):
        raise Exception
    t2 = time.mktime(time.strptime(s2, FORMAT_DATE_TIME))
    t1 = time.mktime(time.strptime(s1, FORMAT_DATE_TIME))
    d = float(t2 - t1)
    return d / (60 * 60 * 24 * DAYS_IN_YEAR)
```

Tanggal dan waktu (datetimeutil_format.py)

```
DAYS_IN_YEAR = 365.2425
DAYS_IN_MONTH_AVERAGE = round( (DAYS_IN_YEAR / float(12)), 2)

def format_date_time_diff(f, year, month, day, ago):
    a = ''
    if f > 0:
        a = ago
    f = abs(f)

    if 'e' in str(f).lower(): return
    f1, f2 = str(f).split('.')
    y = int(f1)

    left_m = f - y
    fm = float(left_m * 12)
    m1, m2 = str(fm).split('.')
    m = int(round(float(m1), 0))

    left_d = fm - m
    d = int(round(left_d * DAYS_IN_MONTH_AVERAGE, 0))

    return '%s%s%s%s%s%s%s' %(y, year, m, month, d, day, a)
```

Tanggal dan waktu

```
import sys
import time
import datetime
import datetime_format

PY3 = sys.version_info[0] == 3
if PY3:
    prompt = input
else:
    prompt = raw_input

FORMAT_DATE_TIME = '%Y-%m-%d %H:%M:%S'

def main():
    now = time.strftime(FORMAT_DATE_TIME)
    date = prompt('Enter date (Y-m-d H:M:S): ')
    diff = datetime.datetime.strptime(date, FORMAT_DATE_TIME) - datetime.datetime.strptime(now, FORMAT_DATE_TIME)
    diff_s = diff.total_seconds()
    diff_s = datetime_format.format_date_time_diff(diff_s,
        'year ', 'month ', 'day ', 'ago')
    print(diff_s)

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


HTTP client

- Mari membuat program untuk bekerja dengan HTTP GET (contoh program terbilang)
- Pembahasan:
 - Memanggil method GET
 - Bekerja dengan JSON

HTTP client

```
import sys
import json

PY3 = sys.version_info[0] == 3
if PY3:
    import urllib.request as module
else:
    import urllib2 as module

def main():
    res = module.urlopen('http://localhost:11738/number_to_words.json')
    content = res.read()
    content_json = json.loads(content)
    for k in content_json.keys():
        print('%s: %s' % (k, content_json.get(k).get('123')))

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

HTTP client

```
import sys

PY3 = sys.version_info[0] == 3
if PY3:
    import urllib.request as module
else:
    import urllib2 as module

LANGUAGES = ['id', 'en']

def get_number_to_words(lang, number):
    res = module.urlopen(
        'http://localhost:11738/number_to_words?lang=%s&number=%s'
        %(lang, number))
    content = res.read()
    return content

def main(number):
    for lang in LANGUAGES:
        print(get_number_to_words(lang, number))

if __name__ == '__main__':
    if not len(sys.argv) > 1:
        sys.exit('usage: %s <number>' %(sys.argv[0]))
    main(sys.argv[1])
```

GUI

- Mari membuat HTTP client GUI sederhana
- Pembahasan:
 - Contoh penggunaan Tk

GUI

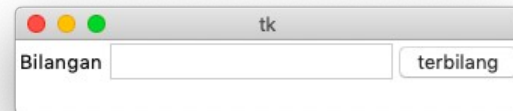
```
import sys
from http_get_param import get_number_to_words

PY3 = sys.version_info[0] == 3
if PY3:
    from tkinter import *
else:
    from Tkinter import *

def get(entry, result):
    result['text'] = get_number_to_words('id', entry.get())

def main():
    Label(text='Bilangan').grid(row=0, column=0, sticky=W)
    entry = Entry()
    entry.grid(row=0, column=1, sticky=W+E)
    button = Button(text='terbilang')
    button.grid(row=0, column=2, sticky=E)
    result = Label()
    result.grid(row=1, column=0, columnspan=3)
    button['command'] = lambda: get(entry, result)

if __name__ == '__main__':
    root = Tk()
    main()
    root.mainloop()
```



Multithreading

- Apabila diperlukan:
 - `noprianto-concurrency_multithreading.pdf`
(`noprianto.com`)
- Contoh: program GUI (Tk) dengan threading sederhana

Multithreading

```
import sys
import time
import threading

PY3 = sys.version_info[0] == 3
if PY3:
    from tkinter import *
else:
    from Tkinter import *

def work():
    for i in range(5):
        print(i)
        time.sleep(1)

def multi():
    threading.Thread(target=work).start()

def main():
    Button(text='single', command=work).grid()
    Button(text='multi', command=multi).grid()

if __name__ == '__main__':
    root = Tk()
    main()
    root.mainloop()
```



Aplikasi web dengan database

- Mari membuat sebuah aplikasi web yang dilengkapi dengan akses database
- Pembahasan:
 - Pustaka web.py (eksternal)
 - Method GET/POST
 - Redirect

Aplikasi web dengan database

```
import web
URLS = ('/', 'index', '/add', 'add')
app = web.application(URLS, globals())
db = web.database(dbn='sqlite', db='web_db.db')

class index:
    def GET(self):
        try:
            res = list(db.select('test'))
            return [r.name for r in res]
        except:
            db.query('create table test(name text)')
            raise web.seeother('/')

class add:
    def GET(self):
        return '''<html><body><form action='/add' method='post'>
Name: <input type='text' name='name'>
<input type='submit'></form></body></html>'''

    def POST(self):
        name = web.input(name='').name.strip()
        if not name: raise web.seeother('/add')
        db.insert('test', name=name)
        raise web.seeother('/')

if __name__ == '__main__': app.run() █
```

Terima kasih

- Diskusi