

BAB 4. CONTOH PROYEK DENGAN PYTHON

Daftar Isi

- 4.1. Pengenalan AIML untuk CHATBOT
- 4.2. Modul Python untuk Aimpl
- 4.3. Proyek Asisten Pribadi

© Copyright by Antonius – www.jasaplus.com All Rights Reserved

Barangsiapa didapati memperjual belikan materi training ini tanpa seizin dari pencipta maka pencipta berhak menuntut ganti rugi yang jumlahnya ditentukan oleh pencipta materi ini.

BAB 4. CONTOH PROYEK DENGAN PYTHON

Pada contoh kali ini, kita akan membuat aplikasi asisten sederhana.

Untuk speech recognition pada contoh aplikasi ini kita memerlukan webspeech yang tadi kita gunakan. Bisa didownload di github repo training ini (alamat github ada pada BAB 1)

Sama seperti proyek speech recognition sebelumnya (speech_selenium.py), kita akan menggunakan webspeech yang sebelumnya sudah kita simpan di httdocs.

4.1. Pengenalan AIML untuk Chatbot

Pada bagian ini kita akan mempelajari dasar aiml. AIML merupakan singkatan dari “Artificial Intelligence Markup Language”. Aiml pertama kali didevelop oleh Alicebot free software community dan Dr. Richard S Wallace sekitar tahun 1995 hingga 2000.

AIML merupakan markup xml yang digunakan untuk mempermudah pembuatan aplikasi chatbot. AIML merupakan contoh dari NLU (Natural Language Understanding). Selain menggunakan NLU, untuk membuat chatbot kita bisa juga menggunakan NLP (Natural Language Processing).

Berikut ini tag tag aiml yang sering digunakan :

<aiml>

merupakan tag yang digunakan sebagai pembuka dan akhir dari dokumen aiml.

<category>

tag ini digunakan untuk mendefinisikan suatu unit pengetahuan.

<pattern>

tag ini digunakan untuk mendefinisikan inputan user yang akan direspon. Isi di dalam tag ini bisa ditambahkan tanda * untuk wildcard.

<template>

tag ini digunakan untuk mendefinisikan respon dari chatbot untuk user input yang sudah didefinisikan pada pattern.

<star>

tag ini digunakan untuk wildcard.

<srai>

tag ini merupakan tag multi purpose yang digunakan untuk mendefinisikan target yang berbeda untuk suatu template.

<random>

tag ini digunakan untuk mendapatkan respons acak

tag ini digunakan untuk mendefinisikan respons lebih dari satu.

<set>

tag ini digunakan untuk menyimpan / mengisi nilai suatu variabel pada aimpl.

<get>

tag ini digunakan untuk mengambil nilai variabel

<that>

tag ini digunakan untuk memberikan respon berdasarkan konteks

<system>

tag ini digunakan untuk menjalankan perintah sistem operasi

<topic>

tag ini digunakan untuk menyimpan suatu topik sehingga percakapan selanjutnya bisa dilakukan berdasarkan konteks yang disimpan.

<think>

tag ini digunakan untuk menyimpan variabel tanpa sepengetahuan user

<condition>

tag ini digunakan untuk memberikan respon berdasarkan kondisi pada input
Untuk mencoba coba beberapa hasil chat bot dari aimpl, anda bisa mengunjungi
<https://botlibre.com/browse?browse-type=Bot&browse=true>

4.2. Modul Python untuk Aimpl

Pada bagian ini kita akan mempelajari modul python yang digunakan untuk aimpl. Kita akan menggunakan modul python : python-aiml. Untuk menginstall modul ini ketikkan :

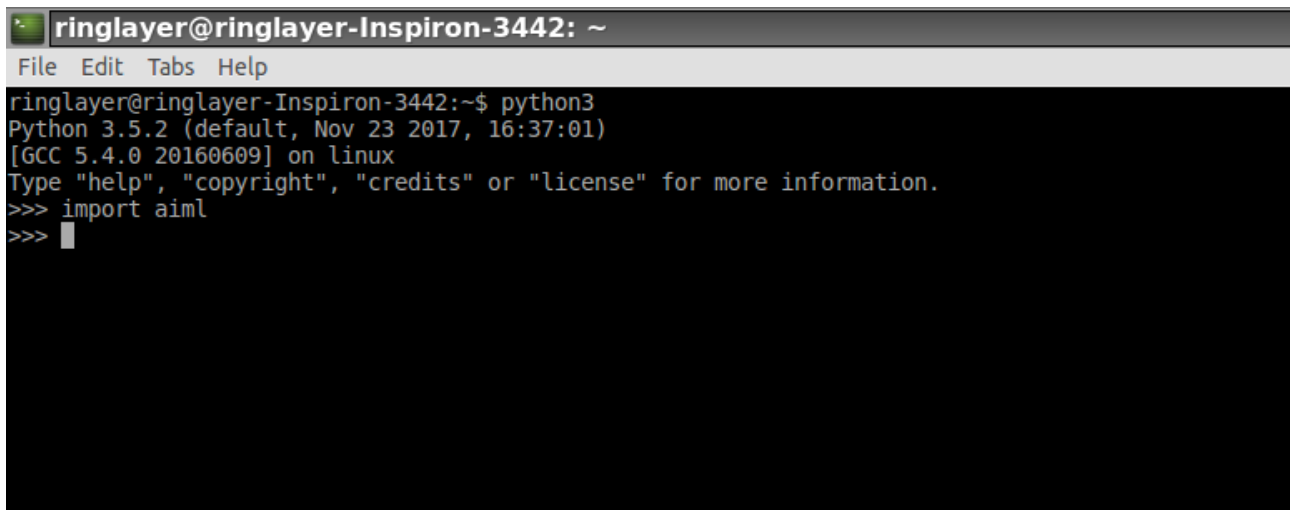
```
pip3 install python-aiml
```

atau

```
pip install python-aiml
```

Setelah menginstall modul ini kita bisa mulai menggunakan modul ini dengan cara mengimport :

```
import aiml
```

A screenshot of a terminal window with a dark background. The title bar reads 'ringlayer@ringlayer-Inspiron-3442: ~'. The menu bar shows 'File Edit Tabs Help'. The terminal text shows the user running 'python3', which opens a Python 3.5.2 shell. The user then enters 'import aiml' and the prompt returns to the shell.

```
ringlayer@ringlayer-Inspiron-3442: ~  
File Edit Tabs Help  
ringlayer@ringlayer-Inspiron-3442:~$ python3  
Python 3.5.2 (default, Nov 23 2017, 16:37:01)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import aiml  
>>>
```

Sebelum membuat aplikasi dengan python-aiml, kita perlu mendownload terlebih dahulu aimpl yang akan kita gunakan. Modul python python-aiml untuk python 3 yang telah kita install ini hanya mendukung aimpl versi 1, oleh karena itu kita akan menggunakan file aimpl versi 1.

Untuk mendownload contoh contoh file aimpl yang sudah jadi bisa didownload di website website berikut ini :

<https://botlibre.com/script?category=AIML>

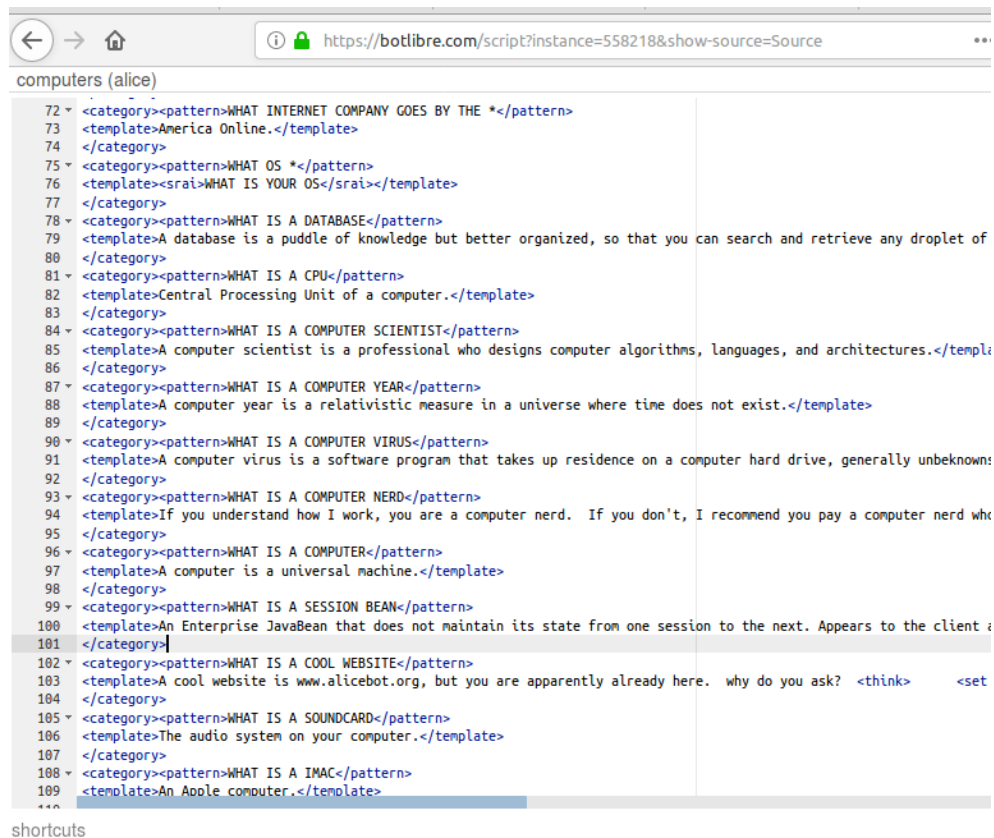
<http://square-bear.co.uk/aiml/>

<https://github.com/drwallace/aiml-en-us-foundation-alice>

versi aimpl yang bisa diparsing oleh modul python-aiml adalah aimpl versi 1, pada contoh kali ini kita akan mencoba menggunakan salah satu aimpl dari chat bot alice : <https://botlibre.com/script?instance=558218&show-source=Source>

Simpan di direktori tempat kita akan menjalankan aplikasi aimpl, simpan dengan nama file : computers.aiml

Pada aiml yang barusan kita download ini, kita bisa melihat ada beberapa pola pola input seputar komputer yang bisa dijawab oleh bot, terlihat pada bagian yang diapit tag pattern :



```
computers (alice)
72 <category><pattern>WHAT INTERNET COMPANY GOES BY THE *</pattern>
73 <template>America Online.</template>
74 </category>
75 <category><pattern>WHAT OS *</pattern>
76 <template><srai>WHAT IS YOUR OS</srai></template>
77 </category>
78 <category><pattern>WHAT IS A DATABASE</pattern>
79 <template>A database is a puddle of knowledge but better organized, so that you can search and retrieve any droplet of
80 </category>
81 <category><pattern>WHAT IS A CPU</pattern>
82 <template>Central Processing Unit of a computer.</template>
83 </category>
84 <category><pattern>WHAT IS A COMPUTER SCIENTIST</pattern>
85 <template>A computer scientist is a professional who designs computer algorithms, languages, and architectures.</templa
86 </category>
87 <category><pattern>WHAT IS A COMPUTER YEAR</pattern>
88 <template>A computer year is a relativistic measure in a universe where time does not exist.</template>
89 </category>
90 <category><pattern>WHAT IS A COMPUTER VIRUS</pattern>
91 <template>A computer virus is a software program that takes up residence on a computer hard drive, generally unbeknown:
92 </category>
93 <category><pattern>WHAT IS A COMPUTER NERD</pattern>
94 <template>If you understand how I work, you are a computer nerd. If you don't, I recommend you pay a computer nerd wh
95 </category>
96 <category><pattern>WHAT IS A COMPUTER</pattern>
97 <template>A computer is a universal machine.</template>
98 </category>
99 <category><pattern>WHAT IS A SESSION BEAN</pattern>
100 <template>An Enterprise JavaBean that does not maintain its state from one session to the next. Appears to the client
101 </category>
102 <category><pattern>WHAT IS A COOL WEBSITE</pattern>
103 <template>A cool website is www.alicebot.org, but you are apparently already here. why do you ask? <think> <set
104 </category>
105 <category><pattern>WHAT IS A SOUND CARD</pattern>
106 <template>The audio system on your computer.</template>
107 </category>
108 <category><pattern>WHAT IS A IMAC</pattern>
109 <template>An Apple computer.</template>
***
shortcuts
```

Beberapa pertanyaan yang bisa dijawab misal :

WHAT IS A CPU

```
<category><pattern>WHAT IS A CPU</pattern>
<template>Central Processing Unit of a computer.</template>
</category>
```

WHAT IS A COMPUTER SCIENTIST

```
<category><pattern>WHAT IS A COMPUTER SCIENTIST</pattern>
<template>A computer scientist is a professional who designs computer algorithms, languages, and
architectures.</template>
</category>
```

WHAT IS A COMPUTER NERD

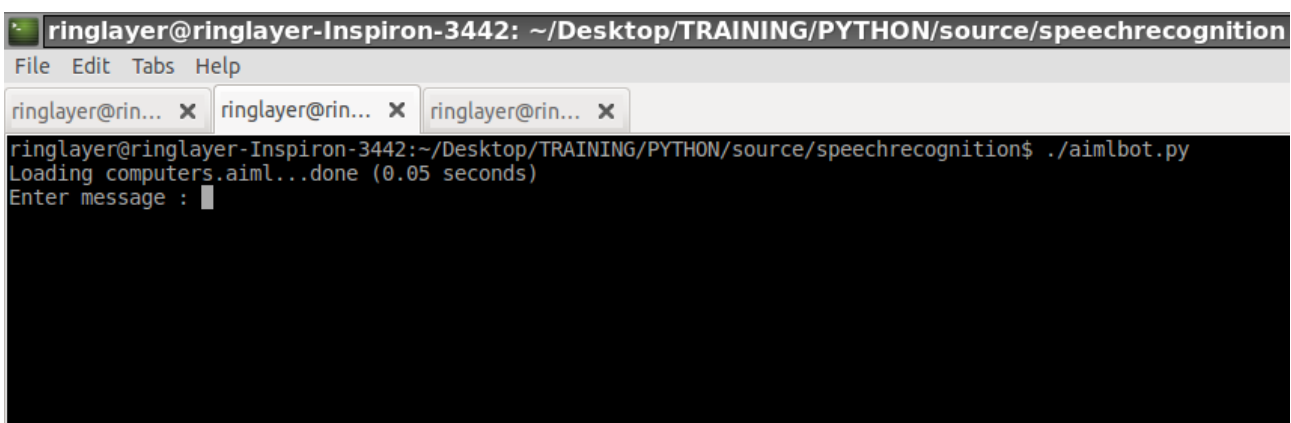
```
<category><pattern>WHAT IS A COMPUTER NERD</pattern>
<template>If you understand how I work, you are a computer nerd. If you don't, I recommend you pay a computer nerd
who does. <think> <set name="it"> <set name="topic">computer nerd</set> </set> </think></template>
</category>
```

dan lain sebagainya.

Untuk mencoba aiml ini kita akan membuat aplikasi sederhana, siapkan aplikasi python baru dengan nama file aimlbot.py

```
#!/usr/bin/env python3
"""
aimlbot.py
sample for python3 training at jasaplus.com
"""
import aiml
kernel = aiml.Kernel()
kernel.learn("computers.xml")
while True:
    print (kernel.respond(input("Enter message : ")))
```

Jalankan aplikasi aimlbot.py di atas.

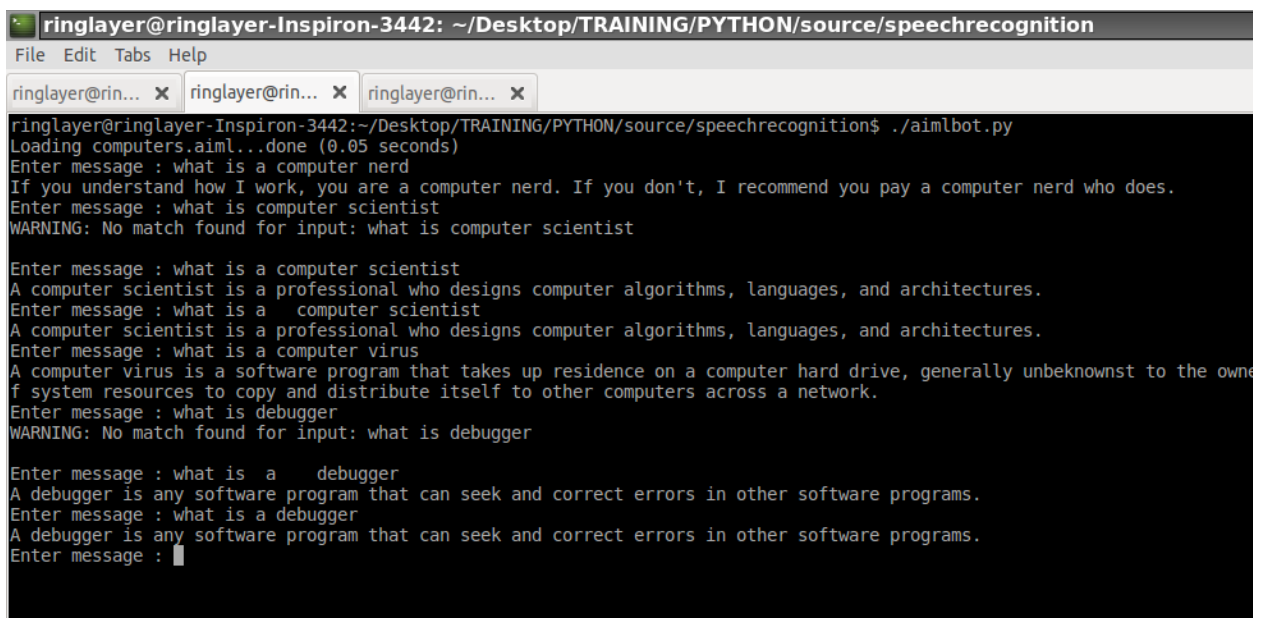


```
ringlayer@ringlayer-Inspiron-3442: ~/Desktop/TRAINING/PYTHON/source/speechrecognition
File Edit Tabs Help
ringlayer@rin... x ringlayer@rin... x ringlayer@rin... x
ringlayer@ringlayer-Inspiron-3442:~/Desktop/TRAINING/PYTHON/source/speechrecognition$ ./aimlbot.py
Loading computers.aiml...done (0.05 seconds)
Enter message : 
```

Saat dijalankan jika aiml berhasil diload maka akan muncul pesan seperti contoh ini :

Loading computers.aiml...done (0.05 seconds)

Pesan di atas merupakan pertanda file computer.aiml berhasil diparsing oleh python-aiml. Berikut ini adalah gambaran kinerja dari aplikasi dengan modul python-aiml di atas :



```
ringlayer@ringlayer-Inspiron-3442: ~/Desktop/TRAINING/PYTHON/source/speechrecognition
File Edit Tabs Help
ringlayer@rin... x ringlayer@rin... x ringlayer@rin... x
ringlayer@ringlayer-Inspiron-3442:~/Desktop/TRAINING/PYTHON/source/speechrecognition$ ./aimlbot.py
Loading computers.aiml...done (0.05 seconds)
Enter message : what is a computer nerd
If you understand how I work, you are a computer nerd. If you don't, I recommend you pay a computer nerd who does.
Enter message : what is computer scientist
WARNING: No match found for input: what is computer scientist

Enter message : what is a computer scientist
A computer scientist is a professional who designs computer algorithms, languages, and architectures.
Enter message : what is a computer scientist
A computer scientist is a professional who designs computer algorithms, languages, and architectures.
Enter message : what is a computer virus
A computer virus is a software program that takes up residence on a computer hard drive, generally unbeknownst to the owner.
f system resources to copy and distribute itself to other computers across a network.
Enter message : what is debugger
WARNING: No match found for input: what is debugger

Enter message : what is a debugger
A debugger is any software program that can seek and correct errors in other software programs.
Enter message : what is a debugger
A debugger is any software program that can seek and correct errors in other software programs.
Enter message : 
```

Pada contoh di atas, kita memberikan beberapa inputan :

what is a computer nerd

Pada contoh di atas aplikasi bisa memberikan respon berdasarkan aturan yang terdapat pada file `aiml computer.aiml` :

```
<category><pattern>WHAT IS A COMPUTER NERD</pattern>
<template>If you understand how I work, you are a computer nerd. If you don't, I recommend you pay a computer nerd
who does. <think>    <set name="it">    <set name="topic">computer nerd</set>    </set>    </think></template>
</category>
```

Respon yang diberikan terdapat pada template. Selanjutnya diberikan inputan :

what is computer scientist

Aplikasi gagal memberikan jawaban, Pada file `computer.aiml` terdapat rule ini :

```
<category><pattern>WHAT IS A COMPUTER SCIENTIST</pattern>
<template>A computer scientist is a professional who designs computer algorithms, languages, and
architectures.</template>
</category>
```

Karena aturan untuk menjawab string input “what is computer scientist” tidak ada, maka jawaban tidak diberikan

Selanjutnya diberikan inputan :

what is a computer scientist

dan selanjutnya dilakukan pengujian dengan menambahkan spasi :

what is a computer scientist

Dari kedua inputan di atas berhasil diberikan response sesuai aturan pada template.

4.3. Proyek Asisten Pribadi

Pada contoh kali ini kita akan membuat aplikasi asisten pribadi yang akan mengikuti beberapa perintah dasar seperti : play music, play song, open google, open twitter, dan open instagram serta melayani chat berbahasa inggris.

Berikut ini source code assistant.py :

```
#!/usr/bin/env python3
"""
assistant.py
sample application
for python training at www.jasaplus.com
"""

from selenium import webdriver
import time, os, inspect, sys, re
import speech_recognition as sr
from gtts import gTTS
import aimpl
from bs4 import BeautifulSoup

#start vars
BRAIN_FILE = "assistant.brain"
COMMAND_FILE = "command.dump"
#eof vars

#start class
class Assistant():
    def speech(self, data):
        try:
            tts = gTTS(text=data, lang='en', slow=True).save("tmp.mp3")
            cmd = "mpg123 tmp.mp3"
            pipe = os.popen(cmd)
        except:
            raise

    def setopt(self):
        opt = webdriver.ChromeOptions()
        opt.add_argument('--alsa-input-device=plughw:1,0')
        opt.add_argument('--mute-audio')
        opt.add_experimental_option("prefs", { \
            "profile.default_content_setting_values.media_stream_mic": 1, \
            "profile.default_content_setting_values.notifications": 1 \
        })
        return opt

    def prepare_kernel(self, FILE, aimpl_path):
        k = aimpl.Kernel()
        try:
            if os.path.exists(FILE):
                k.loadBrain(FILE)
            else:
                k.bootstrap(learnFiles=aiml_path)
                k.saveBrain(FILE)
        except:
            raise
        return k
```



```

def prepare_selenium_browser(self):
    browser = None
    try:
        opti = self.setopt()
        browser = webdriver.Chrome(chrome_options = opti)
        browser.get("http://localhost/webspeech/")
        browser.set_window_position(800, 370)
        browser.set_window_size(520, 350)
    except:
        raise

    return browser

def log(self, data):
    try:
        with open('output.txt', 'w') as file:
            file.write(data)
    except:
        raise

def main_listen_process(self, browser, k, cmd):
    try:
        count = 0
        data_prev = ""
        self.speech("Ok sir I am ready")
        while True:
            data = browser.find_element_by_id("data").get_attribute("value");
            if data != "None":
                data = data.strip()
                if (data != data_prev) and (len(data) > 0):
                    data_prev = data
                    print(data)
                    response = cmd.respond(data)
                    if (response.find("ok sir") == -1) and (response.find("The time") == -1):
                        response = k.respond(data)
                        print(response)
                        self.speech(response)
                    count = 0
                time.sleep(1)
                count+=1
            if (count > 15):
                startme = browser.find_element_by_id("startme")
                if startme != None:
                    startme.click()
        except:
            raise
#eof class

if __name__ == "__main__":
    assistant = Assistant()
    browser = assistant.prepare_selenium_browser()
    kernel = assistant.prepare_kernel(BRAIN_FILE, "aiml/*.aiml")
    command = assistant.prepare_kernel(COMMAND_FILE, "aiml_commands/commands.aiml")
    assistant.main_listen_process(browser, kernel, command)

```

Source code di atas berjalan pada sistem operasi linux, untuk windows memerlukan beberapa penyesuaian.

Berikut ini source code untuk windows :

```
#!/usr/bin/env python3
"""
assistant_windows.py
sample application
for python training at www.jasaplus.com
"""

from selenium import webdriver
import time, os, inspect, sys, re
import speech_recognition as sr
from gtts import gTTS
import aiml
from bs4 import BeautifulSoup

#start vars
BRAIN_FILE = "assistant.brain"
COMMAND_FILE = "command.dump"
#eof vars

#start class
class Assistant():
    def speech(self, data):
        try:
            tts = gTTS(text=data, lang='en', slow=True).save("tmp.mp3")
            cmd = "vlc --intf dummy tmp.mp3 vlc://quit"
            pipe = os.popen(cmd)
        except:
            raise

    def setopt(self):
        opt = webdriver.ChromeOptions()
        #opt.add_argument('--alsa-input-device=plughw:1,0')
        opt.add_argument('--mute-audio')
        opt.add_experimental_option("prefs", { \
            "profile.default_content_setting_values.media_stream_mic": 1, \
            "profile.default_content_setting_values.notifications": 1 \
        })
        return opt

    def prepare_kernel(self, FILE, aiml_path):
        k = aiml.Kernel()
        try:
            if os.path.exists(FILE):
                k.loadBrain(FILE)
            else:
                k.bootstrap(learnFiles=aiml_path)
                k.saveBrain(FILE)
        except:
            raise
        return k

    def prepare_selenium_browser(self):
        browser = None
        try:
            opti = self.setopt()
            browser = webdriver.Chrome(chrome_options = opti)
            browser.get("http://localhost/webspeech/")
            browser.set_window_position(800, 370)
            browser.set_window_size(520, 350)
```

```

except:
    raise

    return browser

def log(self, data):
    try:
        with open('output.txt', 'w') as file:
            file.write(data)
    except:
        raise

def main_listen_process(self, browser, k, cmd):
    try:
        count = 0
        data_prev = ""
        self.speech("Ok sir I am ready")
        while True:
            data = browser.find_element_by_id("data").get_attribute("value");
            if data != "None":
                data = data.strip()
                if (data != data_prev) and (len(data) > 0):
                    data_prev = data
                    print(data)
                    response = cmd.respond(data)
                    if (response.find("ok sir") == -1) and (response.find("The time") == -1):
                        response = k.respond(data)
                        print(response)
                        self.speech(response)
                    count = 0
                    time.sleep(1)
                    count+=1
                if (count > 15):
                    startme = browser.find_element_by_id("startme")
                    if startme != None:
                        startme.click()
            except:
                raise
#eof class

if __name__ == "__main__":
    assistant = Assistant()
    browser = assistant.prepare_selenium_browser()
    kernel = assistant.prepare_kernel(BRAIN_FILE, "aiml/*.aiml")
    command = assistant.prepare_kernel(COMMAND_FILE, "aiml_commands/commands.aiml")
    assistant.main_listen_process(browser, kernel, command)

```

Untuk materi lanjutan pembuatan chat bot akan dibahas dalam kursus selanjutnya “Python for Machine Learning”