

Python (Chapter 1)



Tri Angga Dio Simamora

NIM : 1.18.4.047

Politeknik Pos Indonesia

Applied Bachelor Program of Informatics Engineering

Bandung 2019

‘Jika Kamu tidak dapat menahan lelahnya belajar,
Maka kamu harus sanggup menahan perihnya Kebodohan.’
Imam Syafi’i

Contents

1	<i>Python</i>	1
1.1	Sejarah <i>Python</i>	1
1.2	Perbedaan <i>Python 2.x</i> dan <i>Python 3.x</i>	1
2	Implementasi dan Perusahaan Pengguna <i>Python</i>	11
2.1	Implementasi <i>Python</i>	11
2.2	Perusahaan Pengguna <i>Python</i>	11
3	Instalasi dan Konfigurasi <i>Python</i>	12
3.1	Instalasi <i>Python</i>	12
3.2	Konfigurasi <i>Python</i>	16
3.3	Instalasi <i>pip</i>	16
3.4	Update anaconda dan spyder	17
4	Menjalankan <i>Python</i>	20
4.1	Menjalankan script python dengan perintah CLI	20
4.2	Hello World!	21
4.3	Auto Login SIAP	21
4.4	Menggunakan Variable Explorer	23
4.5	Identasi	23

List of Figures

1.1	Gambar hasil print	2
1.2	Gambar perintah print	2
1.3	Gambar hasil pembagian	3
1.4	Gambar perintah pembagian	4
1.5	Gambar hasil error	5
1.6	Gambar perintah error	5
1.7	Gambar hasil looping	6
1.8	Gambar perintah looping	7
1.9	Gambar hasil unicode (bytes)	8
1.10	Gambar perintah unicode (bytes)	8
1.11	Gambar hasil unicode	9
1.12	Gambar perintah unicode	10
3.1	Gambar halaman download	12
3.2	Gambar install anaconda	13
3.3	Gambar eksekusi anaconda	13
3.4	Gambar anaconda license agreement	14
3.5	Gambar perintah yes or no	14
3.6	Gambar path anaconda	15
3.7	Gambar perintah install spyder3	15
3.8	Gambar setpath	16
3.9	Gambar instal pip	17
3.10	Gambar update anaconda	17
3.11	Gambar update anaconda	18
3.12	Gambar update spyder	19
4.1	Gambar running script dengan CLI	20
4.2	Gambar running script dengan spyder	21
4.3	Gambar halaman github	22

4.4	Gambar open dengan spyder	22
4.5	Gambar running script dengan spyder	23
4.6	Gambar contoh indentasi	24
4.7	Gambar contoh error indentasi	24
4.8	Gambar penanganan indentasi 1	25
4.9	Gambar penanganan indentasi 2	26

List of Tables

4.1 Tabel variabel explorer	23
---------------------------------------	----

Chapter 1

Python

1.1 Sejarah *Python*

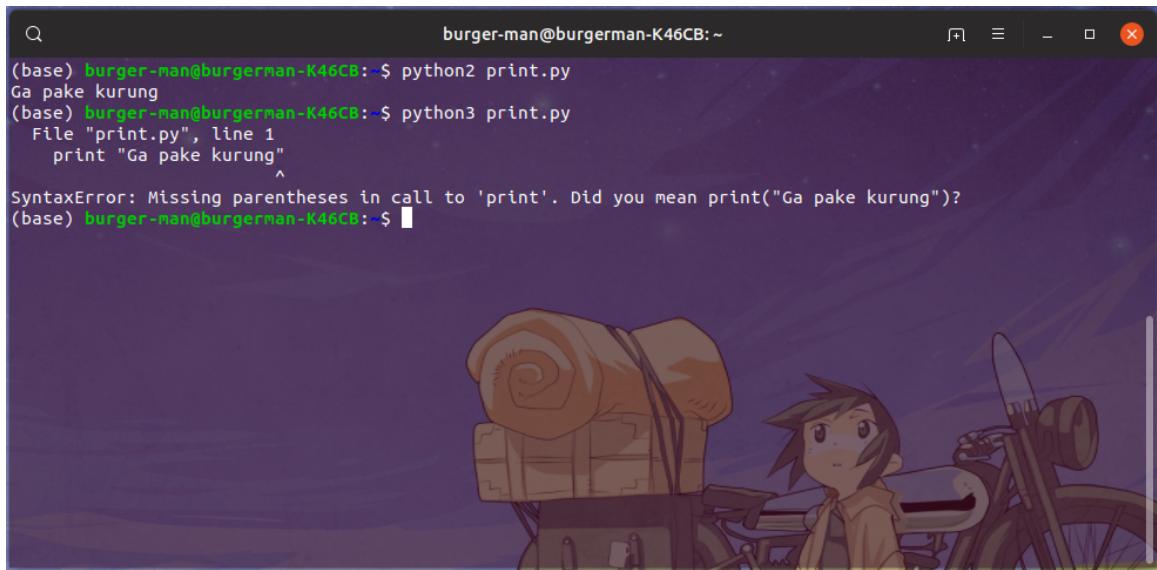
Python merupakan bahasa pemrograman tingkat tinggi yang dapat digunakan banyak hal, *Python* awalnya dirancang oleh **Guido van Rossum** pada tahun 1980 yang mana nama *Python* sebelum sebesar sekarang yaitu *ABC Programming Language* yang dijalankan di sistem operasi bernama *Amoeba Operating System*. Guido merasakan kehebatan dan kemampuan serta fitur yang berada pada bahasa pemrograman ABC ini sehingga Guido mengambil siktaks-sintaks yang berada pada bahasa pemrograman ABC ini, tentu saja banyak komplain yang berdatangan sehingga Guido terus melakukan perbaikan pada bahasa pemrograman yang sedang ia buat kala itu. Lalu, disinilah nama *Python* muncul dimuka bumi sebagai bahasa pemrograman, disaat Guido sedang menonton televisi dan menemukan kata '*Monty Python's Flying Circus*'.

Bahasa *Python* secara resmi dirilis pada tahun 1991, saat rilis pertama kali semua orang terkejut dengan sintaks yang dimiliki oleh *Python* ketika dibandingkan dengan bahasa lain seperti *Java*, *C*, *C++*, dan lain-lain pengekspresian bahasa ini cukup sederhana. Tujuan dari dibuatnya bahasa pemrograman ini adalah untuk mempermudah dalam membaca sebuah kode dari penulisan sintaks dan produktivitas dalam hal pengembangan tingkat lanjut.

1.2 Perbedaan *Python 2.x* dan *Python 3.x*

Banyak perbedaan yang akan kita temui jika kita dahulu pernah menggunakan *python* versi 2.x cukup lama sehingga berpindah ke versi 3.x, berikut contoh perbedaan pada *python* versi 2.x dan 3.x yang sangat penting untuk diketahui:

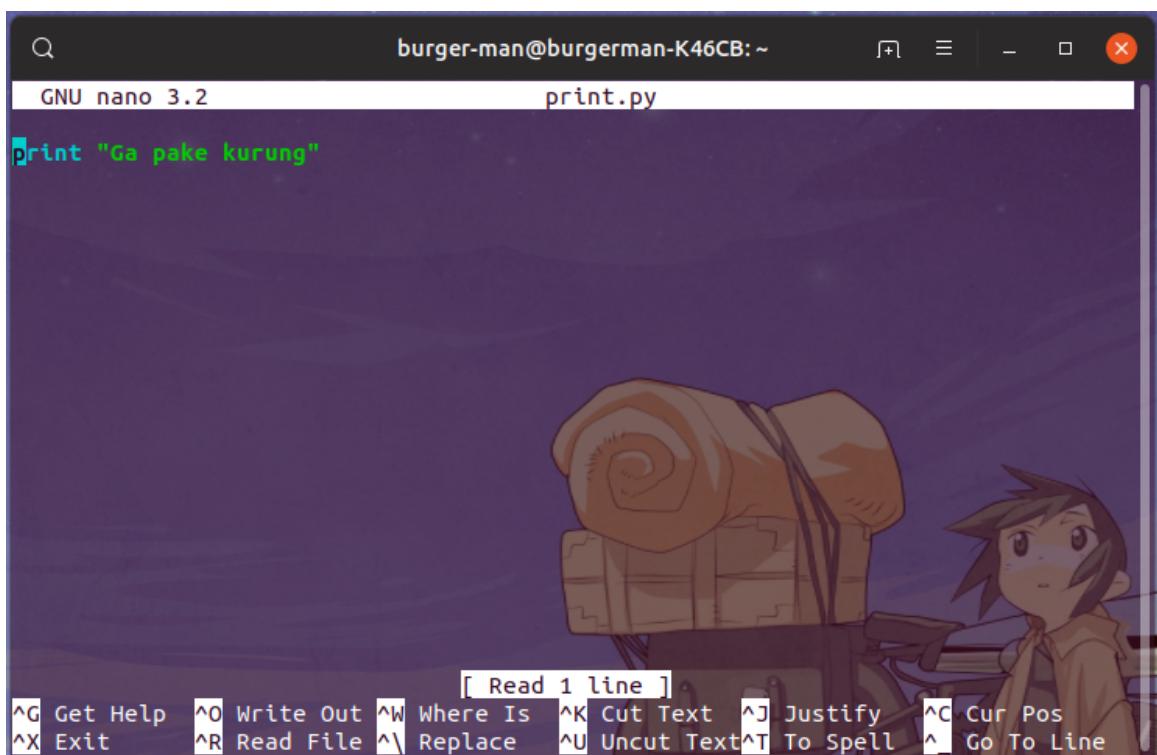
1. Perintah **print** Perbedaan perintah *print* pada dua versi ini adalah python 2.x tidak memakai kurung dan 3.x memakai kurung untuk perintah *print* bisa dilihat pada gambar 1.1 dan 1.2



```
Q burger-man@burgerman-K46CB:~$ python2 print.py
Ga pake kurung
(base) burger-man@burgerman-K46CB:~$ python3 print.py
File "print.py", line 1
    print "Ga pake kurung"
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Ga pake kurung")?
(base) burger-man@burgerman-K46CB:~$
```

The terminal window shows two runs of the script. The first run with Python 2.7 uses the command `python2 print.py` and prints the string "Ga pake kurung" without any parentheses. The second run with Python 3.6 uses the command `python3 print.py` and prints the same string, but it also displays a SyntaxError because the `print` statement lacks parentheses. A tooltip at the bottom of the terminal window says "Did you mean print("Ga pake kurung")?".

Figure 1.1: Gambar hasil print

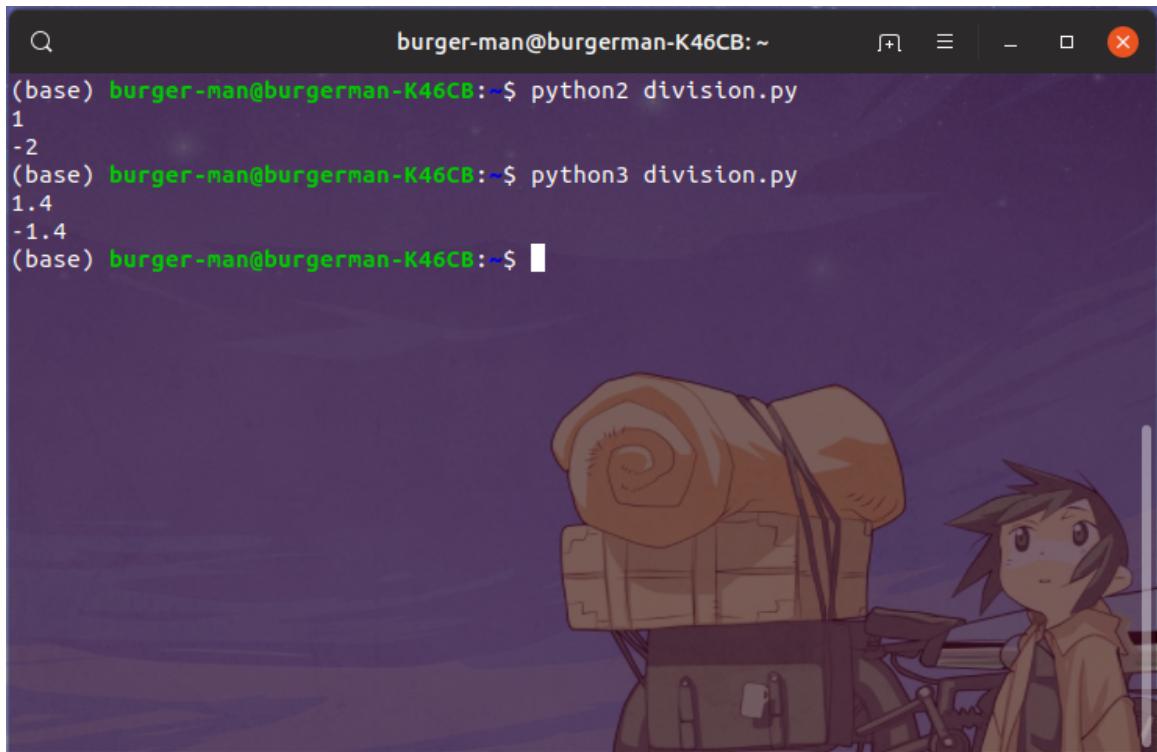


```
Q burger-man@burgerman-K46CB:~$ GNU nano 3.2          print.py
print "Ga pake kurung"
```

The terminal window shows the `GNU nano 3.2` text editor with a single line of Python code: `print "Ga pake kurung"`. The background of the terminal window features a cartoon illustration of a character with brown hair and a large, brown, textured object.

Figure 1.2: Gambar perintah print

2. Perintah pembagian **integer** Hasil dari perintah pembagian cukup jelas berbeda yang mana versi 2.x tidak secara mendetail untuk hasilnya sehingga angka yang dihasilkan bilangan **integer** sedangkan versi 3.x bertipe **float** perbedaannya bisa dilihat pada gambar 1.3 dan 1.4.



```
Q burger-man@burgerman-K46CB:~$ python2 division.py
1
-2
(base) burger-man@burgerman-K46CB:~$ python3 division.py
1.4
-1.4
(base) burger-man@burgerman-K46CB:~$
```

Figure 1.3: Gambar hasil pembagian

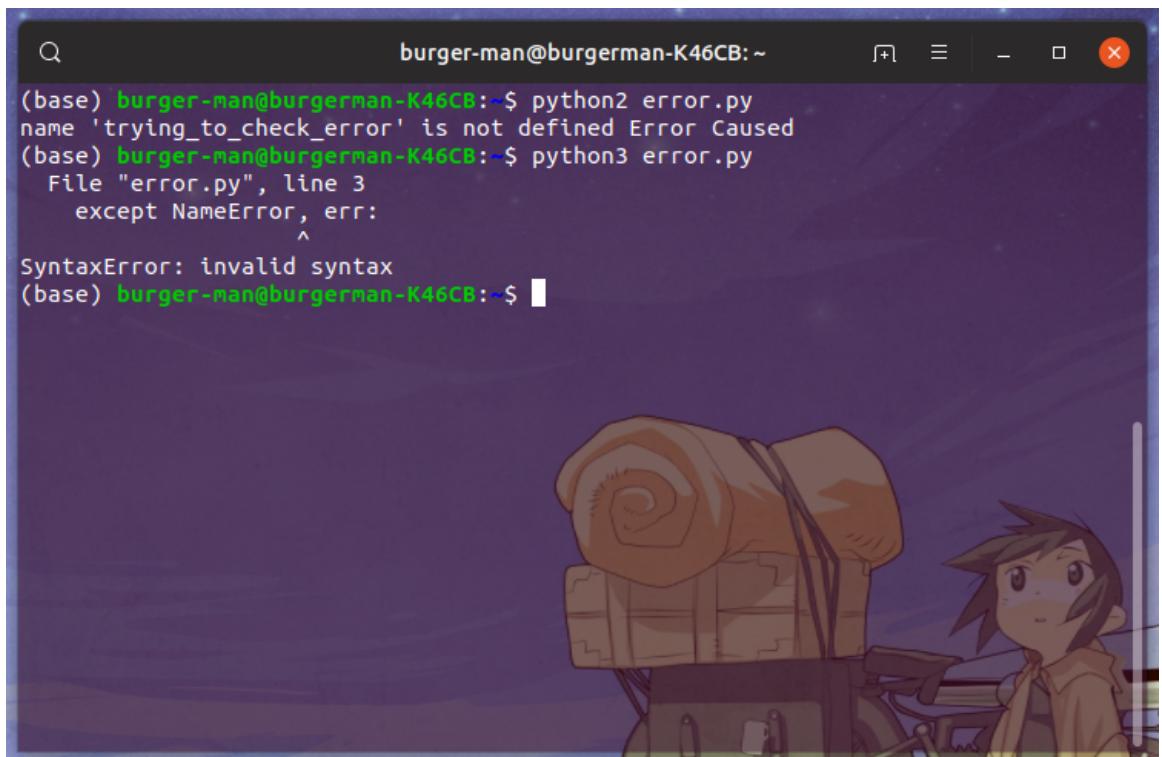
The screenshot shows a terminal window titled "division.py" running on a Linux system. The file contains the following Python code:

```
print (7/5)
print (-7/5)
```

The terminal has a dark purple background with a cartoon character of a boy holding a large sword and a large rose. At the bottom, there is a menu bar with various keyboard shortcuts for nano editor commands.

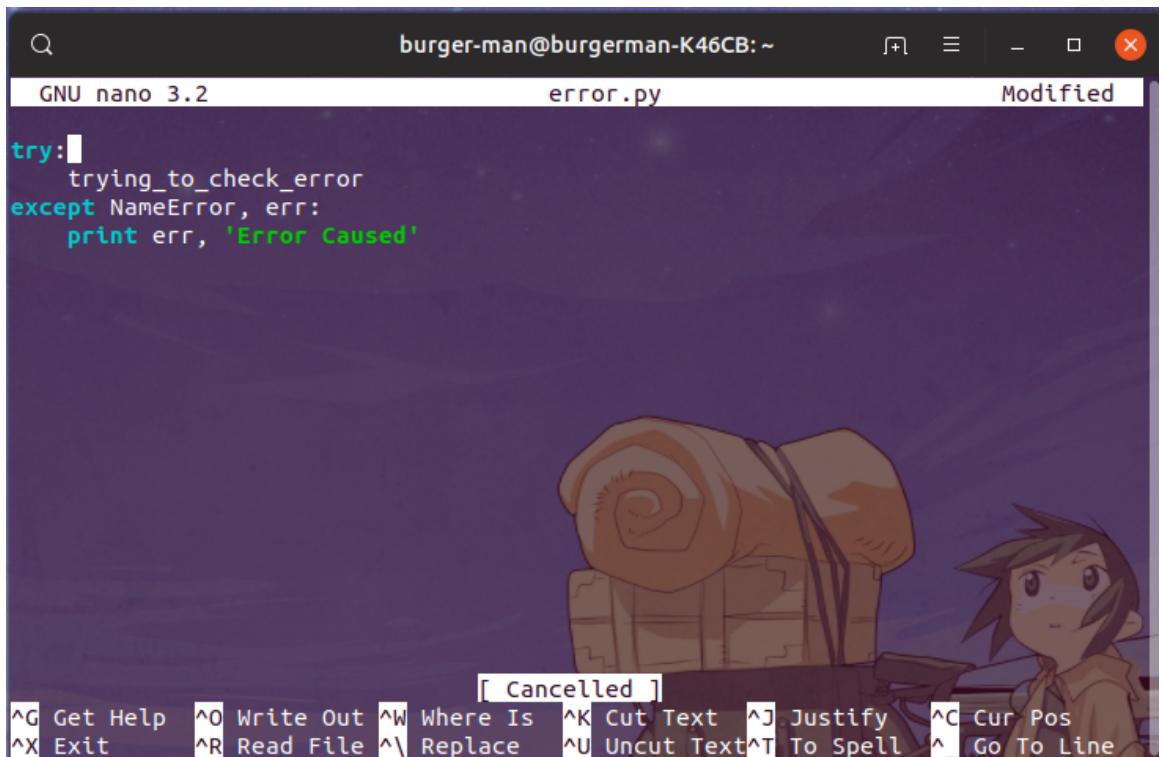
Figure 1.4: Gambar perintah pembagian

3. **Try and Except** Perbedaan pada *try and expcept* hanya berbeda di penggunaan , untuk versi 2.x dan **as** untuk versi 3.x.



```
burger-man@burgerman-K46CB:~$ python2 error.py
name 'trying_to_check_error' is not defined Error Caused
(base) burger-man@burgerman-K46CB:~$ python3 error.py
  File "error.py", line 3
    except NameError, err:
                    ^
SyntaxError: invalid syntax
(base) burger-man@burgerman-K46CB:~$
```

Figure 1.5: Gambar hasil error



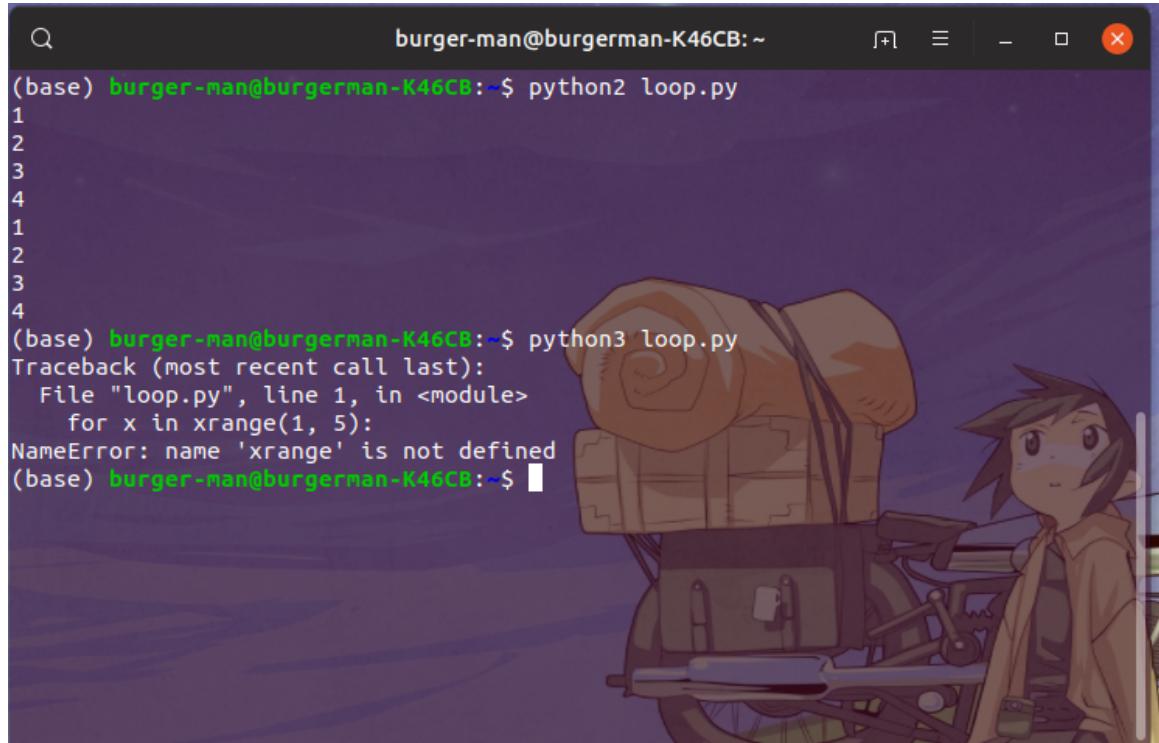
```
GNU nano 3.2          error.py          Modified
try:
    trying_to_check_error
except NameError, err:
    print err, 'Error Caused'
```

[Cancelled]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos
^X Exit **^R** Read File **^** Replace **^U** Uncut Text **^T** To Spell **^** Go To Line

Figure 1.6: Gambar perintah error

4. **Looping** Perbedaan pada looping hanya saja versi 3.x tidak bisa menggunakan sintaks *xrange* lagi.



```
Q burger-man@burgerman-K46CB: ~
(base) burger-man@burgerman-K46CB: ~$ python2 loop.py
1
2
3
4
(base) burger-man@burgerman-K46CB: ~$ python3 loop.py
Traceback (most recent call last):
  File "loop.py", line 1, in <module>
    for x in xrange(1, 5):
NameError: name 'xrange' is not defined
(base) burger-man@burgerman-K46CB: ~$ █
```

Figure 1.7: Gambar hasil looping

The screenshot shows a terminal window titled "loop.py" in the "Modified" state. The file contains two identical loops:`for x in xrange(1, 5):
 print(x)

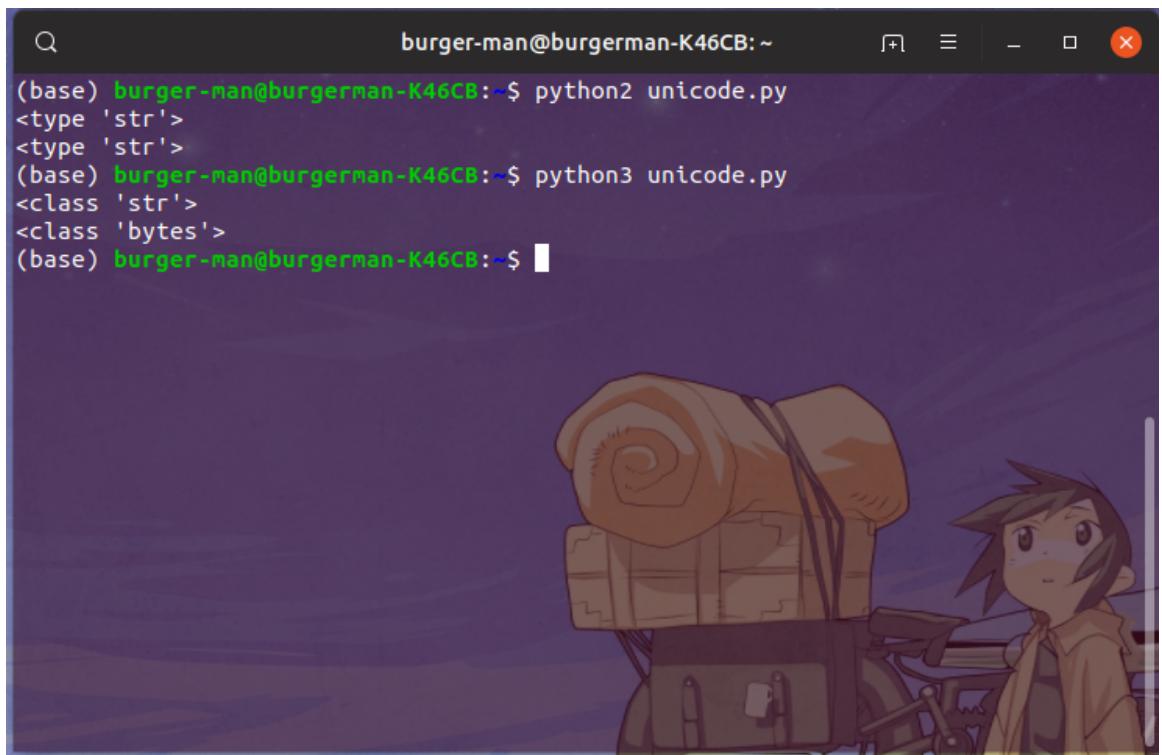
for x in range(1, 5):
 print(x)`

The terminal has a purple background featuring a cartoon character with large brown boots and a brown jacket. The bottom of the screen displays a menu bar with various keyboard shortcuts:

- [Cancelled]
- ^G Get Help**
- ^O Write Out**
- ^W Where Is**
- ^K Cut Text**
- ^J Justify**
- ^C Cur Pos**
- ^X Exit**
- ^R Read File**
- ^| Replace**
- ^U Uncut Text**
- ^T To Spell**
- ^ Go To Line**

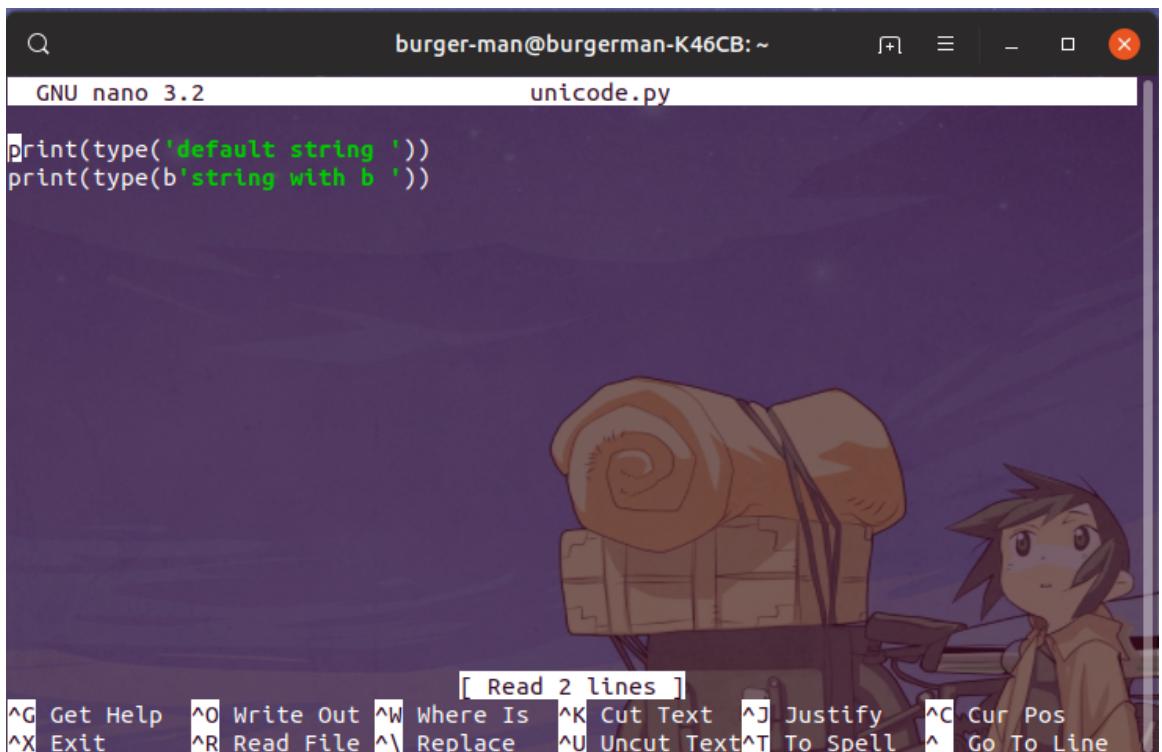
Figure 1.8: Gambar perintah looping

5. **Unicode** Unicode ini cukup penting karena kita mengetahui bagaimana setiap versi merespons setiap unicode yang diberikan.



```
burger-man@burgerman-K46CB:~$ python2 unicode.py
<type 'str'>
<type 'str'>
burger-man@burgerman-K46CB:~$ python3 unicode.py
<class 'str'>
<class 'bytes'>
burger-man@burgerman-K46CB:~$
```

Figure 1.9: Gambar hasil unicode (bytes)



```
GNU nano 3.2                                unicode.py

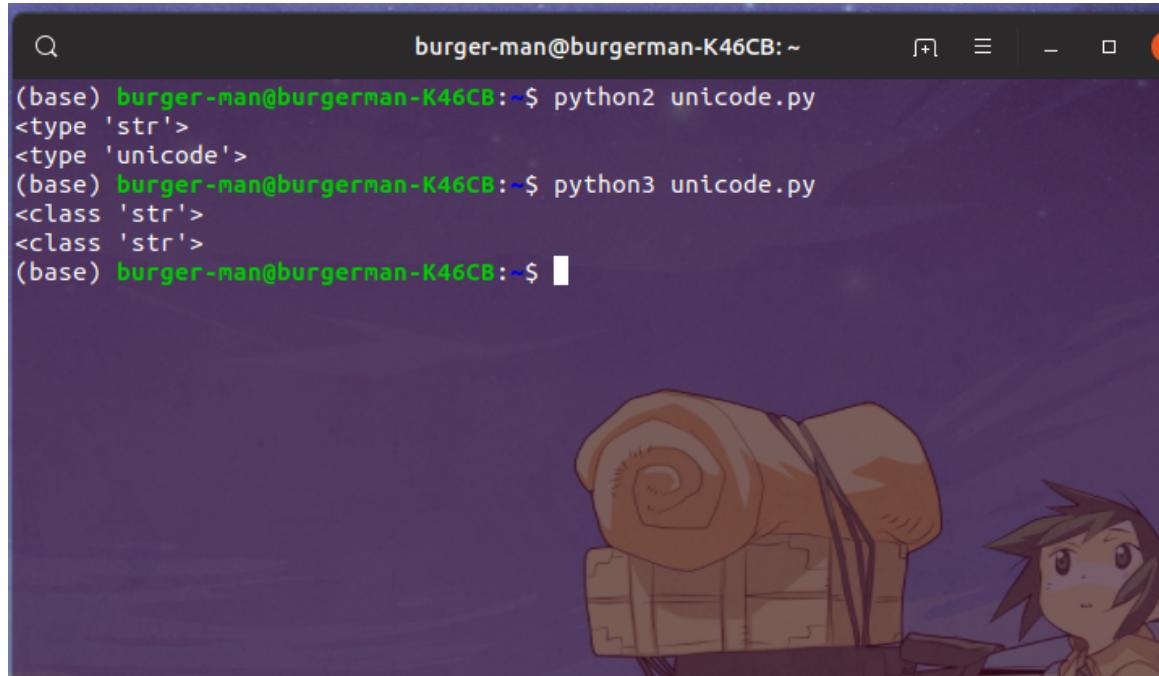
print(type('default string '))
print(type(b'string with b '))
```

[Read 2 lines]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos
^X Exit **^R** Read File **^** Replace **^U** Uncut Text **^T** To Spell **^** Go To Line

Figure 1.10: Gambar perintah unicode (bytes)

Pada gambar 1.9 terlihat jelas bahwa perintah ***bytes*** hanya direspon pada versi 3.x sedangkan versi 2.x merespon ***string***



```
(base) burger-man@burgerman-K46CB:~$ python2 unicode.py
<type 'str'>
<type 'unicode'>
(base) burger-man@burgerman-K46CB:~$ python3 unicode.py
<class 'str'>
<class 'str'>
(base) burger-man@burgerman-K46CB:~$ █
```

Figure 1.11: Gambar hasil unicode

```
burger-man@burgerman-K46CB:~  
GNU nano 3.2          unicode.py  
  
print(type('default string '))  
print(type(u'string with u '))  
  
[ Read 2 lines ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell ^L Go To Line
```

Figure 1.12: Gambar perintah unicode

Chapter 2

Implementasi dan Perusahaan Pengguna *Python*

2.1 Implementasi *Python*

Banyak hal yang dapat dilakukan oleh python, yaitu:

1. Pendidikan
2. Instansi Pemerintahan
3. Kedokteran
4. Transportasi
5. Robotik
6. Dan lain-lain

2.2 Perusahaan Pengguna *Python*

1. Facebook, Facebook menggunakan framework *tornado* untuk menampilkan timeline.
2. Instagram, Instagram menggunakan Django yaitu framework website python sebagai mesin pengolah sisi server dari aplikasinya.
3. Google, pada mesin pencarian Google memakai bahasa pemrograman python untuk mengolahnya.
4. Dropbox, Dropbox menggunakan python baik di sisi server dan di sisi pelayanannya.

Chapter 3

Instalasi dan Konfigurasi *Python*

3.1 Instalasi *Python*

Untuk instalasi python pada Ubuntu 19.04 dibutuhkan sebagai berikut:

1. Internet
2. Anaconda installer (64bit or 32bit)
3. enter, dan yes atau no

Ikuti langkah berikut:

1. Pertama kita kunjungi situs <https://www.anaconda.com/distribution/#download-section> seperti gambar 3.1 dan pilih **64-Bit (x86) Installer (517 MB)**

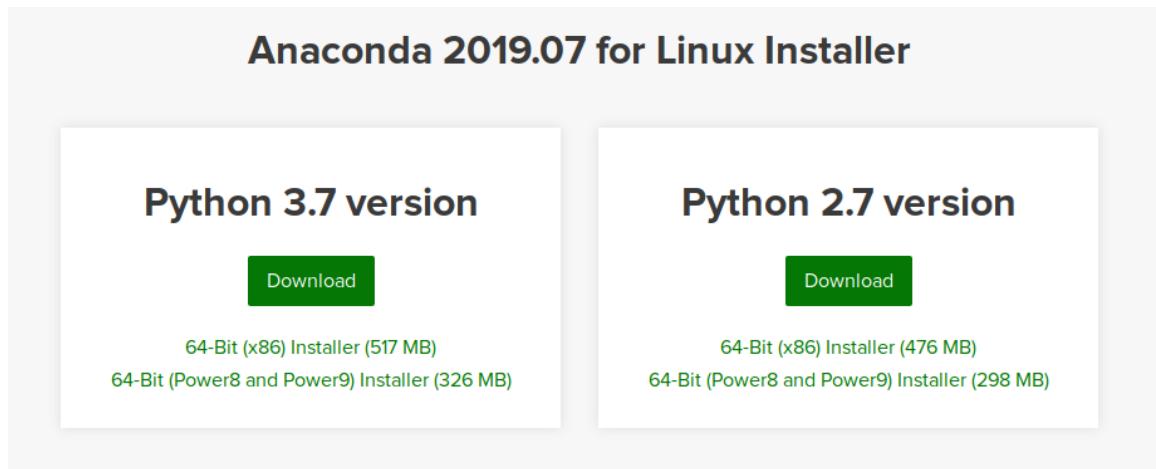


Figure 3.1: Gambar halaman download

2. Kedua kita buka **terminal** kita lalu arahkan ke direktori kita menyimpan file download anaconda

3. Ketiga kita ketikkan sebagai berikut **bash namafileanaconda.sh** lalu enter, contoh seberti gambar 3.2

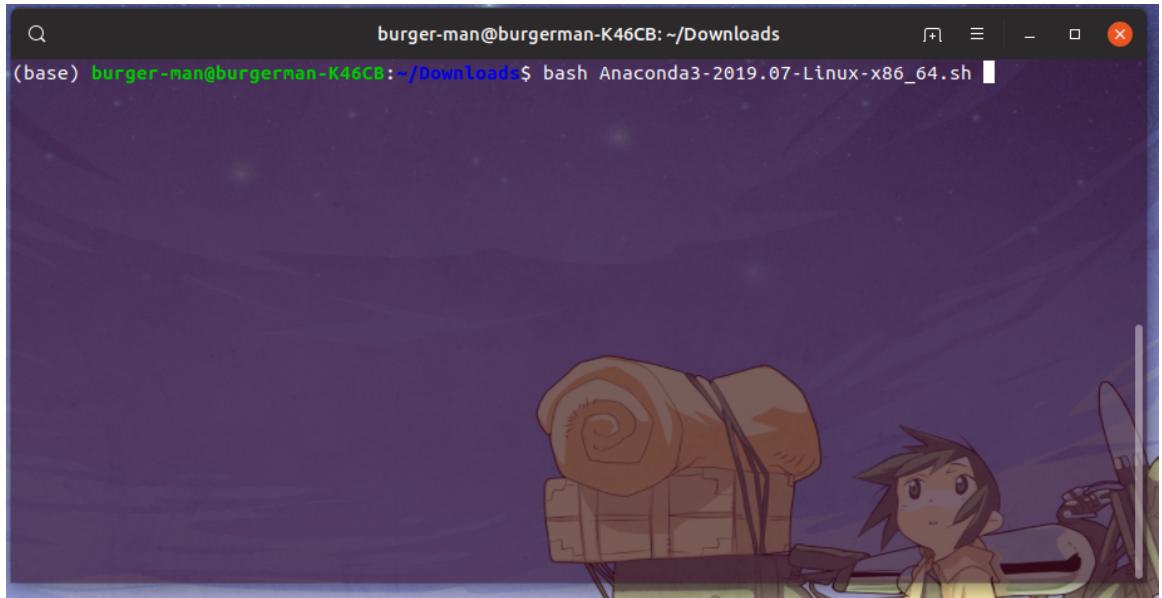


Figure 3.2: Gambar install anaconda

4. Setelah itu, tekan **ENTER** saja seperti gambar 3.3

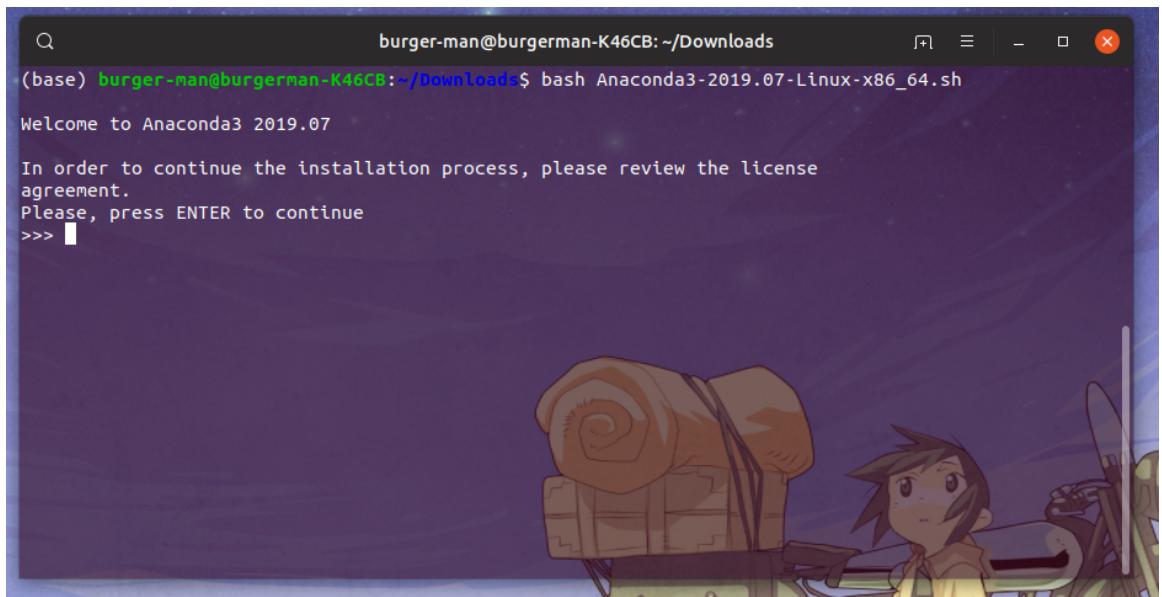
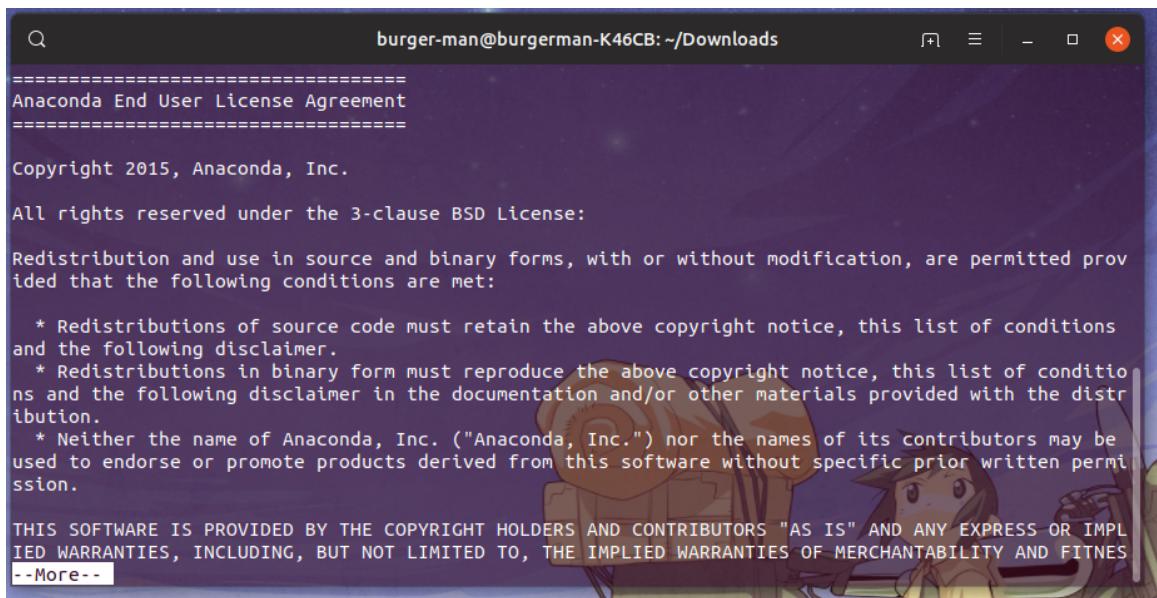


Figure 3.3: Gambar eksekusi anaconda

5. Lalu akan muncul sebuah tulisan **End User License Agreement** seperti gambar 3.4, tekan **ENTER** dan tahan hingga seperti gambar



```
Q burger-man@burgerman-K46CB: ~/Downloads
=====
Anaconda End User License Agreement
=====

Copyright 2015, Anaconda, Inc.

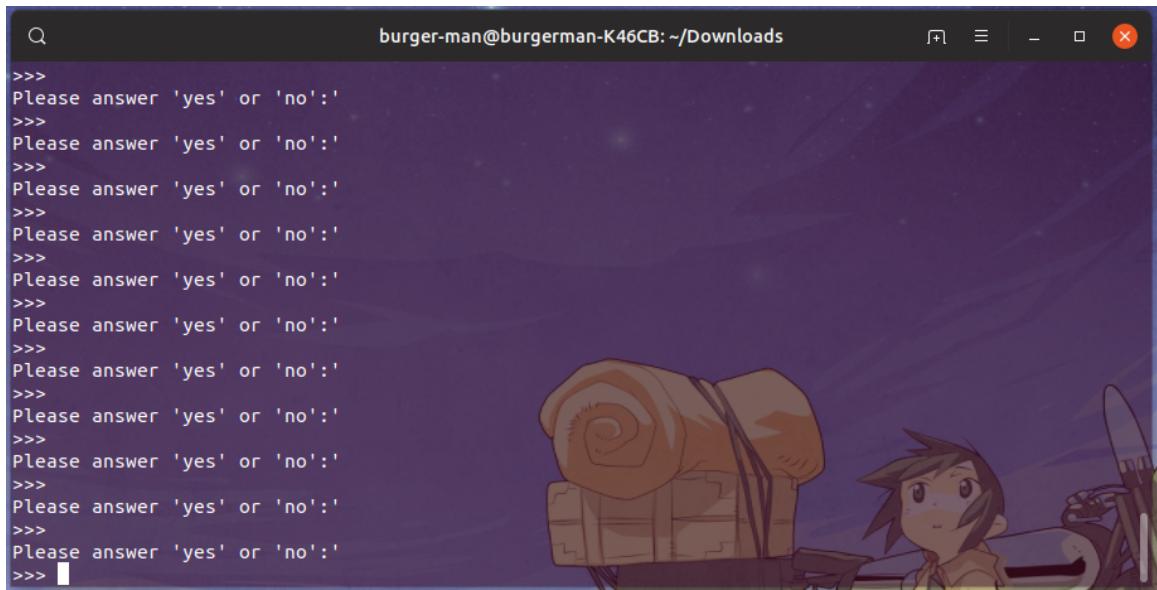
All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
--More--
```

Figure 3.4: Gambar anaconda license agreement



```
>>>
Please answer 'yes' or 'no':'
>>>
```

Figure 3.5: Gambar perintah yes or no

6. Lalu setelah muncul perintah '**yes**' or '**no**' ketik **yes** lalu enter
7. Setelah itu muncul path direktori instalasi anaconda kita seperti gambar 3.6 lalu tekan enter



```
Q burger-man@burgerman-K46CB: ~/Downloads
Please answer 'yes' or 'no':'
>>>
Please answer 'yes' or 'no':'
>>> yes
Anaconda3 will now be installed into this location:
/home/burger-man/anaconda3
- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below
[/home/burger-man/anaconda3] >>> 
```

Figure 3.6: Gambar path anaconda

Setelah kita selesai instalasi anaconda jangan lupa juga untuk menginstal spyder ide, caranya seperti berikut:

- (a) ketikkan perintah ***sudo apt install spyder3 -y*** seperti gambar 3.7



```
Q burger-man@burgerman-K46CB: ~
(base) burger-man@burgerman-K46CB:~$ sudo apt install spyder3 -y
```

Figure 3.7: Gambar perintah install spyder3

(b) lalu jalankan dengan perintah *spyder* atau *spyder3*

3.2 Konfigurasi *Python*

Setelah kita selesai instal Anaconda dan Spyder, selanjutnya kita akan mempelajari bagaimana cara setting environments python kita? caranya sebagai berikut

1. pertama kita buka terminal kita lalu ketikkan perintah **export PYTHONPATH=\$PYTHONPATH:*pathinstallasipythonkalian*** contoh seperti gambar 3.8, lalu enter

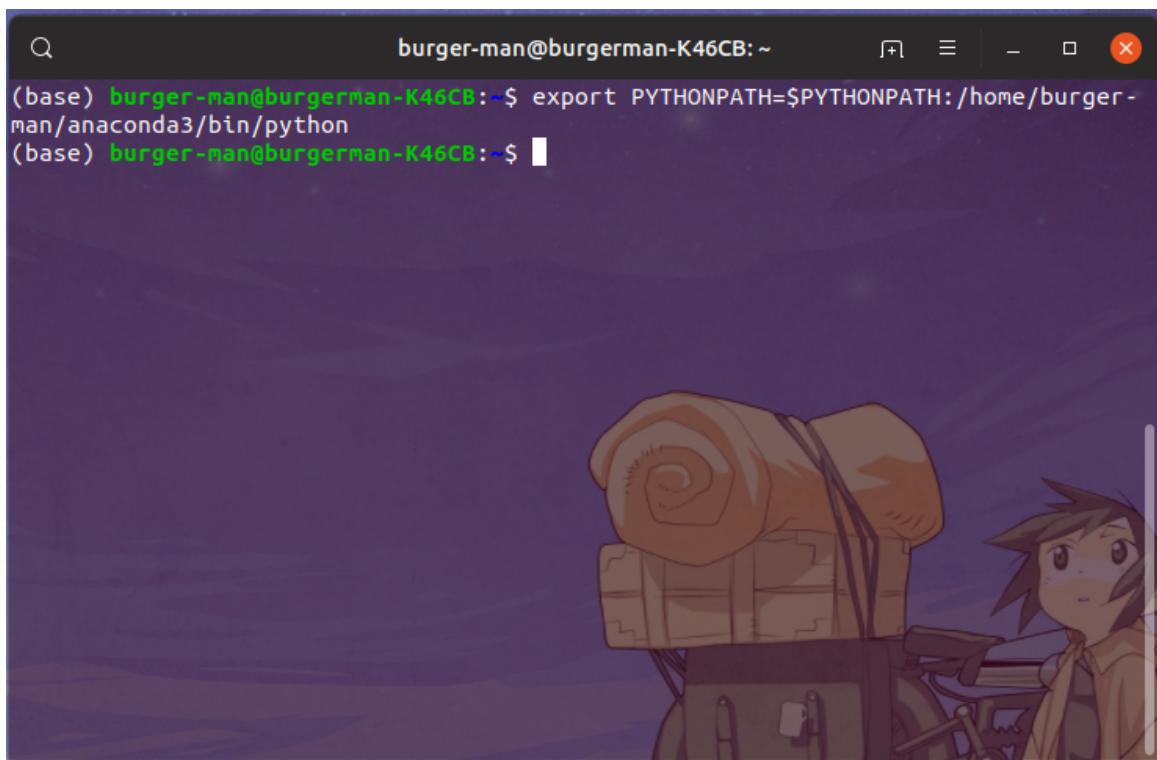
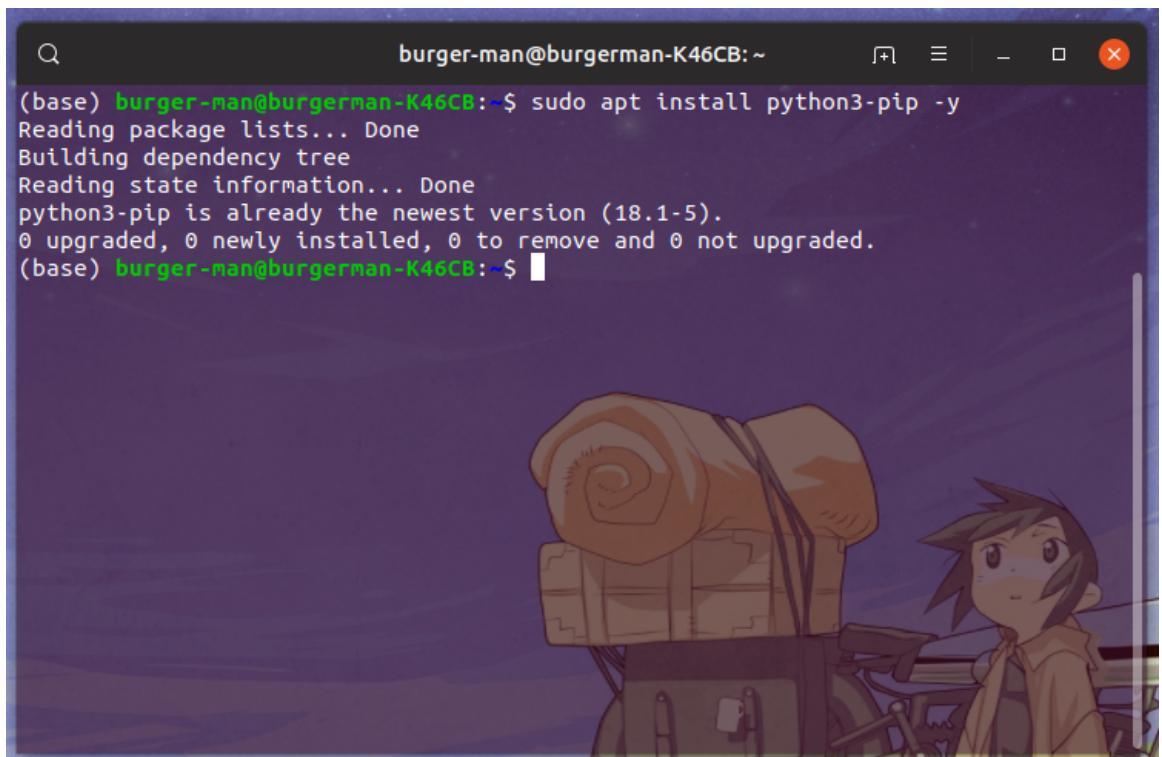


Figure 3.8: Gambar setpath

3.3 Instalasi *pip*

Apa itu pip? pip adalah sebuah standard package atau sebuah perintah yang digunakan untuk menginstal seluruh kebutuhan package yang akan digunakan, caranya sebagai berikut

1. pertama kita buka terminal kita lalu ketikkan perintah **sudo apt install python3-pip -y** untuk pip3 dan **sudo apt install python2-pip -y** untuk pip2 contoh seperti gambar 3.9, lalu enter



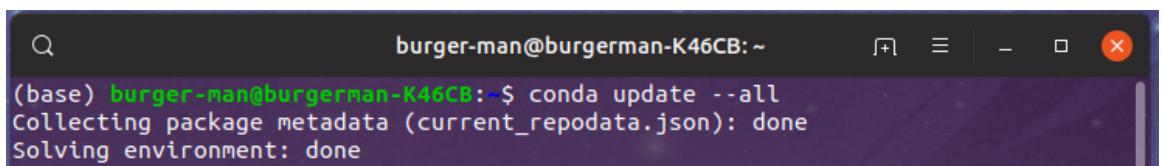
```
Q burger-man@burgerman-K46CB:~
(base) burger-man@burgerman-K46CB:~$ sudo apt install python3-pip -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (18.1-5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
(base) burger-man@burgerman-K46CB:~$
```

Figure 3.9: Gambar instal pip

3.4 Update anaconda dan spyder

Kadang kala anaconda dan spyder melakukan update software terbaru, lalu bagaimana caranya update software tanpa uninstall? berikut caranya

1. Pertama buka terminal dan ketikkan **conda update --all** seperti gambar ??, lalu enter



```
Q burger-man@burgerman-K46CB:~
(base) burger-man@burgerman-K46CB:~$ conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Figure 3.10: Gambar update anaconda

2. Tunggu proses berjalan hingga muncul seperti gambar 3.11, lalu ketik **y** lalu enter

```
burger-man@burgerman-K46CB:~  
The following NEW packages will be INSTALLED:  
  ripgrep          pkgs/main/linux-64::ripgrep-0.10.0-hc07d326_0  
The following packages will be REMOVED:  
  _anaconda_depends-2019.03-py37_0  
The following packages will be UPDATED:  
  anaconda           custom-py37_1 --> 2019.10-py37_0  
  conda-build        3.18.8-py37_0 --> 3.18.9-py37_3  
  numba              0.44.1-py37h962f231_0 --> 0.45.1-py37h962f  
  python             3.7.3-h0371630_0 --> 3.7.4-h265db76_1  
The following packages will be DOWNGRADED:  
  tbb                2019.8-hfd86e86_0 --> 2019.4-hfd86e86_0  
Proceed ([y]/n)?
```

Figure 3.11: Gambar update anaconda

3. Lalu tunggu hingga proses download dan instalasi selesai

Update Spyder: Ada dua cara untuk update spyder yaitu dengan pip atau dengan conda, kali ini saya memakai anaconda sebagai perantara update spyder, sebagai berikut

1. Pertama buka terminal dan ketikkan **conda update spyder** seperti gambar 3.12, lalu enter

```
Q burger-man@burgerman-K46CB: ~ └─ x  
(base) burger-man@burgerman-K46CB:~$ conda update spyder  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
# All requested packages already installed.  
(base) burger-man@burgerman-K46CB:~$ █
```



Figure 3.12: Gambar update spyder

2. Dalam kasus ini spyder saya sudah up-to-date sehingga tidak ada update dari spyder, jikalau ada, maka tinggal ketik **y** lalu enter dan tunggu hingga proses download dan instalasi selesai

Chapter 4

Menjalankan *Python*

4.1 Menjalankan script python dengan perintah CLI

Untuk menjalankan perintah CLI cukup mudah yaitu sebagai berikut

1. Buka terminal lalu ketikkan **python *namafile.py*** seperti gambar 4.1, lalu enter

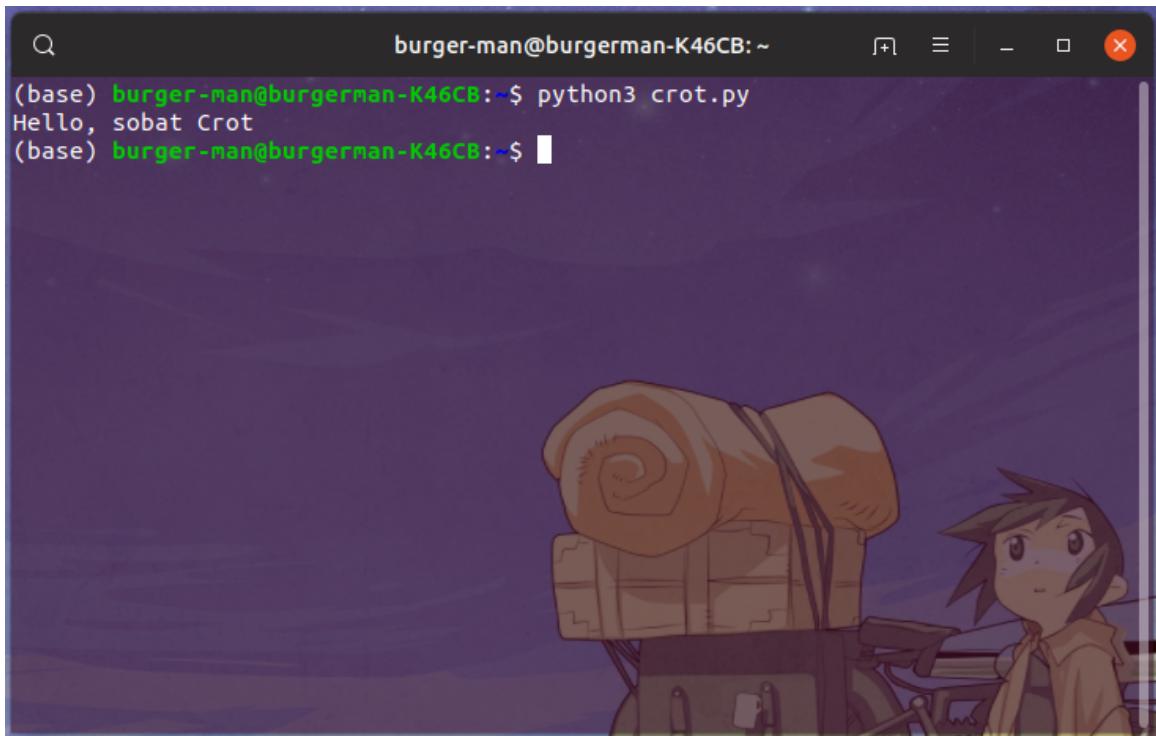


Figure 4.1: Gambar running script dengan CLI

4.2 Hello World!

Sekarang kita akan menjalankan perintah Hello World! pada IDE spyder caranya sebagai berikut

1. Buka IDE spyder
2. Lalu kita ketikkan perintah seperti gambar 4.2, lalu kita run dengan memencet tombol **F5** atau dengan mengklik tombol play di bagian toolbar IDE

The screenshot shows the Spyder IDE interface. On the left is a file browser with icons for Texmaker, a search tool, and a terminal. The main area has tabs for 'Editor' and 'Console'. The 'Editor' tab is active, displaying a file named 'temp.py' with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 hello = "Hello World"
9
10 print(hello)
```

To the right of the editor is a 'Variable explorer' window showing a single variable 'hello' of type str with value 'Hello World'. Below the editor is the 'IPython console' window, which shows the execution of the script:

```
In [1]: runfile('/home/burger-man/.config/spyder-py3/temp.py', wdir='/home/burger-man/.config/spyder-py3')
hello world

In [2]:
```

At the bottom of the interface, status information is displayed: Permissions: RW, End-of-lines: LF, Encoding: UTF-8, Line: 8, Column: 17, Memory: 19 %.

Figure 4.2: Gambar running script dengan spyder

4.3 Auto Login SIAP

Now, kita akan mencoba menjalankan script python auto login ke halaman siap.poltekpos.ac.id, let'ssss goooo

1. kunjungi laman <https://github.com/trianggadios/autologinsiap> seperti gambar 4.3

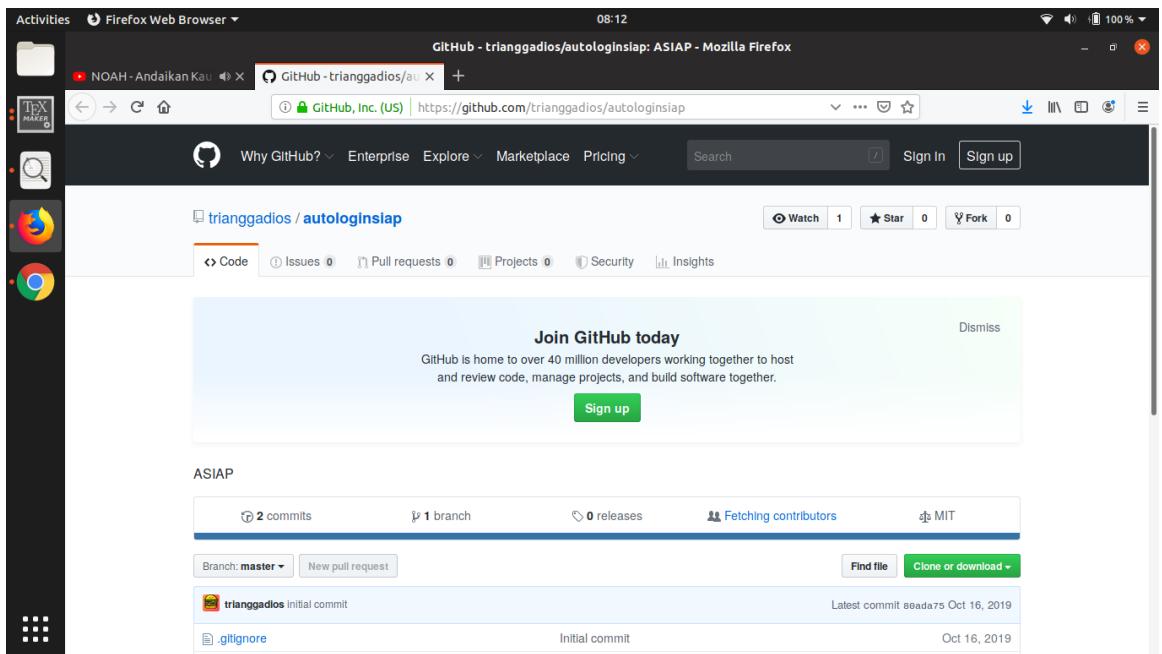


Figure 4.3: Gambar halaman github

2. clone atau download repo tersebut
3. masuk ke direktori repo tersebut dan kita akan menemukan file **main.py** kita klik kanan dan open dengan Spyder IDE seperti gambar 4.4

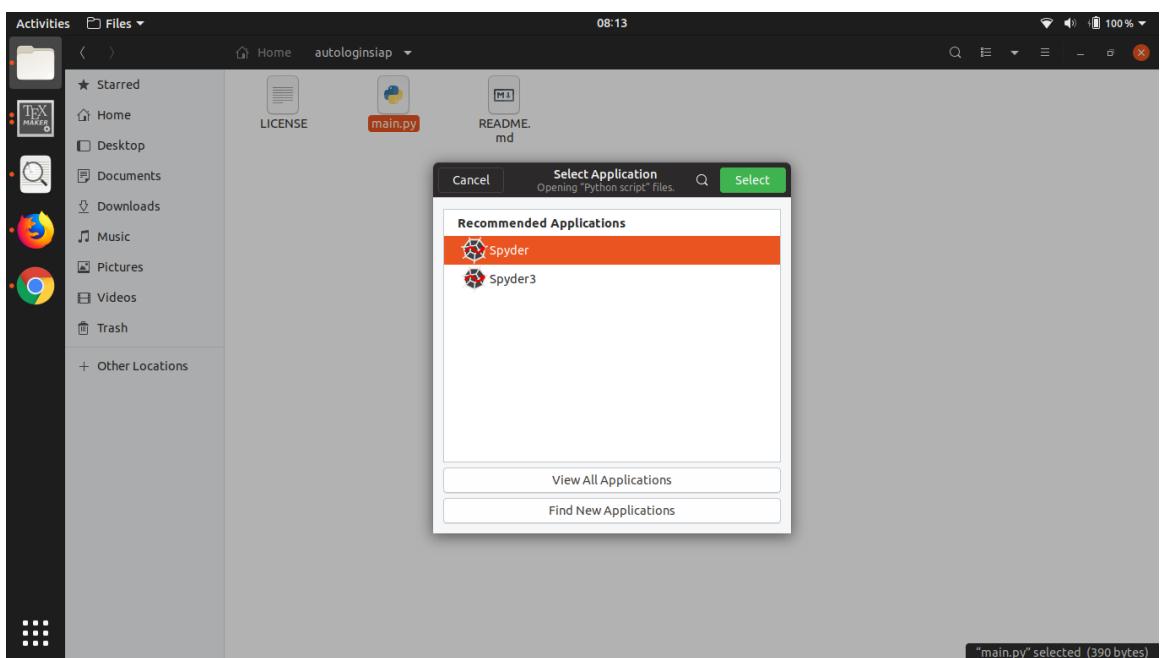


Figure 4.4: Gambar open dengan spyder

- setelah itu kita run file tersebut, dan masukkan npm dan password siap kita dengan benar

4.4 Menggunakan Variable Explorer

Menggunakan Variable Explorer ini cukup simple yaitu fungsinya untuk mengetahui apa nama variabel tersebut, type datanya, jumlahnya, dan isinya apa. Berikut contohnya di gambar 4.5

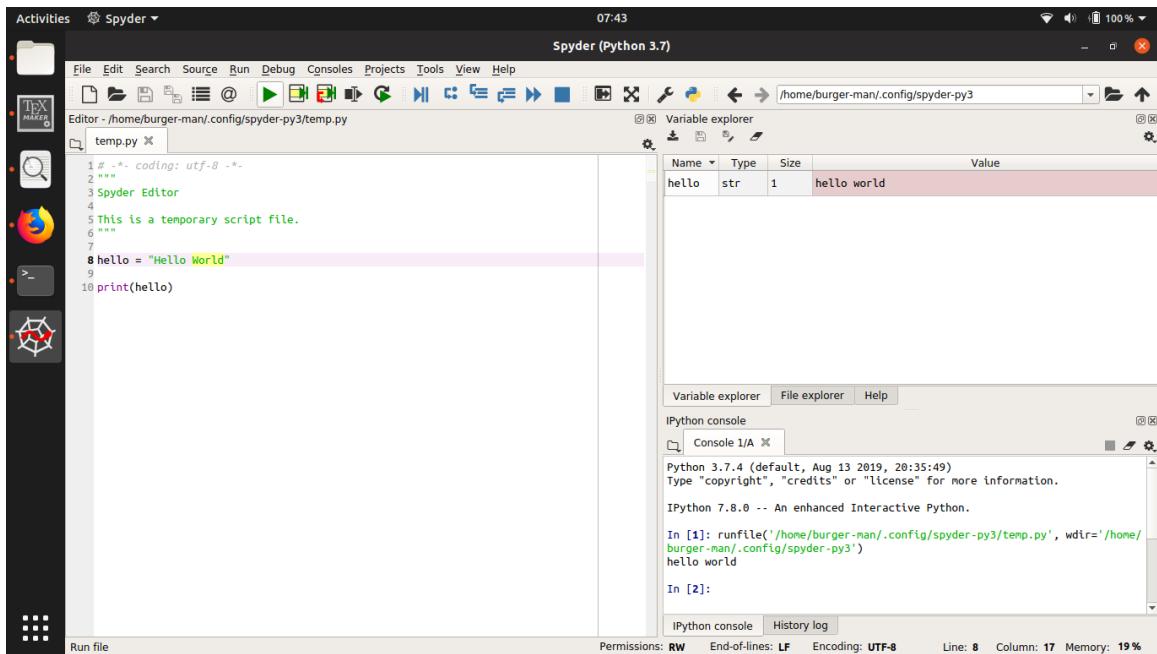


Figure 4.5: Gambar running script dengan spyder

Dikanan atas ada field yang menyebutkan variable explorer dan disitu terlihat bahwa di program yang kita tulis seperti tabel 4.1

name	hello
type data	str
syze	1
value	hello world

Table 4.1: Tabel variabel explorer

4.5 Identasi

Apa itu indentasi? indentasi adalah sebuah aturan python dimana untuk menentukan pembuka dan penutup pada program python biasanya indentasi digunakan setelah

sintaks yang menggunakan tanda :, contoh pada gambar 4.6

The screenshot shows a code editor window with the following Python code:

```
1
2
3
4 if status == 1 :
5     print 'Berhasil'
6     indentasi dengan 4 spasi
7 else :
8     print 'Gagal'
9
10 sama halnya dengan di atas
```

A red callout box highlights the line "indentasi dengan 4 spasi" with the text "indentasi dengan 4 spasi". Another red callout box highlights the line "sama halnya dengan di atas" with the text "sama halnya dengan di atas".

Figure 4.6: Gambar contoh indentasi

Jika indentasi ada yang salah maka akan seperti gambar 4.7

The screenshot shows the Spyder IDE interface with the following code in the editor:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6
7 hello = "Hello World"
8
9 if hello == "Hello World":
10    print (hello)
11 else :
12    print (hello)
```

An error message is displayed in the "Usage" panel:

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

In the IPython console, the following output is shown:

```
execfile(filename, namespace)
File "/home/burger-man/anaconda3/lib/python3.7/site-packages/spyder_kernels/customize/spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)
File "/home/burger-man/.config/spyder-py3/temp.py", line 11
    print (hello)
          ^
IndentationError: expected an indented block
```

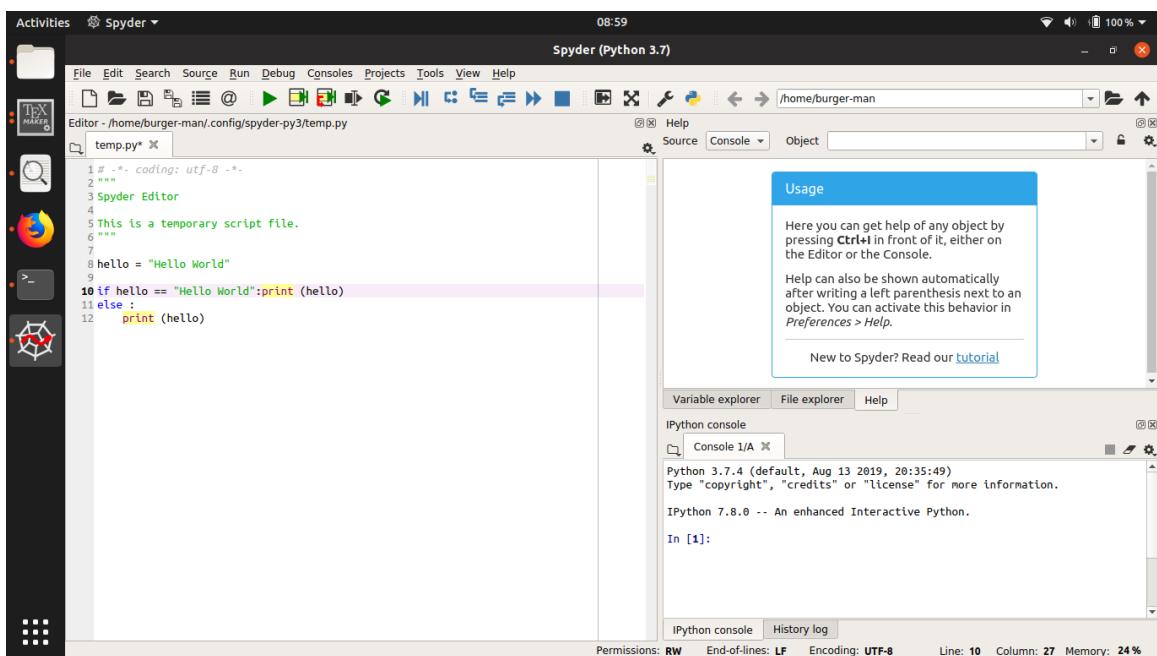
Figure 4.7: Gambar contoh error indentasi

Gambar diatas jelas sekali salah karena setelah sintaks **if hello == "hello world"** : yang memiliki tanda : tidak menjorok sebesar 4 spasi

Cara membaca error tersebut adalah dengan membaca line berapa yang error? pada gambar 4.7 terlihat jelas bahwa yang error **line 11** sehingga kita mengetahui line berapa yang error, selanjutnya kita baca apa yang error? disitu tertulis yang error adalah ***expected and indented block*** yang artinya di line 11 indentasinya salah atau ada yang terblokir di proses itu.

sehingga penanganan errornya seperti berikut

1. Kita sejajarkan dahulu line yang bermasalah dengan line yang diatasnya seperti gambar 4.8



The screenshot shows the Spyder Python IDE interface. On the left is a file browser with a single file named 'temp.py' selected. The main area is an editor containing the following Python code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 hello = "Hello World"
9
10 if hello == "Hello World":print (hello)
11 else :
12     print (hello)
```

A blue tooltip window titled 'Usage' is open over the line '10 if hello == "Hello World":print (hello)'. The tooltip provides information on how to get help for objects in the editor or console. Below the editor is an IPython console window showing the following output:

```
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]:
```

At the bottom of the interface, status bars show 'Permissions: RW', 'End-of-lines: LF', 'Encoding: UTF-8', 'Line: 10', 'Column: 27', and 'Memory: 24 %'.

Figure 4.8: Gambar penanganan indentasi 1

2. setelah itu kita tekan enter hingga menjadi benar seperti gambar 4.9

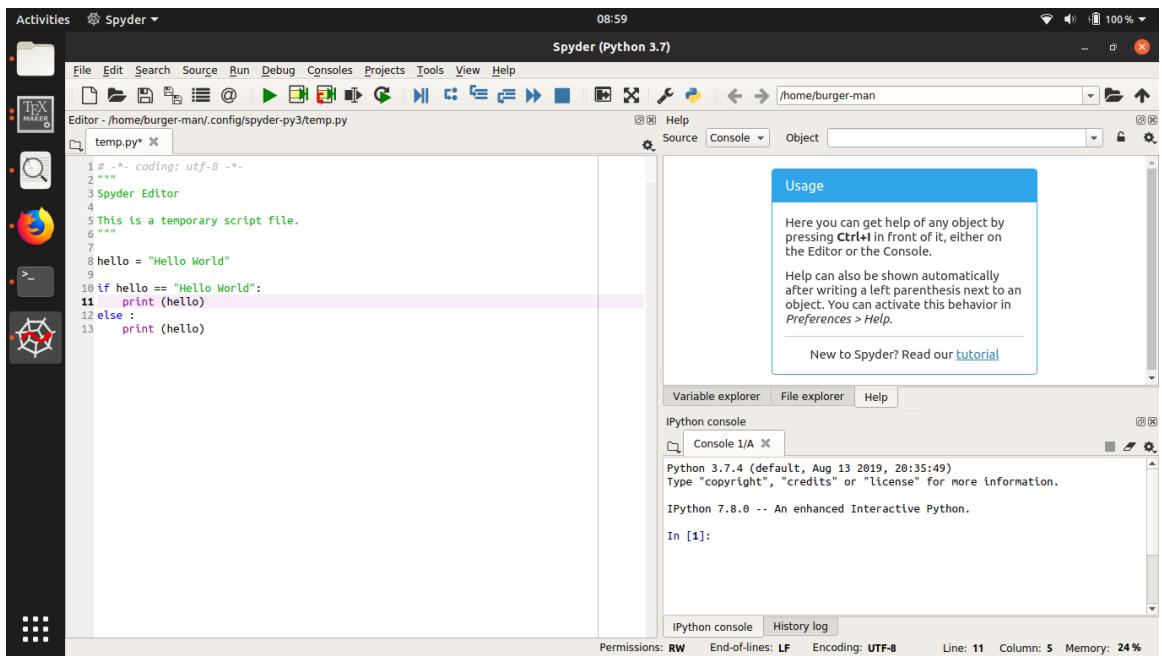


Figure 4.9: Gambar penanganan indentasi 2