

Nama : Fajar Setia Pratama

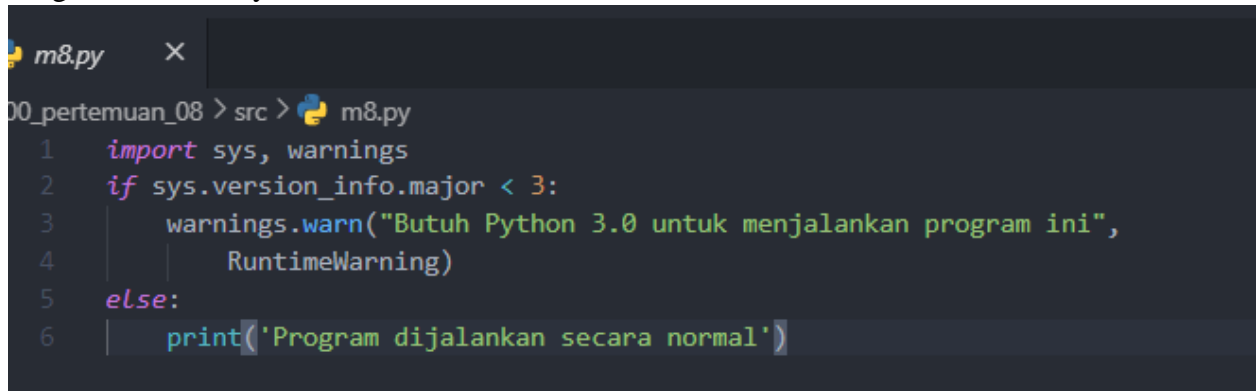
NIM : 145410085

MATKUL : WORKSHOP PYTHON

LINK GITHUB : https://github.com/SetiaP/workshop_python.git

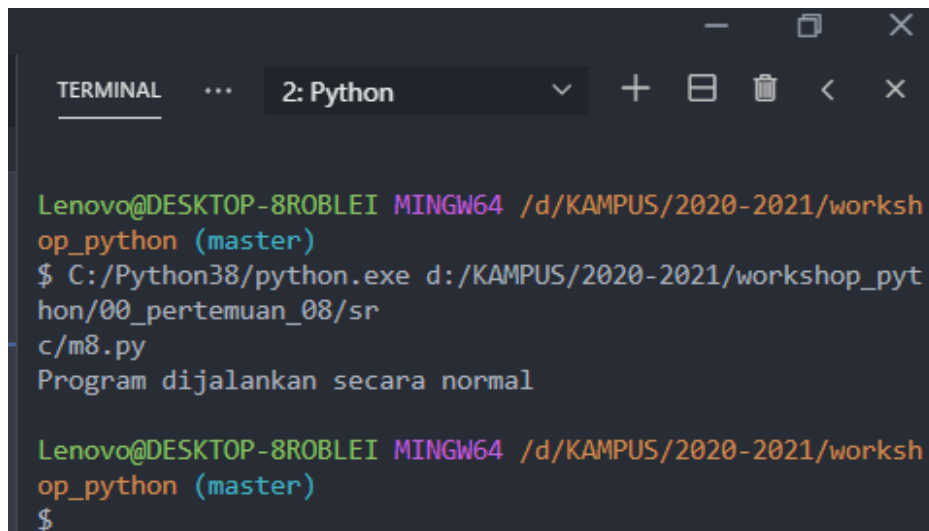
PERTEMUAN 8 : Daftar Pustaka

1. Program 1 modul sys



```
m8.py x
00_pertemuan_08 > src > m8.py
1 import sys, warnings
2 if sys.version_info.major < 3:
3     warnings.warn("Butuh Python 3.0 untuk menjalankan program ini",
4                 RuntimeWarning)
5 else:
6     print('Program dijalankan secara normal')
```

Keluaran



```
TERMINAL ... 2: Python
Lenovo@DESKTOP-8ROBLEI MINGW64 /d/KAMPUS/2020-2021/worksh
op_python (master)
$ C:/Python38/python.exe d:/KAMPUS/2020-2021/workshop_pyth
on/00_pertemuan_08/src
c/m8.py
Program dijalankan secara normal

Lenovo@DESKTOP-8ROBLEI MINGW64 /d/KAMPUS/2020-2021/worksh
op_python (master)
$
```

Bagaimana Cara Kerjanya:

Kita menggunakan modul dari pustaka standar dengan nama warnings yang digunakan untuk menampilkan pesan kesalahan kepada pengguna. Jika versi Python yang digunakan kurang dari 3, kita akan tampilkan pesan kesalahan terkait.

2. Menggunakan Logging

Nama : Fajar Setia Pratama

NIM : 145410085

MATKUL : WORKSHOP PYTHON

LINK GITHUB : https://github.com/SetiaP/workshop_python.git

```
m8.py  gunakan_logging.py X
00_pertemuan_08 > src > gunakan_logging.py > ...
1  import glob
2  import logging
3  import logging.handlers
4
5  LOG_FILENAME = 'logging_rotatingfile_example.out'
6
7  # Set up a specific logger with our desired output level
8  my_logger = logging.getLogger('MyLogger')
9  my_logger.setLevel(logging.DEBUG)
10
11 # Add the log message handler to the logger
12 handler = logging.handlers.RotatingFileHandler(
13     LOG_FILENAME, maxBytes=20, backupCount=5)
14
15 my_logger.addHandler(handler)
16
17 # Log some messages
18 for i in range(20):
19     my_logger.debug('i = %d' % i)
20
21 # See what files are created
22 logfiles = glob.glob('%s*' % LOG_FILENAME)
23
24 for filename in logfiles:
25     print(filename)
26
27
```

Keluaran

```
Lenovo@DESKTOP-8ROBLEI MINGW64 /d/KAMPUS/2020-2021/worksh
op_python (master)
$ C:/Python38/python.exe d:/KAMPUS/2020-2021/workshop_py
hon/00_pertemuan_08/sr
c/gunakan_logging.py
logging_rotatingfile_example.out
logging_rotatingfile_example.out.1
logging_rotatingfile_example.out.2
logging_rotatingfile_example.out.3
logging_rotatingfile_example.out.4
logging_rotatingfile_example.out.5
```

3. Modul penyimpanan pesan (logging)

```
m8.py  gunakan_logging.py  coba_logging.py X
00_pertemuan_08 > src > coba_logging.py > ...
1  import os, platform, logging
2
3  if platform.platform().startswith('Windows'):
4      berkas_logging = os.path.join(os.getenv('HOMEDRIVE'), os.getenv('HOMEPATH'), 'test.log')
5  else:
6      berkas_logging = os.path.join(os.getenv('HOME'), 'test.log')
7
8  print("Logging ke", berkas_logging)
9
10 logging.basicConfig(
11     level=logging.DEBUG,
12     format='%(asctime)s : %(levelname)s : %(message)s',
13     filename = berkas_logging,
14     filemode = 'w',
15 )
16
17 logging.debug("Memulai program")
18 logging.info("Mengerjakan sesuatu")
19 logging.warning("Matikan sekarang")
```

Nama : Fajar Setia Pratama

NIM : 145410085

MATKUL : WORKSHOP PYTHON

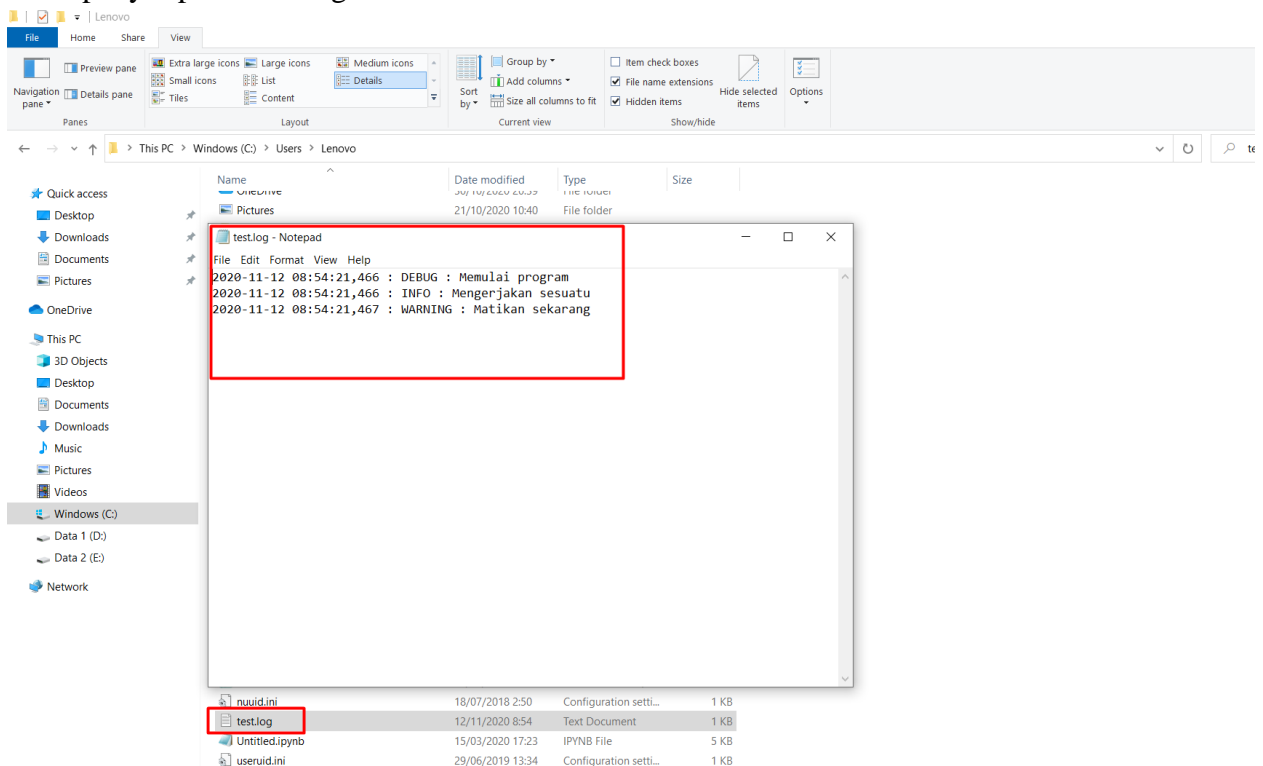
LINK GITHUB : https://github.com/SetiaP/workshop_python.git

Keluaran

```
Lenovo@DESKTOP-8R0BLEI MINGW64 /d/KAMPUS/2020-2021/worksh
op_python (master)
$ C:/Python38/python.exe d:/KAMPUS/2020-2021/workshop_py
hon/00_pertemuan_08/sr
c/coba_logging.py
Logging ke C:\Users\Lenovo\test.log

Lenovo@DESKTOP-8R0BLEI MINGW64 /d/KAMPUS/2020-2021/worksh
op_python (master)
$
```

Hasil penyimpanan tes.log



Bagaimana Cara Kerjanya:

Kita menggunakan tiga modul dari pustaka standar - modul os untuk interaksi dengan sistem operasi, modul platform untuk mencari informasi tentang platform - sistem operasi dan modul logging untuk menyimpan pesan dan informasi.

Nama : Fajar Setia Pratama

NIM : 145410085

MATKUL : WORKSHOP PYTHON

LINK GITHUB : https://github.com/SetiaP/workshop_python.git

Pertama, kita cek sistem operasi apa yang kita gunakan dengan mencatat string kembalian dari fungsi `platform.platform()` (untuk informasi lebih lanjut, jalankan ``import platform; help(platform)``). Jika Windows, kita cari tahu drive (bagian dari harddisk yang digunakan), mencari folder rumah dan nama berkas tempat kita menyimpan informasi. Dengan menyatukan ketiga informasi tersebut bersama-sama, kita akan mendapatkan lokasi lengkap dari berkas. Untuk platform lain, kita hanya perlu tahu folder rumah dari pengguna yang bersangkutan, dan kita akan mendapatkan lokasi lengkap dari berkas yang dituju.

Kita gunakan fungsi `os.path.join()` untuk menyatukan ketiga hal di atas bersamaan. Alasan penggunaan fungsi khusus untuk menyatukan string tersebut bersama-sama, sebab fungsi ini akan memastikan lokasi lengkapnya (separator/pemisah antar folder) sesuai dengan format yang diharapkan oleh sistem operasi.

Kita konfigurasi modul logging untuk menuliskan seluruh pesan dalam format khusus yang telah kita tentukan sebelumnya.

Terakhir, kita dapat menuliskan pesan hanya untuk debugging, informasi atau pesan kesalahan, bahkan pesan kesalahan yang bersifat kritis. Sekali program berjalan, kita dapat melihat berkas ini dan kita tahu apa yang terjadi di dalam program, meskipun tidak ada informasi yang ditampilkan di depan pengguna langsung.