

Pemrograman Python untuk Pengolahan Citra Digital

Diktat kuliah

DR. ARYA ADHYAKSA WASKITA



STMIK Eresha - 2020

Daftar Isi

Daftar Isi	i
Daftar Gambar	ii
Daftar Program	ii
KATA PENGANTAR	iii
1 Sejarah Pemrograman Python	1
2 Instalasi Python	2
2.1 Interpreter Python	2
2.2 Anaconda	7
3 Dasar Pemrograman Python	13
3.1 Pendahuluan	13
4 Pendahuluan Pustaka Scikit-Image	16
5 Sub Modul Pustaka Scikit-Image	17
Bibliografi	18

Daftar Gambar

2.1	Dialog instalasi <i>interpreter</i> Python	2
2.2	Pilihan paket pendukung sebelum instalasi dilakukan	3
2.3	Dialos selama proses instalasi berlangsung	3
2.4	Dialog tanda selesai instalasi	3
2.5	Lokasi instalasi <i>interpreter</i> Python	4
2.6	<i>Interpreter</i> Python siap digunakan	4
2.7	Daftar paket yang terpasang	5
2.8	Hasil upgrade pip	5
2.9	Instalasi pustaka scikit-image menggunakan pip	5
2.10	Instalasi pustaka <i>dependent</i>	6
2.11	Daftar terakhir paket terpasang	6
2.12	Daftar menu aplikasi pendukung Python	6
2.13	Aplikasi IDLE	7
2.14	Pilihan <i>platform</i> instalasi Anaconda	8
2.15	Dialog pembuka instalasi	8
2.16	Menyetujui kesepakatan	8
2.17	Pilihan pengguna Anaconda	9
2.18	Target instalasi	9
2.19	Menjadikan Anaconda sebagai sistem utama Python	9
2.20	Proses instalasi	10
2.21	Instalasi selesai	10
2.22	10
2.23	Aplikasi Jupyter	11
2.24	Terminal pada aplikasi Jupyter	11
2.25	Python Shell pada aplikasi Jupyter	12
2.26	Aplikasi Spyder	12
3.1	Python shell sedang menerima perintah	14
3.2	Variabel a sebagai obyek	14
3.3	Menampilkan dokumentasi obyek integer a	15

Daftar Program

Kata Pengantar

Diktat kuliah ini hanya merupakan pelengkap agar mahasiswa dapat lebih mudah memahami materi pengolahan citra digital. Penggunaan ilustrasi lain dari perangkat lunak berbayar dapat saja diberikan. Tetapi, karena pertimbangan kemandirian dan lisensi, maka saya memutuskan untuk menyusun diktat ini berbasis pada pustaka berlisensi publik dan berbasis bahasa pemrograman Python, **scikit-image**. Python dipertimbangkan karena banyak pustaka ilmiah yang sudah umum digunakan dan terus dikembangkan yang berbasis pada Python. Dalam pengolahan citra, selain **scikit-image**, ada juga OpenCV untuk *Computer Vision*. Dalam pembelajaran mesin, **scikit-learn** adalah pustaka yang juga banyak digunakan. Bahkan **tensorflow**, pustaka yang banyak digunakan dalam penelitian *deep learning* juga berbasis pada Python. Saya yakin, dengan mempelajari diktat ini, mahasiswa mampu mandiri dalam penguasaan bahasa pemrograman Python yang pada akhirnya mampu membuat mahasiswa lebih adaptif terhadap pustaka berbasis python, baik untuk tujuan ilmiah maupun bisnis. Mahasiswapun diharapkan menjadi lebih kreatif dalam melakukan penelitian hingga mengembangkan produk perangkat lunak, maupun prototipe perangkat keras cerdas berbasis Python tanpa harus terbebani masalah lisensi.

Secara umum, diktat ini dibagi ke dalam bagian pendahuluan yang membahas tentang sejarah singkat Python yang dilanjutkan ke bagian instalasi. Instalasi ini, meskipun sangat sederhana, terutama pada sistem operasi Linux, dapat menjadi sangat merepotkan bagi beberapa mahasiswa, terutama ketika mereka menggunakan sistem operasi Windows. Karena itu, instalasi akan dilakukan di sistem operasi Windows. Bagian selanjutnya adalah dasar-dasar pemrograman Python, terutama struktur data (**list**, **tuple** dan **dictionary**), interaksi dengan *file*, hingga mempelajari penggunaan fungsi yang terdapat dalam pustaka tertentu. Sedangkan bagian terakhir dari diktat ini akan sepenuhnya diisi dengan fitur pustaka **scikit-image**, yang saat diktat ini disusun berada pada rilis 0.16.

Diktat ini banyak menggunakan sumber dari situs web dan akan disampaikan secara detil alamat sumber tersebut dalam diktat. Diharapkan, mahasiswa tidak takut mencoba karena ada begitu banyak sumber yang dapat digunakan untuk belajar. Hanya kesungguhan kitalah yang akan menjadi pembeda. Akhirnya, selamat mencoba pengalaman baru.

Serpong, 26 Maret 2020

Dr. Arya Adhyaksa Waskita

Bab 1

Sejarah Pemrograman Python

Bab 2

Instalasi Python

2.1 Interpreter Python

Seperti telah dijelaskan di bagian Pengantar, instalasi *interpreter* Python dilakukan di sistem operasi Windows 7. Tahapan instalasi ini mengasumsikan bahwa tidak ada kendala apapun terkait sistem operasi. Selanjutnya mahasiswa diminta untuk mengunduh *interpreter* Python melalui laman <https://www.python.org/downloads/> sesuai kebutuhannya.

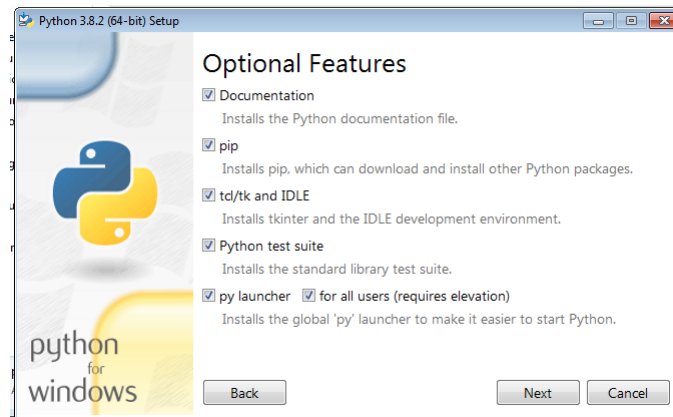
Mengeksekusi unduhan tersebut akan memunculkan dialog seperti pada Gambar 2.1. Pastikan untuk memilih konfigurasi PATH secara otomatis agar ketika proses instalasi selesai, *interpreter* Python dapat dijalankan dari mana saja di sistem komputer masing-masing. Untuk kondisi di mana terjadi kesalahan, akan muncul dialog yang memberi kita kesempatan untuk melihat *log*. Buka log tersebut dan lihat sumber dari kesalahan instalasi yang sedang terjadi.



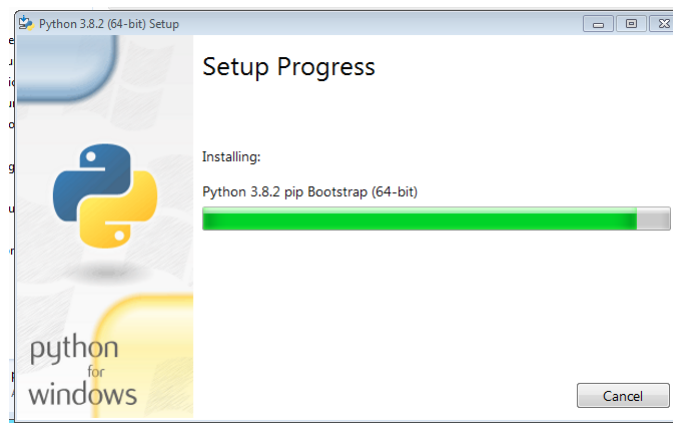
Gambar 2.1: Dialog instalasi *interpreter* Python

Pilihan opsi *Customize installation* akan menampilkan dialog seperti Gambar 2.2. Pastikan semua pilihan dipilih.

Selama proses instalasi berlangsung, pengguna akan disuguhkan dialog seperti Gambar 2.3. Tunggu sampai dialog tanda selesai dikeluarkan seperti pada Gambar 2.4.



Gambar 2.2: Pilihan paket pendukung sebelum instalasi dilakukan



Gambar 2.3: Dialos selama proses instalasi berlangsung



Gambar 2.4: Dialog tanda selesai instalasi

Seperti telah ditunjukkan pada Gambar 2.1 tentang informasi lokasi *interpreter* Python diletakkan, dapat juga dibuktikan melalui aplikasi CMD seperti Gambar 2.5. Sedangkan *interpreter* Python dapat diujicobakan dengan menuliskan perintah `python` di aplikasi CMD. Akan muncul dialog seperti Gambar 2.6. *Interpreter* Python siap digunakan, ditandai dengan munculnya karakter `>>>`.

```

C:\Windows\system32\cmd.exe
C:\Users\arya-win7>cd AppData\Local\Programs\Python\Python38
C:\Users\arya-win7\AppData\Local\Programs\Python\Python38>dir
Volume in drive C has no label.
Volume Serial Number is 1C95-EC23

Directory of C:\Users\arya-win7\AppData\Local\Programs\Python\Python38

03/19/2020  05:57 PM    <DIR>          .
02/25/2020  11:31 PM    <DIR>          ..
02/25/2020  11:31 PM                19,256  api-ms-win-core-console-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-datetime-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-debug-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-errorhandling-l1-1-0.dll
02/25/2020  11:31 PM                22,328  api-ms-win-core-file-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-file-l1-2-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-file-l2-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-handle-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-heap-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-interlocked-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-libraryloader-l1-1-0.dll
02/25/2020  11:31 PM                21,384  api-ms-win-core-localization-l1-2-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-memory-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-namedpipe-l1-1-0.dll
02/25/2020  11:31 PM                19,768  api-ms-win-core-processenvironment-l1-1-0.dll
02/25/2020  11:31 PM                20,792  api-ms-win-core-processthreads-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-processthreads-l1-1-1.dll
02/25/2020  11:31 PM                18,232  api-ms-win-core-profile-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-rtlsupport-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-string-l1-1-0.dll
02/25/2020  11:31 PM                20,792  api-ms-win-core-synch-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-core-synch-l1-2-0.dll
02/25/2020  11:31 PM                19,768  api-ms-win-core-sysinfo-l1-1-0.dll
02/25/2020  11:31 PM                18,744  api-ms-win-core-timezone-l1-1-0.dll
02/25/2020  11:31 PM                17,856  api-ms-win-core-unicode-l1-1-0.dll
02/25/2020  11:31 PM                19,768  api-ms-win-crt-conio-l1-1-0.dll
02/25/2020  11:31 PM                22,840  api-ms-win-crt-convert-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-crt-environment-l1-1-0.dll
02/25/2020  11:31 PM                20,792  api-ms-win-crt-filesystem-l1-1-0.dll
02/25/2020  11:31 PM                19,768  api-ms-win-crt-heap-l1-1-0.dll
02/25/2020  11:31 PM                19,256  api-ms-win-crt-locale-l1-1-0.dll
02/25/2020  11:31 PM                27,064  api-ms-win-crt-math-l1-1-0.dll
02/25/2020  11:31 PM                26,936  api-ms-win-crt-multibyte-l1-1-0.dll
02/25/2020  11:31 PM                71,480  api-ms-win-crt-private-l1-1-0.dll
02/25/2020  11:31 PM                19,768  api-ms-win-crt-process-l1-1-0.dll

```

Gambar 2.5: Lokasi instalasi *interpreter* Python

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\arya-win7>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>>

```

Gambar 2.6: *Interpreter* Python siap digunakan

Tahapan selanjutnya adalah instalasi pustaka `scikit-image`. Proses instalasinya dilakukan dengan aplikasi pengelola paket Python yang bernama `pip`. Silakan lihat Gambar 2.2. `pip` ada di urutan kedua dari fitur tambahan. `pip` dapat digunakan untuk melihat paket apa saja yang telah terpasang di sistem kita. Caranya dengan menjalankan perintah `python -m pip list` seperti ditunjukkan Gambar 2.7.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\arya-win7>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> exit()

C:\Users\arya-win7>python -m pip list
Package Version
-----
pip      19.2.3
setuptools 41.2.0
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\arya-win7>
```

Gambar 2.7: Daftar paket yang terpasang

pip dapat juga digunakan untuk meng-upgrade paket yang telah terpasang, bahkan dirinya sendiri. Untuk meng-upgrade paket pip itu sendiri, dapat dilakukan dengan menjalankan perintah `python -m pip install --upgrade pip` seperti Gambar 2.8. Perhatikan versi pip yang ada di Gambar 2.7 dan Gambar 2.8.

```
C:\Windows\system32\cmd.exe
pip      19.2.3
setuptools 41.2.0
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\arya-win7>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a58f431eb594a17495f15b88da142ce14b5845662c13f3/pip-20.0.2-py2.py3-none-any.whl (1.4MB)
    ! 1.4MB 544kB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
  Successfully installed pip-20.0.2

C:\Users\arya-win7>python -m pip list
Package Version
-----
pip      20.0.2
setuptools 41.2.0

C:\Users\arya-win7>
```

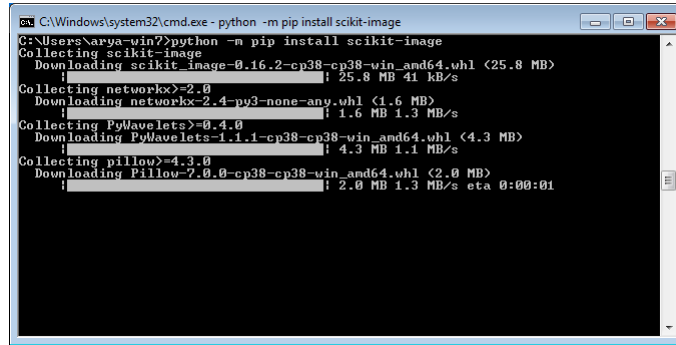
Gambar 2.8: Hasil upgrade pip

Sedangkan untuk memasang pustaka `scikit-image`, jalankan perintah `python -m pip install scikit-image` pada aplikasi CMD seperti Gambar 2.9.

```
C:\Windows\system32\cmd.exe - python -m pip install scikit-image
C:\Users\arya-win7>python -m pip install scikit-image
Collecting scikit-image
  Downloading scikit_image-0.16.2-cp38-cp38-win_amd64.whl (25.8 MB)
    ! 3.4 MB 1.3 MB/s eta 0:00:18
```

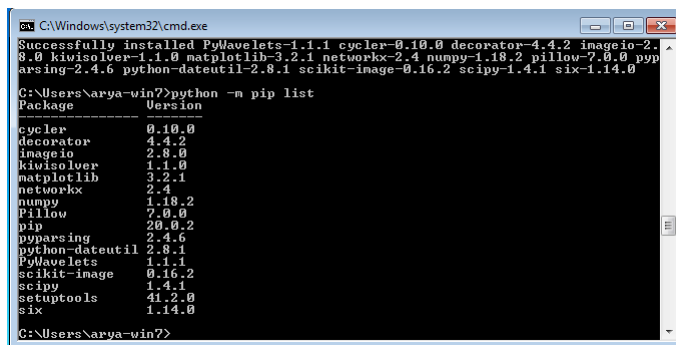
Gambar 2.9: Instalasi pustaka `scikit-image` menggunakan pip

Jika ada pustaka lain yang menjadi ketergantungan dari pustaka yang akan diinstal, pip akan melakukan instalasi secara otomatis. Gambar 2.10 menunjukkan proses tersebut. Hal ini akan sangat memudahkan pengguna mengelola pustaka Python yang digunakan.



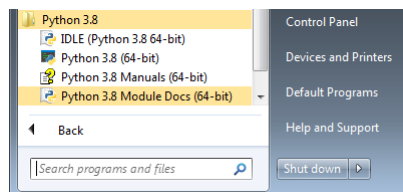
Gambar 2.10: Instalasi pustaka *dependent*

Setelah selesai, kita dapat kembali melihat daftar paket yang terpasang melalui pengelolaan pip yang ditunjukkan Gambar 2.11.



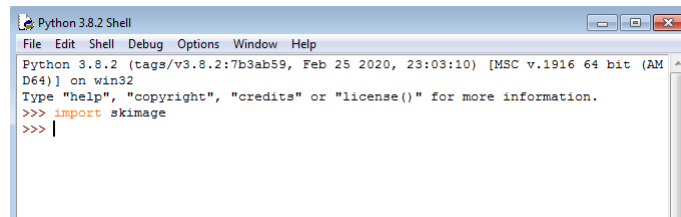
Gambar 2.11: Daftar terakhir paket terpasang

Menu aplikasi pendukung Python akan muncul seperti Gambar 2.12. Menu kedua pada Gambar 2.12 akan memunculkan aplikasi CMD yang sama dengan yang ditunjukkan Gambar 2.6, tetapi tanpa perlu memanggil perintah `python` terlebih dahulu. CMD secara otomatis akan memunculkan Python `shell` seperti Gambar 2.6.



Gambar 2.12: Daftar menu aplikasi pendukung Python

IDLE adalah antarmukan *interpreter* Python seperti ditunjukkan Gambar 2.13. Dalam Gambar 2.13 juga terlihat bahwa kita berhasil meng-*import* pustaka **scikit-image**, yang dalam IDLE di Windows 7 disebut sebagai **skimage**. Jika Anda sedang menggunakan Ubuntu, kemudian menggunakan pustaka **scikit-image** yang diperoleh dari *repository* Ubuntu (bukan dari **pip**), pustaka **scikit-image** juga di-*import* dengan nama **skimage**. Berhasilnya sebuah pustaka Python di-*import* adalah ketika tidak ada komentar yang muncul setelah perintah **import** tersebut.



Gambar 2.13: Aplikasi IDLE

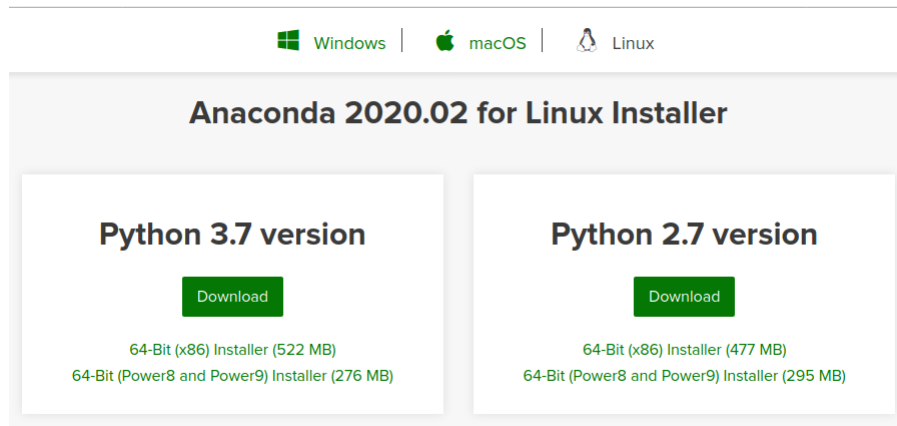
Selanjutnya, jika ditemukan petunjuk untuk masuk ke Python **Shell**, Anda dapat menggunakan aplikasi IDLE, atau menggunakan terminal (di Linux)/CMD (di Windows) dengan terlebih dahulu menjalankan perintah **python**.

2.2 Anaconda

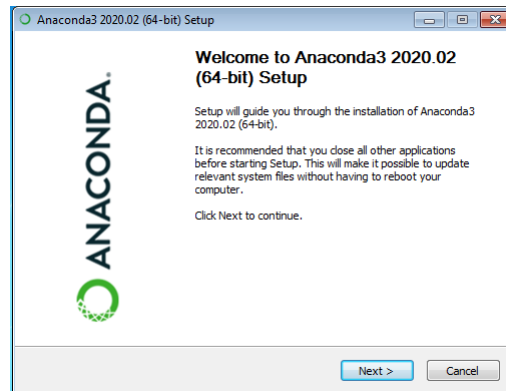
Selain pilihan manual seperti yang telah dijelaskan di Sub bab 2.1, Anaconda bisa menjadi opsi lain yang lebih bersifat otomatis. Saya menyebutnya otomatis karena Anaconda sejumlah pustaka Python, terutama yang banyak digunakan di *Data Mining*, *Machine Learning* atau *Data Science* telah dikemas di dalam Anaconda. Bahkan beberapa editor yang populer untuk Python juga dikemasnya. Anaconda bahkan mengemasnya khusus untuk *platform* yang berbeda. Anda dapat menghubungi alamat <https://www.anaconda.com/> untuk mengunduh aplikasinya. Sesuaikan kebutuhan Anda dengan pilihan yang ada seperti ditunjukkan Gambar 2.14.

Instalasi Anaconda akan menghadirkan dialog seperti ditunjukkan Gambar 2.15 - Gambar 2.21. Anaconda akan meletakkan pustaka di lokasi `C:\\ProgramData\\Anaconda3` yang berbeda dengan **pip** seperti terlihat di Gambar 2.18. Sedangkan di Gambar 2.20 terlihat sejumlah pustaka penting seperti **scikit-image** dan **scikit-learn** tengah diinstal.

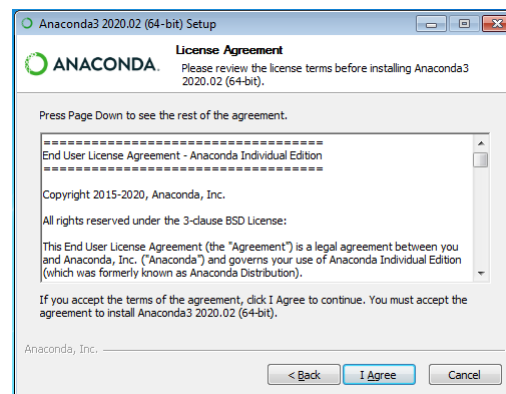
Instalasi Anaconda akan membuat menu seperti pada Gambar 2.22. Di situ terlihat sejumlah aplikasi yang dapat digunakan untuk mengembangkan kode komputer berbasis Python seperti Jupyter dan Spyder. Untuk Jupyter, aplikasi ini akan menghadirkan antarmuka seperti tampak pada Gambar 2.23. Di sisi kanan atas terlihat beberapa opsi antarmuka untuk mengelola proyek Python dengan Jupyter, seperti Terminal Gambar 2.24 atau Python **Shell** di bawah Jupyter seperti Gambar 2.25 yang perannya seperti IDLE di Gambar 2.13. Sedangkan untuk Spyder, akan tampak antarmuka seperti Gambar 2.26.



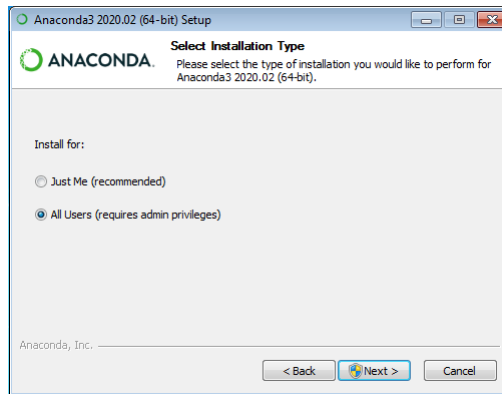
Gambar 2.14: Pilihan *platform* instalasi Anaconda



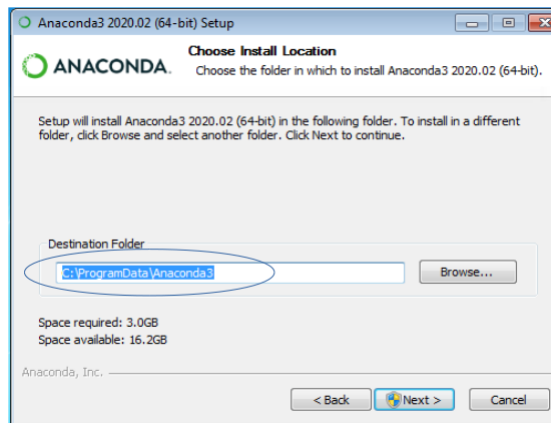
Gambar 2.15: Dialog pembuka instalasi



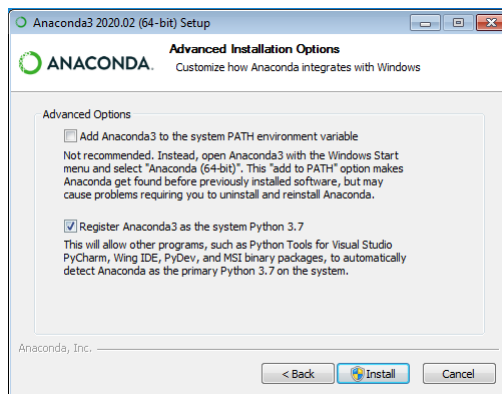
Gambar 2.16: Menyetujui kesepakatan



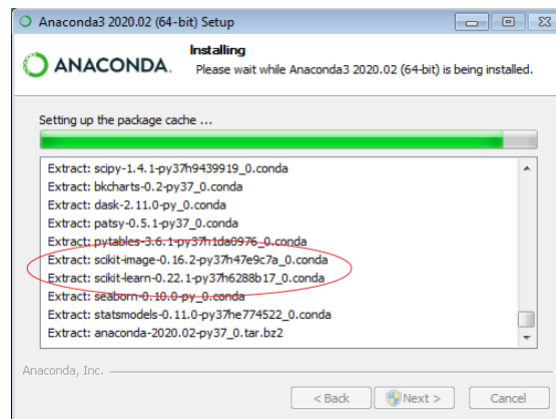
Gambar 2.17: Pilihan pengguna Anaconda



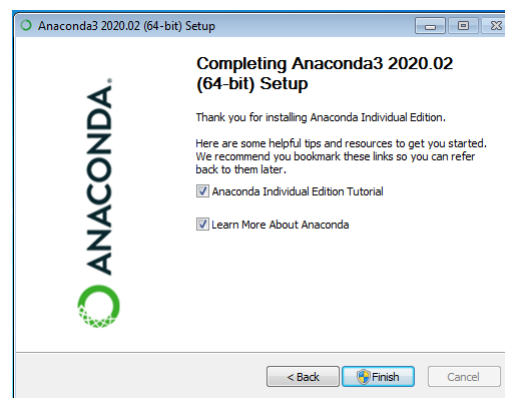
Gambar 2.18: Target instalasi



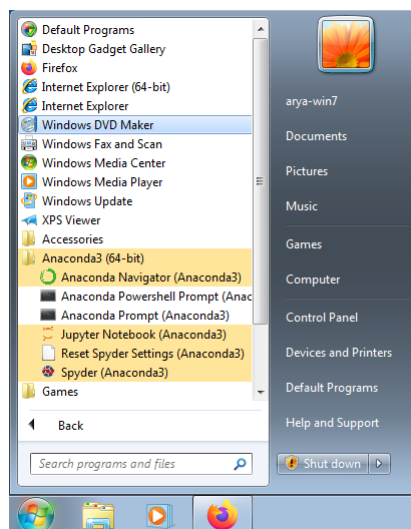
Gambar 2.19: Menjadikan Anaconda sebagai sistem utama Python



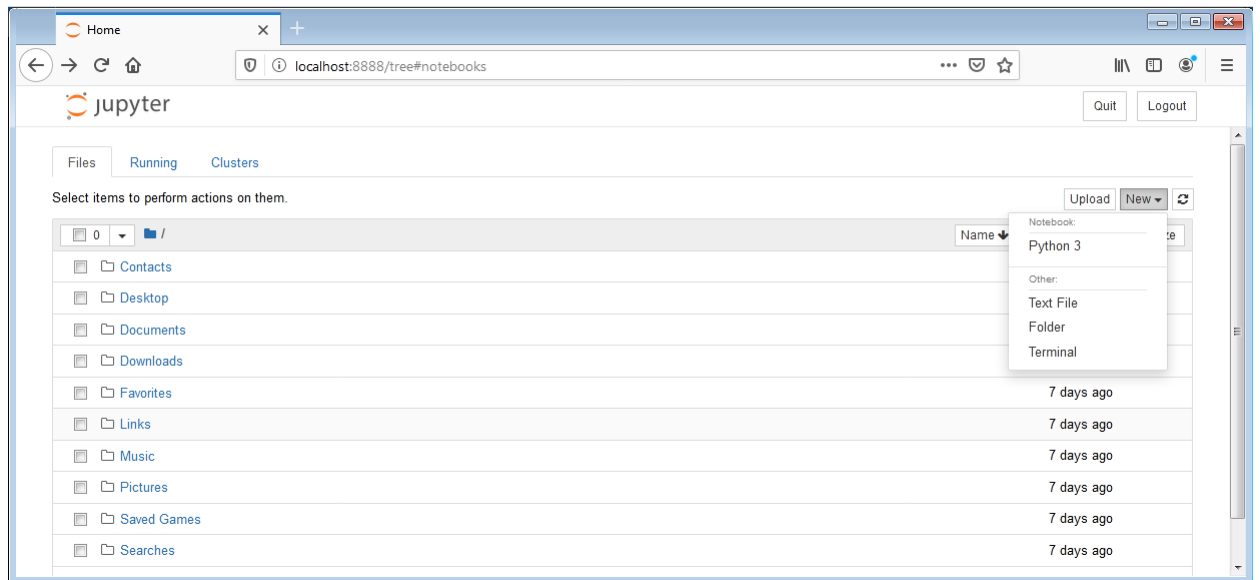
Gambar 2.20: Proses instalasi



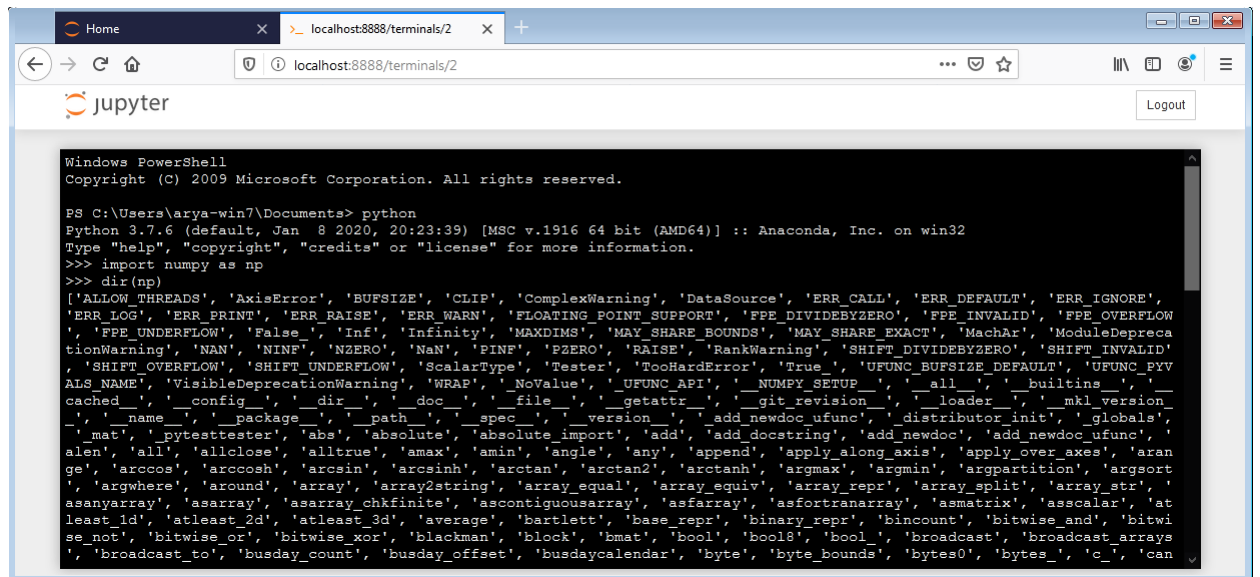
Gambar 2.21: Instalasi selesai



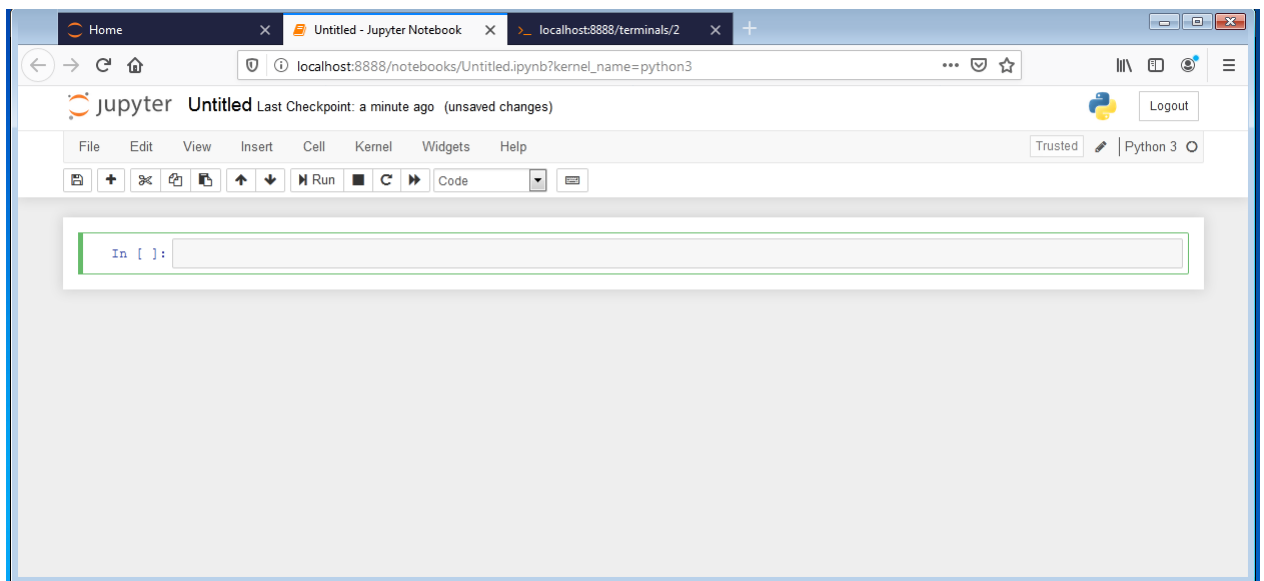
Gambar 2.22



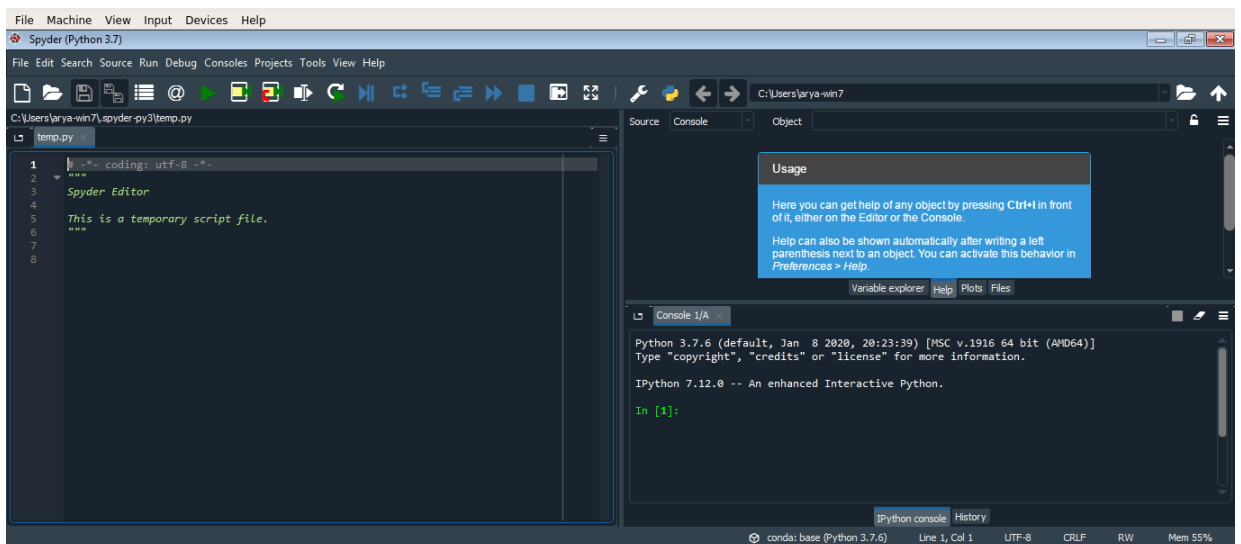
Gambar 2.23: Aplikasi Jupyter



Gambar 2.24: Terminal pada aplikasi Jupyter



Gambar 2.25: Python Shell pada aplikasi Jupyter



Gambar 2.26: Aplikasi Spyder

Bab 3

Dasar Pemrograman Python

3.1 Pendahuluan

Bahasa pemrograman Python memiliki 4 sifat dasar berikut¹.

1. *Interpreter*. Python diproses oleh *interpreter*, sehingga tidak perlu dikompilasi untuk menjalankannya. Hal ini seperti dijumpai pada bahasa pemrograman PHP yang sangat populer itu.
2. Interaktif. Anda dapat berinteraksi dengan Python dengan memberikannya perintah satu per satu melalui Python `shell`. Setiap perintah yang diberikan langsung akan direspon. Selain itu, Python bersifat *self explained*. Jika ada fungsi dari suatu obyek yang tidak kita ketahui, kita bisa mempelajarinya langsung dari dokumentasi di Python `shell`.
3. Berorientasi obyek. Ada semacam slogan bahwa "*Everything is object in Python*". Seperti telah dipahami melalui kuliah Rekayasa Perangkat Lunak, orientasi obyek menyebabkan variabel dan fungsi (sering disebut sebagai *state* dan *behavior*) terkemas dalam sebuah obyek, sehingga memudahkan pengelolaan variabel. Fungsi yang melekat pada sebuah obyek juga dapat diturunkan dari satu obyek ke obyek lain sehingga tidak perlu dideklarasikan ulang. Namun, fitur orientasi obyek ini pemberlakuannya bagi pemrogram tidak seketat seperti yang dilakukan di Java. Jika Java mengharuskan pemrogram mendeklarasikan kelas untuk membuat program yang bahkan sangat sederhana, maka Python tidak mengharuskannya.
4. Bahasa pemrograman untuk pemula. Hal ini disebabkan karena Python sangat sederhana, tidak memerlukan banyak deklarasi yang seringkali menyulitkan, bahkan menakutkan bagi pemula. Selain itu, Python juga mendukung pengembangan aplikasi untuk banyak *platform*, dari aplikasi *embedded* hingga *web* dan *mobile*.

Untuk sifat dasar pertama dan kedua, dapat dilihat ilustrasinya di Gambar 3.1. Dalam Gambar 3.1, Python `shell` dipanggil dengan perintah `python3`. Hal tersebut disebabkan karena

¹<https://www.tutorialspoint.com/python/index.htm>

Ubuntu (yang sedang digunakan adalah Ubuntu 18.04) secara *default* menyertakan Python versi 2.x. Sedangkan untuk Python versi 3.x harus dijalankan dengan perintah `python3`. Di Gambar 3.1 terlihat bahwa ada dua perintah yang diberikan secara berurutan. Tetapi, Python akan meresponnya satu per satu. Sedangkan untuk keluar dari Python `shell`, berikan perintah `exit()`.

```
arya@arya-pc:~$ python3
Python 3.6.9 (default, Nov 7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello world!')
Hello world!
>>> 3+7
10
>>> exit()
_
```

Gambar 3.1: Python `shell` sedang menerima perintah

Untuk sifat dasar ketiga dapat diilustrasikan melalui Gambar 3.2. Kita dapat mengetahui jenis obyek dari variabel `a` dengan fungsi `type(a)`. Sedangkan untuk melihat fungsi dan variabel apa saja yang terkandung pada variabel `a`, kita dapat menggunakan fungsi `dir(a)`. Tetapi, meskipun semuanya di dalam Python adalah obyek, penggunaan Python tidak mengharuskan kita mendeklarasi kelas secara eksplisit. Dengan menuliskan perintah `a=3`, Python tahu bahwa obyek `a` adalah obyek dari kelas `integer`. Bahkan, di Gambar 3.1, operasi aritmatika dapat dilakukan tanpa mendeklarasi variabel.

```
arya@arya-pc:~$ python3
Python 3.6.9 (default, Nov 7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a=3
>>> dir(a)
['_abs_', '_add_', '_and_', '_bool_', '_ceil_', '_class_', '_delatt',
'_dir_', '_divmod_', '_doc_', '_eq_', '_float_', '_floor_', '_f',
'_floordiv_', '_format_', '_ge_', '_getattr_', '_getnewargs_', '_g',
'_hash_', '_index_', '_init_', '_init_subclass_', '_int_', '_in',
'_invert_', '_le_', '_lshift_', '_lt_', '_mod_', '_mul_', '_ne_', '_ne',
'_new_', '_or_', '_pos_', '_pow_', '_radd_', '_rand_', '_rdi',
'_mod_', '_reduce_', '_reduce_ex_', '_repr_', '_rfloordiv_', '_rshift',
'_rmod_', '_rmul_', '_ror_', '_round_', '_rpow_', '_rrshift_', '_r',
'_rshift_', '_rsub_', '_rtruediv_', '_rxor_', '_setattr_', '_sizeof_',
'_str_', '_sub_', '_subclasshook_', '_truediv_', '_trunc_', '_xor_',
'_bit_length_', '_conjugate_', '_denominator_', '_from_bytes_', '_imag_', '_numerator_',
'_real_', '_to_bytes_']
>>> type(a)
<class 'int'>
>>>
```

Gambar 3.2: Variabel `a` sebagai obyek

Di Gambar 3.2 terlihat ada entitas yang diawali dan/atau diakhiri dengan karakter dua *underscore* (`'_'`) atau sering disebut sebagai *dunder*² (*double underscore*) oleh komunitas pemrogram Python. Hal tersebut merupakan bagian dari PEP (*Python Enhancement Proposals*) ke-8 tentang *Style Guide for Python Code*³.

Di Gambar 3.2 juga terlihat bahwa obyek `a` memiliki fungsi `__doc__`. Fungsi inilah yang akan memberikan penjelasan singkat kepada kita tentang obyek yang sedang menjadi perhatian.

²<https://dbader.org/blog/meaning-of-underscores-in-python>

³<https://www.python.org/dev/peps/pep-0008/>

Untuk menggunakannya, jalankan perintah `a.__doc__` seperti ditunjukkan Gambar 3.3. Dengan `a` adalah nama variabel untuk obyek yang sedang menjadi perhatian.

```
arya@arya-pc:~$ python3
Python 3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a=3
>>> a.__doc__
'int(x=0) -> integer\nint(x, base=10) -> integer\n\nConvert a number or string to an integer, or return 0 if no arguments\nare given. If x is a number, return x.__int__(). For floating point\nnumbers, this truncates towards zero.\n\nIf x is not a number or if base is given, then x must be a string,\nbytes, or bytearray instance representing an integer literal in the\ngiven base. The literal can be preceded by '+' or '-' and be surrounded\nby whitespace. The base defaults to 10. Valid bases are 0 and 2-36.\nBase 0 means to interpret the base from the string as an integer literal.\n>>> int('0b100', base=0)\n4"
>>>
```

Gambar 3.3: Menampilkan dokumentasi obyek `integer a`

Bab 4

Pendahuluan Pustaka Scikit-Image

Bab 5

Sub Modul Pustaka Scikit-Image

Bibliografi