

# **LAPORAN PRAKTIKUM**

## **“PRAKTIKUM 10: IMPLEMENTASI CUSTOM PACKAGE PYTHON”**

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Praktikum Kecerdasan Buatan yang di ampu oleh:

Ir. Sri Winiarti, S.T., M.Cs.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A / Senin 16.45 – 18.15 Lab. Basis Data

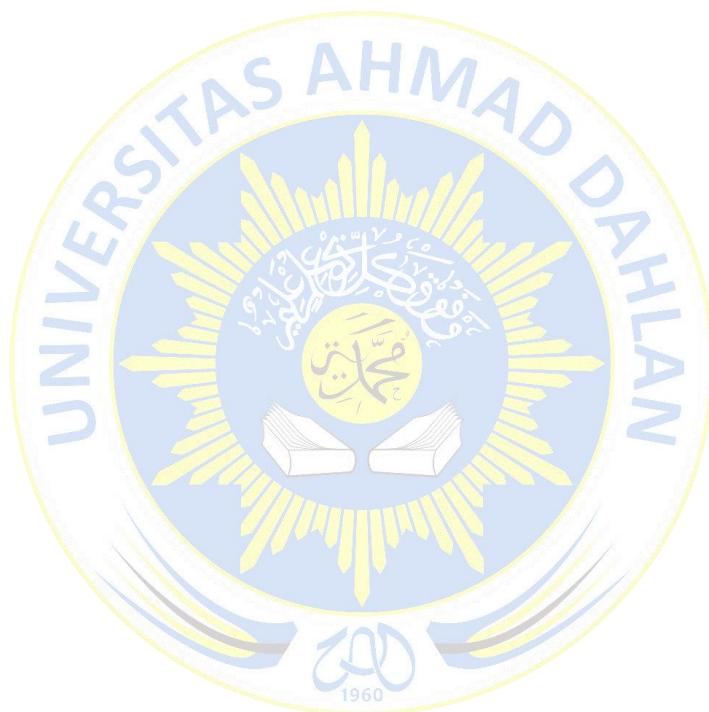
PJ Penilai:

Dwi Priambodo Prayoga (ke Mas Reza aja deh :3 karena gak ada pilihan dia T\_T)

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS AHMAD DAHLAN  
FAKULTAS TEKNOLOGI INDUSTRI  
TAHUN 2024**

## DAFTAR ISI

<b>PRETEST .....</b>	3
<b>LANGKAH PRAKTIKUM.....</b>	6
<b>POST TEST .....</b>	15



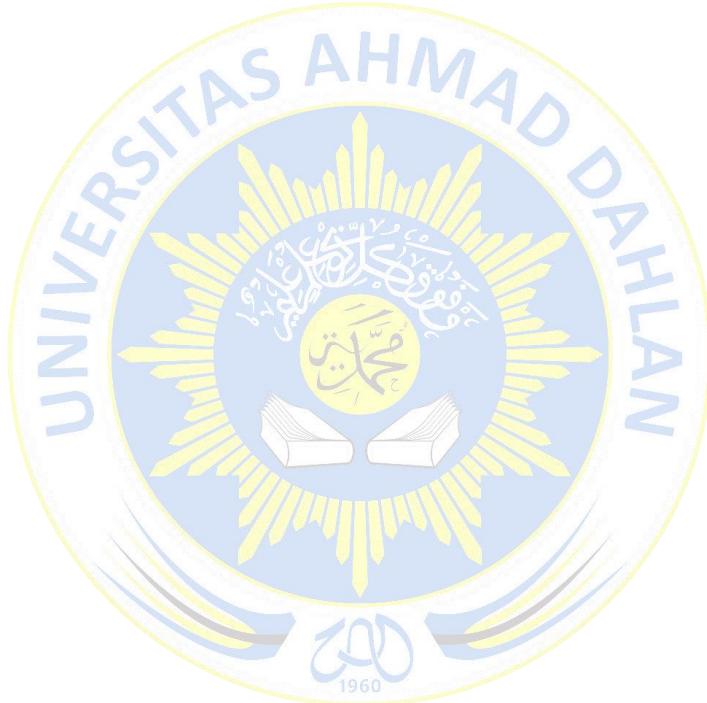
## PRETEST

Kerjakan soal pretest wajib menggunakan lembar kerja praktikum dan ditulis tangan.

Soal:

1. Jelaskan apa yang dimaksud dengan package-package dalam pemrograman Phyton.
2. Jelaskan tujuan adanya package dalam pemrograman Phyton bagi pengguna.
3. Jelaskan saat pembuatan package dalam Phyton hal-hal apa saja yang harus dipersiapkan!

Jawab:



## LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama: Mohammad Farid Hendianto NIM: 2200018401	Asisten: Paraf Asisten:	Tanggal: 19/06/2024 Nilai:
---	----------------------------	-------------------------------

1. Dalam pemrograman Python, sebuah package adalah kumpulan modul yang disimpan dalam sebuah direktori dengan struktur tertentu untuk mengorganisir modul-modul tersebut diatur dan diakses secara hierarkis. Package digunakan untuk mereorganisirkan kode program menjadi bagian-bagian yang lebih terstruktur dan mudah. Pada dasarnya, sebuah package adalah direktori yang berisi file modul Python serta sebuah file khusus '`__init__.py`'. File '`__init__.py`' bisa saja kosong, namun kehadirannya memungkinkan direktori tersebut diakui sebagai package oleh Python. Ini membantu menghindari konflik nama antara modul yang berbeda dan mempermudah penyelesaian kode dalam projek yang lebih besar.
2. Tujuan utama adanya package dalam pemrograman Python adalah untuk reseptivitas modulitas, keteraturan, dan keterbacaan program. Benar-benar adalah beberapa poin penting dari tujuan penggunaan package.
- Dengan menggunakan package, kode dapat diatur dalam struktur yang logis dan terorganisir, menjadikan penyelesaian projek besar dengan banyak modul.
  - Mendukung penulisan kode karena fungsi-fungsi yang serupa dikelompokkan bersama sehingga lebih mudah untuk dipahami dan memperbaiki kesalahan atau mengupdate fitur tertentu.
  - Mengintegrasikan penggunaan bantuan kode ketika package dapat didistribusikan dan diimpor ke dalam projek lain tanpa perlu menulis kode yang sama kembali.
  - Memudahkan menghindari konflik namabel dengan mengorganisir kodé ke dalam ruang nama yang berbeda, sehingga fungsi atau class dengan nama yang sama dapat ada di package yang berbeda tanpa konflik.
  - Memudahkan distribusi kodé kepada pengguna lain melalui repositori seperti PyPI (Python Package Index).
  - Memudahkan orang lain untuk mudah menginstall dan menggunakan package yang telah dibuat, memungkinkan orang lain untuk dengan mudah menginstall dan menggunakan package yang telah dibuat.
3. Saat membuat package dalam Python, beberapa hal penting yang harus diperhatikan dan dipertimbangkan:
- 1) Struktur direktori: Buat struktur direktori yang sebagus dan terorganisir untuk package. Struktur ini akan mendukung penulisan dan pengembangan package. Biasanya struktur direktori minimal akan mencakup direktori utama untuk package dan subdirektori untuk modul-modul yang ada di dalamnya.
  - 2) File '`__init__.py`': File ini harus ada di dalam direktori package dan digunakan untuk mensinialisasi package. Ini juga memungkinkan direktori untuk diakui sebagai package oleh Python. Didalamnya, biasanya dicantumkan variabel, fungsi, atau kelas yang ingin dibagikan oleh package, atau memimpin modul atau sub-package yang dibutuhkan.

3) File Setup ('setup.py'): File ini adalah script untuk membangun package menggunakan setuptools.  
Di dalam file ini di definisikan semua informasi tentang package, seperti nama, versi, penulis, deskripsi, URL, dan lisensi.

Contoh informasi yang dimanfaatkan

```
import setuptools
```

```
with open("README.md", "r") as fh:
```

```
    long_description = fh.read()
```

```
setuptools.setup(
```

```
    name = "my_package",
```

```
    version = "0.1.0",
```

```
    author = "Your Name",
```

```
    author_email = "your_email@example.com",
```

```
    description = "a small example package",
```

```
    long_description = long_description,
```

```
    long_description_content_type = "text/markdown",
```

```
    url = "https://github.com/yourusername/my_package",
```

```
    packages = setuptools.find_packages(),
```

```
    classifiers=[
```

```
        "Programming Language::Python :: 3",
```

```
        "License :: OSI Approved :: MIT License",
```

```
        "Operating System :: OS Independent",
```

```
    ],
```

```
)
```

4) File README.md : File ini berisi deskripsi package termasuk tujuan, cara instalasi, dan contoh penggunaan. Informasi ini sangat membantu pengguna lain yang ingin memahami dan menggunakan package Anda.

Contoh README.md

**My Package**

This is a simple example package for demonstration purposes. You can use this package to understand how to create and distribute Python packages.

5) File Lisensi ('LICENSE'): File ini berisi informasi lisensi yang membatasi hak-hak dan izin penggunaan package. Memilih lisensi yang tepat sangat penting untuk menentukan bagaimana orang lain bisa menggunakan dan mendistribusikan package Anda. Ada beberapa jenis lisensi seperti MIT, GPL, LGPL, Apache License 2.0, BSD License, MPL, dsb.

6) Distribusi Package: Setelah semua file siap, langkah selanjutnya adalah membuat distribusi package menggunakan tools seperti setuptools dan wheel.

Urutan untuk menginstall kedua package tersebut

```
pip install --user --upgrade setuptools wheel
```

Jalankan perintah berikut pada direktori yang memuat file setup.py untuk membuat distribusi package:

```
python setup.py sdist bdist_wheel
```

Perintah tersebut akan menghasilkan dua file di dalam folder dist, yaitu file .tar.gz (tar.gz file) dan file .whl (file disambut). File - File ini bisa diinstall untuk menginstall package menggunakan pip.

# LANGKAH PRAKTIKUM

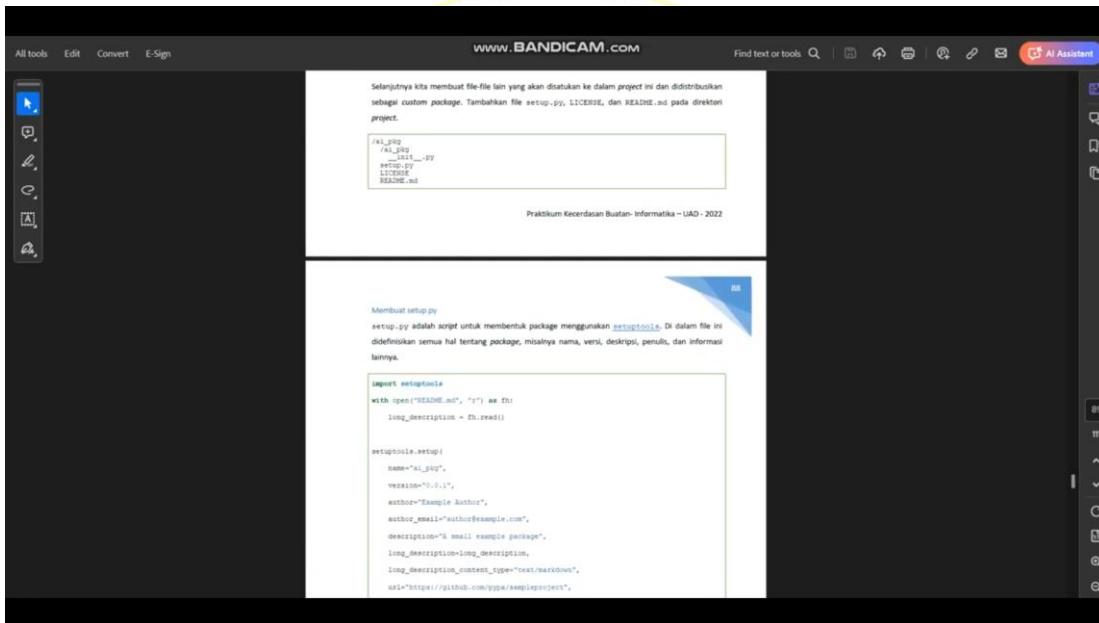
## LANGKAH PRAKTIKUM

Simak materi berikut :

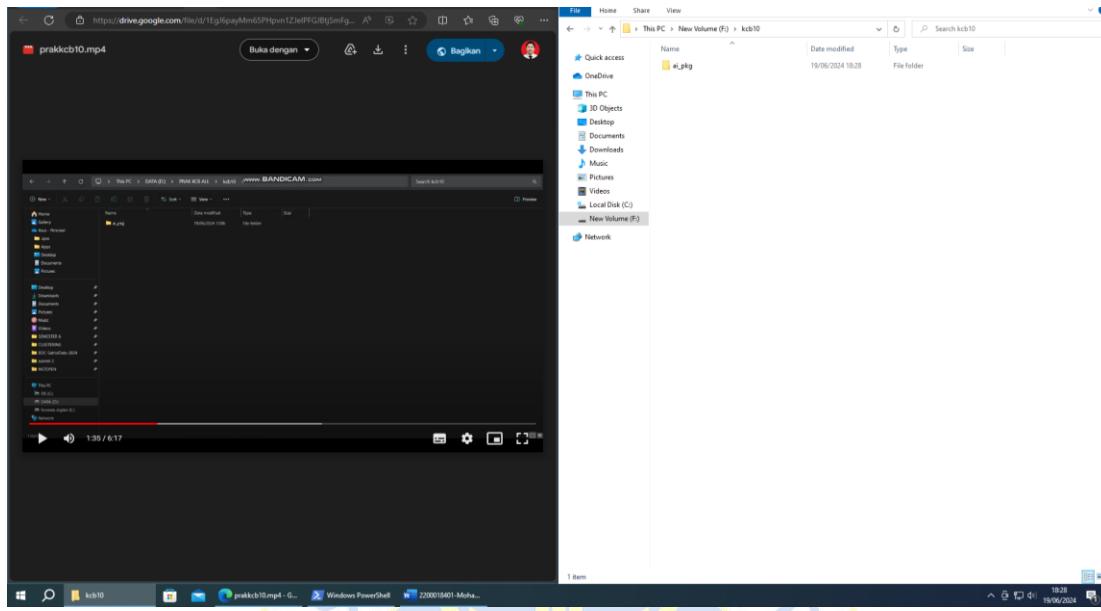
<https://drive.google.com/file/d/1EgJ6payMm65PHpvn1ZJelPFGJBtjSmFg/view?usp=sharing>

Silahkan kerjakan langkah praktikum sesuai video materi diatas, SS setiap langkahnya, dan Bagaimana hasilnya..? Tunjukkan hasil dari pembuatan package-package tersebut!

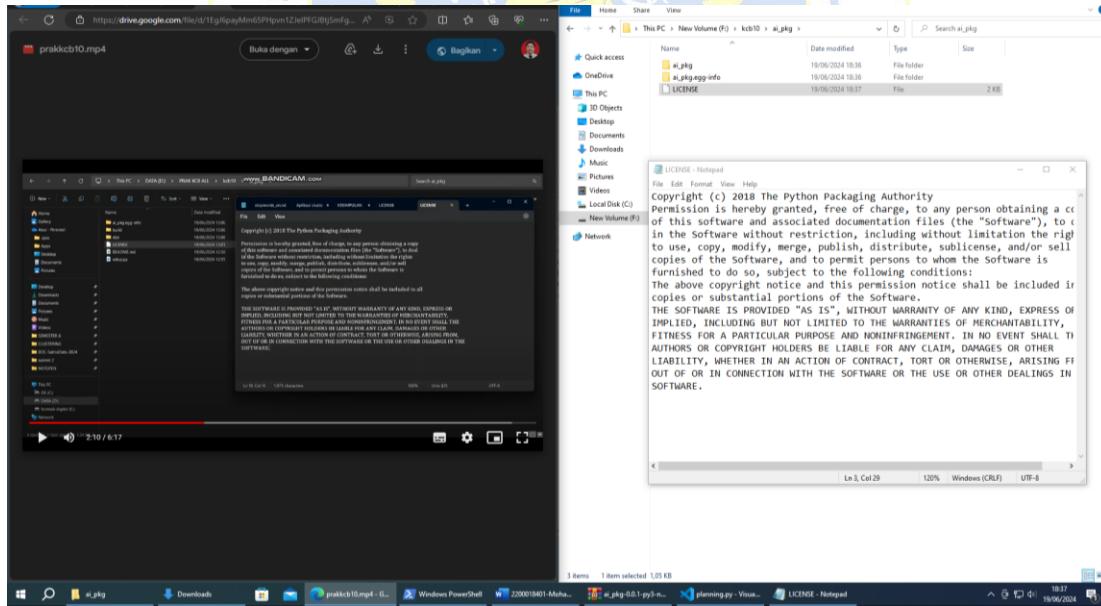
Menit 0:00-1:13



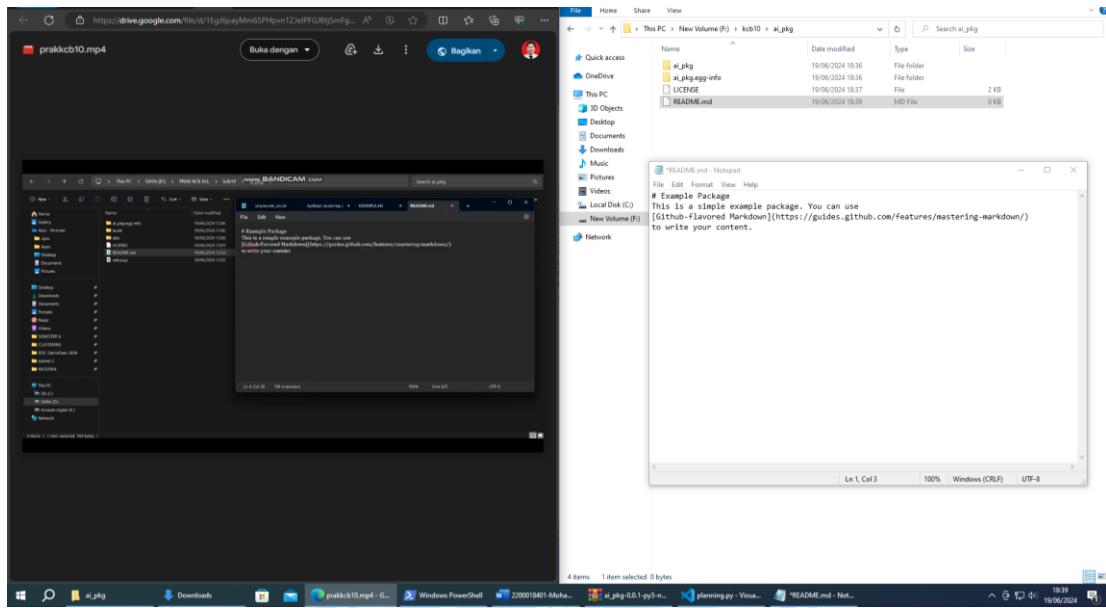
Disini asisten praktikum, menjelaskan materi pembuatan package, dari pembuatan folder “ai\_pkg”, kemudian di dalam folder tersebut ada setup.py, dimasukkan sesuai “modul praktikum”, lalu akan menambahkan README.md, hingga pembuatan pembuatan LICENSE



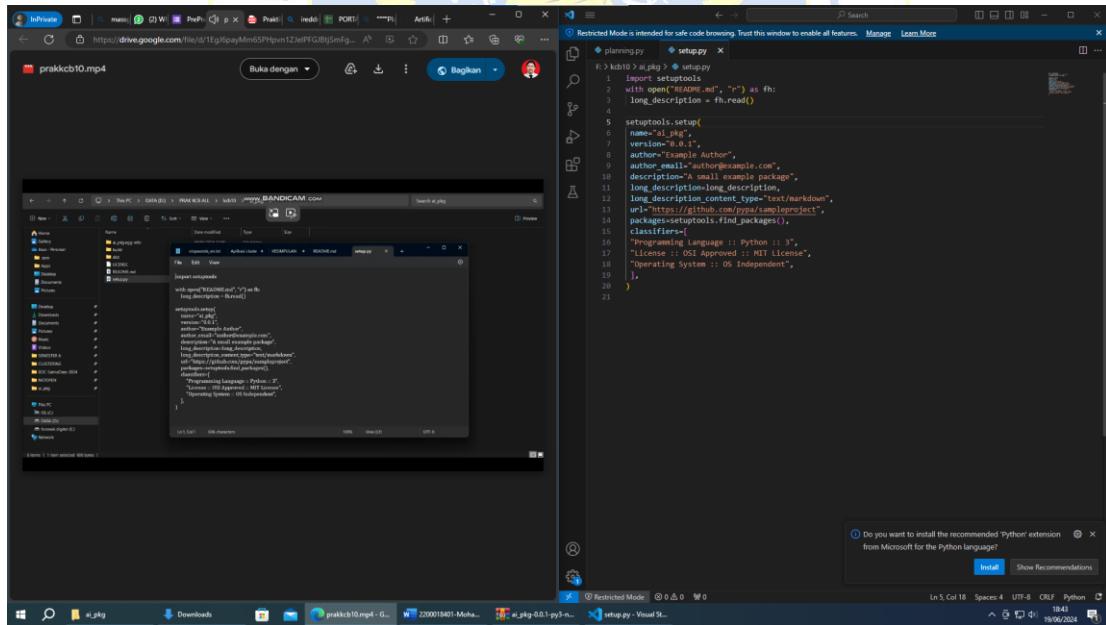
Membuat file directory D: yaitu folder kcb10/ai\_pkg. Dikarenakan saya menggunakan komputer laboratorium, sehingga saya akan menyesuaikan partisi selain D: yaitu F:



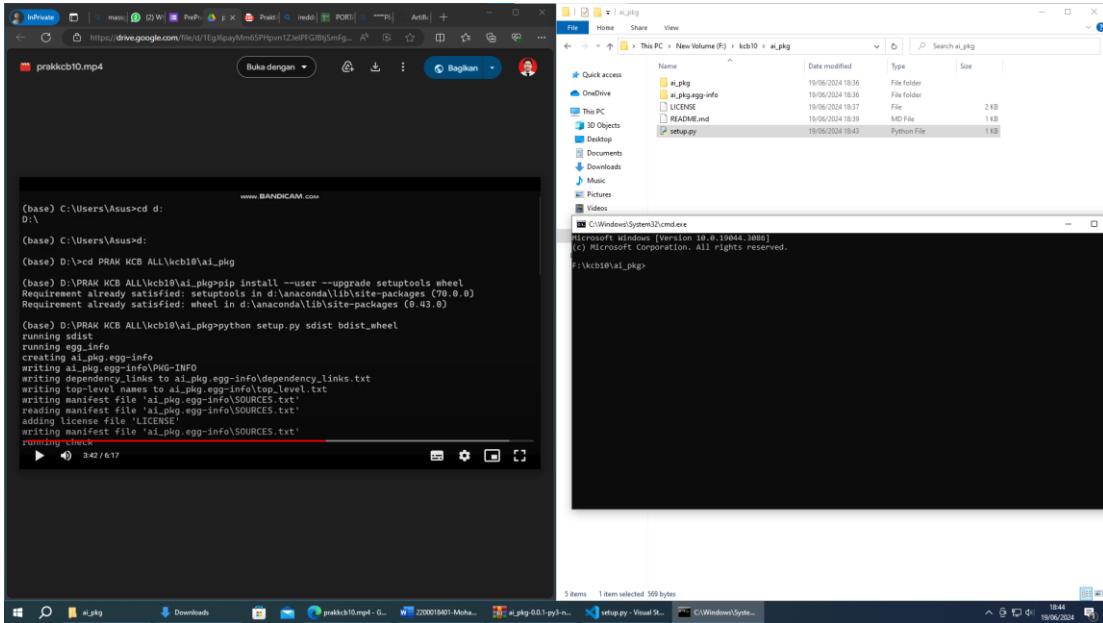
Di dalam ai\_pkg, kita mengisi LICENSE. Isi LICENSE akan disesuaikan dengan modul dan disimpan dengan nama LICENSE.



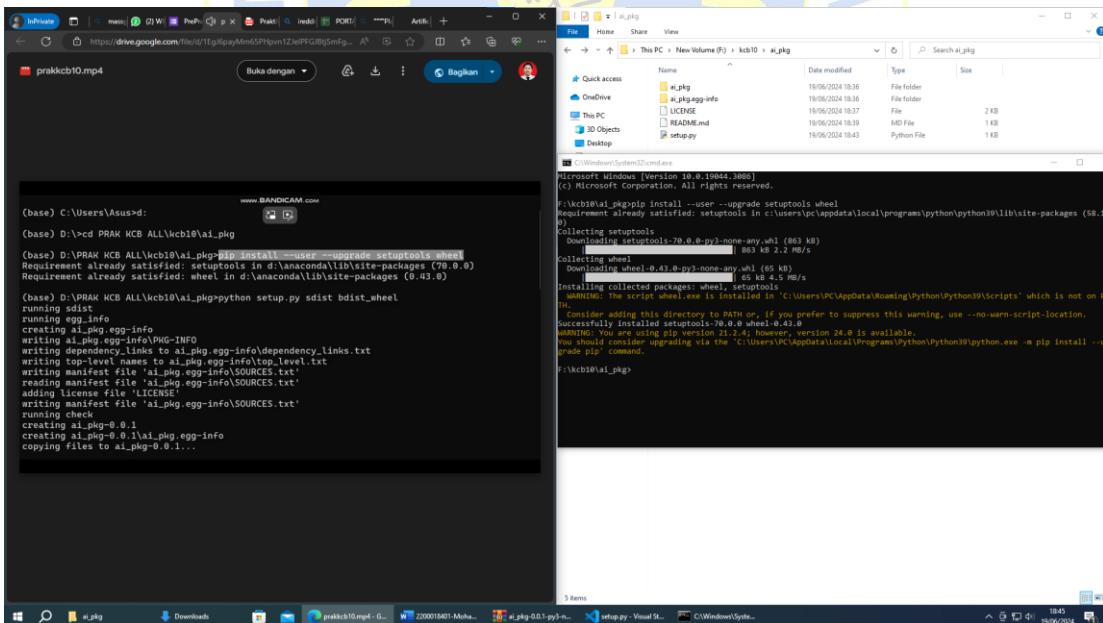
Kita akan membuat README.md, gunanya README.md sebagai penjelas project module Python kita untuk orang lain dapat memahami tata kegunaan module Python tersebut.



Membuat setup.py sesuai yang ada modul praktikum.



Tidak perlu menggunakan Anaconda, karena dari python sudah cukup bisa dijalankan. Masuk ssuai directory! Bedanya Anaconda dengan Python yaitu kalo di Anaconda bisa memiliki custom version Python sesuai env di Anaconda. Jadi menggunakan Python tanpa Anaconda **sudah cukup.**



Membuat distribusi package melakukan instalasi.

```

python setup.py sdist bdist_wheel
running sdist
running egg_info
writing ai_pkg.egg-info/PKG-INFO
writing dependency_links to ai_pkg.egg-info\dependency_links.txt
writing top-level names to ai_pkg.egg-info\top_level.txt
writing manifest file 'ai_pkg.egg-info\SOURCES.txt'
reading manifest file 'ai_pkg.egg-info\SOURCES.txt'
adding license file 'LICENSE'
writing manifest file 'ai_pkg.egg-info\SOURCES.txt'
running check
creating ai_pkg-0.0.1
creating ai_pkg-0.0.1\ai_pkg
creating ai_pkg-0.0.1\ai_pkg.egg-info
copying files to ai_pkg-0.0.1...
copying LICENSE -> ai_pkg-0.0.1
copying README.md -> ai_pkg-0.0.1
copying setup.py -> ai_pkg-0.0.1
copying ai_pkg\__init__.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\agents.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\logic.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\models.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\planning.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\search.py -> ai_pkg-0.0.1\ai_pkg

```

Melakukan running command “python setup.py sdist bdist\_wheel”

Output keseluruhan

```

F:\kcb10\ai_pkg>python setup.py sdist bdist_wheel
running sdist
running egg_info
writing ai_pkg.egg-info/PKG-INFO
writing dependency_links to ai_pkg.egg-info\dependency_links.txt
writing top-level names to ai_pkg.egg-info\top_level.txt
writing manifest file 'ai_pkg.egg-info\SOURCES.txt'
reading manifest file 'ai_pkg.egg-info\SOURCES.txt'
adding license file 'LICENSE'
writing manifest file 'ai_pkg.egg-info\SOURCES.txt'
running check
creating ai_pkg-0.0.1
creating ai_pkg-0.0.1\ai_pkg
creating ai_pkg-0.0.1\ai_pkg.egg-info
copying files to ai_pkg-0.0.1...
copying LICENSE -> ai_pkg-0.0.1
copying README.md -> ai_pkg-0.0.1
copying setup.py -> ai_pkg-0.0.1
copying ai_pkg\__init__.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\agents.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\logic.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\models.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\planning.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg\search.py -> ai_pkg-0.0.1\ai_pkg

```

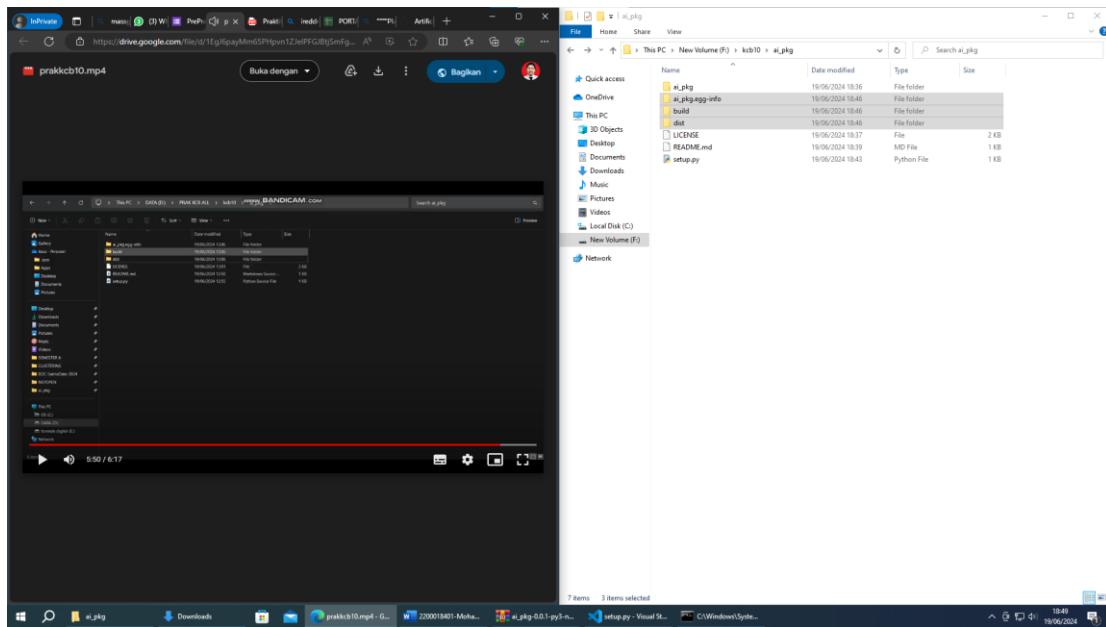
```
copying ai_pkg\utils.py -> ai_pkg-0.0.1\ai_pkg
copying ai_pkg.egg-info\LICENSE -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\METADATA -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\PKG-INFO -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\RECORD -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\SOURCES.txt -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\WHEEL -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\dependency_links.txt -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\desktop.ini -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\top_level.txt -> ai_pkg-0.0.1\ai_pkg.egg-info
copying ai_pkg.egg-info\SOURCES.txt -> ai_pkg-0.0.1\ai_pkg.egg-info
Writing ai_pkg-0.0.1\setup.cfg
creating dist
Creating tar archive
removing 'ai_pkg-0.0.1' (and everything under it)
running bdist_wheel
running build
running build_py
creating build
creating build\lib
creating build\lib\ai_pkg
copying ai_pkg\agents.py -> build\lib\ai_pkg
copying ai_pkg\logic.py -> build\lib\ai_pkg
copying ai_pkg\models.py -> build\lib\ai_pkg
copying ai_pkg\planning.py -> build\lib\ai_pkg
copying ai_pkg\search.py -> build\lib\ai_pkg
copying ai_pkg\utils.py -> build\lib\ai_pkg
copying ai_pkg\__init__.py -> build\lib\ai_pkg
C:\Users\PC\AppData\Roaming\Python\Python39\site-
packages\setuptools\_distutils\cmd.py:66: SetuptoolsDeprecationWarning: setup.py install
is deprecated.
!!
*****
*****
Please avoid running ``setup.py`` directly.
Instead, use pypa/build, pypa/installer or other
standards-based tools.

See https://blog.ganssle.io/articles/2021/10/setup-py-deprecated.html for details.
*****
*****
!!
```

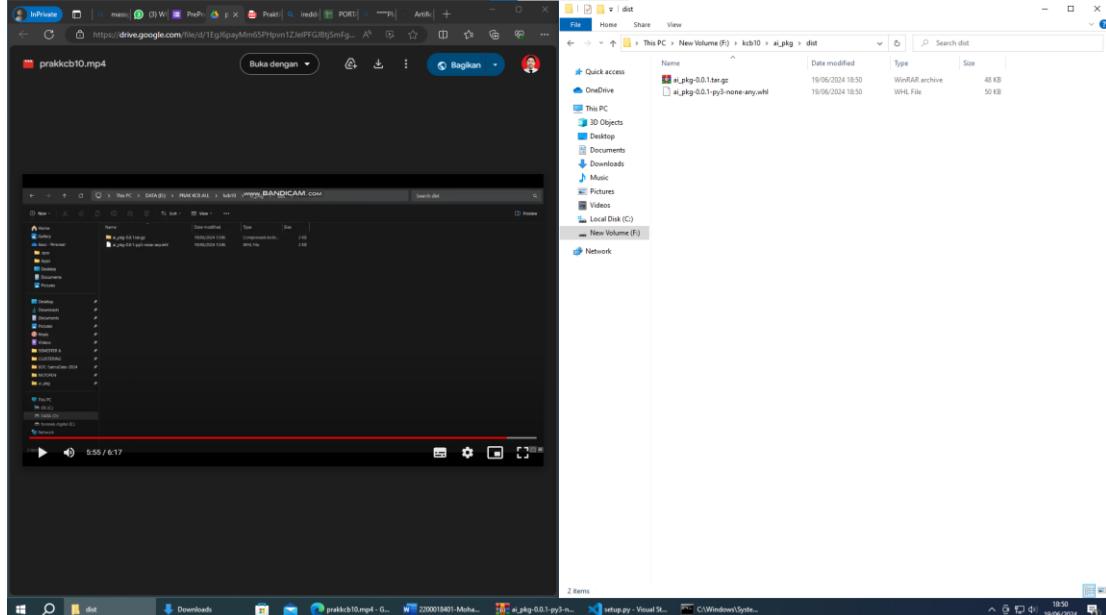
```
self.initialize_options()
installing to build\bdist.win-amd64\wheel
running install
running install_lib
creating build\bdist.win-amd64
creating build\bdist.win-amd64\wheel
creating build\bdist.win-amd64\wheel\ai_pkg
copying build\lib\ai_pkg\agents.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\logic.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\models.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\planning.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\search.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\utils.py -> build\bdist.win-amd64\wheel\.\ai_pkg
copying build\lib\ai_pkg\_\_init__.py -> build\bdist.win-amd64\wheel\.\ai_pkg
running install_egg_info
Copying ai_pkg.egg-info to build\bdist.win-amd64\wheel\.\ai_pkg-0.0.1-py3.9.egg-info
running install_scripts
creating build\bdist.win-amd64\wheel\ai_pkg-0.0.1.dist-info\WHEEL
creating 'dist\ai_pkg-0.0.1-py3-none-any.whl' and adding 'build\bdist.win-amd64\wheel' to it
adding 'ai_pkg/\_\_init__.py'
adding 'ai_pkg/agents.py'
adding 'ai_pkg/logic.py'
adding 'ai_pkg/models.py'
adding 'ai_pkg/planning.py'
adding 'ai_pkg/search.py'
adding 'ai_pkg/utils.py'
adding 'ai_pkg-0.0.1.dist-info/LICENSE'
adding 'ai_pkg-0.0.1.dist-info/METADATA'
adding 'ai_pkg-0.0.1.dist-info/WHEEL'
adding 'ai_pkg-0.0.1.dist-info/desktop.ini'
adding 'ai_pkg-0.0.1.dist-info/top_level.txt'
adding 'ai_pkg-0.0.1.dist-info/RECORD'
removing build\bdist.win-amd64\wheel

F:\kcb10\ai_pkg>
```

Apabila output seperti ini maka sudah berhasil install

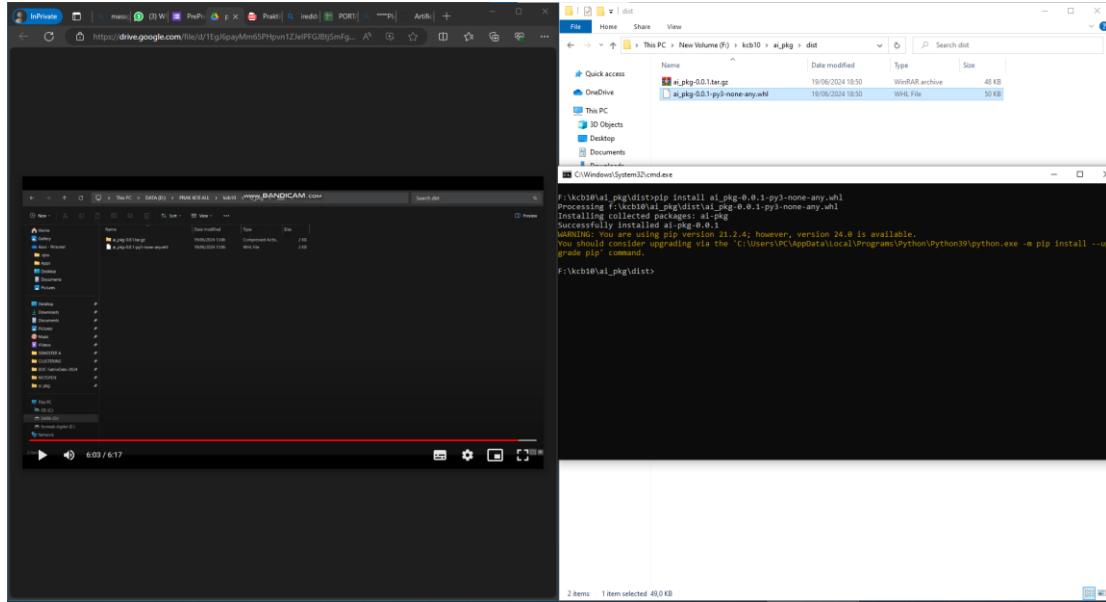


Ketiga folder tersebut DIBUAT OTOMATIS terbuat sendiri hasil dari output command. Yaitu ai\_pkg.egg.info, build, dan dist

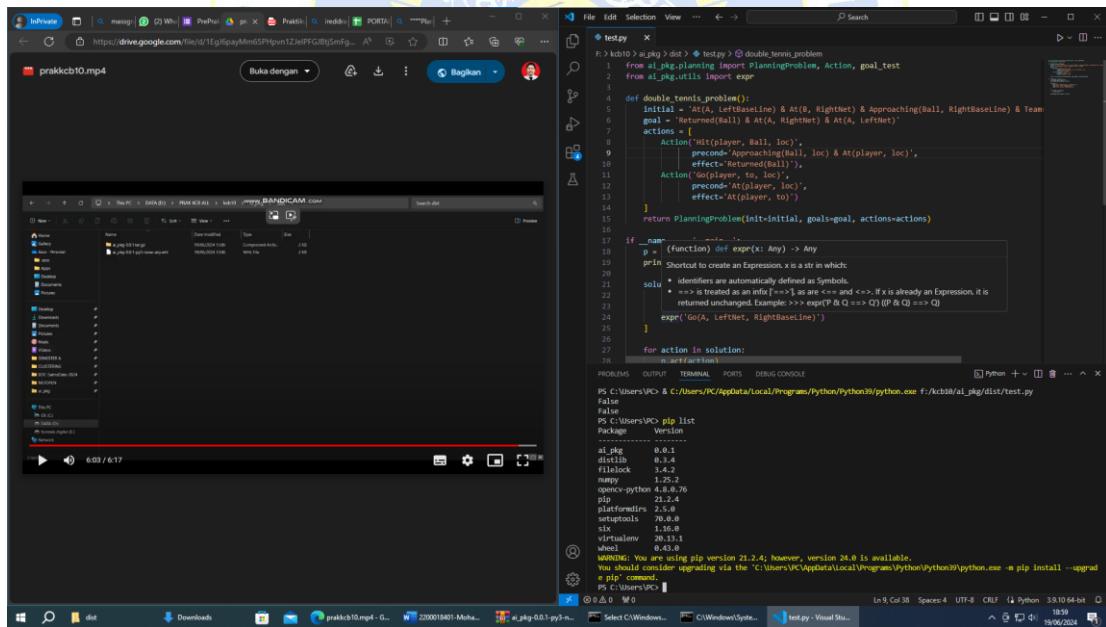


Mengecek dist, ada package yang sudah kita custom. Seperti itu hasilnya

Bonus tidak ada video, instalasi package hasil praktikum



Mencoba hasil installasi custom pkg yang sudah dibuat sebelumnya



Instalasi dari package custom sendiri berhasil <3

## POST TEST

Silahkan kerjakan postest sesuai soal dimodul dengan membuat package versi 0.0.2

Sesuai modul:

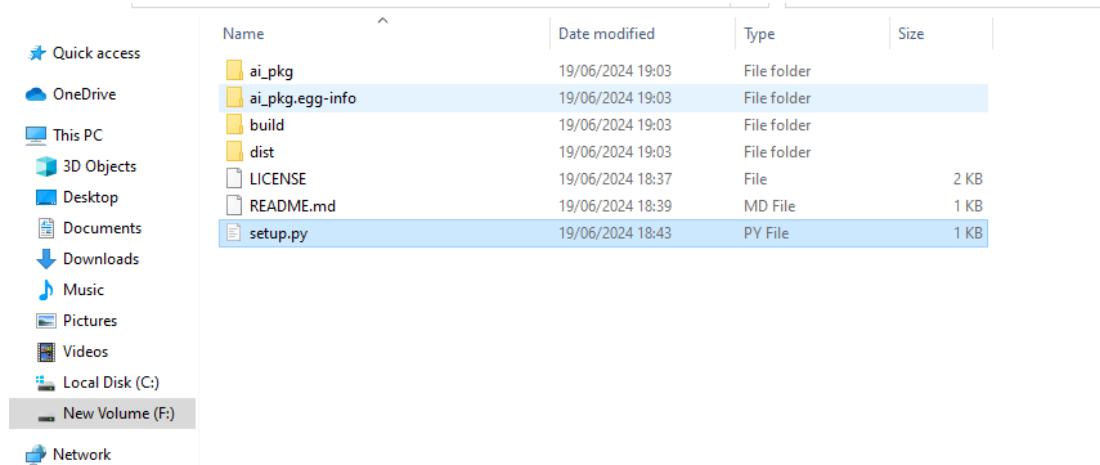
No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL 08- KK01	CPMK- 04	<p>Cobalah membuat file setup.py dengan menuliskan koding berikut ini:</p> <pre>import setuptools with open("README.md", "r") as fh:     long_description = fh.read()  setuptools.setup(     name="ai_pkg",     version="0.0.1",     author="Example Author",     author_email="author@example.com",     description="A small example package",     long_description=long_description,     long_description_content_type="text/markdown",     url="https://github.com/pypa/sampleproject",     packages=setuptools.find_packages(),     classifiers=[         "Programming Language :: Python :: 3",     ]</pre>		30

Kecerdasan Buatan - Informatika – UAD - 2024

			"License :: OSI Approved :: MIT License", "Operating System :: OS Independent", )  Eksekusilah program tersebut di atas. Tuliskan hasilnya			103
2	CPL 08- KK01	CPMK- 05	Buatlah package License dan red.Me seperti yang ada dalam materi tersebut di atas. Tuliskan hasilnya	Hasil Analisa	70	
<b>Total Nilai</b>						<b>100</b>

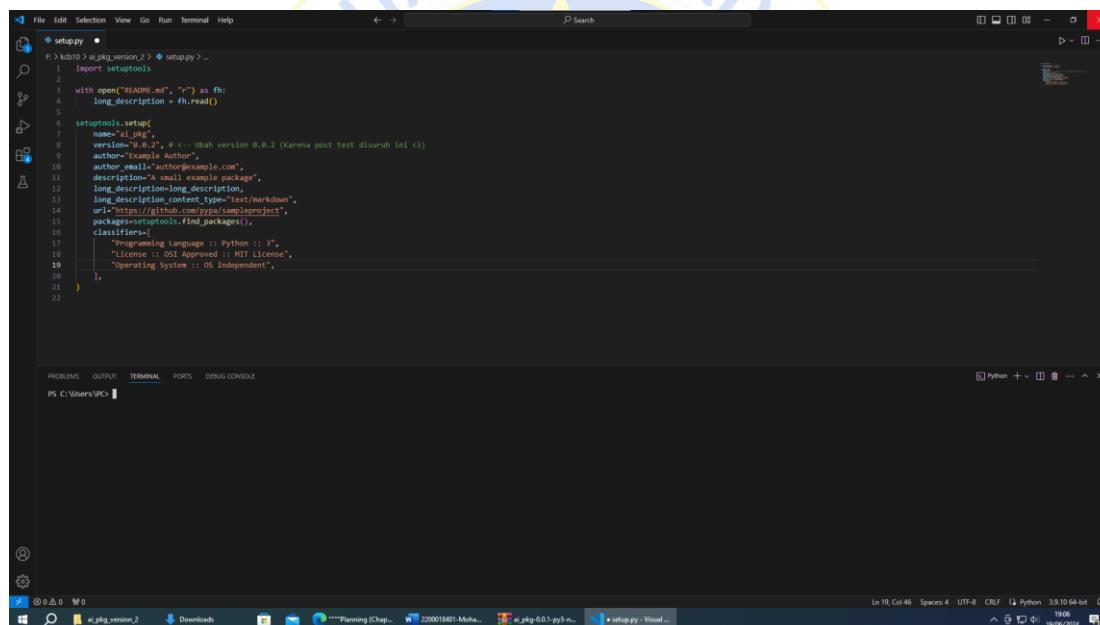
Name	Date modified	Type	Size
ai_pkg	19/06/2024 18:50	File folder	
ai_pkg_version_2	19/06/2024 19:03	File folder	

Kita copy aja terlebih dahulu dari folder langkah praktikum, misalkan tambahkan dibelakangnya \_version\_2



	Name	Date modified	Type	Size
	ai_pkg	19/06/2024 19:03	File folder	
	ai_pkg.egg-info	19/06/2024 19:03	File folder	
	build	19/06/2024 19:03	File folder	
	dist	19/06/2024 19:03	File folder	
	LICENSE	19/06/2024 18:37	File	2 KB
	README.md	19/06/2024 18:39	MD File	1 KB
	setup.py	19/06/2024 18:43	PY File	1 KB

Buka setup.py, bisa menggunakan vscode biar enak dipandang

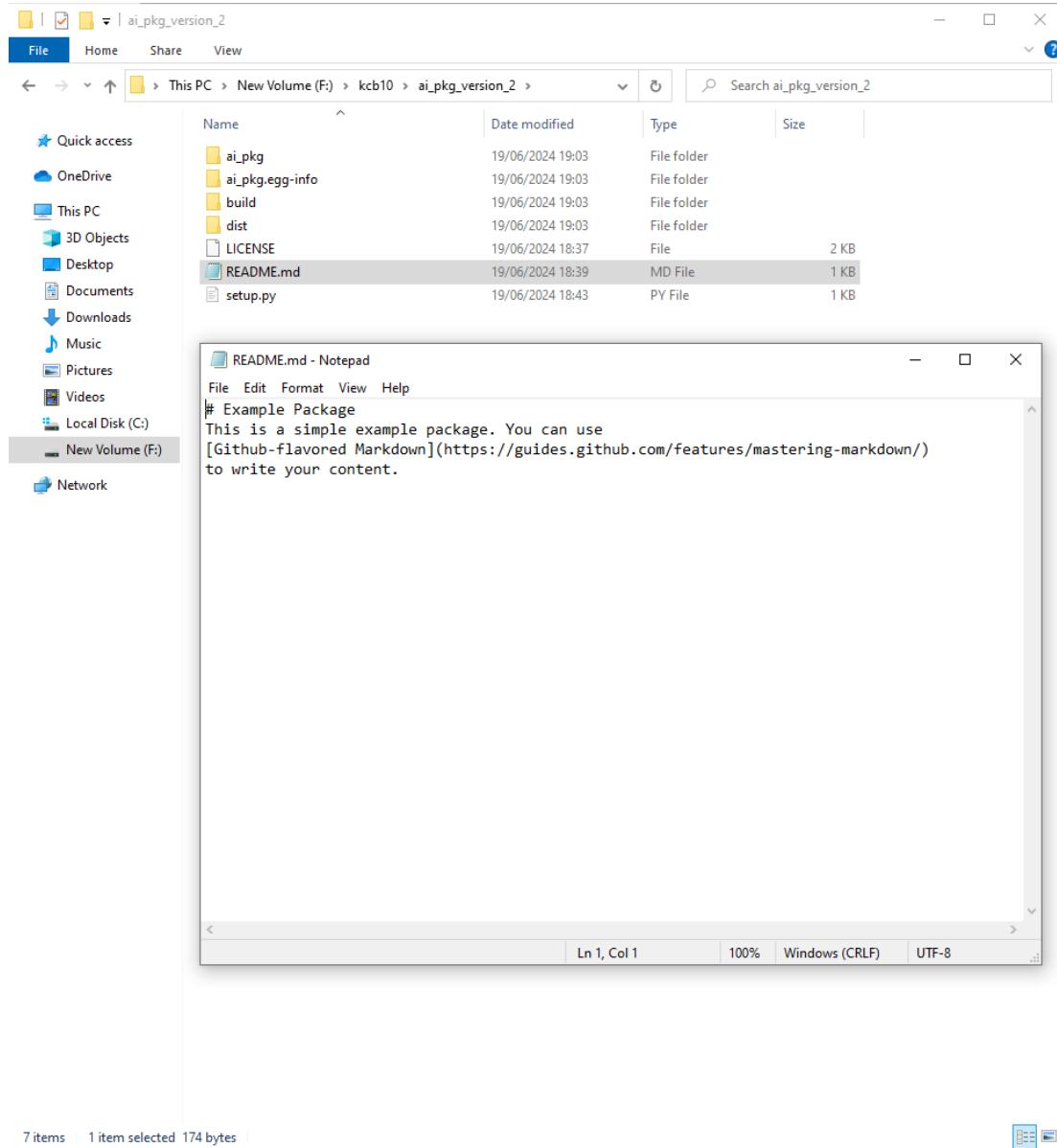


```

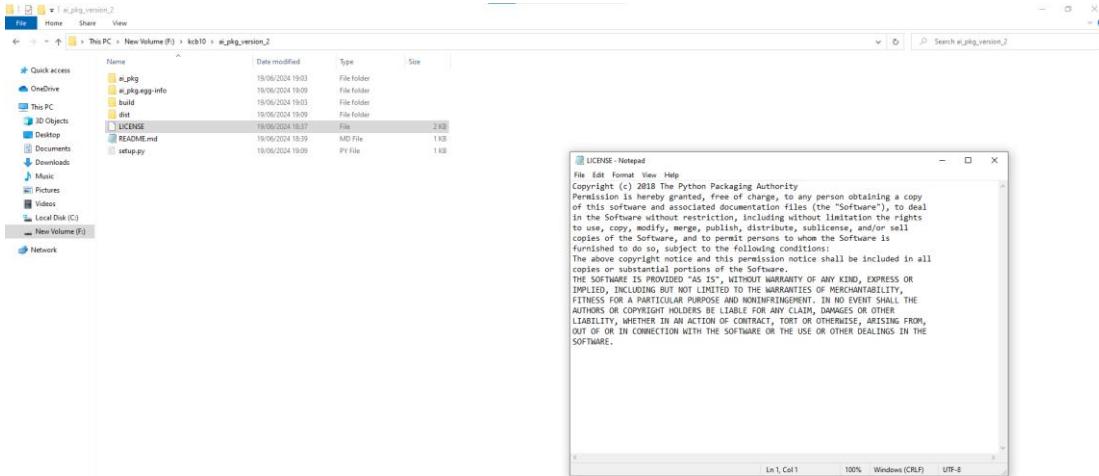
F:\kobido\ai_pkg>cd .. & python setup.py ...
1  import setuptools
2
3  with open("README.md", "r") as fh:
4      long_description = fh.read()
5
6  setuptools.setup(
7      name="ai_pkg",
8      version="0.0.2", # c : Ubah versi 0.0.2 (Karena post test disuruh lini <>)
9      author="authorexample.com",
10     author_email="authorexample.com",
11     description="A small example package",
12     long_description=long_description,
13     long_description_content_type="text/markdown",
14     url="https://github.com/authorexample/ai_pkg",
15     packages=setuptools.find_packages(),
16     classifiers=[
17         "Programming Language :: Python :: 3",
18         "License :: OSI Approved :: MIT License",
19         "Operating System :: OS Independent",
20     ],
21 )
22

```

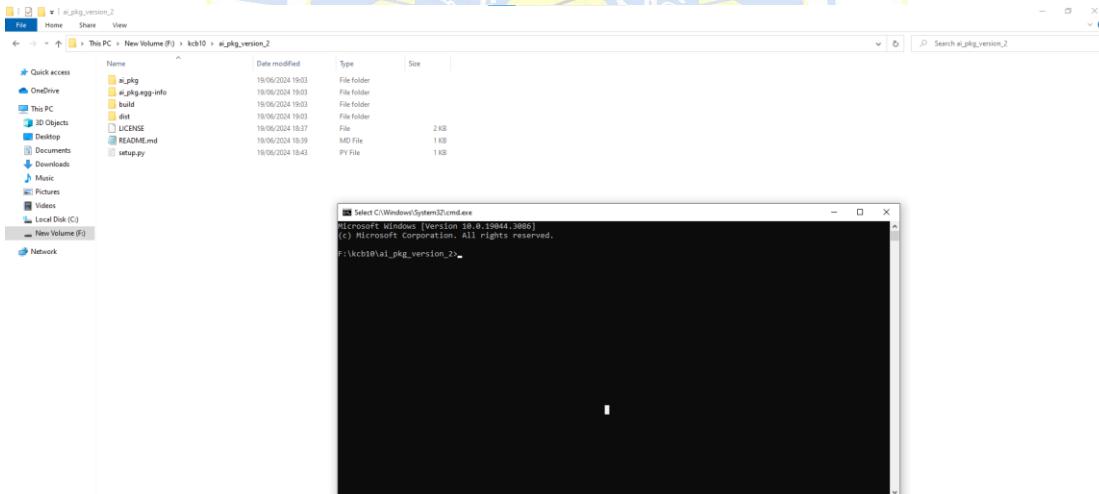
Tuliskan ulang sesuai permintaan Post Test pertemuan 10 setup.py Lalu jangan lupa ganti version dari 0.0.1 menjadi 0.0.2



Untuk red.Me harusnya README.md!!!! karena langkah praktikum sudah buat, kita buat samakan saja.



LICENSE pun sama aja. Karena gak diminta LICENSE berjenis apa, mungkin MIT yang sesuai dengan modul langkah praktikum



Buka cmd dan masuk ke foldernya  
biar tidak usah cd ada cara cepatnya yaitu

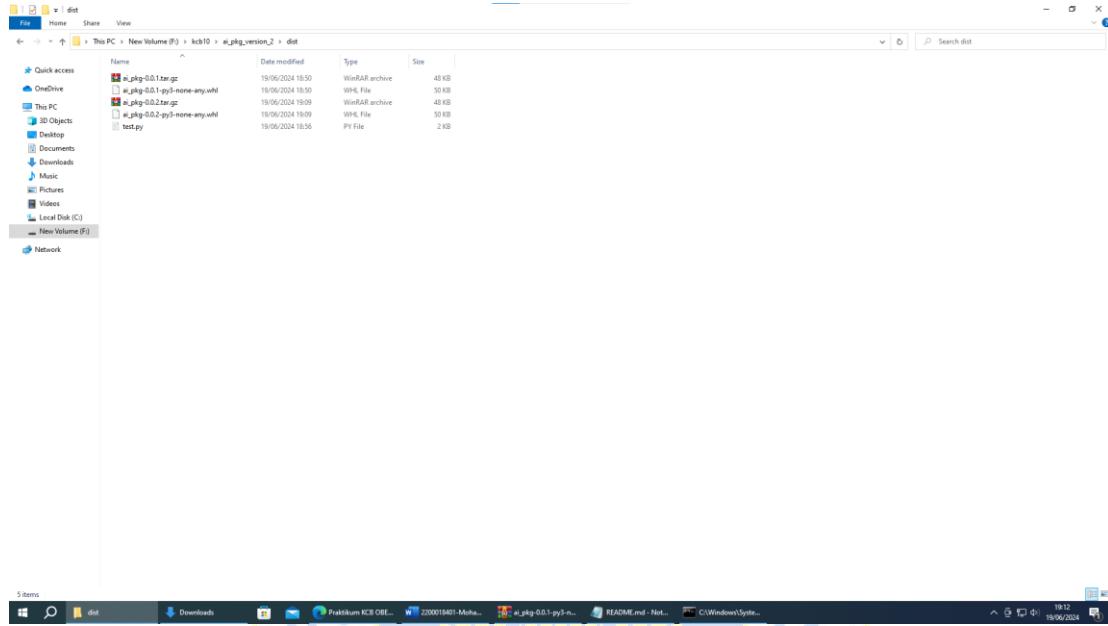


Ketik aja cmd di folder pathnya

Langsung jalankan command

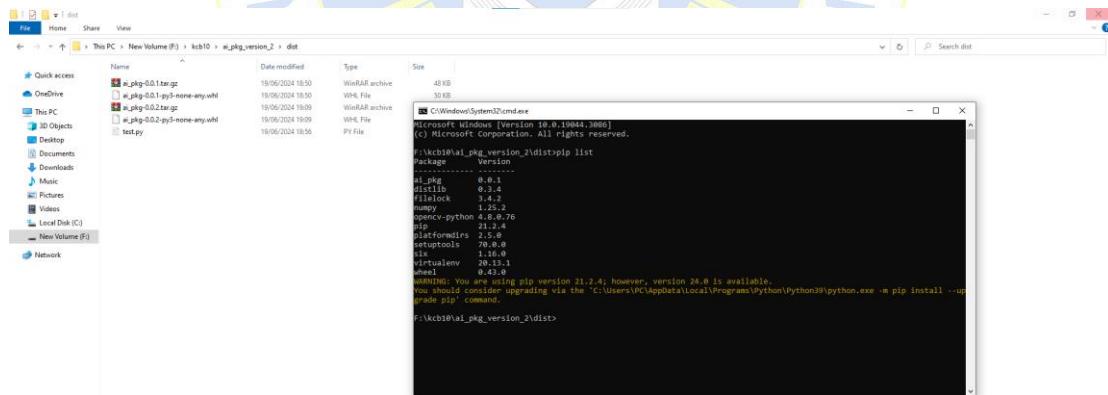
```
python setup.py sdist bdist_wheel
```

Tidak perlu **pip install --user --upgrade setuptools wheel** karena sebelumnya di langkah praktikum sudah

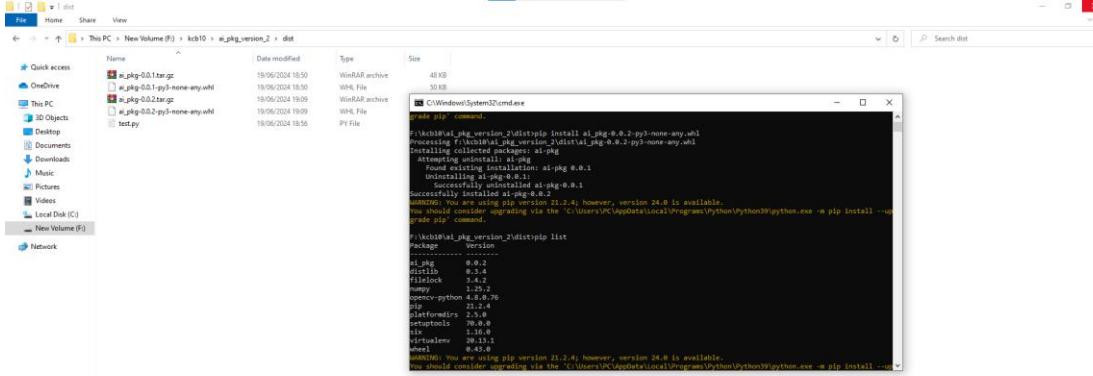


Terlihat ada 2 package, package pertama itu adalah langkah praktikum dengan versi 0.0.1 sedangkan post test adalah 0.0.2

Mari kita coba install (update versi terbaru)



Mengecek versi sebelumnya yaitu masih 0.0.1 pada ai\_pkg



Kita install versi terbarunya yaitu 0.0.2 dan berhasil install dengan versi yang paling terbarunya

```

File Edit Selection View Go Run Terminal Help
File Explorer Search
F1 Help x
F2 New Volume (F) > dist > testpy > ...
from al_pkPlanning import PlanningProblem, Action, goal_test
from al_pkUtil import expr
def doubleTennisProblem():
    initial = At(A, LeftBaseline) & At(B, RightNet) & Approaching(Ball, RightBaseline) & Team(A, B) & Team(B, A)
    goal = At(Returned(Ball)) & At(A, RightNet) & At(A, LeftNet)
    actions = [
        Action("Hit(player, Ball, loc)", 
            precond="At(player, Ball, loc) & At(Ball, RightNet)", 
            effects="Returned(Ball)"),
        Action("Go(player, to, loc)", 
            precond="At(player, loc)", 
            effects="At(player, to)")
    ]
    return PlanningProblem(initial, goals=goal, actions=actions)
if __name__ == '__main__':
    p = DoubleTennisProblem()
    print(goal_test(p.goals, p.initial))
    solution = [
        expr('And(LeftBaseline, RightBaseline)'), 
        expr('Not(At(A, Ball), RightBaseline)'), 
        expr('Not(At(A, LeftNet), RightBaseline)')
    ]
    for action in solution:
        n = act + action
        n.act + action
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\PC> & C:\Users\PC\AppData\Local\Programs\Python\Python39\python.exe f:/kcb10/al_pk_version_2/dist/test.py
False
False
PS C:\Users\PC> pip list
Package Version
-----
ai_pkg 0.0.2
distlib 0.3.4
filelock 3.4.2
numpy 1.25.2
openpyxl 3.0.8
pip 21.2.4
plataformdirs 2.1.0
setuptools 59.0.8
six 1.16.0
virtualenv 20.13.1
wheel 0.45.0
WARNING: You are using pip version 21.2.4; however, version 26.0 is available.
You should consider upgrading via the 'C:\Users\PC\AppData\Local\Programs\Python\Python39\python.exe -m pip install --upgrade pip' command.
PS C:\Users\PC>

```

Mencoba jalankan dengan versi terbarunya, ternyata WORK! <3

Kesimpulannya, perbedaan pada langkah praktikum dan post test adalah letak version packagenya. Langkah praktikum kita membuat dengan versi 0.0.1 sedangkan post test membuat versi 0.0.2

Untuk mengakses kodingan package dapat melihat github berikut:

[Artificial-Intelligence/practice/lecture-10 at main · IRedDragonICY/Artificial-Intelligence \(github.com\)](https://github.com/IRedDragonICY/Artificial-Intelligence)

