**DASAR PEMROGRAMAN**

**JOBSHEET 13**

**GIT & PYPI**



{nama mahasiswa}

{nim mahasiswa}

**JURUSAN TEKNIK ELEKTRO**

**PRODI TEKNOLOGI REKAYASA KOMPUTER**

**POLITEKNIK NEGERI SEMARANG**

**2023**

# Materi

* 1. GIT

Git adalah sistem kontrol versi yang paling umum digunakan. Git melacak perubahan yang Anda lakukan pada file, sehingga Anda memiliki catatan tentang apa yang telah dilakukan, dan Anda dapat kembali ke versi tertentu jika diperlukan. Git juga membuat kolaborasi menjadi lebih mudah, memungkinkan perubahan yang dilakukan oleh beberapa orang digabungkan ke dalam satu sumber.

Jadi, terlepas dari apakah Anda menulis kode yang hanya akan dilihat oleh Anda sendiri, atau bekerja sebagai bagian dari sebuah tim, Git akan berguna bagi Anda.

Seperti halnya Google Drive yang memungkinkan beberapa kontributor untuk menulis, mengedit, dan menambah isi satu file teks, git adalah program komputer yang memungkinkan beberapa pembuat kode (dan manajer proyek, penguji, penyedia konten, dan siapa pun yang ada dalam tim) untuk berkolaborasi dalam satu proyek atau untuk proyek Anda pribadi.

Jadi: git adalah sebuah perangkat lunak yang Anda instal di komputer Anda, yang kemudian menangani kontrol versi untuk Anda. Lalu, apa yang dimaksud dengan kontrol versi?

Bayangkan Anda memiliki sebuah proyek baru. Biasanya, Anda berencana untuk menyimpan semua file terkait dalam satu direktori baru. Seiring dengan berjalannya pekerjaan, file-file ini akan berubah. Banyak sekali. Segalanya akan menjadi tidak teratur, bahkan berantakan, dan pada titik tertentu bahkan akan menjadi sangat kacau. Pada saat itu, Anda pasti ingin kembali ke masa lalu ke versi terbaru yang tidak berantakan dan masih berfungsi dari proyek Anda - seandainya saja itu mungkin!

Nah, berkat git, hal itu bisa dilakukan. Kontrol versi terjadi ketika Anda menginstal git di komputer Anda. Git dibuat untuk membuat direktori proyek baru, dan untuk melacak semua perubahan yang Anda lakukan pada setiap dan semua file yang Anda masukkan ke dalam direktori tersebut. Seiring dengan perkembangannya dan Anda membuat penambahan dan perubahan, git mengambil "snapshot" dari versi saat ini. Dan itu, teman-teman, adalah kontrol versi: buat perubahan kecil, ambil snapshot, buat perubahan kecil lagi, ambil snapshot... Dan simpan semua snapshot ini dalam urutan kronologis. Anda kemudian dapat menggunakan git untuk bolak-balik seperlunya melalui setiap versi direktori proyek Anda.

Jadi, ketika kamu mengacaukannya, git seperti memiliki kemampuan ajaib untuk kembali ke masa lalu ke versi terakhir yang bagus sebelum kamu mengacaukannya. Jadi, kontrol versi. git bukan satu-satunya sistem kontrol versi di luar sana, tetapi mungkin yang paling banyak digunakan. Ini juga penting untuk menggunakan GitHub, yang merupakan situs web dan platform publik paling populer untuk menghosting dan berbagi proyek. GitHub tidak sama dengan git - ini adalah hub untuk proyek yang menggunakan kontrol versi git.

* 1. GitHub

GitHub adalah perusahaan nirlaba yang menawarkan layanan hosting repositori Git berbasis cloud. Pada dasarnya, ini membuatnya jauh lebih mudah bagi individu dan tim untuk menggunakan Git untuk kontrol versi dan kolaborasi.

Antarmuka GitHub cukup ramah pengguna sehingga bahkan pembuat kode pemula pun dapat memanfaatkan Git. Tanpa GitHub, menggunakan Git umumnya membutuhkan sedikit lebih banyak pemahaman teknis dan penggunaan baris perintah.

Namun, GitHub sangat ramah pengguna sehingga beberapa orang bahkan menggunakan GitHub untuk mengelola jenis proyek lain - seperti menulis buku.

Selain itu, siapa pun dapat mendaftar dan meng-host repositori kode publik secara gratis, yang membuat GitHub sangat populer di kalangan proyek sumber terbuka.

* 1. PyPi

PyPI, kependekan dari Python Package Index, adalah repositori paket pihak ketiga resmi untuk bahasa pemrograman Python. Ini adalah tempat penyimpanan pusat yang menampung dan mendistribusikan paket perangkat lunak untuk digunakan oleh para pengembang Python.

Developer dapat mengunggah paket Python ke PyPI agar dapat dengan mudah diakses oleh orang lain. Paket-paket ini dapat mencakup pustaka, kerangka kerja, utilitas, dan alat untuk memperluas dan meningkatkan fungsionalitas Python.

PyPI digunakan untuk menemukan dan menginstal paket untuk proyek. PyPI memungkinkan para pengembang untuk dengan mudah mengelola ketergantungan dan merampingkan proses instalasi untuk aplikasi mereka. Dengan PyPI, mereka dapat menghemat waktu dan tenaga dengan menggunakan kembali kode yang sudah ada daripada menulis semuanya dari awal.

Dalam Python, package adalah cara untuk mengatur modul, fungsi, dan kelas yang terkait secara hierarkis. Package menyediakan metode terstruktur untuk menggabungkan kode dan dapat dianggap sebagai direktori yang berisi satu atau lebih modul Python. Package membantu mengatur kode dan membuatnya lebih mudah untuk dikelola dan didistribusikan.

# Langkah Kerja

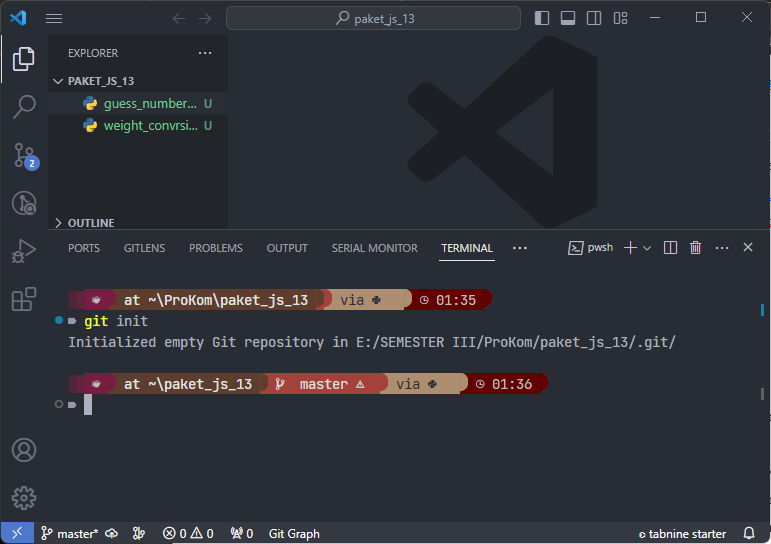
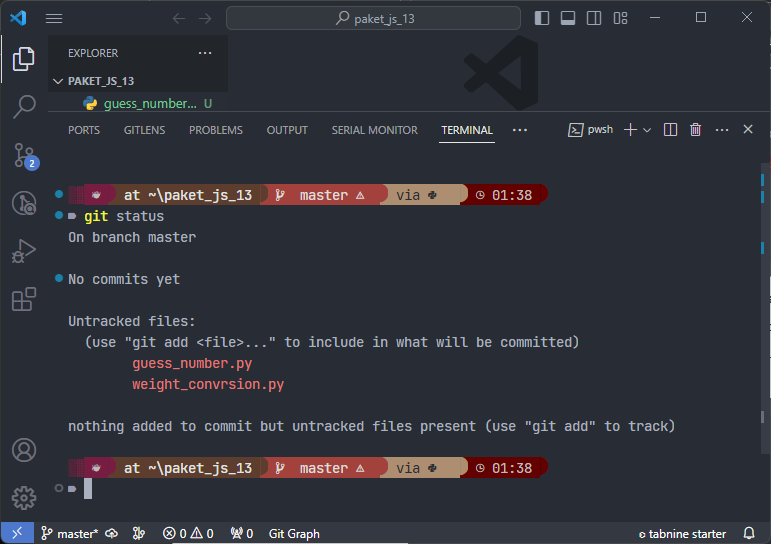
1. Git
   1. Install Git versi terbaru dari website resminya <https://git-scm.com/downloads> , lalu ikut alur penginstallan dari Git tersebut.
   2. Kita perlu memberi tahu Git siapa kita, ini seperti ketika Anda membuat akun Facebook dan Anda perlu mengisikan kredensial Anda seperti nama, alamat, jenis kelamin, dan lainnya. Yang diperlukan oleh git adalah nama serta email, sehingga ketika Anda melakukan proses git, proses tersebut terkait dengan nama dan email yang Anda berikan.

Setelah berhasil menginstall git, buka command prompt, dan ketik command ini.satu-satu

git config --global user.name 'My\_Name'

git config --global user.email 'myEmail@wherever.com'

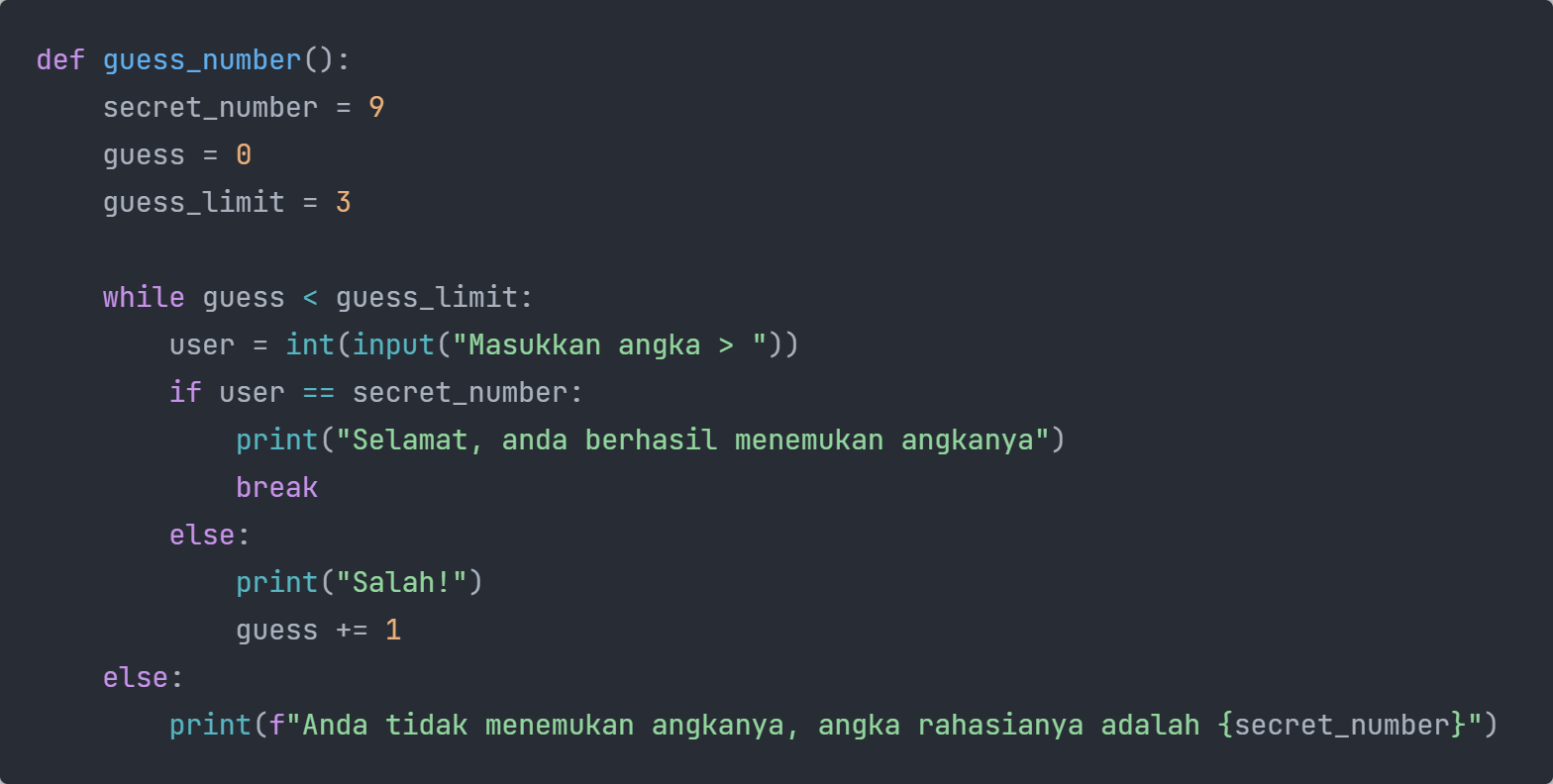
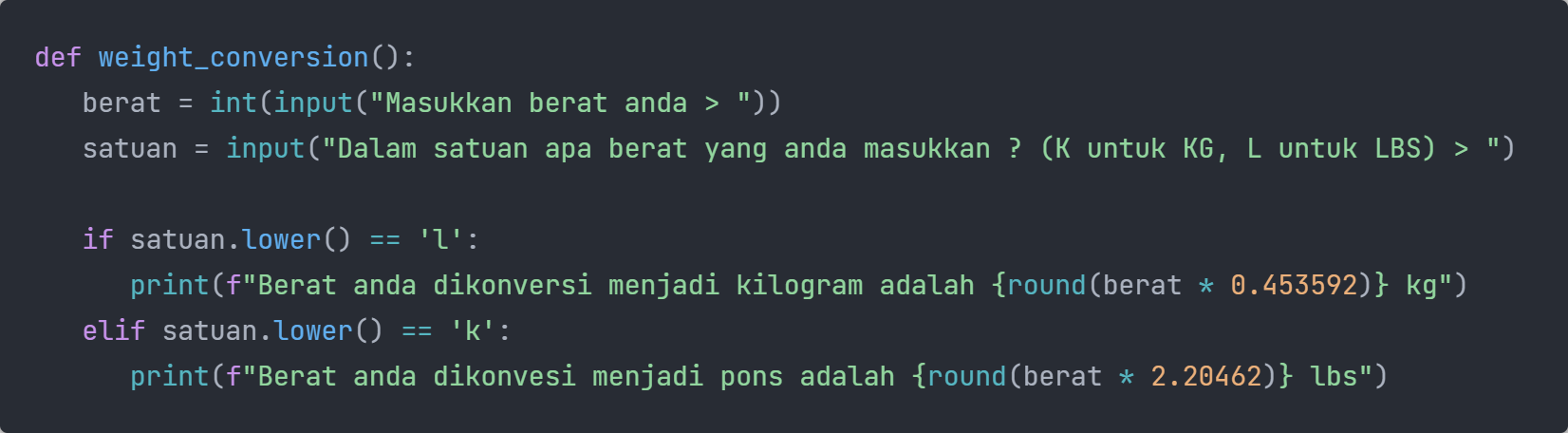
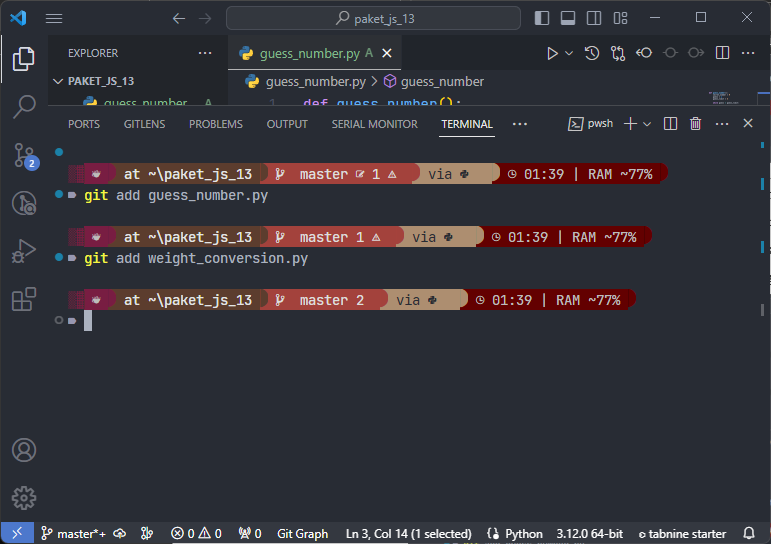
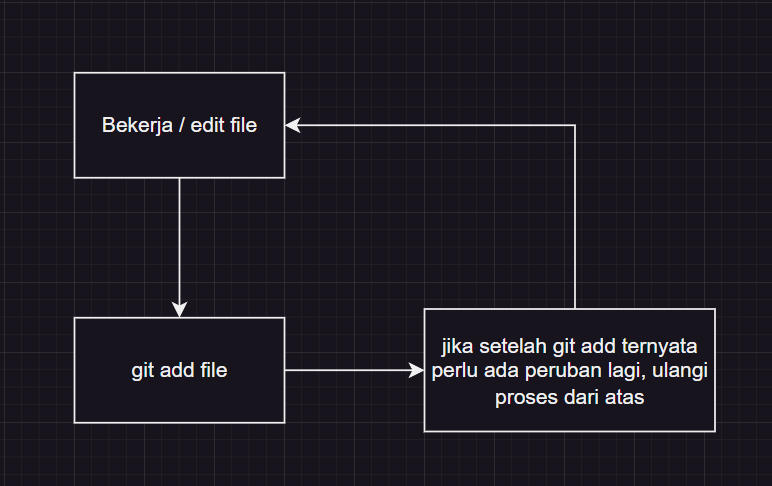
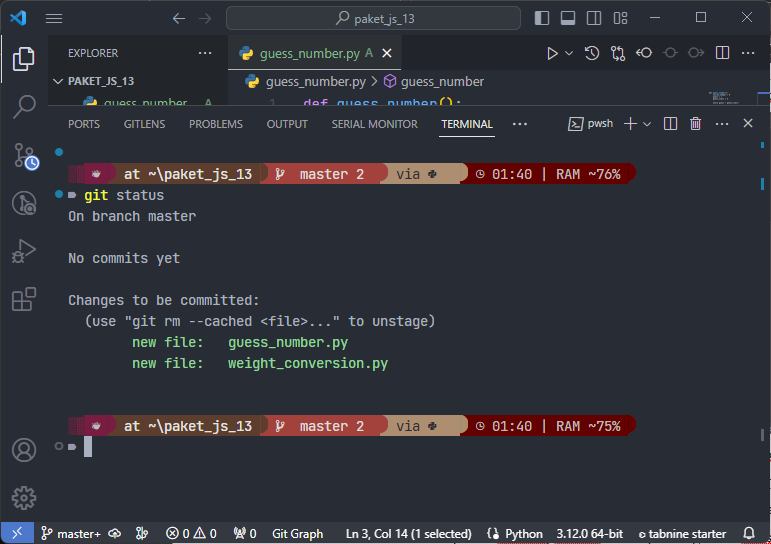
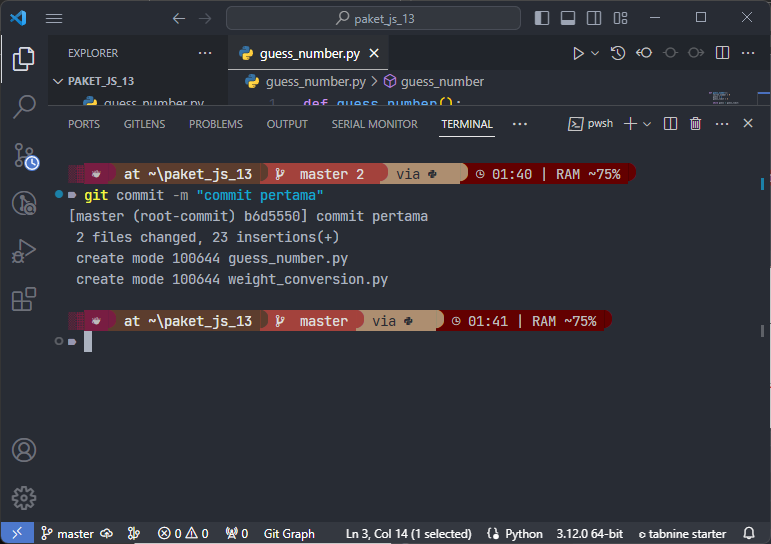
Baris pertama digunakan untuk memberikan nama, baris kedua untuk memberikan email.

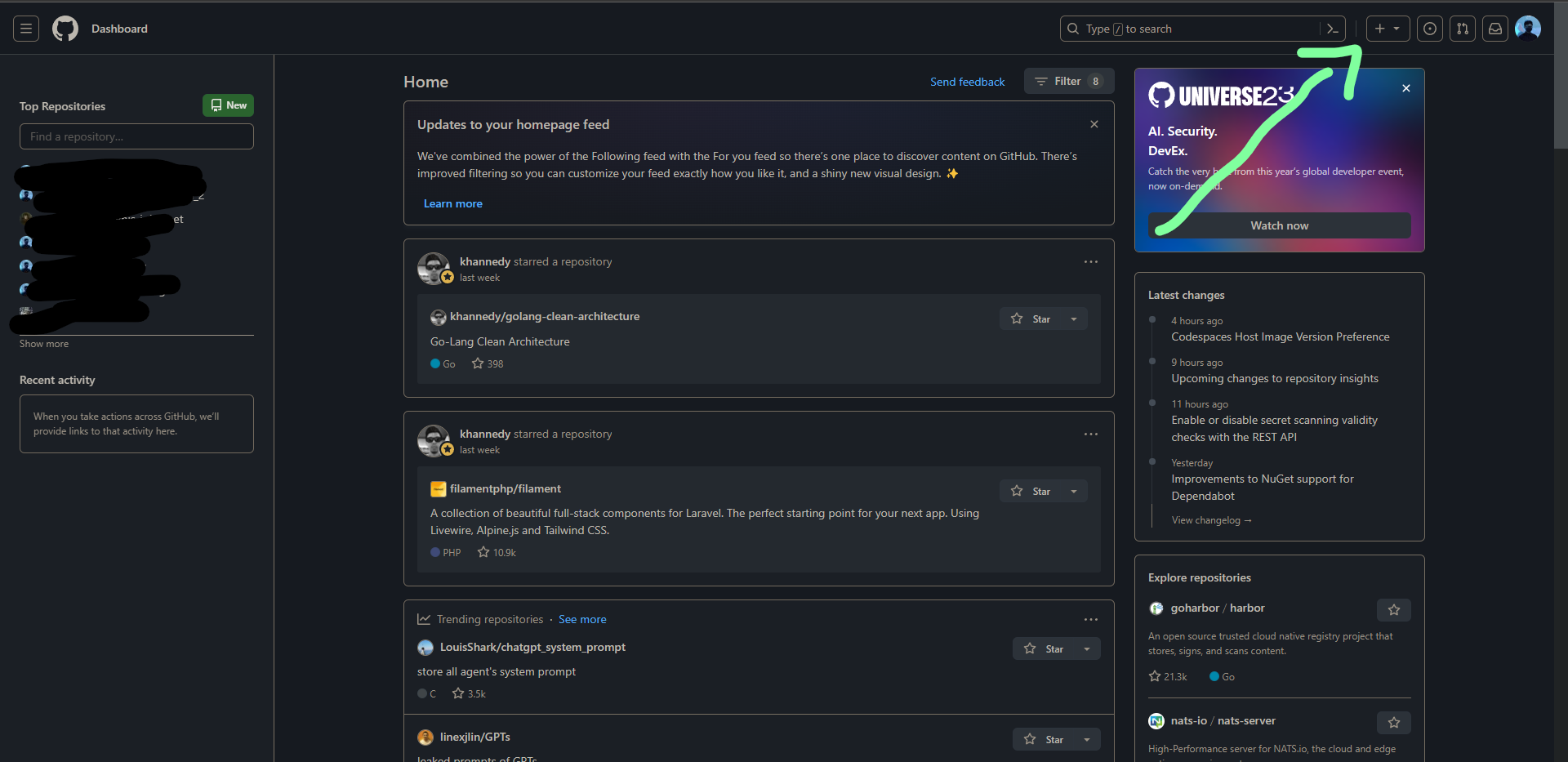
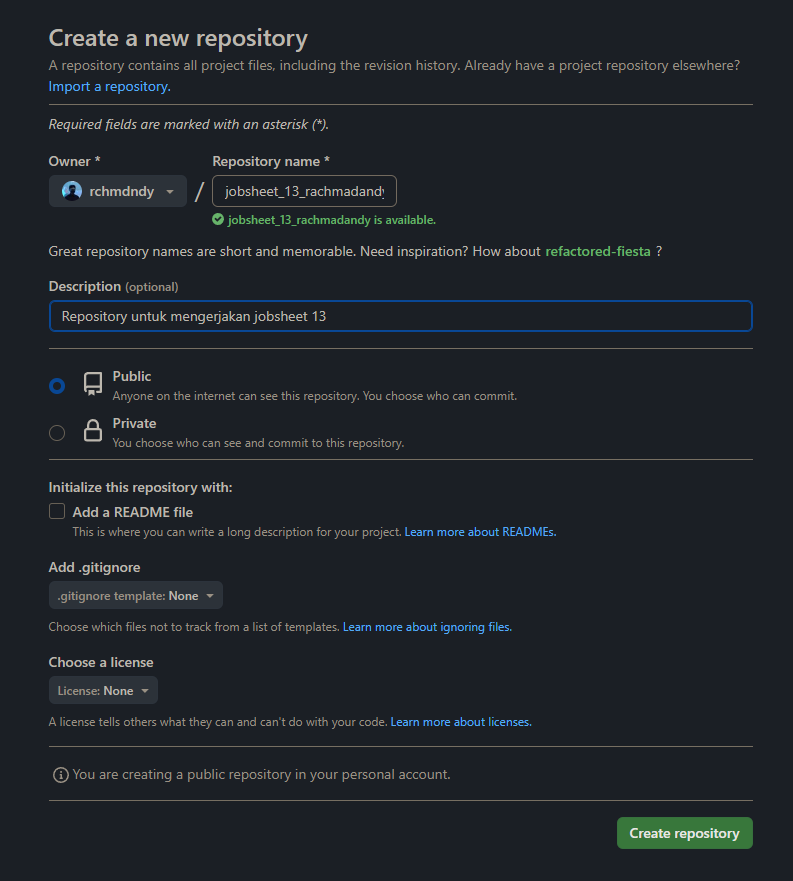
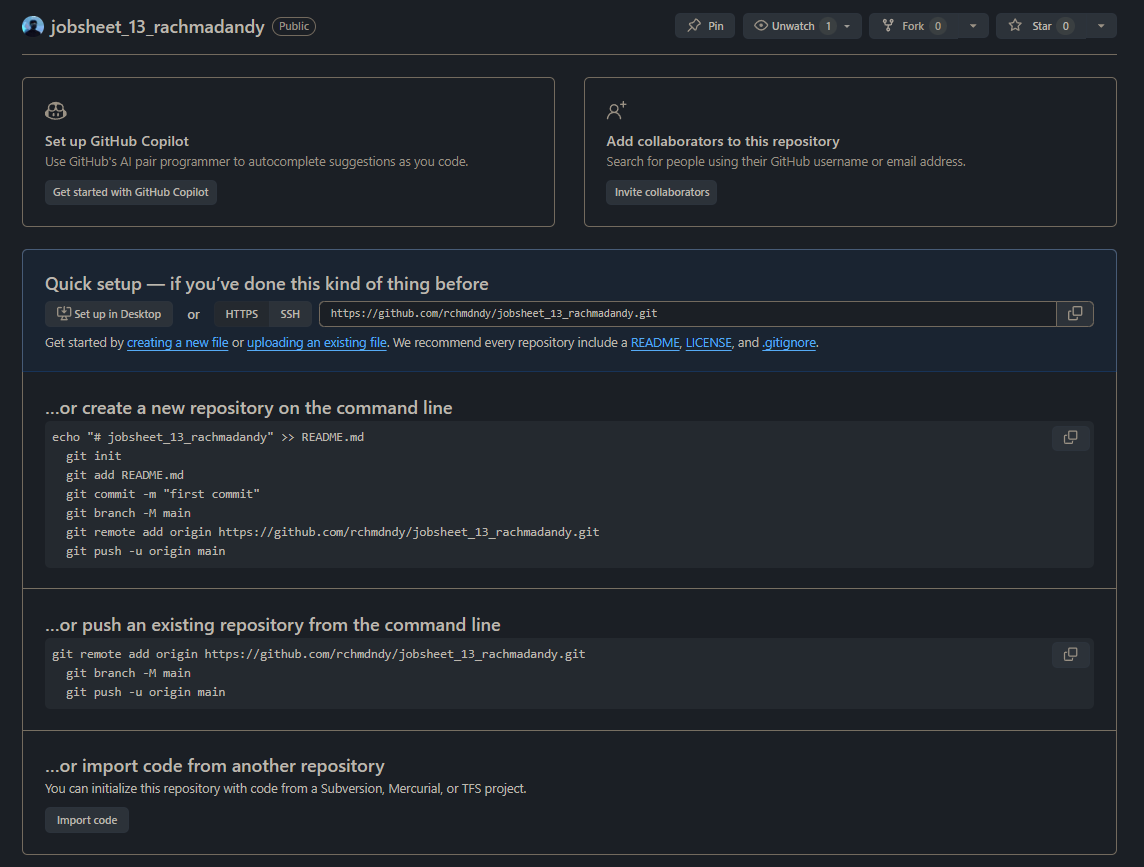
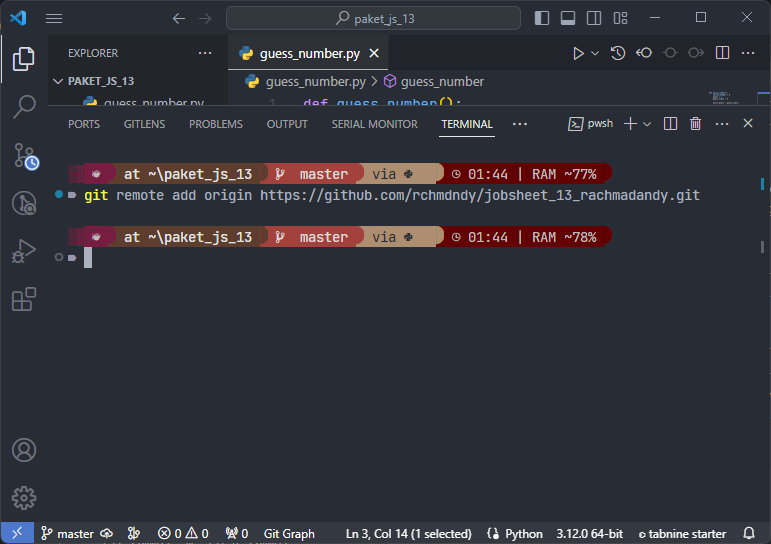
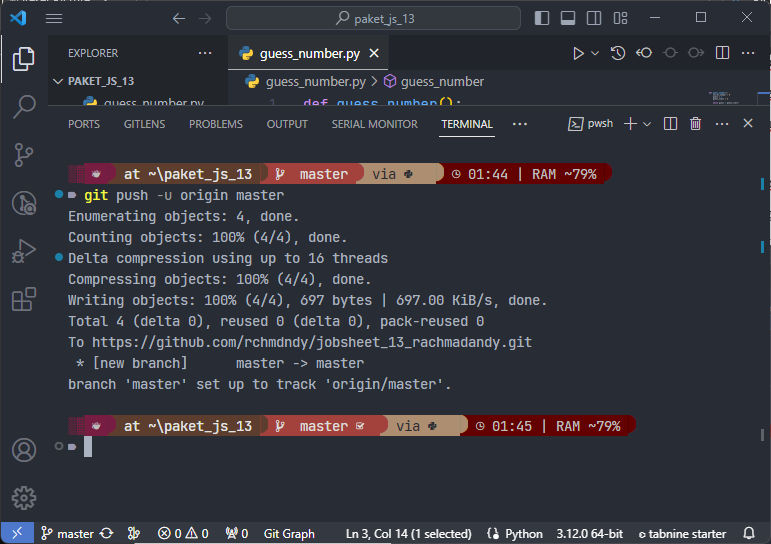
* 1. Buat folder baru **(pada penamaan folder, jangan gunakan dash ( - ) untuk memberi jarak antara kata, gunakan underscore, untuk menghindari error selama pengerjaan jobsheet ini)**, copy paste game yang telah Anda buat pada minggu sebelumnya ke dalam folder baru, lalu buka folder pada text editor kalian (PyCharm, VSCode, dll)
  2. Setelah terbuka, buka terminal (Ctrl + ` (tombol disebelah kiri angka 1)), lalu ketikkan command git init seperti berikut Jika berhasil, maka file Python kalian akan berwarna hijau dan memiliki huruf U, untuk kedepannya, saya akan mulai menyebut folder yang ada pada jobsheet ini sebagai repositori, penjelasan apa itu repositori ada pada bagian materi di atas.
  3. Check status repositori kalian 
  4. Modifikasi isi kedua file python tersebut, dengan menambahkan syntax

def nama\_fungsi():

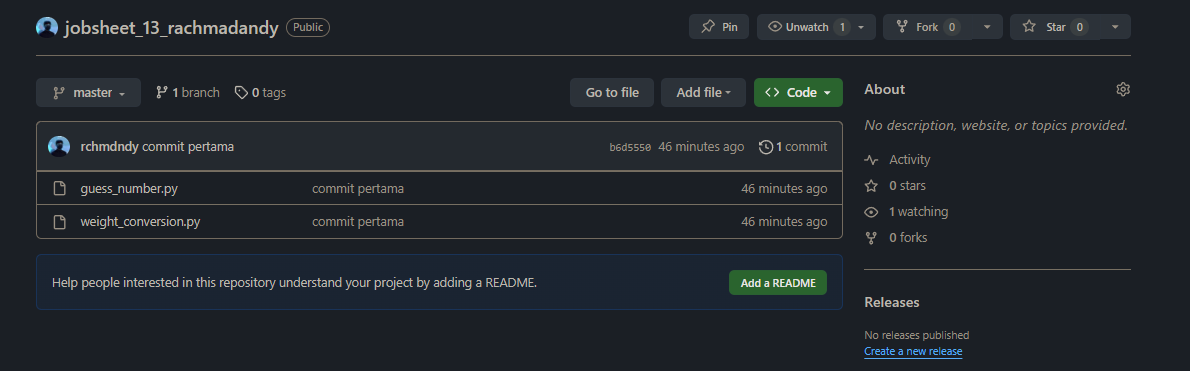
# kode program

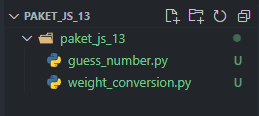
pada paling atas file. contoh seperti ini, sesuaikan dengan kode kalian;

* + 1. Untuk file penebak angka 
    2. Untuk file konversi berat 
  1. Jika Anda lihat, git memberi tahu kita bahwa ada file baru yang belum kita masukkan pada tahap “Staging”, staging sendiri adalah tahap dimana kita mempersiapkan file apa saja yang akan kita simpan untuk kondisi akhir file kita **pada git**, gunakan command git add {nama\_file} untuk memasukkan file ke tahap “Staging” tersebut
  2. Jika File yang sudah masuk pada tahap “Staging” (sudah di git add) mengalami perubahan (di edit dalamnya), maka kita perlu melakukan proses git add lagi, sehingga workflownya kurang lebih seperti ini 
  3. Jika kita cek repositosi menggunakan git status, maka akan tampil sebagai berikut 
  4. Lalu apa fungsi dari git add sebenarnya jika setiap ada perubahan perlu kita git add? Command tersebut berfungsi untuk mempersiapkan file yang akan di “commit” atau sederhananya di-final-kan perubahannya, dalam dunia game bisa diandaikan seperti “checkpoint” terakhir, setelah semua proses git add selesai dan kita ingin membuat “checkpoint” tesrsebut, gunakan command berikut  git commit berfungsi untuk membentuk “checkpoint”, dimana isi “checkpoint” tersebut adalah file yang ada pada tahap “Staging” atau file yang sudah dieksekusi menggunakan command git add.

1. GitHub
   1. Register akun pada website github.com jika sudah memiliki akun, login ke GitHub
   2. Buat repository dengan langkah berikutPilih “new repository”, Isi form yang diperlukan Setelah repository berhasil terbuat, copy text url yang pada form sesuai pada gambar di bawah ini, dengan cara klik tombol copy di sebelah kanan atau select text dan copy
   3. Setelah repository di GitHub berhasil dibuat, dan text link .git telah di copy, kembali ke terminal text editor kalian, lalu ketikkan perintah dibawah Command tersebut berfungsi untuk membuat koneksi dari repository git local ke repositori remote, lalu pada kata **origin**, itu adalah sebuah “nama” untuk url tersebut, atau andaikan saja seperti ini, variabel bernama **origin** bernilai url yang kita berikan setelahnya.
   4. Setelah menambah repositori remote, kita bisa “mendorong” kode kita ke repository remote yang telah kita tambahkan tadi menggunakan git dengan command berikut Command git **push** … selaras dengan kata “mendorong” tadi, secara harfiah memasukkan kode ke repositori yang bernama origin (yang bernilai url tadi) pada branch master menggunakan option -u , option tersebut berguna untuk mengingat parameter yang kita terakhir berikan, dalam kasus ini adalah origin master .
   5. Anggaplah branch sebagai tempat bereksperimen. Anda bisa membuat branch baru untuk mencoba hal-hal baru tanpa mengacaukan branch pekerjaan utama. Jika berhasil, Anda bisa push ke dalam pekerjaan utama; jika gagal, maka kerusakan hanya terjadi pada branch tempat bereksperimen.



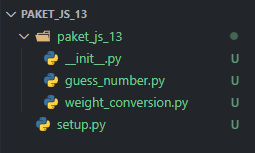
* 1. Setelah push berhasil, lihat laman github repositori di browser (refresh halaman) 

1. PyPi
   1. Alur pembuatan folder package dari awal seperti ini, buat package (folder pertama), lalu buat lagi folder didalamnya dengan nama yang sama, lalu taruh file utama di folder paling dalam, pada kodingan kita, kita hanya perlu menambah 1 folder baru yang bernama sama seperti folder utama kita, contoh paket\_js\_13 
   2. Install package setuptools dengan command

pip install setuptools

pada command prompt

* 1. Buat 2 file dengan ketentuan sebagai berikut
     1. setup.py di direktori utama
     2. \_\_init.py\_\_ (menggunakan dua underscore, 2 di awal 2 di depan) di direktori paling dalam;

Seperti berikut

* 1. Isikan file setup.py dengan kode berikut, ganti nilai variabel name sama seperti nama folder, ganti nilai variabel author, author email, dan description sesuai keinginan

from setuptools import setup, find\_packages

setup(

name='mypackage',

version='0.1.0',

author='Your Name',

author\_email='your.email@example.com',

description='A short description of your package',

packages=find\_packages(),

classifiers=[

'Programming Language :: Python :: 3',

'License :: OSI Approved :: MIT License',

'Operating System :: OS Independent',

],

python\_requires='>=3.6',

)

* 1. Isikan file \_\_init\_\_.py dengan kode berikut, ganti nilai variabel \_\_author\_\_ dengan nama anda, dan nilai variabel \_\_credits\_\_ menjadi “Politeknik Negeri Semarang”

"""

pyexample.

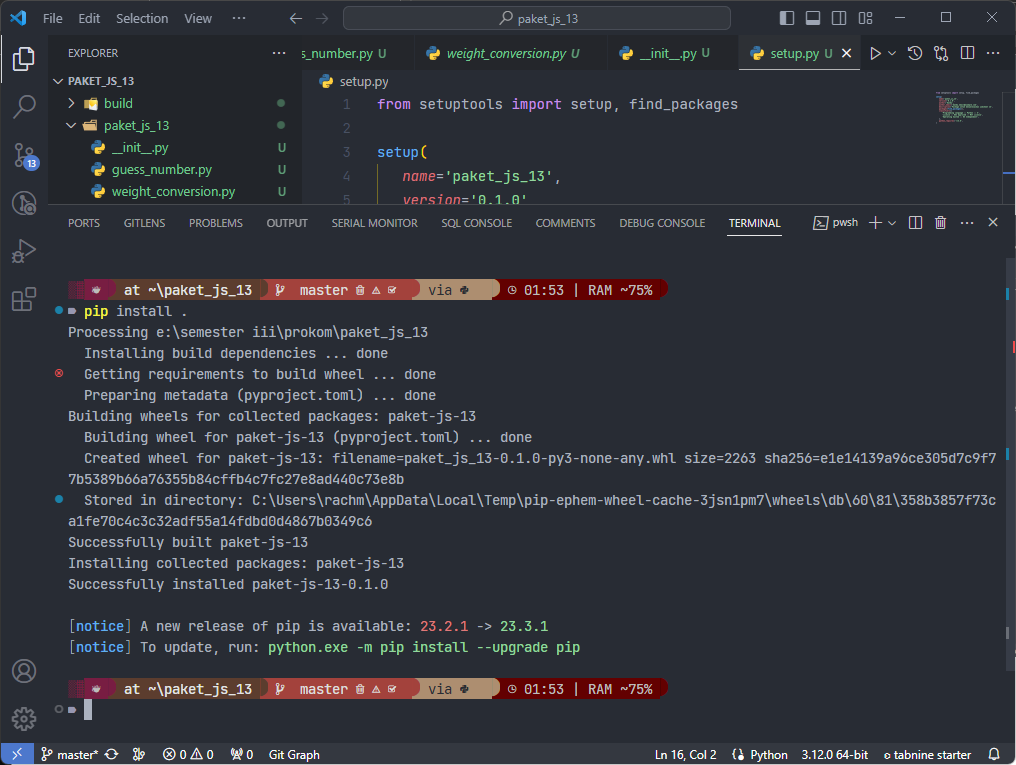
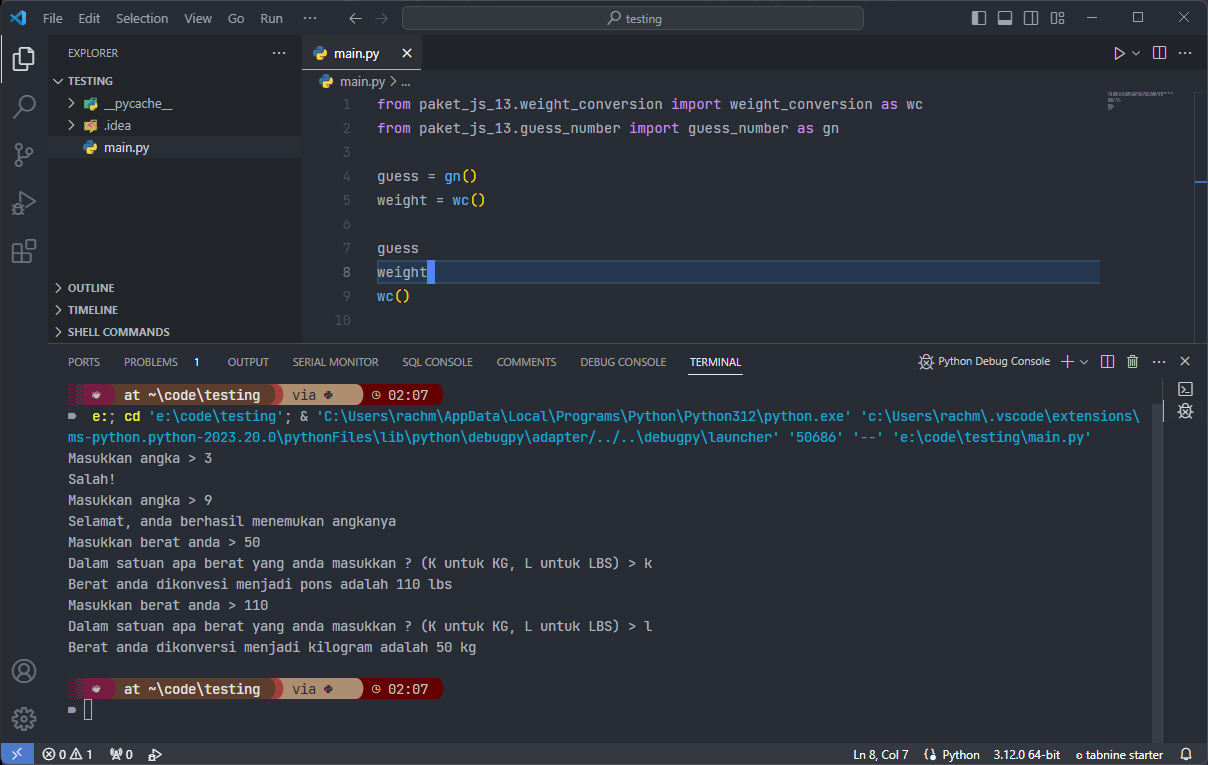
An example python library.

"""

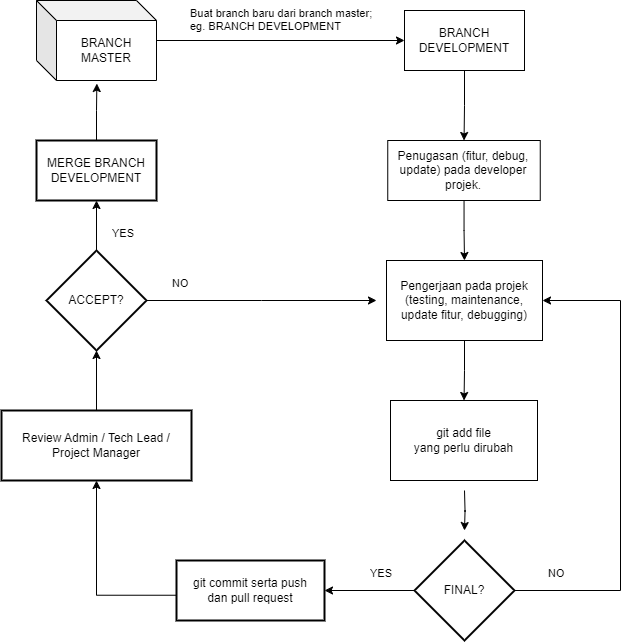
\_\_version\_\_ = "0.1.0"

\_\_author\_\_ = 'Stephen Hudson'

\_\_credits\_\_ = 'Argonne National Laboratory'

* 1. Jalankan perintah pip install . (jangan lupakan titiknya!) pada terminal VSCode 
  2. Coba import nama paket (nama folder) pada file python di direktori yang berbeda dari file paket, lalu test paket kita, contoh 
  3. Selamat, Anda berhasil membuat library / package Anda sendiri!

1. Alur pengerjaan projek menggunakan GitHub



# Latihan Praktikum

|  |  |
| --- | --- |
| git init |  |
| git add |  |
| git commit |  |
| Pembuatan repositori GitHub |  |
| git push |  |
| pip install . |  |
| from {namaPackage.namaFile} import {namaFunction} |  |
| pip show {namaPackage} |  |

# Tugas

-

# V. Sumber referensi

<https://www.nobledesktop.com/learn/git/what-is-git>

<https://kinsta.com/knowledgebase/what-is-github/>

<https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>

<https://betterscientificsoftware.github.io/python-for-hpc/tutorials/python-pypi-packaging/>

<https://www.turing.com/kb/how-to-create-pypi-packages>