Topik : 3.1. Dataset Tabular Simulasi

Objective : Buat dataset dummy subsidi (Dinsos, Dukcapil, Kemenkes)

Task : Format CSV per client → load ke TFF

Source : <https://www.w3schools.com/python/pandas/default.asp>

**PANDAS**

* Pandas Adalah library python yang digunakan untuk bekerja dengan Kumpulan data
* Library ini memiliki fungsi untuk menganalisis, membersihkan , mengeksplorasi, dan memanipulasi data
* Nama “Pandas” merujuk pada “Panel Data” dan “Python Data Analysis” yang diciptakan oleh Wes McKinney pada tahun 2008.

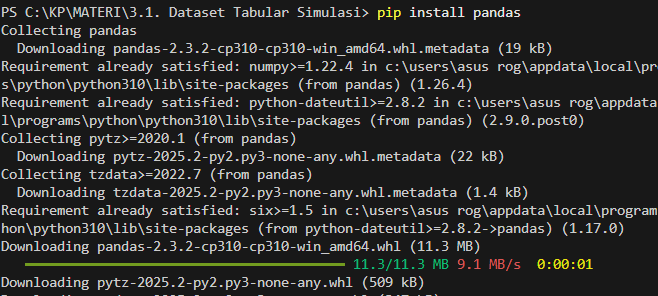
Mengapa menggunakan Pandas ?

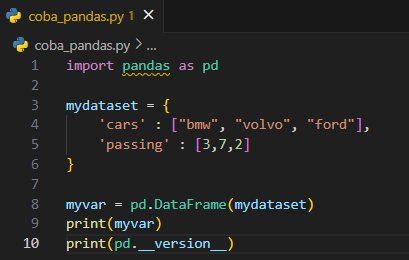
* Pandas memungkinkan kita untuk menganalisis data besar dan menarik Kesimpulan berdasarkan teori statistic
* Pandas dapat membersihkan dataset yang berantakan dan membuatnya menjadi mudah dibaca dan relevan.
* Data yang relevan sangat penting dalam ilmu data

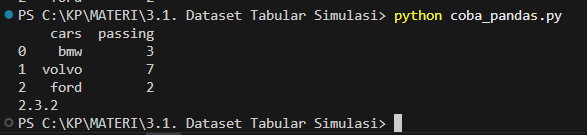
Apa yang dapat dilakukan oleh pandas ?

* Pandas dapat memberikan jawaban tentang korelasi data antara dua kolom atau lebih kolom
* Pandas dapat memberikan jawaban tentang data yang memiliki nilai rata – rata , nilai maksimum, dan nilai minimum.
* Pandas juga dapat menghapus baris yang tidak relevan atau mengandung nilai yang salah seperti nilai kosong atau null (ini disebut pembersihan data).

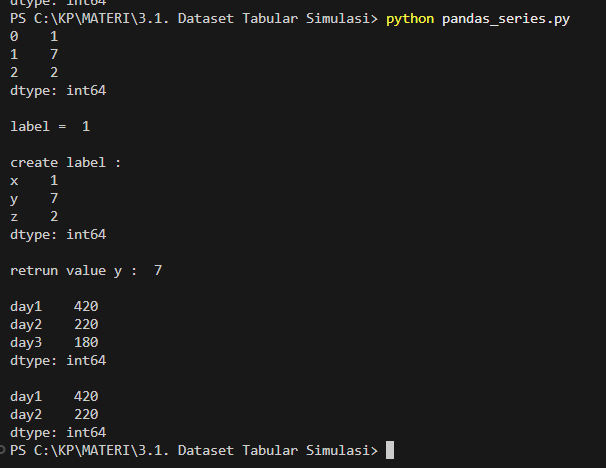
**Installation of pandas**

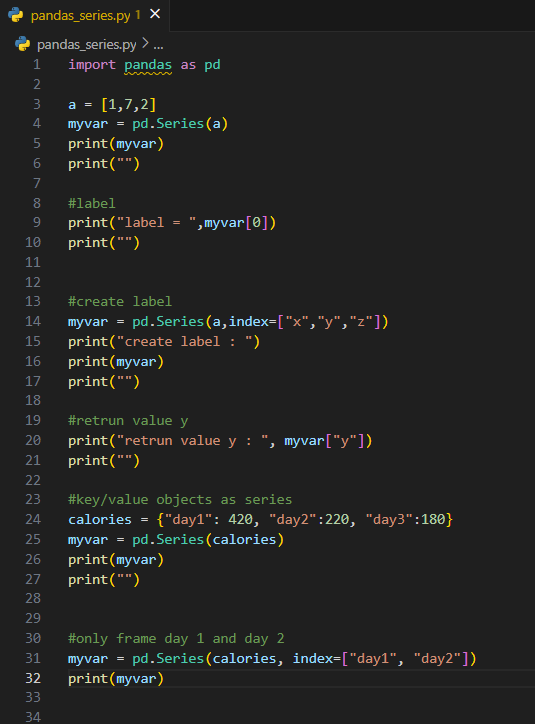






**Pandas Series**

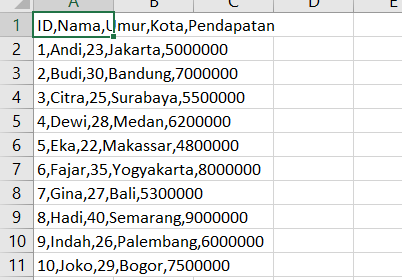
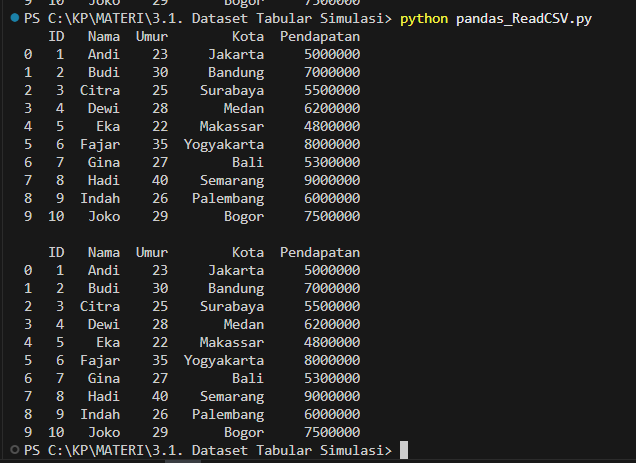
 Panda series mirip dengan kolom dalam table. Ini Adalah array satu dimensi yang menyimpan data dari jenis apapun.

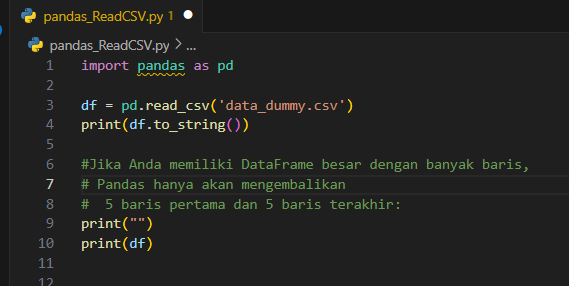


**Pandas Read CSV**

Cara sederhana untuk menyimpan set data besar Adalah dengan menggunakan berkas CSV (berkas yang dipisahkan koma). Berkas CSV berisi teks biasa dan merupakan format yang umum digunakan yang dapat dibaca oleh semua orang , termasuk pandas.

Contoh data\_dummy.csv :

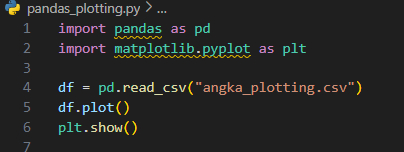
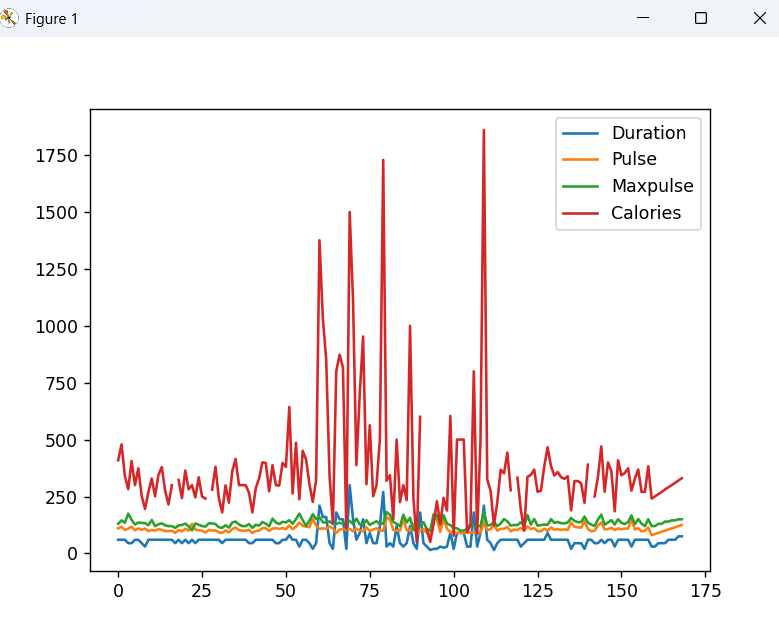




**Pandas Plotting**

1. Plotting

Pandas menggunakan metode plot() untuk membuat diagram. Kita dapat menggunakan Pyplot, sebuah submodule dari perpustakaan Matplotlib untuk menampilkan diagram di layer.

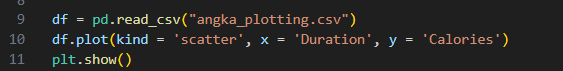


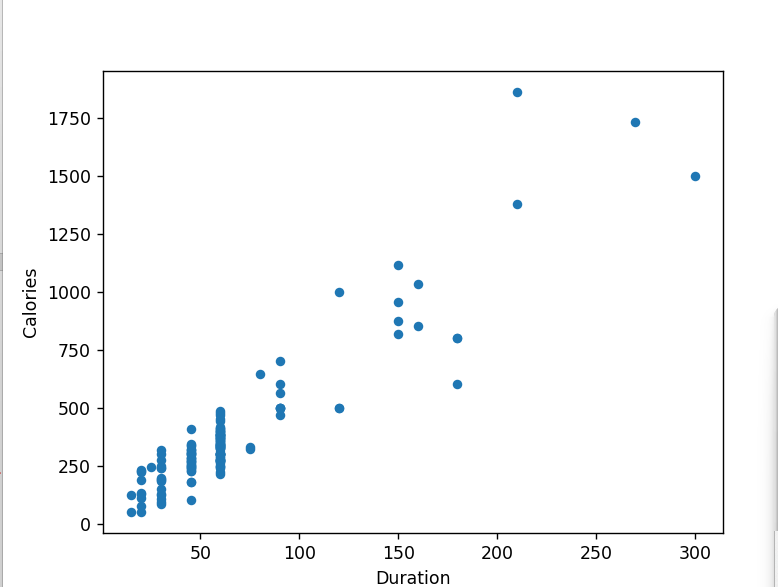
1. Scatter plot

Tentukan bahwa kita ingin membuat scatter plot dengan kind argument :

* + Kind = ‘Scatter’

Scatter plot memerlukan sumbu x dan y. dalam contoh ini, kita akan menggunakan “durasi” untuk sumbu x dan “kalori” untuk sumbu y.



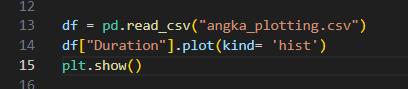
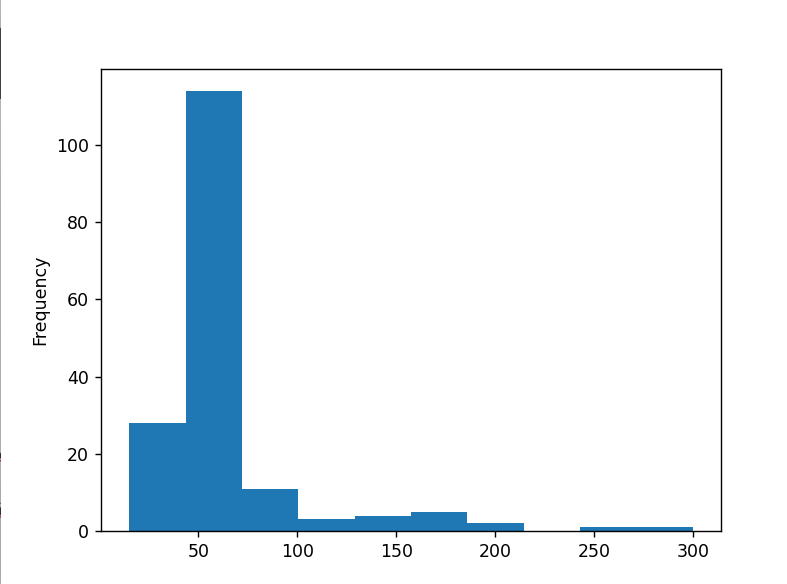


1. Histogram

Gunakan argument ‘kind’ untuk menentukan bahwa anda ingin membuat histogram :

* + Kind = ‘hist’

Histogram hanya memerlukan satu kolom. Histogram menunjukkan frekuensi setiap interval, misalnya, berapa banyak latihan yang berlangsung antara 50 dan 60 menit?



Source : <https://scikit-learn.org/stable/api/sklearn.datasets.html>

**sklearn.datasets**

|  |  |
| --- | --- |
| **LOADERS** | |
|  | Hapus semua akonten dari chace data home |
|  | Simpan dataset dalam format svmlight / libsvm |
|  | Muat nama file dan data dari dataset 20 newsgroup (klasifikasi) |
|  | Muat dan vektorkan dataset 20 newsgroup (klasikasi) |
|  | muat dataset perumahan California (regresi) |
|  | Muat dataset covtype (klasifikasi) |
|  | Ambil file dari web jika belum tersedia di folder local |
|  | Muat dataset kddcup99 (klasifikasi) |
|  | Muat dataset pasangan wajah Labeled Faces in the Wild (LFW) (klasifikasi) |
|  | Muat dataset orang Labeled Facess in the Wild (LFW) (klasifikasi) |
|  | Muat dataset wajah olivetti dari AT&T (klasifikasi) |
| **.** | Ambil dataset dari openml berdasarkan nama atau id dataset |
|  | Mulai dataset multilabel RCV1 (klasifikasi) |
|  | Loader untuk dataset distribusi spesies dari Phillips dkk/ |
|  | Kembalikan path direktori data scikit-learn |
|  | Muat dan kembalikan dataset kanker payudara Wisconsin (klasifikasi) |
|  | Muat dan kembalikan dataset diabetes (regresi) |
|  | Muat dan kembalikan dataset digit (klasifikasi) |
|  | Muat file teks dengan kategori sebagai nama subfolder |
|  | Muat dan kembalikan dataset iris (klasifikasi) |
|  | Muat dan kembalikan dataset Latihan fisik Linnerud |
|  | Muat array numpy dari satu gambar sampel |
|  | Muat gambar contoh untuk manipulasi gambar |
|  | Muat dataset dalam format svmlight / libsvm ke dalam matriks CSR sparse |
|  | Muat dataset dari beberapa file dalam format SVMlight |
|  | Muat dan kembalikan dataset wine (klasifikasi) |

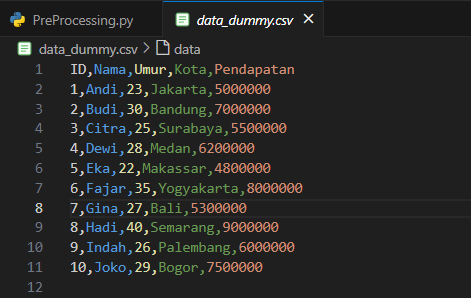
|  |  |
| --- | --- |
| **Sample Generators** | |
|  | Hasilkan array struktur blog diagonal konstan untuk biclustering |
|  | Hasilkan blob Gaussian isotropic untuk clustering |
|  | Hasilkan array dengan structure papan kotak – kotak untuk biclustering |
|  | Buat lingkaran besar yang berisi lingkaran kecil di dalamnya dalam 2D |
|  | Hasilkan masalah klasifikasi acak dengan n kelas |
|  | Hasilkan masalah regresi “Friedman #1” |
|  | Hasilkan masalah regresi “Friedman #2” |
|  | Hasilkan masalah regresi “Friedman #3” |
|  | Hasilkan Gaussian isotropic dan beri label sampel berdasarkan kuantil |
|  | Hasilkan data untuk klasifikasi biner yang digunakan pada Hestie dkk. 2009 |
|  | Hasilkan matriks peringkat rendah dengan nilai singular berbentuk lonceng |
|  | Buat dua setengah lingkaran yang saling bertautan |
|  | Hasilkan masalah klasifikasi multilabel acak |
|  | Hasilkan masalah regresi acak |
|  | Hasilkan dataset kurva berbentuk S |
|  | Hasilkan sinyal sebagai kombinasi jarang (sparse) dari elemen kamus |
|  | Hasilkan matriks jarang yang simetris dan positif-definit |
|  | Hasilkan masalah regresi acak dengan desain jarang yang tidak berkorelasi |
|  | Hasilkan matriks simetris acak yang positif-definit |
|  | Hasilkan dataset berbentuk gulungan Swiss |

Source : <https://www.youtube.com/watch?v=I5mntelNnsM>

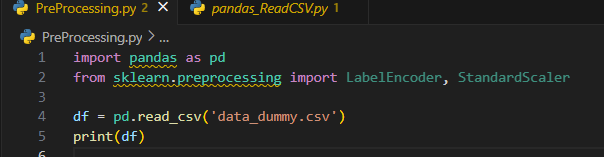
[**https://www.kaggle.com/code/faviansulthanwafi/tugas-eda-kelompok-b/notebook**](https://www.kaggle.com/code/faviansulthanwafi/tugas-eda-kelompok-b/notebook)

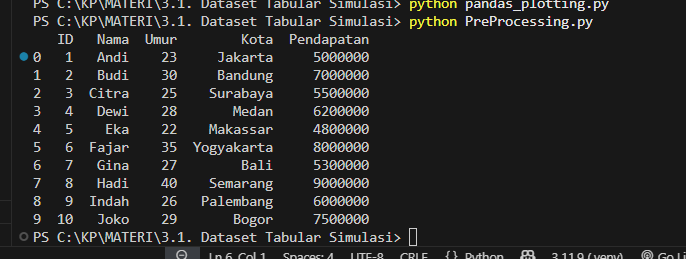
**PREPROCESSING TABULAR DATA DENGAN PYTHON MENGGUNAKAN PANDAS DAN SCIKIT-LEARN :**

1. Menggunakan data Dummy



1. Import Data

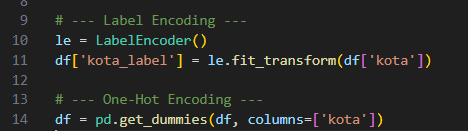




1. Encoding column categorical (kota)

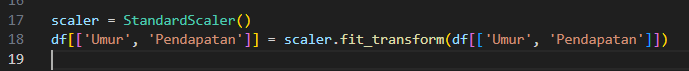
Ada 2 pilihan :

1. Label Encoding 🡪 setiap kota jadi angka unik
2. One-Hot Encoding 🡪 tiap kota jadi kolom biner



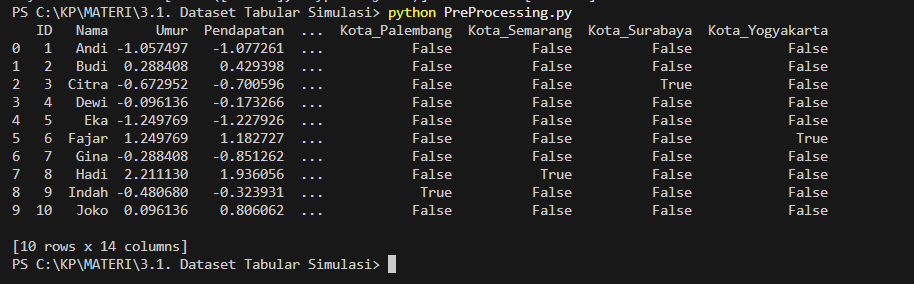
1. Normalisasi / Standarisasi Data Numerik

Kita scaling Umur dan Pendapatan biar punya skala yang sama



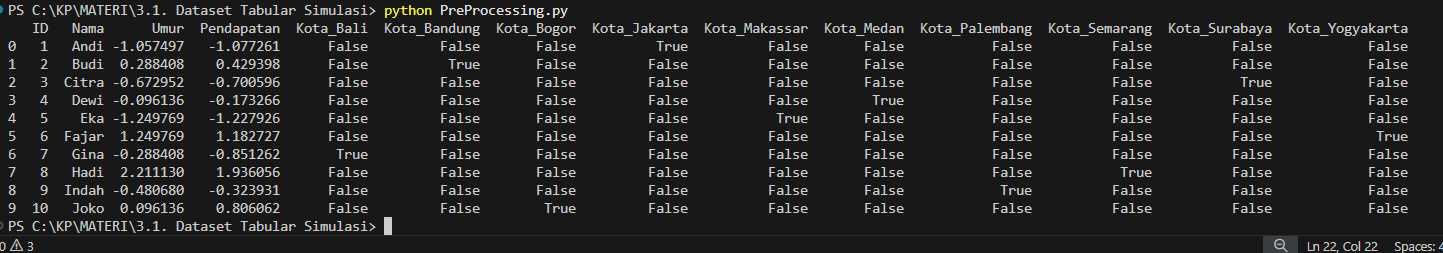
1. Output akhir





Kalau mau kelihatan semua outputnya , maka :





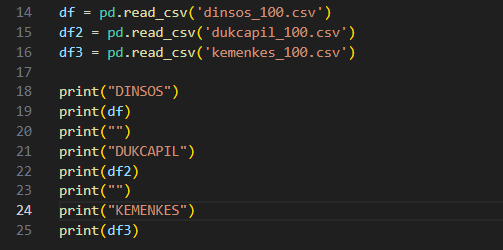
**Task : CSV per Client 🡪 Load ke TFF**

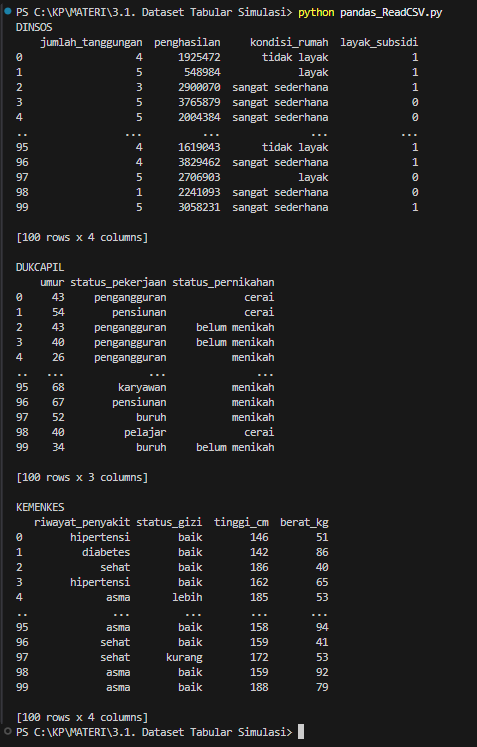
1. Format CSV per client

Dalam federated learning, data biasanya tidak digabung menjadi satu, tetapi tersimpan di masing – masing client. Artinya :

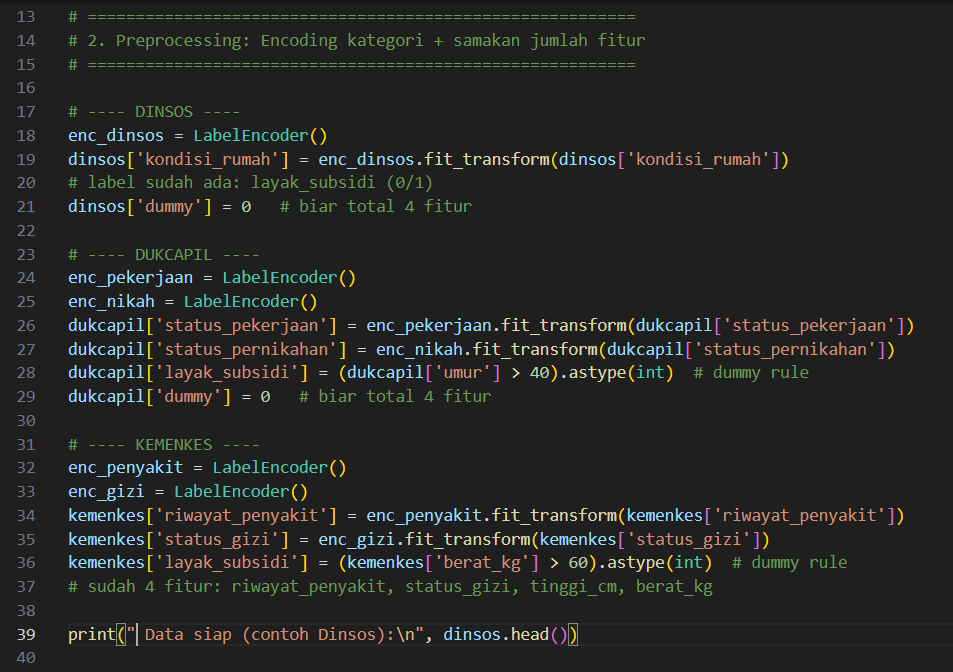
* + Dinsos.csv (100 baris data) 🡪 client 1
  + Dukcapil.csv (100 baris data) 🡪 client 2
  + Kemenkes.csv (100 baris data) 🡪 client 3

1. Buat File CSV nya



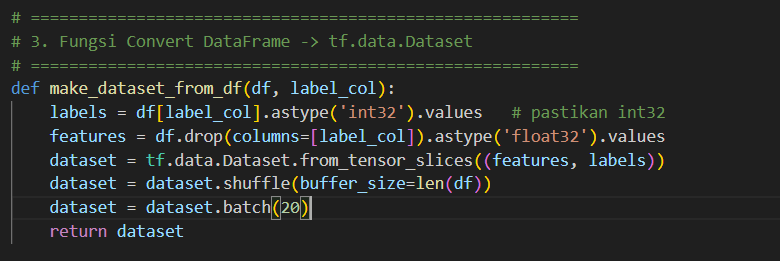


1. Preprocessing : Encoding + samakan jumlah fitur



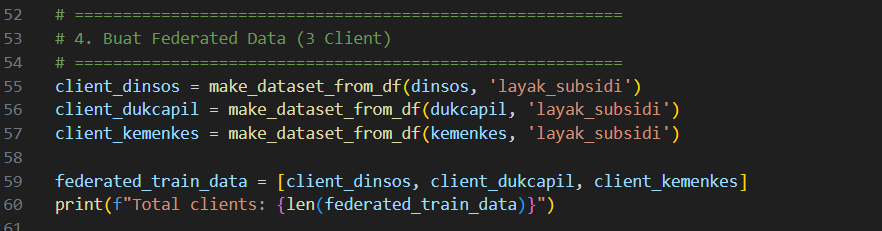
* + Dinsos 🡪 encode kondisi\_rumah (kategori teks jadi angka) + tambahkan kolom dummy biar total fiturnya 4.
  + Dukcapil 🡪 encode status\_pekerjaan & status\_pernikahan, buat label layak\_subsidi (rule: umur > 40), tambah kolom dummy.
  + Kemenkes 🡪 encode riwayat\_penyakit & status\_gizi, buat label layak\_subsidi (rule: berat\_kg > 60), sudah punya 4 fitur jadi tidak perlu dummy.

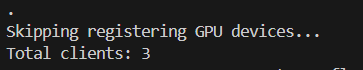
1. Fungsi convert DataFrame 🡪 tf.dataset



* + Mengubah pandas.DataFrame jadi tf.data.Dataset 🡪 format yang dipahami oleh TFF
  + Dataset di-shuffle agar training tidak bias urutan
  + Dibagi batch ukuran 20
  + Output per elemen: (fitur, label) → (4 angka, 1 target biner).

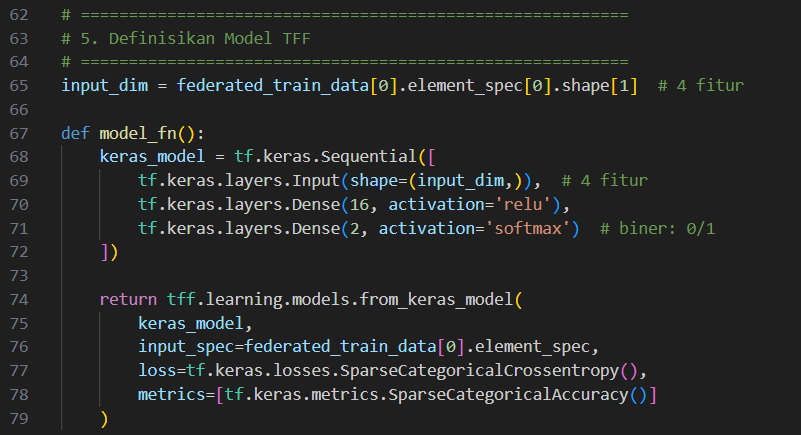
1. Buat Federated Data (3 client)





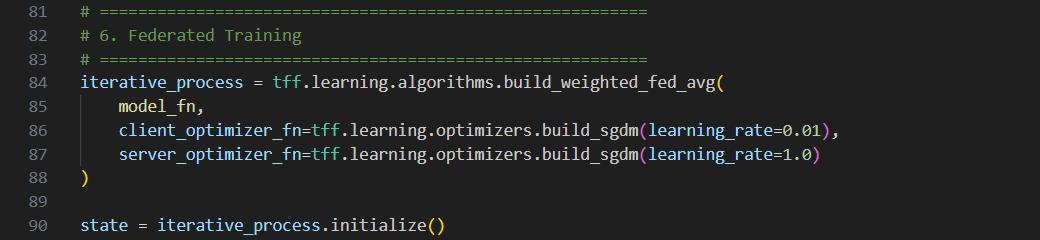
* + Tiap dataset dijadikan 1 client
  + Semua client digabungkan ke list federated\_train\_data
  + Jadi, federated learning akan melatih 3 model

1. Definisikan model



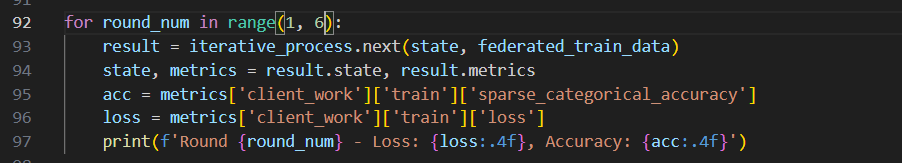
* + Model Keras Sederhana
    - Input : 4 fitur
    - Hidden Layer : Dense 16 neuron dengan aktivasi ReLU.
    - Output Layer : Dense 2 neuron dengan softmax → klasifikasi biner (tidak layak=0, layak=1).

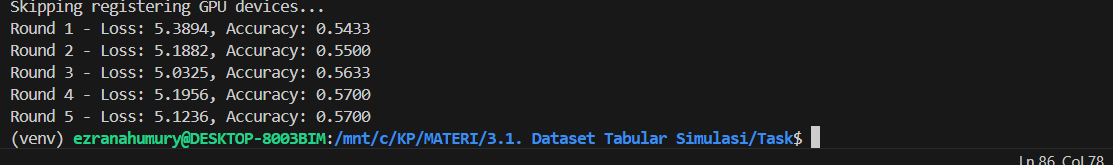
1. Federated Training



* + Menggunakan Federated Averaging (FedAvg).
  + Optimizer :
    - **Client** 🡪 SGD learning rate 0.02 (update model lokal).
    - **Server** 🡪 SGD learning rate 1.0 (gabungkan model dari client).
  + state = iterative\_process.initialize() 🡪 inisialisasi model global.\

1. Training loop





* + Round Federated Learning
    - Model global dikirim ke client
    - Client melatih model dengan datanya masing – masing
    - Update model dengan datanya masing – masing
    - Server menggabungkan (average) update 🡪 model global baru