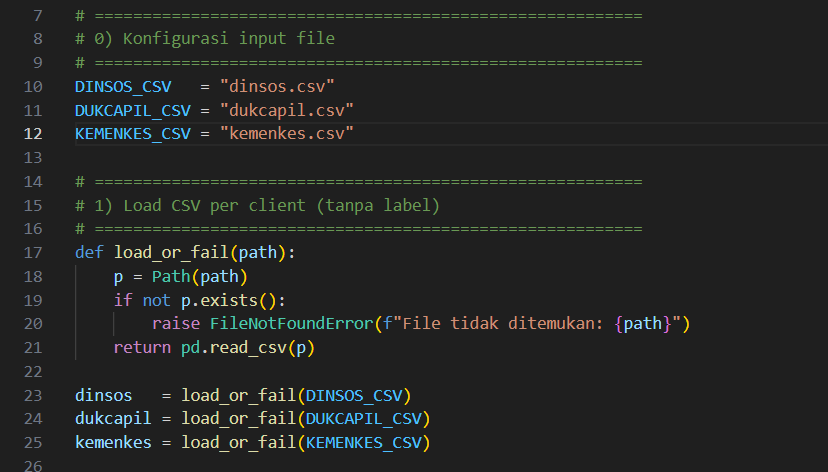
**TASK GABUNGAN 3.1 – 3.3**

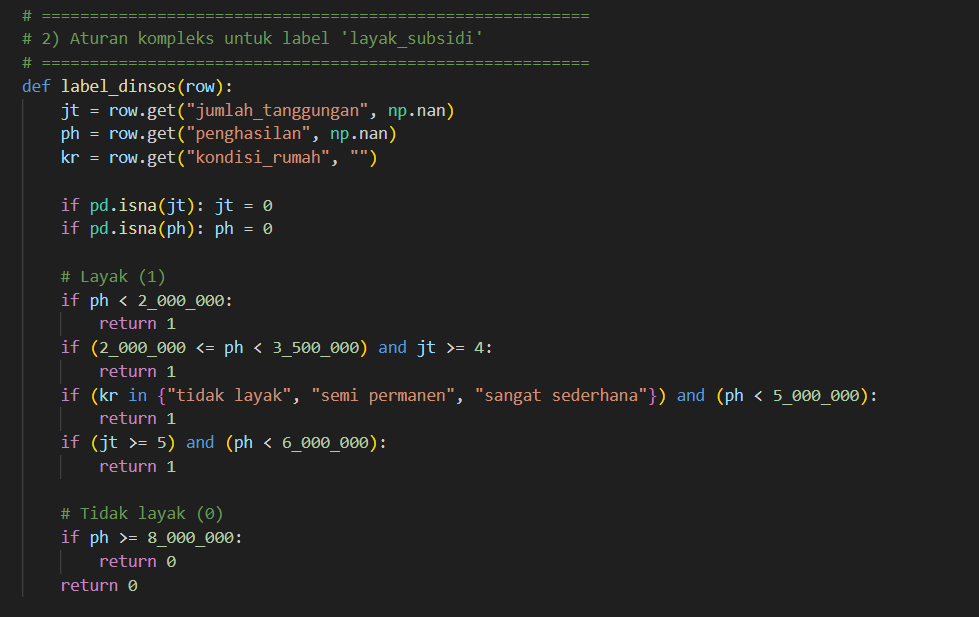
1. Load data per client



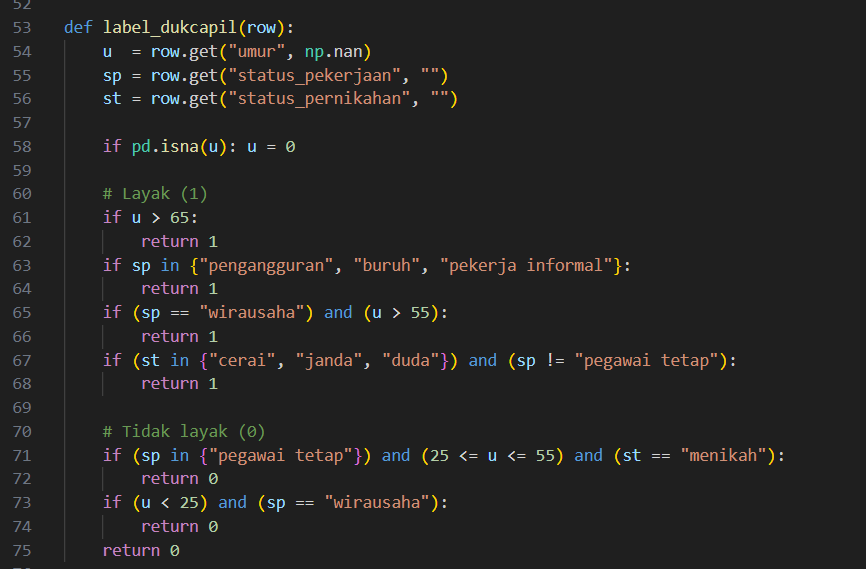
Baca tiga CSV (Dinsos, Dukcapil, Kemenkes) jadi DataFrame.

1. Buat Label biner local

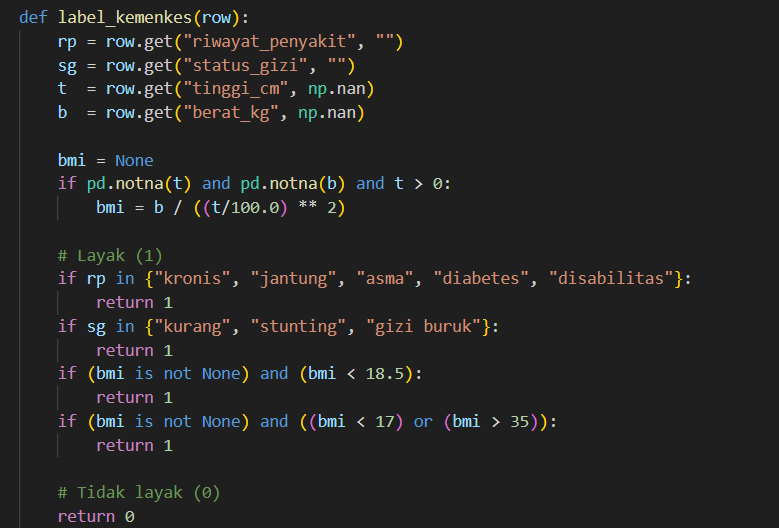
Dinsos :



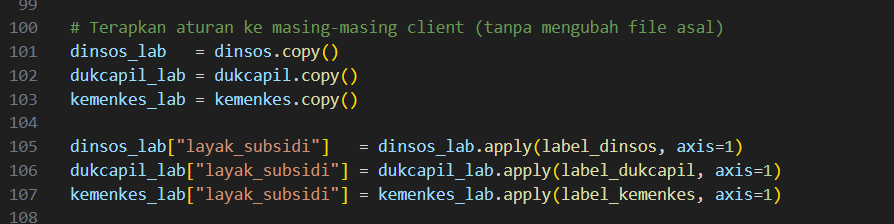
Dukcapil :



Kemenkes :

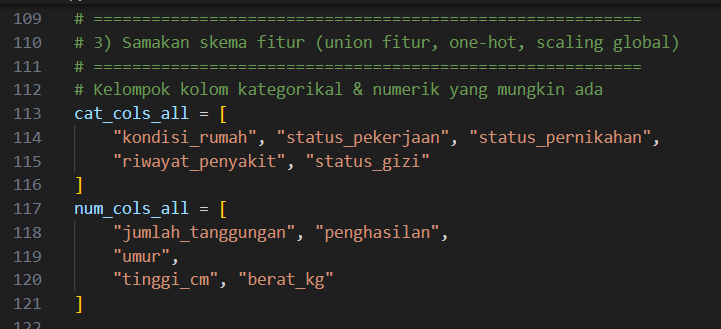


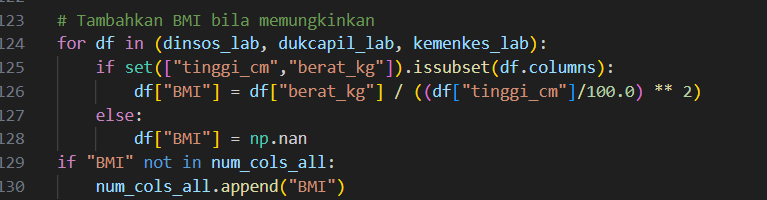
Terapkan aturan domain per sumber untuk menghasilkan layak\_subsidi (0/1) tanpa mengubah file asli.

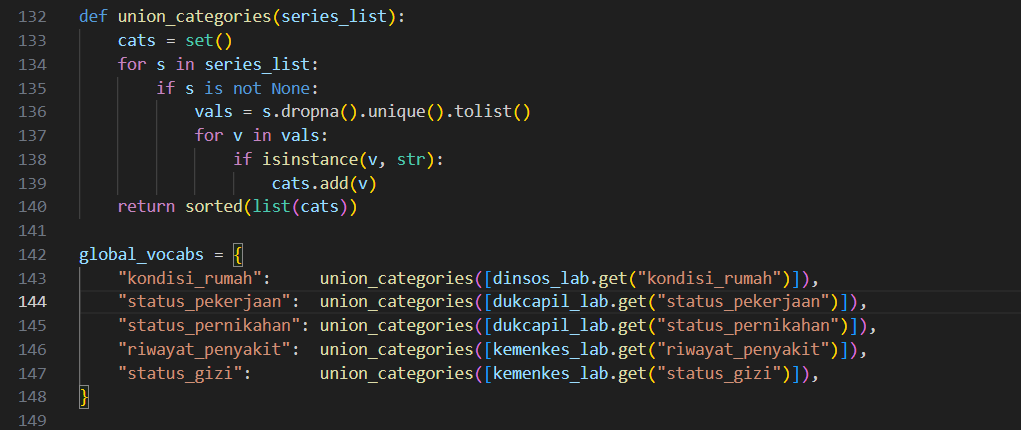


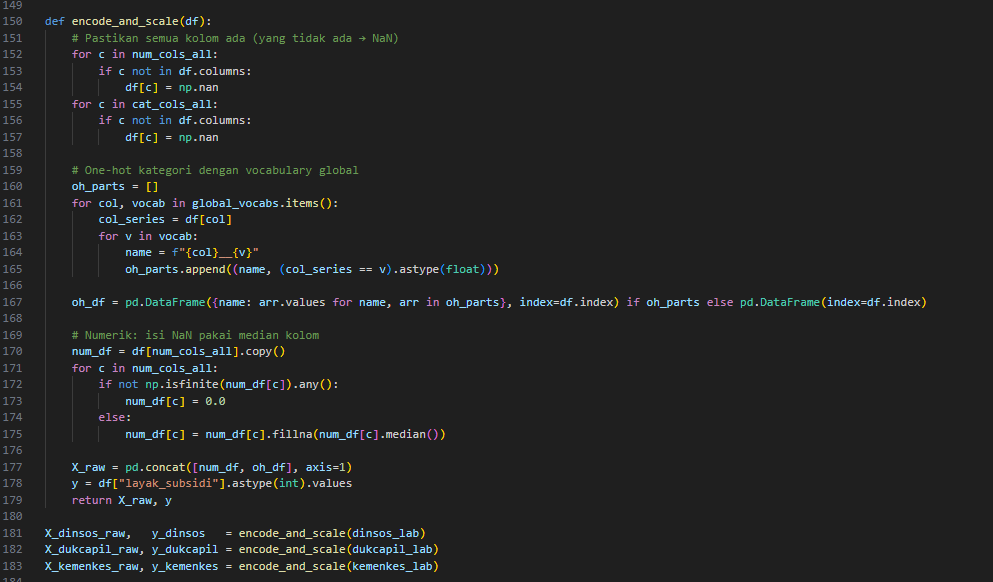
1. Samakan skema fitur lintas-klien

* Tambah fitur turunan **BMI** bila ada tinggi/berat.
* Bentuk **vocabulary global** untuk kolom kategorikal, lalu **one-hot** agar urutan & jumlah kolom konsisten.
* Imputasi numerik (median) untuk NaN.

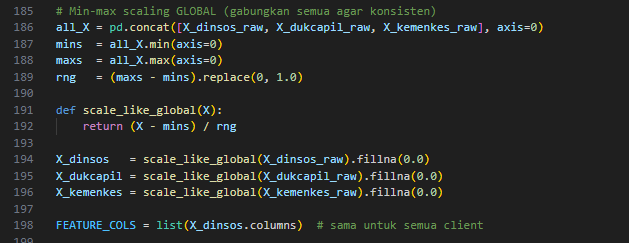




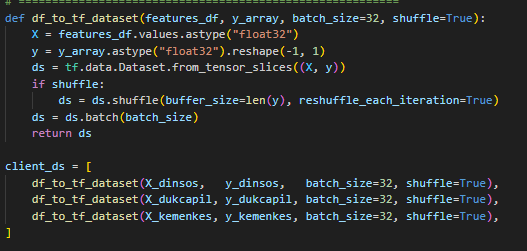




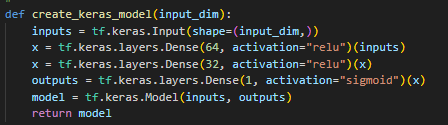
1. Scaling global yang konsisten  
   Gabungkan semua fitur → hitung min/max global → terapkan min-max scaling ke tiap klien.



1. Siapkan dataset Tensors  
   Konversi tiap (X, y) klien ke tf.data.Dataset (shuffle + batch).

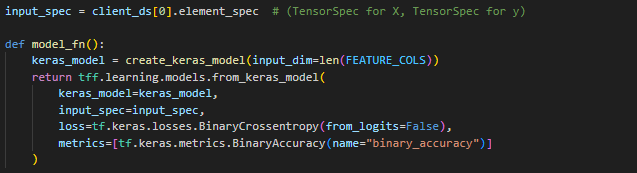


1. Bangun model Keras sederhana  
   MLP: Dense(64) → Dense(32) → Dense(1, sigmoid) untuk klasifikasi biner.

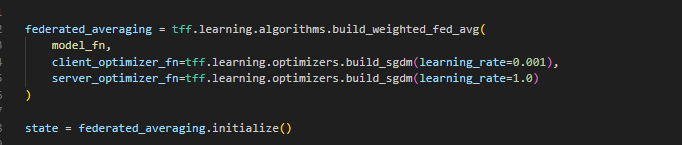


1. Wrap ke TFF

Gunakan tff.learning.models.from\_keras\_model dengan input\_spec, BinaryCrossentropy, dan metrik BinaryAccuracy.



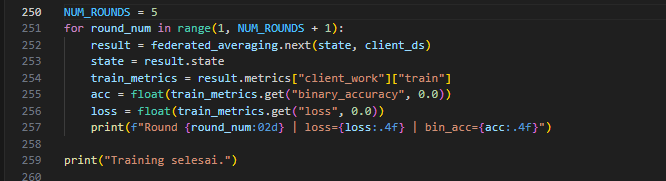
1. **Konfigurasi Federated Averaging**  
   Build FedAvg dengan optimizer klien (SGD lr=0.001) dan server (SGD lr=1.0).

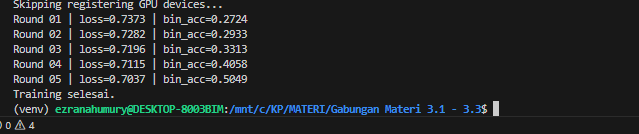


1. Training federated

Inisialisasi state → loop beberapa ronde → tiap ronde: latih di klien, agregasi ke server, update model global, dan log loss & binary\_accuracy.

Menggunakan 5 round :





Menggunakan 10 round :

