Topik : 3.3. Training Federated dengan Tabular Data

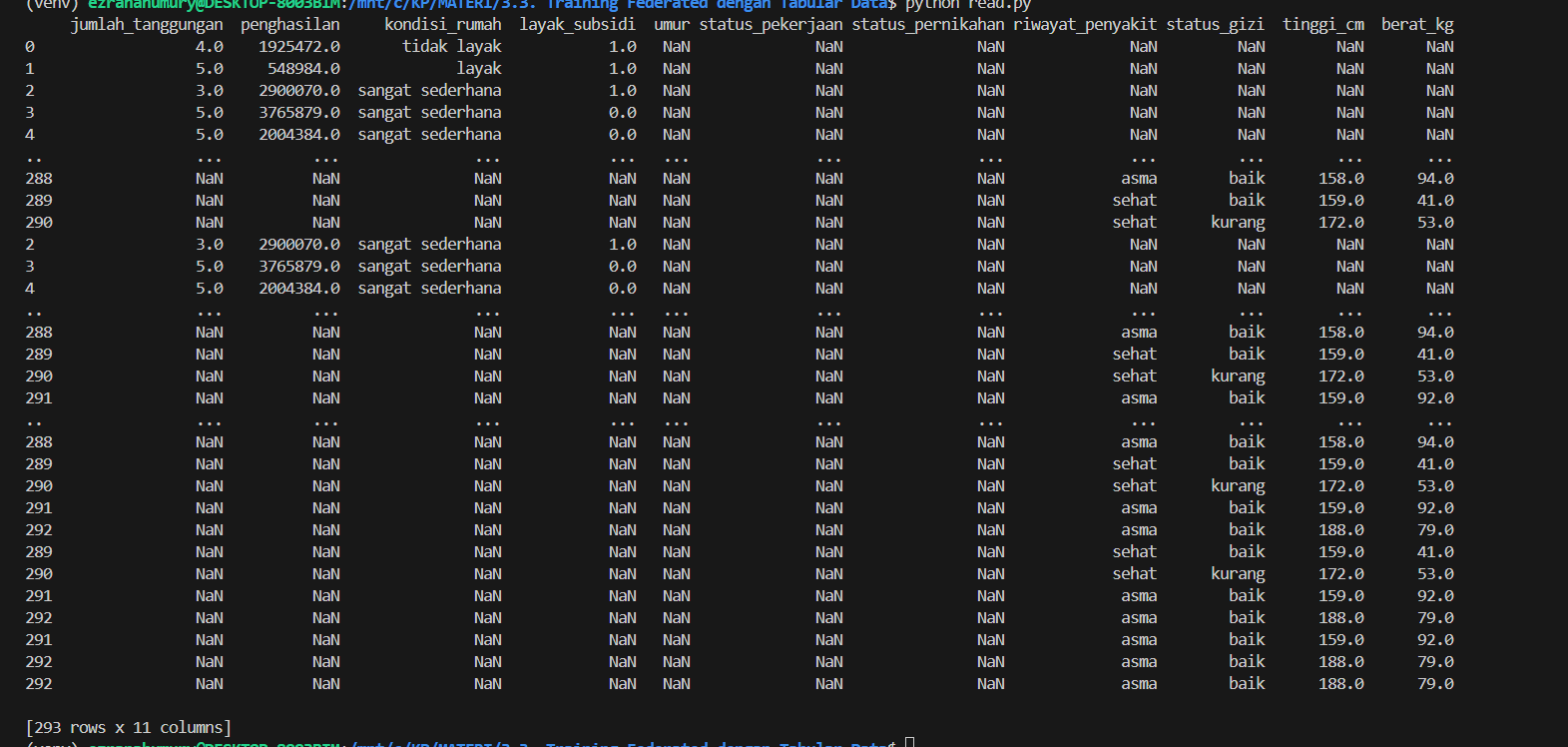
Objective : Latih FL model dengan dataset dummy 3 client

Task : Simulasikan 5-10 round + logging akurasi

Source : <https://medium.com/@irfandy.thalib/menggabungkan-2-dataset-dengan-pandas-dan-eksport-ke-csv-167b535c4f45>

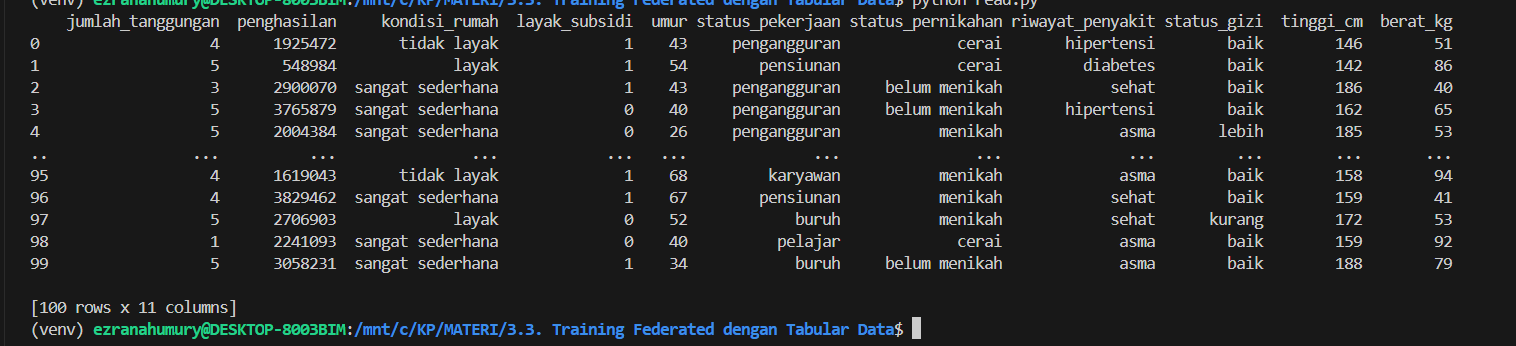
**Menggabungkan 3 Dataset Dengan Pandas dan Ekspor ke CSV**

OPSI 1 : Pandas akan meng-*union* semua kolom dari ketiga file

Hasil akhirnya: file gabungan.csv memang punya semua kolom, tapi sebagian besar nilai jadi NaN karena tiap instansi hanya punya subset kolom tertentu.

OPSI 2 : Column-Wise 🡪 ketiga client digabung menjadi baris gabungan



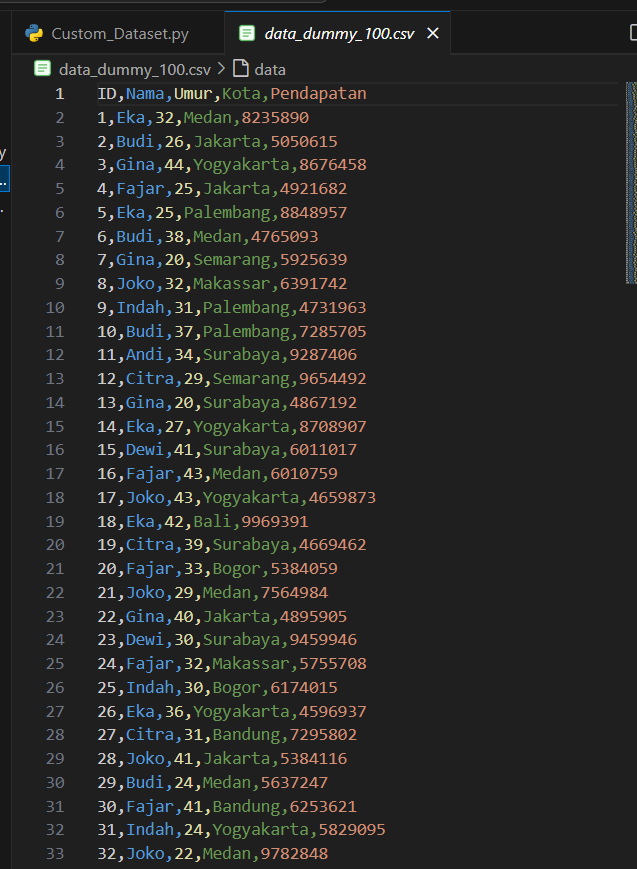
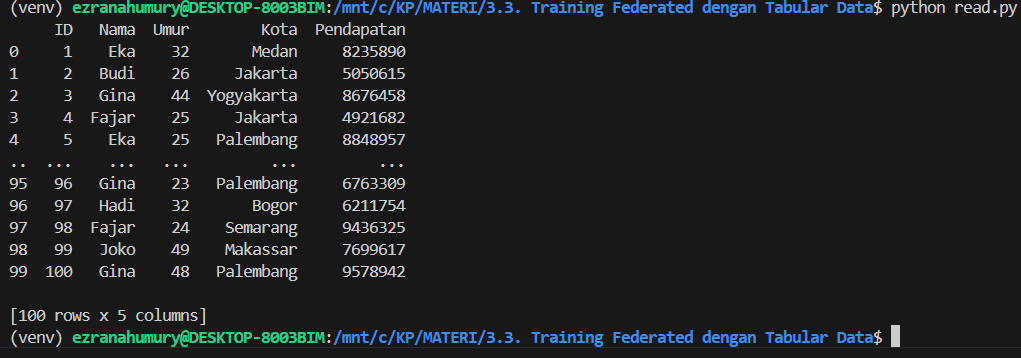


**Pandas + TFF custom dataset examples**

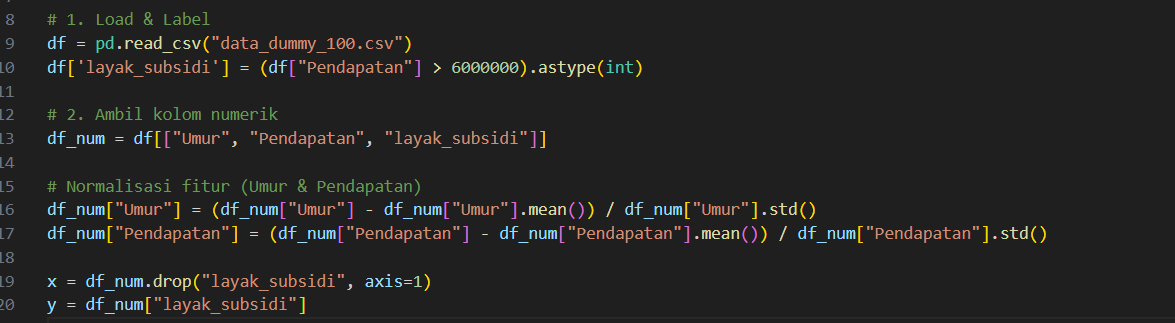
Referensi : <https://colab.research.google.com/github/tensorflow/federated/blob/v0.88.0/docs/tutorials/federated_learning_for_image_classification.ipynb>

Federated Learning for custom datasets, bagian custom dataset itu Adalah Ketika kita tidak memakai dataset bawaan TFF (Seperti EMNIST, Shakespeare, StackOverflow) , tapi justru mempersiapkan sendiri dataset kita (misalnya CSV dengan Pandas) lalu mengubahnya menjadi tf.data.Dataset untuk tiap klient.

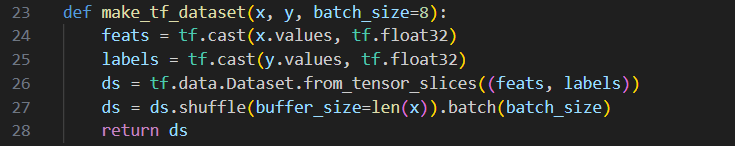
Menggunakan Dataset Dummy :



1. Menyiapkan Dataset dengan pandas

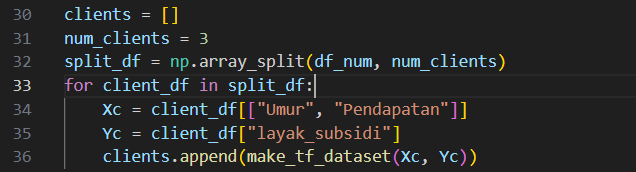


1. Konversi ke Tensorflow Dataset



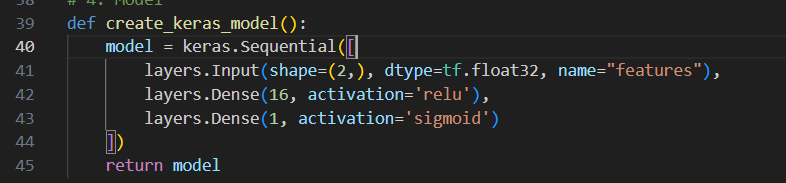
* tf.data.Dataset dipakai TFF untuk baca data.

1. Bungkus menjadi federated Dataset



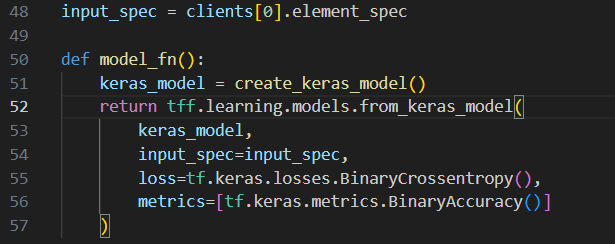
* Dalam Federated Learning, data dibagi per klien.
* Misalnya kita punya 3 klien → dataset dibagi jadi 3.
* Setiap klien punya tf.data.Dataset masing-masing.
* Hasil akhirnya: clients = list of dataset per client.

1. Definisikan model keras



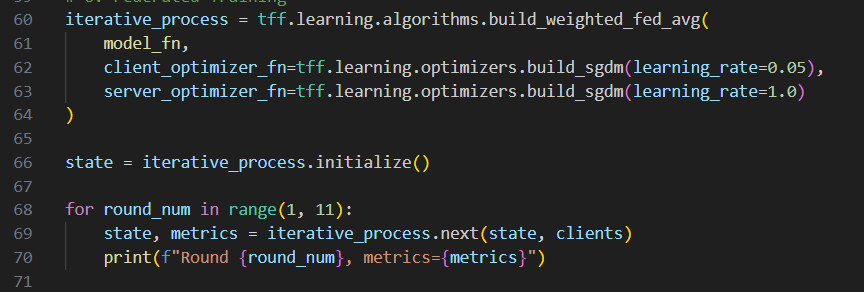
* Model sederhana: input → hidden layer → output biner.
* Input shape (2,) karena kita pakai **2 fitur** (Umur, Pendapatan).
* Hidden layer: 16 neuron dengan aktivasi ReLU.
* Output: sigmoid → cocok untuk klasifikasi biner.

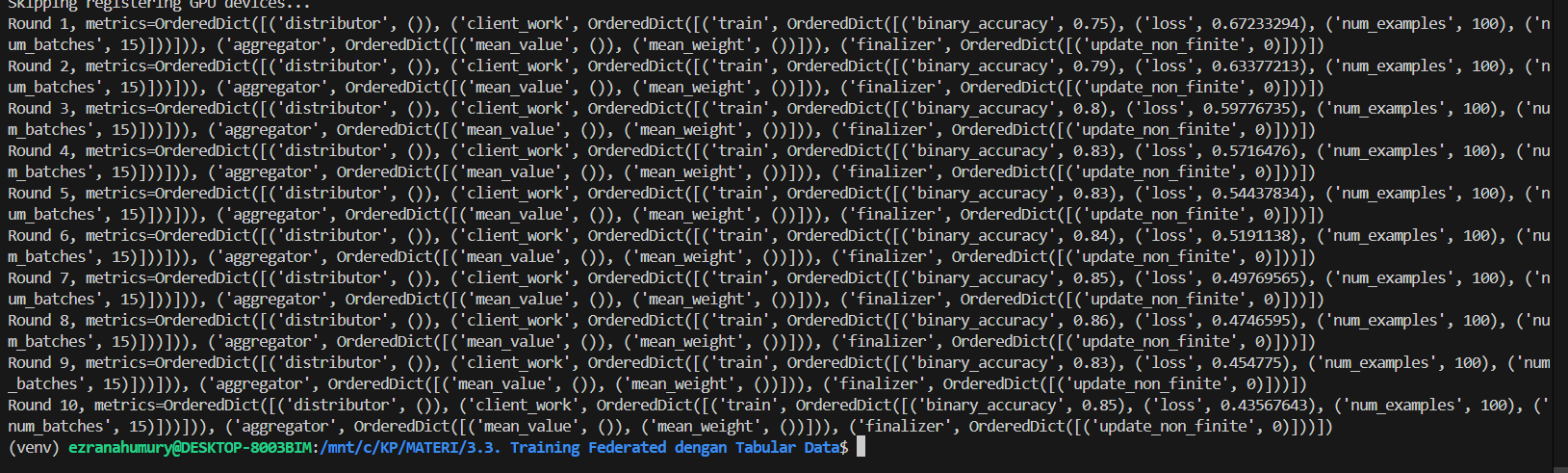
1. Wrap ke TFF



* input\_spec diambil dari dataset klien → memberitahu TFF bentuk data.
* from\_keras\_model membungkus model Keras ke dalam format TFF.
* Kita tentukan **loss** (Binary Crossentropy) dan **metrics** (Binary Accuracy).

1. Training dengan Federated Averaging





**Ringkasan Alur**

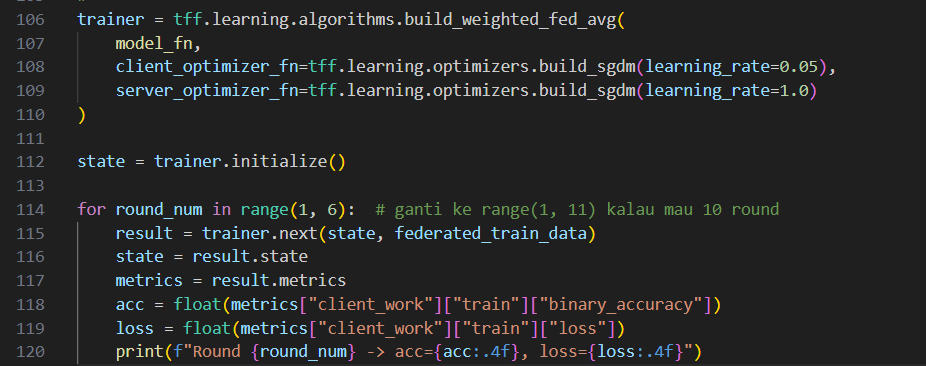
1. **Pandas** 🡪 load CSV, bersihkan, buat label.
2. **TensorFlow Dataset** 🡪 konversi tabular ke tf.data.Dataset.
3. **Federated Dataset** 🡪 split dataset per klien.
4. **Model Keras** 🡪 desain NN sederhana.
5. **Wrapper TFF** 🡪 bungkus model supaya TFF paham input spec.
6. **FedAvg Training** 🡪 jalankan federated learning dengan beberapa round.

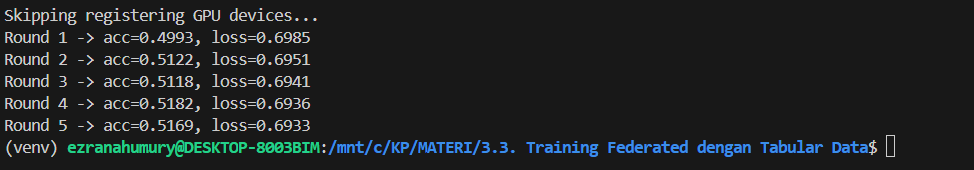
**Task :**

**Ringkasan Dataset**

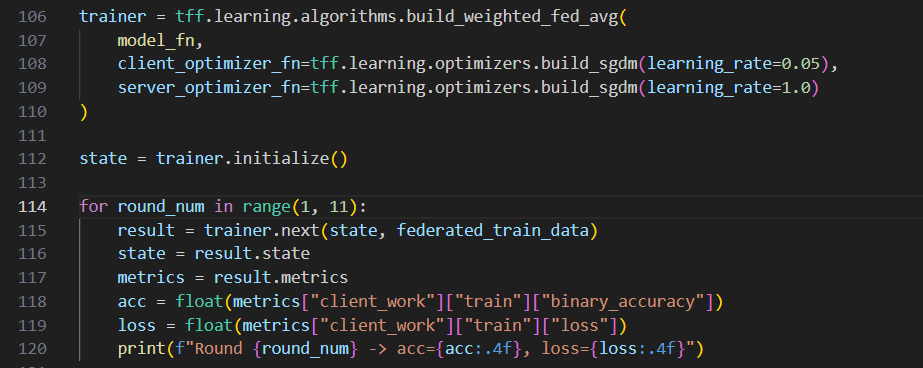
1. **Dinsos**
   * Fitur: jumlah\_tanggungan, penghasilan, kondisi\_rumah
   * Label: layak\_subsidi
2. **Dukcapil**
   * Fitur: umur, status\_pekerjaan, status\_pernikahan
   * Label: layak\_subsidi
3. **Kemenkes**
   * Fitur: riwayat\_penyakit, status\_gizi, tinggi\_cm, berat\_kg
   * Label: layak\_subsidi

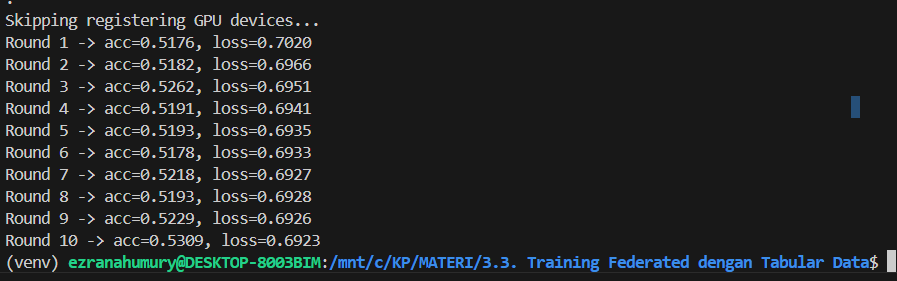
Menggunakan 5 Round :





Menggunakan 10 Round :





* **Simulasi 5–10 round** 🡪 berarti kita latih model federated selama beberapa putaran (round).
* **Logging akurasi** 🡪 setiap round, kita cetak akurasi (dan loss) supaya bisa memantau apakah model makin bagus.