

Topik : 1.4 Federated Computation Primitives

Objective : Memahami `tff.federated_map`, `tff.federated_mean`

Task : Implementasi operasi sederhana FL: sum, mean antar client

Source : https://www.tensorflow.org/federated/tutorials/custom_federated_algorithms_1

Intended Uses

Bayangkan Anda ingin melatih sebuah model AI (kecerdasan artifisial), tetapi datanya tidak terkumpul di satu server pusat. Data tersebut tersebar di ribuan perangkat milik pengguna, seperti ponsel atau laptop. **Federated Learning** adalah teknik yang melatih model secara langsung di perangkat-perangkat tersebut tanpa harus mengirim data pribadi pengguna ke server pusat.

Federated Core (FC) Adalah “Mesin” atau lingkungan pemrograman di dalam TensorFlow Federated (TFF) yang memungkinkan kita menulis logika untuk Scenario seperti ini. FC memberikan kita alat untuk mengekspresikan dua hal penting :

1. Kode Komputasi (Menggunakan Tensorflow) : misalnya, kode untuk melatih model sedikit demi sedikit di ponsel seorang pengguna
2. Kode Komunikasi Terdistribusi : Misalnya, cara mengumpulkan hasil pelatihan dari ribuan ponsel, kemudian menggabungkannya (rata – rata), dan mengirimkan model yang telah diperbarui Kembali ke semua ponsel.

Perbedaan Federated Core (FC) & `tf.contrib.distribute`

	Federated Core	Tf.contrib.distribute
Tujuan	Memberikan control penuh untuk menciptakan algoritma baru yang berjalan di atas data yang memang sengaja terpecah dan tidak boleh di pindahkan	Membuat Pelatihan yang sudah ada menjadi lebih cepat dan efisien
Cara Kerja	Data tetap di perangkat pengguna (di luar "gedung"). FC menyediakan bahasa pemrograman untuk mengatur bagaimana model berkelana ke perangkat, berhitung di sana, dan bagaimana hasilnya dikombinasikan.	Anda punya model dan data yang sudah terkumpul di sebuah pusat data (data center) yang penuh dengan GPU/TPU. Framework ini membagi-bagi data dan komputasi ke across semua prosesor tersebut secara otomatis.

Federated Core ditujukan untuk para peneliti dan engineer yang ingin menciptakan algoritma Federated Learning yang baru. Misalnya, jika anda ingin mencoba cara baru untuk merata – ratakan model, atau pola komunikasi yang berbeda antar perangkat.

FC beroperasi pada level yang setara dengan **pseudocode** (kode semu) dalam makalah penelitian. Anda bisa mendeskripsikan ide ("kirim model ke klien, latih, ambil update, rata-ratakan") tanpa harus pusing dengan detail teknis jaringan seperti protokol socket dan pertukaran pesan point-to-point.

Pengujian TFF

Mencoba menjalankan “Hello World” untuk memastikan bahwa lingkungan sudah diatur dengan benar :

1. Melakukan pengecekan Versi Python dikarenakan TFF hanya mensupport Python Versi 3.9 – 3.11

```
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primit  
03BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python3 --version  
Python 3.12.3
```

2. Melakukan penginstallan python terbaru yang mensupport TFF

```
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primit  
ives$ sudo apt update  
[sudo] password for ezranahumury:  
Hit:1 https://storage.googleapis.com/bazel-apt stable InRelease  
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease  
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Hit:5 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu noble InRelease  
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [ 21,6 kB]
```

```
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primit  
ives$ sudo apt install python3.11 python3.11-venv python3.11-dev -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3.11 is already the newest version (3.11.13-1+noble1).  
python3.11-venv is already the newest version (3.11.13-1+noble1).  
python3.11-dev is already the newest version (3.11.13-1+noble1).  
The following packages were automatically installed and are no longer required  
:  
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-amber  
  libllvml17t64 libxcb-dri2-0  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primit  
ives$
```

3. Membuat dan mengaktifkan environment

```
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python3.11 -m venv venv  
ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ source venv/bin/activate  
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

```
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ pip install --upgrade pip
Requirement already satisfied: pip in ./venv/lib/python3.11/site-packages (24.0)
Collecting pip
  Using cached pip-25.2-py3-none-any.whl.metadata (4.7 kB)
Using cached pip-25.2-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
Successfully installed pip-25.2
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

4. Melakukan pengecekan versi Numpy, SciPy, TFF

```
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python -c "import numpy, scipy, tensorflow_federated as tff; print(numpy.__version__, s
cipy.__version__, tff.__version__)"
2025-08-25 21:03:48.045688: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-25 21:03:48.450336: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN whe
n one has already been registered
2025-08-25 21:03:48.451084: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when
one has already been registered
2025-08-25 21:03:48.453168: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS
when one has already been registered
2025-08-25 21:03:48.712724: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-25 21:03:48.715982: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical ope
rations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-25 21:03:58.988919: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
1.25.2 1.9.3 0.87.0
```

5. Membuat file untuk melakukan pengecekan TFF dengan mencoba “Hello World”

```
1.4 FEDERATED COMPUTA... test.py > ...
> folder venv
test.py 3
1 import collections
2
3 import numpy as np
4 import tensorflow as tf
5 import tensorflow_federated as tff
6
7 @tff.federated_computation
8 def hello_world():
9     return 'Hello, World!'
10
11 print(hello_world())
```

```
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python test.py
2025-08-25 21:07:07.263378: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-25 21:07:07.629236: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN whe
n one has already been registered
2025-08-25 21:07:07.630084: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT whe
n one has already been registered
2025-08-25 21:07:07.632225: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS
when one has already been registered
2025-08-25 21:07:07.881370: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-25 21:07:07.883885: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical op
erations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-25 21:07:17.744160: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
b'Hello, World!'
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

FEDERATED DATA

Salah satu fitur pembeda dalam TFF Adalah kemampuannya untuk mengeskpresikan komputasi berbasis TensorFlow pada data terfederasi secara ringkas. Dalam tutorial ini, kita akan menggunakan istilah Data Terfederasi untuk merujuk pada Kumpulan item data yang dihosting di sekelompok perangkat dalam system terdistribusi. Misalnya, aplikasi yang berjalan pada perangkat seluler dapat mengumpulkan data dan menyimpannya secara lokal, tanpa mengunggahnya ke lokasi terpusat. Atau, sejumlah sensor terdistribusi dapat mengumpulkan dan menyimpan pembacaan suhu di lokacsi masing-masing.

Data terfederasi seperti contoh diatas diperlakukan dalam TFF sebagai entitas utama (First-class citizens), artinya data tersebut dapat muncul sebagai parameter dan hasil fungsi, dan mereka memiliki tipe

Poin penting yang harus dipahami adalah bahwa kita memodelkan seluruh kumpulan item data di semua perangkat (misalnya, seluruh kumpulan pembacaan suhu dari semua sensor dalam sebuah array terdistribusi) sebagai satu nilai terfederasi tunggal.

Contoh cara mendefinisikan tipe float terfederasi yang dihosting oleh sekelompok perangkat klien dalam TFF. Kumpulan pembacaan suhu yang terwujud di sepanjang array sensor terdistribusi dapat dimodelkan sebagai nilai dari tipe terfederasi ini.

```
FL.py 3 x
FL.py > ...
1 import collections
2
3 import numpy as np
4 import tensorflow as tf
5 import tensorflow_federated as tff
6 federated_float_on_clients = tff.FederatedType(np.float32, tff.CLIENTS)
7 print(str(federated_float_on_clients.member))
8 print(str(federated_float_on_clients.placement))
9
```

```
(venv) ezranahumy@DESKTOP-800301M:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ pyth
eady been registered
2025-08-28 13:24:24.910393: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already be
en registered
2025-08-28 13:24:25.385562: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 13:24:25.390733: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-28 13:24:39.807622: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
float32
CLIENTS
(venv) ezranahumy@DESKTOP-800301M:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Jenis federasi di TFF didefinisikan dengan menentukan jenis T dan konstituen anggotanya (item data yang berada pada perangkat individu) dan jenis G yaitu perangkat yang nilai federasi jenis ini di-host (ditambahkan sedikit informasi opsional)

	Jenis T	Jenis G
Code	<code>str(federated_float_on_clients.member)</code>	<code>str(federated_float_on_clients.placement)</code>
Pengertian	Tipe dari data individual yang berada pada setiap perangkat secara terpisah.	Kelompok atau lokasi di mana nilai federasi tersebut "hidup" atau dikumpulkan. Ini menunjukkan cakupan dari nilai tersebut.
Contoh	Float, String	Client, Server

Jenis Federasi dengan konstituen anggota T dan Penempatan G dapat direpresentasikan kompak (digabung) sebagai $\{T\}@G$, seperti di bawah ini

```
print(str(federated_float_on_clients))
```

```
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python FL.py
2025-08-28 13:43:55.040920: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 13:43:55.410700: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already
registered
2025-08-28 13:43:55.411468: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:690] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already
registered
2025-08-28 13:43:55.411462: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already
registered
2025-08-28 13:43:55.680809: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 13:43:55.683437: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-28 13:44:05.433077: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
float32
CLIENTS
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Tipe Federasi du TFF hadir dalam dua rasa :

1. Tipe Dimana konstituen anggota dari nilai federasi mungkin berbeda (ini Adalah default dan paling umum) code nya (all_equal=False)
2. Tipe yang diketahui semuanya sama (harus sama semua) codenya (all_equal=True)

```
FL.py > ...
1 import collections
2
3 import numpy as np
4 import tensorflow as tf
5 import tensorflow_federated as tff
6 federated_float_on_clients = tff.FederatedType(np.float32, tff.CLIENTS)
7 # print(str(federated_float_on_clients.member)) #T
8 # print(str(federated_float_on_clients.placement)) #G
9 # print(str(federated_float_on_clients))
10
11 print(str(federated_float_on_clients.all_equal))
12
```

```
CLIENTS
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python FL.py
2025-08-28 14:01:21.408168: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to re
en registered
2025-08-28 14:01:21.957795: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your m
2025-08-28 14:01:21.962557: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropria
2025-08-28 14:01:33.535782: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could
False
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

$T@G$ vs $\{T\}@G$

$\{T\}@G$	$T@G$
all_equal=False	all_equal=True
multi-set (kumpulan di mana anggota bisa berulang) dari nilai-nilai bertipe T yang berasal dari penempatan G. Nilai-nilai ini boleh berbeda-beda antara satu perangkat dengan perangkat lainnya.	Sebuah nilai bertipe T yang sama persis ada (atau dibroadcast) ke setiap perangkat dalam penempatan G.
Kurung Kurawal { } melambangkan sebuah kumpulan atau tas (bag) yang berisi banyak item.	Tidak ada kurung kurawal menandakan bahwa ini adalah sebuah nilai tunggal , bukan sebuah kumpulan. Nilai tunggal ini diduplikasi dan hadir di semua perangkat di G.

```

1 import collections
2 import numpy as np
3 import tensorflow as tf
4 import tensorflow_federated as tff
5
6
7 federated_float_on_clients_all_equal = tff.FederatedType(np.float32, tff.CLIENTS, all_equal=True)
8 print("All Equal:", federated_float_on_clients_all_equal)

```

```

(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python Federated_language.py
2025-08-28 14:42:11.114043: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 14:42:11.535756: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
2025-08-28 14:42:11.536625: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
2025-08-28 14:42:11.539038: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
2025-08-28 14:42:11.826886: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 14:42:11.830454: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-28 14:42:22.197520: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
All Equal: float32@CLIENTS
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```

Salah satu contoh nilai gabungan dari jenis tersebut yang mungkin muncul dalam scenario praktis Adalah hyperparameter (seperti kecepatan belajar, norma klipng, dll) yang telah disiarkan oleh server ke sekelompok perangkat yang berpartisipasi dalam pelatihan gabungan.

Contoh lain adalah seperangkat parameter untuk model pembelajaran mesin yang telah dilatih sebelumnya di server, yang kemudian disiarkan ke sekelompok perangkat klien, di mana parameter tersebut dapat dipersonalisasi untuk setiap pengguna.

Sebagai contoh, misalkan kita memiliki sepasang float32 parameter a dan b untuk model regresi linier sederhana satu dimensi. Kita dapat membangun tipe (non-federasi) model tersebut untuk digunakan dalam TFF sebagai berikut. Kawat gigi sudut < > di tipe string dicetak adalah notasi TFF kompak untuk bernama atau tidak disebutkan namanya tuple.

```

9
10 simple_regression_model_type = (
11     tff.StructType([('a', np.float32), ('b', np.float32)])
12 )
13
14 print(str(simple_regression_model_type))

```

```

(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python Federated_language.py
2025-08-28 14:52:24.778692: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 14:52:25.256377: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
2025-08-28 14:52:25.256664: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
2025-08-28 14:52:25.260858: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
2025-08-28 14:52:25.560320: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-28 14:52:25.563344: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-28 14:52:36.470357: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
<a=float32,b=float32>
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```


Penting untuk dipahami bahwa `tf.float32` hanyalah notasi singkat yang mewakili skenario paling dasar—sebuah tensor dengan bentuk kosong `shape=[]` (hanya satu angka skalar). Dalam dunia nyata, data federated learning hampir selalu lebih kompleks dari sekadar satu angka; data tersebut bisa berupa vektor bobot model, gambar, atau kumpulan teks. Oleh karena itu, untuk kejelasan dan ketepatan, selalu gunakan `tff.TensorType` secara eksplisit dengan menentukan shape-nya.

Ketika model ini disiarkan ke klien, jenis nilai gabungan yang dapat di hasilkan dapat direpresentasikan seperti di bawah ini :

```
15 simple_regression_model_type = (  
16     tff.StructType([('a', np.float32), ('b', np.float32)])  
17 )  
18 print(str(tff.FederatedType(  
19     simple_regression_model_type,  
20     placement=tff.CLIENTS,  
21     all_equal=True  
22 )))
```

```
(venv) ezranahumy@DESKTOP-8003B2M: /mnt/c/KP/WATERI/1.4 Federated Computation Primitives$ python Federated_language.py  
2025-08-28 15:10:18.170900: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.  
2025-08-28 15:10:18.086274: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered  
2025-08-28 15:10:18.607156: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:689] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered  
2025-08-28 15:10:18.609286: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered  
2025-08-28 15:10:18.834076: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.  
2025-08-28 15:10:18.836378: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.  
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.  
2025-08-28 15:10:23.913090: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: could not find TensorRT  
c=a=float32,b=float32@CLIENTS  
(venv) ezranahumy@DESKTOP-8003B2M: /mnt/c/KP/WATERI/1.4 Federated Computation Primitives$
```

Per simetri dengan *mengapung Federasi* di atas, kita akan merujuk pada jenis seperti *tupel federasi*. Lebih umum, kita akan sering menggunakan *XYZ Federasi* istilah untuk merujuk ke nilai federasi di mana konstituen anggota yang *XYZ* -seperti. Dengan demikian, kita akan berbicara tentang hal-hal seperti *tupel federasi*, *urutan federasi*, *model federasi*, dan sebagainya.

Apa itu "Isomorfisme" antara T dan $T@G$?

Bayangkan Anda memiliki **sebuah file PDF** (T). Itu hanya ada di komputer server Anda.

Sekarang, Anda **mengirim file PDF yang sama persis itu ke 100 laptop karyawan** ($T@tff.CLIENTS$). Sekarang ada 100 salinan file yang identik.

"**Isomorfisme**" artinya kedua keadaan ini pada intinya adalah setara untuk banyak keperluan:

1. **Memiliki 1 file di server** (T)
2. **Memiliki 100 salinan file yang sama di laptop karyawan** ($T@tff.CLIENTS$)

Kedua keadaan merepresentasikan **informasi yang sama persis**. Jika isi file di server berubah, semua salinan di laptop juga harus diperbarui agar tetap sama.

Kegunaan Membuat Jenis T@G?

Membuat jenis T@G itu seperti ** memberikan alamat pengiriman pada sebuah paket**.

- **Isi paketnya** adalah T (contoh: sebuah buku).
- **Alamatnya (G)** menentukan apakah paket itu ada di **gudang pusat (@SERVER)** atau sudah **diantarkan ke setiap toko cabang (@CLIENTS)**.

Placements

TFF dirancang dengan filosofi yang berbeda dari kebanyakan framework. Alih-alih berfokus pada di mana suatu operasi dijalankan, TFF berfokus pada di mana data berada dan bagaimana data itu berubah. **Placement** dalam TFF adalah sifat dari **data itu sendiri**, bukan dari operasinya. Data bisa berada di server (@SERVER) atau di kumpulan klien (@CLIENTS).

Dengan secara eksplisit menandai data sebagai {T}@CLIENTS (data sensitif di perangkat user), TFF dapat secara otomatis mencegah operasi yang salah. Framework ini dirancang untuk **tidak pernah mengizinkan** data mentah dari klien dibawa keluar dari konteks yang aman. Semua operasi yang Anda tulis akan berlaku untuk **seluruh grup klien secara kolektif**, bukan untuk individu tertentu.

TFF menyediakan dua "alamat" dasar yang sudah jadi untuk memudahkan pemrograman:

1. tff.SERVER → **"Pusat Komando"**. Biasanya hanya ada satu. Di sinilah model global disimpan dan proses koordinasi dilakukan.
2. tff.CLIENTS → **"Lapangan"** atau **"Perangkat User"**. Ini bisa mewakili ribuan ponsel, sensor, atau database yang terpisah-pisah.

TFF menggunakan model **client-server** yang intuitif:

- **Server** bertindak sebagai koordinator pusat.
- **Klien** adalah perangkat-perangkat individual yang memiliki data.

Fleksibilitas yang Penting:

Yang menarik, TFF **tidak mendikte secara kaku** apa sebenarnya CLIENTS dan SERVER itu. Ini memberimu kebebasan:

- tff.CLIENTS bisa berarti 1.000 **ponsel**, atau 50 **database** yang tersebar, atau bahkan 3 **komputer** dalam lab.
- tff.SERVER bisa berupa sebuah **aplikasi cloud**, sebuah **komputer tunggal**, atau bahkan sebuah **proses dalam simulator**.

Federated Computations

TFF dirancang sebagai lingkungan pemrograman fungsional bertipe kuat yang mendukung pengembangan modular. Unit dasar dari komposisi di TFF merupakan Federated Computations yang merupakan bagian dari logika yang dapat menerima nilai – nilai federasi sebagai masukan dan Kembali nilai – nilai federasi sebagai sebuah output.

Inilah cara menentukan perhitungan yang menghitung rata – rata suhu yang di laporkan oleh larik sensor :

```
1 import collections
2
3 import numpy as np
4 import tensorflow as tf
5 import tensorflow_federated as tff
6
7 @tff.federated_computation(tff.FederatedType(np.float32, tff.CLIENTS))
8 def get_average_temperature(sensor_readings):
9     return tff.federated_mean(sensor_readings)
```

Kode yang dihasilkan oleh `tff.federated_computation` bukanlah tensorflow, juga bukan python. Melainkan spesifikasi system terdistribusi dalam internal platform-independent glue language.

Pertama, mari kita bermain dengan definisi sedikit. Perhitungan TFF umumnya dimodelkan sebagai fungsi - dengan atau tanpa parameter, tetapi dengan tanda tangan tipe yang terdefinisi dengan baik. Anda dapat mencetak tanda tangan jenis perhitungan dengan query yang `type_signature` properti, seperti yang ditunjukkan di bawah ini.

```
7 @tff.federated_computation(tff.FederatedType(np.float32, tff.CLIENTS))
8 def get_average_temperature(sensor_readings):
9     return tff.federated_mean(sensor_readings)
10
11 print(str(get_average_temperature.type_signature))
```

```
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python Placement.py
2025-08-28 17:41:57.428935: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used
2025-08-28 17:41:29.764800: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempt
registered
2025-08-28 17:41:29.765593: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempt
registered
2025-08-28 17:41:29.767692: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempt
en registered
2025-08-28 17:41:30.008828: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used
2025-08-28 17:41:30.011508: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-28 17:41:40.905184: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
({float32}@CLIENTS -> float32@SERVER)
(venv) ezranahumury@DESKTOP-80038IM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Tanda tangan tipe memberi tahu kita bahwa komputasi menerima kumpulan pembacaan sensor yang berbeda pada perangkat klien, dan mengembalikan rata-rata tunggal di server.

Dalam banyak scenario praktis, perhitungan yang mewakili tugas singkat atas cenderung menerima masukan dan melaporkan keluaran di server- hal ini mencerminkan gagasan bahwa perhitungan mungkin dipicu oleh query yang berasal dan berakhir di server.

Secara umum, Anda tidak boleh memandang komputasi federasi sebagai sesuatu yang berjalan di server atau dieksekusi oleh server. Server hanyalah salah satu jenis peserta dalam komputasi federasi. Dalam memikirkan mekanisme komputasi semacam ini, sebaiknya selalu berangkat dari perspektif jaringan global secara keseluruhan, bukan perspektif koordinator terpusat tunggal.

Secara umum, jenis tanda tangan fungsional kompak direpresentasikan sebagai (T -> U) untuk jenis T dan U input dan output, masing-masing. Jenis parameter formal (seperti `sensor_readings` dalam hal ini) ditetapkan sebagai argumen untuk dekorator. Anda tidak perlu menentukan jenis hasil - itu ditentukan secara otomatis.

Executing Federated Computations

Untuk mendukung pengembangan dan debugging, TFF memungkinkan kita untuk secara langsung menjalankan komputasi yang didefinisikan dengan cara ini sebagai fungsi python, seperti yang ditunjukkan di bawah ini. Dimana perhitungan mengharapkan nilai tipe Federasi dengan `all_equal` bit set ke `False`, Anda dapat memberikannya sebagai daftar biasa dalam Python. Sedangkan untuk tipe federasi dengan bit `all_equal` disetel ke `True`, Anda dapat langsung memberikan konstituen anggota tunggalnya. Inilah cara hasilnya dilaporkan kembali kepada Anda.

```
6
7 @tff.federated_computation(tff.FederatedType(np.float32, tff.CLIENTS))
8 def get_average_temperature(sensor_readings):
9     return tff.federated_mean(sensor_readings)
10
11 print(str(get_average_temperature([68.5, 70.3, 69.8])))
```

```
Your kernel may have been built without NUMA support.
2025-08-28 18:00:32.966980: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2211] Cannot dlopen s
ou would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to down
Skipping registering GPU devices...
69.53333
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Ketika interpreter python bertemu fungsi dihiasi dengan `tff.federated_computation`, interpreter tersebut akan menelusuri pernyataan – pernyataan dalam tubuh fungsi tersebut sekali (pada saat definisi) dan kemudian membangun representasi serial dari logika komputasi tersebut untuk digunakan dimasa depan (baik untuk eksekusi, maupun untuk dimasukkan sebagai sub-komponen ke dalam komputasi lain).

Anda dapat memverifikasi ini dengan menambahkan pernyataan cetak, sebagai berikut:

```
13 @tff.federated_computation(tff.FederatedType(np.float32, tff.CLIENTS))
14 def get_average_temperature(sensor_readings):
15     print('Getting traced, the argument is "{}".format(
16         type(sensor_readings).__name__
17     ))
18
19     return tff.federated_mean(sensor_readings)
20
21 print(str(get_average_temperature()))
```

```
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python Average.py
2025-08-28 18:53:44.308664: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine,
2025-08-28 18:53:44.941104: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cu
registered
2025-08-28 18:53:44.941848: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuF
egistered
2025-08-28 18:53:44.943902: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register c
en registered
2025-08-28 18:53:45.200925: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine,
2025-08-28 18:53:45.203971: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimiz
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compi
2025-08-28 18:53:56.464613: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find
Getting traced, the argument is "Value".
```

Perhitungan TFF mengekspresikan perilaku global system terdistribusi, berbeda dengan program python yang mengekspresikan perilaku local masing – masing peserta. Kita dapat melihat bahwa dalam contoh sederhana di atas, dengan operator khusus `tff.federated_mean` yang menerima data pada perangkat klien, tapi deposit hasil pada server.

Operator `tff.federated_mean` tidak dapat dengan mudah dimodelkan sebagai operator biasa di Python, karena tidak mengeksekusi secara lokal - seperti disebutkan sebelumnya, itu merupakan sistem terdistribusi yang koordinat perilaku beberapa peserta sistem. Kita akan mengacu ke operator seperti *operator federasi*, untuk membedakan mereka dari biasa operator (lokal) di Python.

Composing federated computations

Perhitungan `get_average_computation` dapat di panggil dalam tubuh fungsi python lain yang dihiasi dengan `tff.federated_computation`. Hal tersebut akan menyebabkan itu tertanam dalam tubuh induk, banyak cara yang sama `tff.federated_mean` tertanam dalam tubuh sendiri sebelumnya.

Pembatasan penting untuk menyadari adalah bahwa tubuh fungsi Python dihiasi dengan `tff.federated_computation` harus terdiri hanya dari operator federasi, yaitu, mereka tidak dapat secara langsung mengandung operasi TensorFlow. Misalnya, Anda tidak bisa langsung menggunakan `tf.nest` antarmuka untuk menambah sepasang nilai federasi. Kode TensorFlow harus terbatas pada blok kode dihiasi dengan `tff.tf_computation` dibahas dalam bagian berikut. Hanya ketika dibungkus dengan cara ini dapat kode TensorFlow dibungkus dipanggil dalam tubuh seorang `tff.federated_computation`.

Alasan untuk pemisahan ini adalah teknis (sulit untuk mengelabui operator seperti `tf.add` bekerja dengan non-tensor) serta arsitektur. Bahasa perhitungan federasi (yaitu, logika yang dibangun dari tubuh serial fungsi Python dihiasi dengan `tff.federated_computation`) dirancang untuk melayani sebagai bahasa lem platform-independen. Bahasa lem ini saat ini digunakan untuk membangun sistem terdistribusi dari bagian tertanam kode TensorFlow (terbatas `tff.tf_computation` blok). Dalam kepenuhan waktu, kami mengantisipasi kebutuhan untuk bagian menanamkan dari lainnya, logika non-TensorFlow, seperti query database relasional yang mungkin mewakili pipa input, semua terhubung bersama-sama menggunakan bahasa lem sama (`tff.federated_computation` blok).

Logika TensorFlow

TFF dirancang untuk digunakan dengan tensorflow. Dengan demikian, Sebagian besar kode yang akan ditulis dalam TFF kemungkinan Adalah TensorFlow biasa (yaitu, yang dijalankan secara local). Untuk menggunakan kode tersebut dengan TFF, seperti disebutkan di atas, hanya perlu dihiasi dengan `tff.tf_computation`.

Sebagai contoh, berikut Adalah cara untuk menerapkan fungsi yang mengambil nomor dan menambahkan 0.5 :

```
Sum.py 3 x
Sum.py > add_half
1 import collections
2
3 import numpy as np
4 import tensorflow as tf
5 import tensorflow_federated as tff
6
7 @tff.tensorflow.computation(np.float32)
8 def add_half(x):
9     return tf.add(x, 0.5)
10 print(add_half(10.0))
11
```

```
2023-08-28 15:40:47.784974: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2211] cannot open some GPU devices.
You would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and
install GPUs on Windows, Mac, and Linux.
Skipping registering GPU devices...
10.5
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Untuk menyematkan blok penyusun yang dapat digunakan kembali yang diimplementasikan menggunakan kode TensorFlow di badan komputasi gabungan, blok penyusun tersebut harus memenuhi properti tertentu - seperti dilacak dan diserialisasikan pada waktu definisi, memiliki tanda tangan tipe, dll. Ini umumnya memerlukan beberapa bentuk dekorator.

Secara khusus, ia memiliki tanda tangan tipe TFF.

```
7 @tff.tensorflow.computation(np.float32)
8 def add_half(x):
9     return tf.add(x, 0.5)
10 # print(add_half(10.0))
11 print(str(add_half.type_signature))
12
```

```
ing. Could not find TensorRT
(float32 -> float32)
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Perhatikan tanda tangan jenis ini tidak memiliki penempatan. Komputasi TensorFlow tidak dapat menggunakan atau mengembalikan tipe federasi.

Anda sekarang dapat juga menggunakan `add_half` sebagai sebuah blok bangunan dalam perhitungan lainnya. Sebagai contoh, berikut adalah cara Anda dapat menggunakan `tff.federated_map` operator untuk menerapkan `add_half` pointwise ke semua konstituen anggota pelampung Federasi pada perangkat klien.

```

7  @tff.tensorflow.computation(np.float32)
8  def add_half(x):
9      return tf.add(x, 0.5)
10
11  # print(add_half(10.0))
12
13
14  # print(str(add_half.type_signatu
15
16  @tff.federated_computation(tff.FederatedType(np.float32,tff.CLIENTS))
17  def add_half_on_clients(x):
18      return tff.federated_map(add_half, x)
19
20  print(str(add_half_on_clients.type_signature))

```

```

2025-08-28 20:25:23.890756: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
({float32}@CLIENTS -> {float32}@CLIENTS)
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```

Menjalankan Komputasi Tensorflow

Pelaksanaan perhitungan didefinisikan dengan `tff.tf_computation` mengikuti aturan yang sama seperti yang dijelaskan untuk `tff._federated_computation`. Mereka dapat dipanggil sebagai callable biasa dengan python, sebagai berikut :

```

7  @tff.tensorflow.computation(np.float32)
8  def add_half(x):
9      return tf.add(x, 0.5)
10
11  # print(add_half(10.0))
12
13
14  # print(str(add_half.type_signatu
15
16  @tff.federated_computation(tff.FederatedType(np.float32,tff.CLIENTS))
17  def add_half_on_clients(x):
18      return tff.federated_map(add_half, x)
19
20  # print(str(add_half_on_clients.type_signature))
21  result = add_half_on_clients([1.0, 3.0, 2.0])
22  print(result)

```

```

Skipping registering GPU devices...

```

```

[1.5, 3.5, 2.5]

```

```

(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```

perhitungan `add_half_on_clients` dengan cara ini mensimulasikan proses didistribusikan. Data dikonsumsi pada klien, dan dikembalikan pada klien. Memang, perhitungan ini membuat setiap klien melakukan tindakan lokal. Tidak ada `tffSERVER` disebutkan secara eksplisit dalam sistem ini (bahkan jika dalam prakteknya, mendalangi pemrosesan tersebut mungkin melibatkan satu). Pikirkan perhitungan didefinisikan dengan cara ini sebagai konseptual analog dengan Map panggung MapReduce .

Perbedaan satu-satunya antara metode Python yang diberi dekorasi dengan `federated_language.federated_computation` dan yang diberi dekorasi dengan `tff.tensorflow.computation` adalah bahwa yang terakhir diserialisasikan sebagai grafik

TensorFlow (sedangkan yang pertama tidak diperbolehkan mengandung kode TensorFlow yang tertanam langsung di dalamnya).

```
7 try :
8     constant_10 = tf.constant(10)
9
10    @tff.tensorflow.computation(np.float32)
11    def add_ten(x):
12        return x + constant_10
13
14 except Exception as err :
15     print(err)
```

```
y: Attempting to capture an EagerTensor without building a function.
```

```
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Di atas gagal karena `constant_10` telah dibangun di luar grafik yang `tff.tf_computation` konstruksi internal di tubuh `add_ten` selama proses serialisasi.

Di sisi lain, memanggil fungsi python yang memodifikasi grafik saat saat dipanggil di dalam `tff.tf_computation` baik-baik saja:

```
17 def get_constant_10():
18     return tf.constant(10.0)
19 @tff.tensorflow.computation(np.float32)
20 def add_ten(x):
21     return x + get_constant_10()
22
23 print(add_ten(5.0))
```

Working with `tf.data.Datasets`

Sebuah fitur unik dari `tff.tf_computation` adalah bahwa mereka memungkinkan Anda untuk bekerja dengan `tf.data.Dataset` s didefinisikan secara abstrak sebagai parameter formal dengan kode Anda. Parameter yang akan diwakili dalam TensorFlow sebagai data set harus dideklarasikan menggunakan `tff.SequenceType` konstruktor.

Sebagai contoh, jenis spesifikasi `tff.SequenceType(tf.float32)` mendefinisikan urutan abstrak elemen mengapung di TFF. Urutan dapat berisi tensor, atau struktur bersarang yang kompleks (kita akan melihat contohnya nanti). The ringkas representasi dari urutan T-typed item adalah `T*`.

```
7 float32_sequence = tff.SequenceType(np.float32)
8 print(str(float32_sequence))
```

```
float32*
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```


Misalkan dalam contoh sensor suhu kita, setiap sensor tidak hanya menyimpan satu pembacaan suhu, tetapi beberapa pembacaan. Berikut adalah cara Anda dapat mendefinisikan perhitungan TFF di TensorFlow yang menghitung rata-rata suhu dalam satu set data lokal menggunakan operator `tf.data.Dataset.reduce`.

```
10 @tff.tensorflow.computation(tff.SequenceType(np.float32))
11 def get_local_temperature_average(local_temperatures):
12     sum_and_count = (
13         local_temperatures.reduce((0.0, 0), lambda x, y: (x[0] + y, x[1] + 1))
14     )
15     return sum_and_count[0] / tf.cast(sum_and_count[1], np.float32)
16
17 print(str(get_local_temperature_average.type_signature))
```

```
Skipping registering GPU devices...
(float32* -> float32)
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$
```

Dalam tubuh metode yang diberi dekorasi dengan `tff.tf_computation`, parameter formal dari tipe urutan TFF diwakili sebagai objek yang berperilaku seperti `tf.data.Dataset`, yaitu mendukung properti dan metode yang sama (saat ini mereka tidak diimplementasikan sebagai subkelas dari tipe tersebut - hal ini mungkin berubah seiring dengan perkembangan dukungan untuk kumpulan data di TensorFlow).

Anda dapat dengan mudah memverifikasi hal ini sebagai berikut :

```
19 @tff.tensorflow.computation(tff.SequenceType(np.int32))
20 def foo(x) :
21     return x.reduce(np.int32(0), lambda x, y: x + y)
22 print(foo([1,2,3]))
```

```
Skipping registering GPU devices...
6
\\(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ \\
```

Berbeda dengan `tf.data.Dataset` biasa, objek-objek serupa dataset ini hanyalah placeholder. Mereka tidak mengandung elemen apa pun, karena mewakili parameter bertipe urutan abstrak yang akan diikat ke data konkret saat digunakan dalam konteks konkret. Dukungan untuk dataset placeholder yang didefinisikan secara abstrak masih terbatas pada tahap ini, dan pada awal pengembangan TFF, Anda mungkin menemui beberapa batasan, tetapi kita tidak perlu khawatir tentang hal itu dalam tutorial ini (silakan merujuk ke halaman dokumentasi untuk detailnya).

Saat menjalankan perhitungan secara lokal yang menerima urutan dalam mode simulasi, seperti dalam tutorial ini, Anda dapat memasukkan urutan sebagai daftar Python, seperti di bawah ini (serta dengan cara lain, misalnya sebagai `tf.data.Dataset` dalam mode eager, tetapi untuk saat ini, kita akan menjaga kesederhanaannya).

```
27 @tff.tensorflow.computation(tff.SequenceType(np.float32))
28 def get_local_temperature_average(local_temperatures):
29     sum_and_count = (
30         local_temperatures.reduce((0.0, 0), lambda x, y: (x[0] + y, x[1] + 1))
31     )
32     return sum_and_count[0] / tf.cast(sum_and_count[1], np.float32)
33
34 print(str(get_local_temperature_average([68.5, 70.3, 69.8])))
```

```

69.53333
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ S
0 0 3

```

Seperti semua jenis TFF lainnya, urutan seperti yang didefinisikan di atas dapat menggunakan konstruktor `tff.StructType` untuk mendefinisikan struktur bersarang. Misalnya, berikut adalah cara mendeklarasikan perhitungan yang menerima urutan pasangan A, B, dan mengembalikan jumlah hasil perkaliannya. Kami menyertakan pernyataan pelacakan dalam tubuh perhitungan agar Anda dapat melihat bagaimana tanda tangan jenis TFF diterjemahkan menjadi `output_types` dan `output_shapes` dataset.

```

36
37 @tff.tensorflow.computation(tff.SequenceType(collections.OrderedDict([('A', np.int32), ('B', np.int32)])))
38 def foo(ds):
39     print('element_structure = {}'.format(ds.element_spec))
40     return ds.reduce(np.int32(0), lambda total, x:total + x['A'] * x['B'])
41
42 print(str(foo.type_signature))
43
44 hasil = foo([['A':2, 'B':3], ['A':4, 'B':5]])
45 print(hasil)

```

```

(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$ python Dataset_S.py
2025-08-29 12:20:05.053754: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-29 12:20:05.505936: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attempting
registered
2025-08-29 12:20:05.506755: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempting
egistered
2025-08-29 12:20:05.508733: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempt
en registered
2025-08-29 12:20:05.789888: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2025-08-29 12:20:05.792791: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available C
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-08-29 12:20:17.778250: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
element_structure = OrderedDict([('A', TensorSpec(shape=(), dtype=tf.int32, name=None)), ('B', TensorSpec(shape=(), dtype=tf.int32, nam
2025-08-29 12:21:11.682283: I tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:880] could not open file to read NUMA m
Your kernel may have been built without NUMA support.
2025-08-29 12:21:11.684329: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2211] Cannot dlopen some GPU libraries. Please make sure
ou would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required library
Skipping registering GPU devices...
(<A=int32,B=int32>* -> int32)
2025-08-29 12:21:11.724363: I tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:880] could not open file to read NUMA n
Your kernel may have been built without NUMA support.
2025-08-29 12:21:11.724431: I tensorflow/core/grappler/devices.cc:66] Number of eligible GPUs (core count >= 8, compute capability >= 0
2025-08-29 12:21:11.724797: I tensorflow/core/grappler/clusters/single_machine.cc:361] Starting new session
2025-08-29 12:21:11.726388: I tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:880] could not open file to read NUMA n
Your kernel may have been built without NUMA support.
2025-08-29 12:21:11.726448: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2211] Cannot dlopen some GPU libraries. Please make sure
ou would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required library
Skipping registering GPU devices...
26
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```

PENGGABUNGAN SEMUANYA

Menggunakan komputasi tensorflow dalam pengaturan federasi. Misalkan kita memiliki sekelompok sensor yang masing – masing memiliki urutan pembacaan suhu local. Kita dapat menghitung rata- rata suhu global dengan mengumpulkan rata – rata local dari sensor – sensor tersebut sebagai berikut :

```

gabung.py / get_global_temperature_average
1  import collections
2
3  import numpy as np
4  import tensorflow as tf
5  import tensorflow_federated as tff
6
7
8  @tff.tensorflow_computation(tff.SequenceType(np.float32))
9  def get_local_temperature_average(local_temperatures):
10     sum_and_count = local_temperatures.reduce(
11         (0.0, 0),
12         lambda x, y: (x[0] + y, x[1] + 1)
13     )
14     return sum_and_count[0] / tf.cast(sum_and_count[1], np.float32)
15
16 @tff.federated_computation(
17     tff.FederatedType(tff.SequenceType(np.float32), tff.CLIENTS))
18 def get_global_temperature_average(sensor_readings):
19     return tff.federated_mean(
20         tff.federated_map(get_local_temperature_average, sensor_readings)
21     )

```

Ini bukan rata-rata sederhana dari semua pembacaan suhu lokal dari semua klien, karena hal itu akan memerlukan penimbangan kontribusi dari klien yang berbeda berdasarkan jumlah pembacaan yang mereka simpan secara lokal. Kami menyisakan hal ini sebagai latihan bagi pembaca untuk memperbarui kode di atas; operator `tff.federated_mean` menerima bobot sebagai argumen kedua opsional (diharapkan berupa federated float).

Input untuk `get_global_temperature_average` kini menjadi urutan float federated. Urutan federated adalah cara umum untuk mewakili data di perangkat dalam pembelajaran federated, dengan elemen urutan biasanya mewakili batch data.

```

23  print(str(get_global_temperature_average.type_signature))

```

```

Skipping registering GPU devices...
({float32*}@CLIENTS -> float32@SERVER)
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```

Menjalankan perhitungan secara lokal pada sampel data di Python. Perhatikan bahwa cara kita memberikan input sekarang berupa daftar dari daftar. Daftar luar mengiterasi perangkat dalam grup yang diwakili oleh `tff.CLIENTS`, sedangkan daftar dalam mengiterasi elemen dalam urutan lokal masing-masing perangkat.

```

24
25  hasil = get_global_temperature_average([[68.0, 70.0], [71.0], [68.0, 72.0, 70.0]])
26  print(hasil)

```

```

Skipping registering GPU devices...
70.0
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/1.4 Federated Computation Primitives$

```