

Ringkasan Chapter 1

Nama Buku: Tensorflow in Actions

Team: Wilhelmina Arlene, Luthfiah Maulidya, Fadhilah Dwi Istiani

Chapter 3 — *Keras and Data Retrieval in TensorFlow 2*

Tujuan Utama Chapter 2

Chapter 2 bertujuan membangun **fondasi teknis TensorFlow 2**. Jika Chapter 1 menjawab "*mengapa TensorFlow*", maka Chapter 2 menjawab:

"Bagaimana TensorFlow bekerja di balik layar?"

Fokus utamanya:

- perbedaan TensorFlow 1 vs TensorFlow 2
- konsep eksekusi (eager execution)
- struktur data inti TensorFlow
- operasi matematis yang menjadi dasar neural network

2.1 First Steps with TensorFlow 2

Evolusi dari TensorFlow 1 ke TensorFlow 2

Masalah utama TensorFlow 1:

- berbasis **static computational graph**
- harus:
 1. mendefinisikan graph
 2. menjalankan graph di dalam Session
- debugging sulit
- kode terasa tidak "Pythonic"

Contoh konsep lama:

- graph dibuat dulu
- data baru dijalankan belakangan

Solusi TensorFlow 2: Eager Execution

TensorFlow 2 secara default menggunakan **eager execution**, artinya:

- operasi langsung dieksekusi
- mirip NumPy
- hasil langsung terlihat
- debugging jauh lebih mudah

Manfaat utama:

- kode lebih intuitif

- lebih cepat belajar
- cocok untuk eksperimen & riset

⚠ Namun:

- TensorFlow tetap bisa membuat **graph** untuk optimasi performa (dibahas nanti dengan `@tf.function`)

TensorFlow di Balik Layar

Walaupun terlihat seperti Python biasa:

- TensorFlow tetap membangun **data-flow graph**
- graph inilah yang:
 - dioptimasi
 - dipindahkan ke GPU/TPU
 - dieksekusi secara efisien

Jadi:

- **developer experience = Python**
- **execution engine = graph-based**

2.2 TensorFlow Building Blocks (Fondasi Utama)

Chapter ini menekankan bahwa TensorFlow dibangun dari **tiga komponen inti**:

2.2.1 tf.Tensor

Apa itu Tensor?

Tensor adalah:

- array n-dimensi
- generalisasi dari:
 - scalar (0D)
 - vector (1D)
 - matrix (2D)
 - tensor (3D+)

Contoh:

- gambar → tensor 3D (height × width × channel)
- batch gambar → tensor 4D

Karakteristik Tensor:

- **immutable** (tidak bisa diubah langsung)
- punya:
 - shape
 - dtype
 - device (CPU/GPU)

TensorFlow tensor mirip NumPy array, tapi:

- bisa dijalankan di GPU/TPU

- bisa masuk ke computational graph

2.2.2 tf.Variable

Perbedaan Tensor vs Variable

Tensor	Variable
Immutable	Mutable
Biasanya input	Biasanya parameter model
Tidak bisa di-update	Bisa di-update

tf.Variable digunakan untuk:

- weight
- bias
- parameter yang dipelajari model

Contoh konsep:

- gradient descent mengubah nilai variable
- tapi tidak mengubah tensor input

2.2.3 tf.Operation

Operation adalah:

- fungsi yang mengubah tensor
- contoh:
 - penjumlahan
 - perkalian
 - matmul
 - convolution

TensorFlow program pada dasarnya adalah:

rangkaian operasi yang mengalirkan tensor

2.3 Neural Network–Related Computations

Bagian ini menjelaskan **operasi matematis utama** yang membentuk neural network.

2.3.1 Matrix Multiplication

Kenapa penting?

- hampir semua neural network = rangkaian matrix multiplication
- fully connected layer = matmul + bias

TensorFlow:

- sangat dioptimasi untuk operasi ini
- memanfaatkan GPU/TPU

Konsep penting:

- shape harus kompatibel
- error shape adalah error paling umum di TensorFlow

2.3.2 Convolution Operation

Digunakan pada:

- CNN
- computer vision

Konsep inti:

- kernel/filter digeser di atas input
- menghasilkan feature map
- mengekstrak pola lokal (edge, texture)

TensorFlow menyediakan:

- fungsi convolution yang sudah dioptimasi
- mendukung stride, padding, dll

2.3.3 Pooling Operation

Tujuan pooling:

- mengurangi dimensi data
- mempertahankan fitur penting
- mengurangi overfitting

Jenis umum:

- max pooling
- average pooling

Pooling:

- tidak memiliki parameter yang dilatih
- murni operasi matematis

Ringkasan Alur Neural Network Menurut Chapter 2

1. Data masuk sebagai tf.Tensor
2. Parameter model disimpan sebagai tf.Variable
3. Operasi (matmul, conv, pooling) mengubah tensor
4. Graph dieksekusi secara eager
5. TensorFlow mengoptimasi eksekusi di hardware

Filosofi TensorFlow 2

Penulis menekankan filosofi berikut:

- **mudah digunakan seperti Python**
- **kuat seperti framework graph-based**
- **fleksibel dari riset sampai produksi**

TensorFlow 2 mencoba menyeimbangkan:

- kemudahan eksperimen

- performa tinggi
- skalabilitas

Inti Chapter 2 (Key Takeaways)

- TensorFlow 2 menggunakan **eager execution**
- Tetap berbasis **computational graph** di balik layar
- Tiga fondasi utama:
 - tf.Tensor
 - tf.Variable
 - tf.Operation
- Neural network dibangun dari operasi matematika sederhana
- Pemahaman chapter ini **krusial** sebelum masuk ke Keras & model kompleks