

Ringkasan Buku 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 3

Chapter 3 adalah **jembatan dari fondasi ke praktik nyata**.

Setelah kamu memahami:

- apa itu *TensorFlow* (Chapter 1)
- bagaimana *TensorFlow* bekerja (Chapter 2)

maka Chapter 3 menjawab pertanyaan inti:

“Bagaimana cara membangun model ML sungguhan dan memberi mereka data?”

Fokus utama:

1. **Keras sebagai high-level API**
2. **Berbagai cara mengambil dan mengalirkan data ke model**

BAGIAN A — KERAS: MEMBANGUN MODEL DENGAN MUDAH

3.1 Keras Model-Building APIs

Apa itu Keras?

Keras adalah:

- **high-level API** di dalam *TensorFlow*
- dirancang agar:
 - kode lebih ringkas
 - lebih mudah dibaca
 - lebih aman dari kesalahan umum

Keras menyembunyikan kompleksitas:

- tf.Variable
- graph construction
- low-level ops

tanpa menghilangkan fleksibilitas.

Dataset yang Digunakan (Konteks Praktik)

Menjelaskan dataset contoh (klasifikasi), untuk menunjukkan:

- bagaimana data masuk ke model
- bagaimana API Keras digunakan secara nyata

Tujuannya bukan akurasi tertinggi, tapi **alur kerja yang benar**.

3.1.1 Sequential API

Konsep Dasar

Sequential API:

- model dibangun **layer demi layer**
- alur data **linear**
- cocok untuk:
 - model sederhana
 - prototyping cepat

Visualisasi mental:

Input → Layer 1 → Layer 2 → Output

Kelebihan

- sangat mudah digunakan
- sedikit kode
- cocok untuk pemula

Keterbatasan

✗ Tidak bisa:

- multiple input
- multiple output
- skip connection
- arsitektur kompleks

👉 **Pesan penting:**

Sequential API bagus, tapi **tidak universal**.

3.1.2 Functional API

Kenapa Functional API Penting?

Sebagian besar model modern:

- **tidak linear**
- memiliki:
 - cabang
 - penggabungan
 - shared layers

Functional API:

- memperlakukan layer sebagai **fungsi**
- memungkinkan arsitektur kompleks

Contoh mental:

Input → Layer A → Layer B

 ↓ Layer C ↑

Keunggulan

- fleksibel
- scalable
- cocok untuk:
 - CNN kompleks
 - encoder-decoder
 - multi-input models

Kekurangan

- sedikit lebih kompleks dari Sequential
- butuh pemahaman alur data

3.1.3 Subclassing API

Apa itu Subclassing?

Subclassing API:

- kamu membuat class Python sendiri
- mewarisi tf.keras.Model
- mendefinisikan:
 - `__init__`
 - `call()`

Kapan Digunakan?

Digunakan jika:

- butuh kontrol penuh
- arsitektur sangat custom
- logika forward pass dinamis

Trade-off

Fleksibel maksimal

Sulit:

- dibaca
- divisualisasikan
- di-debug oleh orang lain

Best practice:

- gunakan **Sequential** → **Functional** → **Subclassing**
- naik tingkat **hanya jika perlu**

BAGIAN B — DATA RETRIEVAL: MEMBERI MAKAN MODEL

3.2 Retrieving Data for TensorFlow/Keras Models

Menekankan satu prinsip besar:

Model secanggih apa pun tidak berguna tanpa pipeline data yang baik

3.2.1 tf.data API

Fungsi Utama

tf.data digunakan untuk:

- membuat pipeline data yang:
 - scalable
 - efisien
 - bisa di-stream

Pipeline umum:

1. load data
2. map (preprocessing)
3. shuffle
4. batch
5. prefetch

Keunggulan

- sangat cepat
- bisa paralel
- cocok untuk data besar

Kapan Dipakai?

- dataset besar
- training serius
- production-ready pipeline

3.2.2 Keras DataGenerators

Apa itu DataGenerator?

Keras menyediakan generator siap pakai untuk:

- image
- time series

Contoh use case:

- membaca gambar dari folder
- melakukan augmentasi otomatis

Kelebihan

- sangat praktis
- sedikit kode
- ideal untuk computer vision

Kekurangan

- kurang fleksibel dibanding tf.data
- tidak selalu optimal untuk pipeline kompleks

3.2.3 tensorflow-datasets (TFDS)

Apa itu TFDS?

TFDS menyediakan:

- dataset populer
- siap pakai
- satu baris kode

Contoh dataset:

- MNIST
- CIFAR
- IMDB
- dll

Kelebihan

- cepat untuk eksperimen
- konsisten
- reproducible

Kekurangan

- tidak cocok untuk data custom perusahaan