

# Ringkasan Buku 1

Nama Buku: Tensorflow in Actions

Team: Wilhelmina Arlene, Luthfiah Maulidya, Fadhilah Dwi Istiani

## Chapter 15 — *TFX: MLOps and Deploying Models with TensorFlow*

### Tujuan Utama Chapter 15

Chapter 15 menjawab pertanyaan yang sangat krusial di dunia nyata:

**"Bagaimana model TensorFlow dibawa dari notebook → production secara rapi, terstandar, dan dapat dipercaya?"**

Di sinilah buku **beralih dari modeling ke MLOps**.

Fokus utama chapter ini:

- memahami **TFX (TensorFlow Extended)**
- membangun **pipeline ML end-to-end**
- training model secara terstandar
- **deployment model ke production**
- serving model melalui **API**

### **GAMBARAN BESAR: APA ITU TFX?**

#### **Apa itu TFX?**

**TensorFlow Extended (TFX)** adalah:

- framework MLOps dari TensorFlow
- untuk membangun **pipeline ML production-ready**
- berbasis **komponen modular**

TFX dirancang untuk:

- reproducibility
- scalability
- reliability
- automation

 TFX memaksa ML project mengikuti **best practice industri**.

### **PERBEDAAN EXPERIMENTAL ML vs PRODUCTION ML**

#### **ML Eksperimental**

- manual
- banyak notebook
- preprocessing terpisah
- sulit direproduksi

## ML Production (TFX)

- pipeline terstruktur
- preprocessing konsisten
- model tervalidasi
- deployment terkontrol

👉 **TFX menjembatani jurang ini.**

## STRUKTUR BESAR CHAPTER 15

Chapter 15 dibagi menjadi **4 bagian utama**:

1. Data pipeline dengan TFX
2. Training model dengan TFX Trainer
3. Model serving dengan Docker & TensorFlow Serving
4. Deployment & inference via API

### 15.1 Writing a Data Pipeline with TFX

#### Konsep Pipeline TFX

Pipeline TFX:

- terdiri dari **komponen**
- setiap komponen punya tugas spesifik
- output satu komponen → input komponen berikutnya

#### Komponen Data Utama TFX

##### 1 ExampleGen

- membaca data mentah (CSV, dll)
- mengubah ke format standar TFX

##### 2 StatisticsGen

- menghitung statistik data
- distribusi fitur
- missing values

Tujuan:

- data understanding
- data validation

##### 3 SchemaGen

- membuat schema otomatis
- mendefinisikan:
  - tipe fitur
  - range nilai
  - constraint

Schema ini menjadi **kontrak data**.

#### **Transform**

- preprocessing fitur
- normalisasi
- feature engineering

 **Penting:**

preprocessing dilakukan **di dalam pipeline**, bukan di notebook terpisah.

Ini menjamin:

- training = inference preprocessing

## **15.2 Training a Model with TFX Trainer API**

### **TFX Trainer**

Trainer bertugas:

- melatih model TensorFlow/Keras
- menggunakan data hasil Transform
- menghasilkan model siap production

### **Defining the Model**

Model didefinisikan dengan:

- Keras API
- tanpa bergantung pada pipeline manual

### **SignatureDefs**

SignatureDefs mendefinisikan:

- bagaimana model dipanggil dari luar
- input & output API

Ini penting untuk:

- serving
- interoperability

### **Training dalam Pipeline**

Training:

- otomatis
- reproducible
- bisa dijalankan ulang tanpa error inkonsistensi

## **15.3 Setting up Docker to Serve a Trained Model**

### **Kenapa Docker?**

Docker:

- mengemas model + environment
- konsisten di semua mesin
- memudahkan deployment

### **TensorFlow Serving**

TensorFlow Serving:

- server khusus untuk model TensorFlow
- high performance
- scalable
- production-tested

Pipeline:

SavedModel

→ Docker Image

→ TensorFlow Serving

## **15.4 Deploying the Model and Serving It through an API**

### **Validating Infrastructure**

Sebelum deploy:

- pastikan environment benar
- model bisa dimuat
- dependency lengkap

### **Resolving the Correct Model**

TFX memastikan:

- model terbaik yang dipilih
- berdasarkan evaluasi
- bukan model sembarangan

### **Model Evaluation & Pushing**

TFX bisa:

- membandingkan model lama vs baru
- hanya deploy jika:
  - performa meningkat
  - lolos threshold

 Ini **guardrail production ML**.

### **Prediction via TensorFlow Serving API**

Setelah deploy:

- model diakses via REST API
- client mengirim request
- server mengembalikan prediksi

Ini adalah **titik akhir lifecycle ML**.