

Nama : Mauricio Bethoven Tigauw  
NIM : 1103204099  
Kelas : TK-44-04

## **PyTorch Going Modular**

PyTorch dirancang sebagai framework deep learning yang fleksibel dan mudah digunakan. Namun, seiring waktu, PyTorch terus berkembang dan memperkenalkan perubahan signifikan dalam strukturnya untuk meningkatkan modularitas dan fleksibilitasnya. Transformasi ini dikenal sebagai "PyTorch Going Modular."

Berikut adalah beberapa aspek kunci dari perubahan ini:

### **1. Libtorch dan ATen (Tensors and Operations Library):**

- Sebelum modularisasi, implementasi PyTorch didasarkan pada monolitik C++ library yang dikenal sebagai "TH" (Torch). Dengan PyTorch Going Modular, komponen inti seperti tensor dan operasi-operasi fundamental dipisahkan ke dalam proyek yang disebut Libtorch dan ATen.
- Libtorch bertanggung jawab untuk mengekspor API PyTorch ke C++ untuk integrasi yang lebih baik dengan proyek-proyek berbasis C++.
- ATen (short for "A Tensor") menyediakan representasi tensor yang bersifat independen dari backend dan mendukung operasi tensor pada tingkat rendah.

### **2. C10 (C++ Core Library):**

- C10 adalah komponen inti yang menyediakan fungsionalitas inti dan infrastruktur untuk PyTorch.
- C10 juga memfasilitasi migrasi ke berbagai backend perangkat keras dan memastikan fleksibilitas serta portabilitas PyTorch.

### **3. PyTorch sebagai frontend:**

- Dengan modularisasi, PyTorch berperan sebagai frontend yang terpisah dari backend. Ini memungkinkan integrasi yang lebih baik dengan backend yang berbeda, termasuk backend yang dioptimalkan untuk perangkat keras tertentu.

### **4. TorchScript dan TorchScript IR (Intermediate Representation):**

- Modularisasi mencakup pengembangan dan penyempurnaan TorchScript, yang memungkinkan pengguna untuk mengonversi model PyTorch ke format serializable dan menjalankannya di berbagai lingkungan, termasuk lingkungan tanpa Python.

### **5. Plugin Sistem Grafis:**

- Sistem grafis PyTorch diubah untuk mendukung plugin, memungkinkan integrasi mudah dengan berbagai backend grafis.

Dengan modularisasi ini, PyTorch menjadi lebih fleksibel, dapat diintegrasikan dengan backend yang berbeda, dan memungkinkan pengembang untuk lebih mudah berkontribusi pada

bagian-bagian tertentu dari framework. Ini juga mendukung tujuan PyTorch untuk menjadi alat yang lebih kuat dan tangguh dalam ekosistem deep learning.