

# Teknologi Tepat Guna

## PEMBUATAN APLIKASI UNTUK KLASIFIKASI TUBERKULOSIS DENGAN MULTIMODAL DEEP LEARNING



Disusun Oleh:

**Dr. Nia Saurina, SST., M.Kom.**

**Dr. Noven Indra Prasetya, S.Kom., M.Kom.**



Tahun 2026

**TEKNOLOGI TEPAT GUNA  
PEMBUATAN APLIKASI UNTUK KLASIFIKASI TUBERKULOSIS DENGAN  
MULTIMODAL DEEP LEARNING**

**Dr. Nia Saurina, S.ST., M.Kom.  
Dr. Noven Indra Prasetya, S.Kom., M.Kom.**



**PENERBIT  
UWKS PRESS**

**TEKNOLOGI TEPAT GUNA  
PEMBUATAN APLIKASI UNTUK KLASIFIKASI TUBERKULOSIS DENGAN  
MULTIMODAL DEEP LEARNING**

ISBN .....  
Ukuran buku 21x 29 cm  
64 hlm Cetakan ke-1, Juni 2026

**Penulis:**

Dr. Nia Saurina, S.ST., M.Kom.  
Dr. Noven Indra Prasetya, S.Kom., M.Kom.

**Editor:**

Diyas Age Larasati

**Penerbit:**

UWKS PRESS

Anggota IKAPI No.206/Anggota Luar Biasa/JTI/2018 Anggota APPTI No.002.071.1.12019

Jl. Dukuh Kupang XXV/54 Surabaya Jawa Timur 60225  
Telp. (031) 5677577  
Hp. 085745182452  
Email : [uwkspress@gmail.com](mailto:uwkspress@gmail.com) / [uwkspress@uwks.ac.id](mailto:uwkspress@uwks.ac.id)

**Dilarang mengutip sebagian atau seluruh isi buku ini  
dengan cara apapun, termasuk dengan penggunaan mesin  
fotokopi, tanpa izin sah dari penerbit**

## KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT, karena atas berkat dan rahmat-Nya, buku yang berjudul **Pembuatan Aplikasi untuk Klasifikasi Tuberkulosis dengan Multimodal Deep Learning** ini dapat diselesaikan dengan baik. Buku ini disusun sebagai salah satu kontribusi dalam bidang teknologi tepat guna, khususnya dalam pemanfaatan kecerdasan buatan (*artificial intelligence*) untuk mendukung layanan kesehatan yang lebih akurat, cepat, dan mudah diakses.

Perkembangan teknologi *deep learning* telah membuka peluang besar dalam dunia medis, terutama dalam membantu diagnosis berbagai penyakit. Tuberkulosis (TBC) merupakan salah satu penyakit menular yang masih menjadi tantangan besar di negara berkembang. Melalui pendekatan multimodal deep learning yang menggabungkan berbagai jenis data seperti citra medis dan data klinis, diharapkan aplikasi yang dikembangkan dapat menjadi alat bantu yang efektif bagi tenaga kesehatan dalam melakukan klasifikasi tuberkulosis secara lebih objektif.

Buku ini hadir sebagai panduan praktis bagi para mahasiswa, peneliti, praktisi teknologi informasi, maupun tenaga kesehatan yang ingin memahami proses pengembangan aplikasi berbasis multimodal deep learning. Materi yang disajikan meliputi konsep dasar tuberkulosis, tinjauan tentang deep learning, metode multimodal, serta langkah-langkah teknis dalam pembuatan aplikasi mulai dari pengumpulan data, pra-pemrosesan, pengembangan model, hingga implementasi antarmuka pengguna. Buku Teknologi Tepat Guna ini disusun berdasarkan hasil penelitian yang dibiayai oleh PUSLAPDIKTI Beasiswa Pendidikan Indonesia (BPI) dengan Nomor 00170/BPPT/BPI.06/9/2023.

Kami menyadari bahwa buku ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan untuk perbaikan di masa mendatang. Ucapan terima kasih kami sampaikan kepada semua pihak yang telah membantu dalam penyusunan buku ini, terutama kepada tim peneliti, institusi mitra, serta keluarga yang senantiasa memberikan dukungan.

Akhir kata, semoga buku ini dapat memberikan manfaat nyata bagi pengembangan teknologi tepat guna di Indonesia, serta turut berkontribusi dalam upaya percepatan penanggulangan tuberkulosis melalui inovasi teknologi yang aplikatif dan berkelanjutan.

Tim Penyusun

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI .....	iv
BAB I PENDAHULUAN .....	1
BAB II BAHAN DAN PERALATAN .....	8
BAB III PROSES PEMBUATAN APLIKASI .....	19
BAB IV ANALISIS KESEHATAN .....	56
DAFTAR PUSTAKA .....	62

# BAB I PENDAHULUAN

## 1.1 Mengapa Buku Ini Dibuat

Di era digital yang serba cepat ini, teknologi tidak lagi menjadi barang asing di ruang-ruang pelayanan kesehatan. Namun, masih ada kesenjangan antara perkembangan kecerdasan buatan (AI) yang pesat dengan penerapannya di lapangan, terutama di fasilitas kesehatan tingkat pertama. Buku ini lahir dari satu keyakinan sederhana: **teknologi canggih harus bisa menjadi alat yang benar-benar berguna, mudah digunakan, dan menyentuh kebutuhan nyata masyarakat.**

Tuberkulosis (TBC) adalah salah satu penyakit menular yang masih menjadi momok di Indonesia. Setiap tahun, ratusan ribu kasus baru ditemukan, dan tidak sedikit yang terlambat terdiagnosis karena keterbatasan akses terhadap tenaga ahli atau alat diagnostik yang memadai. Padahal, diagnosis dini adalah kunci utama untuk memutus rantai penularan dan meningkatkan kesembuhan. Di sinilah peran teknologi tepat guna menjadi sangat penting.

**Apa yang istimewa dari pendekatan dalam buku ini?** Kami tidak sekadar membahas teori deep learning secara abstrak. Buku ini mengajak Mahasiswa atau Peneliti untuk membangun sebuah aplikasi nyata yang menggabungkan dua jenis data sekaligus yaitu gambar dahak dari Tuberkulosis dari tiga kanal warna lalu memprosesnya dengan model multimodal deep learning. Mengapa multimodal? Karena dalam dunia nyata, dokter tidak hanya melihat foto dahak, tetapi juga mempertimbangkan gejala, riwayat kesehatan, dan faktor risiko lainnya. Aplikasi yang kami buat meniru cara kerja profesional kesehatan: melihat dari berbagai sudut sebelum mengambil kesimpulan.

Buku ini disusun untuk menjawab pertanyaan-pertanyaan praktis:

- **Bagaimana cara memulai membuat aplikasi AI untuk kesehatan, terutama jika saya bukan ahli pemrograman?**
- **Data apa saja yang dibutuhkan dan dari mana mendapatkannya?**
- **Langkah demi langkah membangun model multimodal yang tidak rumit namun tetap akurat?**
- **Bagaimana cara mengemas model tersebut menjadi aplikasi yang bisa dijalankan di komputer biasa?**

Kami percaya bahwa inovasi tidak harus lahir dari laboratorium riset yang mahal. Dengan perangkat keras yang terjangkau, perangkat lunak sumber terbuka, dan panduan yang tepat, siapa pun baik. Mahasiswa, tenaga kesehatan, atau pengembang pemula juga dapat menciptakan solusi yang berdampak.

Lebih dari sekadar buku panduan teknis, buku ini juga mengajak Mahasiswa dan Peneliti untuk memahami konteks kesehatan dari aplikasi yang dibuat. Setiap bab tidak hanya menjelaskan *how*, tetapi juga *why*, mengapa langkah ini penting, bagaimana hasilnya bisa diinterpretasikan secara medis, dan apa yang harus diperhatikan agar aplikasi benar-benar aman dan bermanfaat.

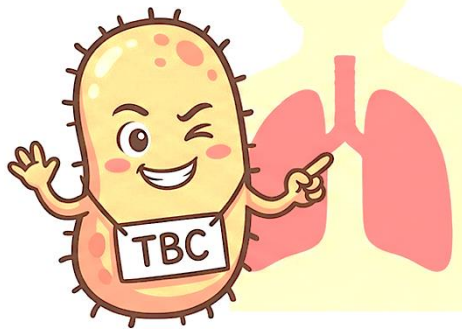
Dengan membaca buku ini, Mahasiswa dan Peneliti akan:

- Memahami konsep dasar multimodal deep learning secara sederhana.
- Mempraktikkan langsung pembuatan aplikasi dari awal hingga siap pakai.
- Mengetahui cara menguji dan mengevaluasi aplikasi di lingkungan kesehatan sungguhan.
- Memiliki bekal untuk mengembangkan solusi serupa di bidang kesehatan lainnya.

Akhirnya, buku ini adalah undangan untuk ikut serta dalam gerakan teknologi tepat guna. Kita tidak perlu menunggu solusi jadi dari luar; kita bisa membangunnya sendiri, menyesuaikan dengan kondisi lokal, dan bersama-sama berkontribusi pada upaya penanggulangan tuberkulosis di Indonesia. Mari, mulai langkah pertama.

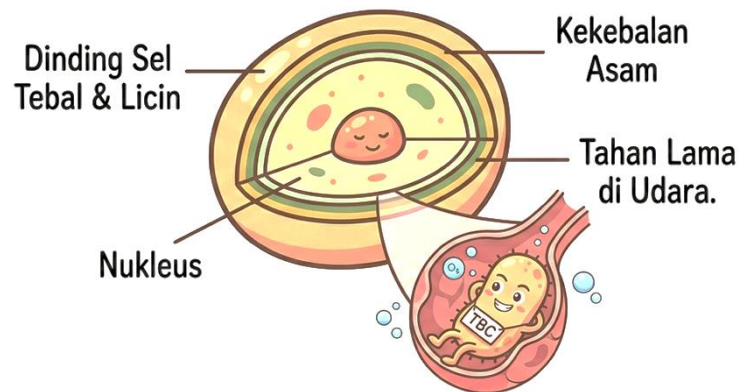
## **1.2 Apa Itu Tuberkulosis dan Tantangan Diagnosisnya**

Tuberkulosis, atau yang lebih akrab disebut TBC, adalah penyakit infeksi yang disebabkan oleh bakteri *Mycobacterium tuberculosis*. Penyakit ini paling sering menyerang paru-paru, tetapi sebenarnya bisa juga menyerang organ lain seperti tulang, kelenjar getah bening, hingga selaput otak. TBC menyebar melalui udara, ketika penderita batuk, bersin, atau berbicara, percikan dahak yang mengandung bakteri dapat terhirup oleh orang lain di sekitarnya.



Perkenalkan, aku Tuberkulosis. Tapi panggil saja aku TBC. Aku bukan monster dari film fiksi ilmiah—aku nyata, dan aku sudah hidup berdampingan dengan manusia selama ribuan tahun. Aku senang bersembunyi di dalam tubuh, terutama di paru-paru, dan aku bisa menyebar begitu saja saat salah satu “kendaraanku” batuk atau bersin di tengah keramaian.

Di Indonesia, TBC masih menjadi salah satu masalah kesehatan masyarakat terbesar. Menurut data global, Indonesia menempati peringkat kedua dengan jumlah kasus TBC terbanyak di dunia. Setiap harinya, ribuan orang terinfeksi, dan sayangnya, tidak sedikit yang tidak terdeteksi hingga kondisinya memburuk. Padahal, TBC sebenarnya bisa disembuhkan dengan obat-obatan yang tersedia secara gratis di fasilitas kesehatan pemerintah. Kuncinya ada pada satu kata: **diagnosis dini**.



Aku berasal dari makhluk bernama *Mycobacterium tuberculosis*. Ukuranku sangat kecil, tidak terlihat mata telanjang, tapi jangan remehkan aku. Aku punya dinding sel yang tebal dan licin, membuatku kebal terhadap kondisi asam, tahan lama di udara, dan sulit dihancurkan oleh sistem kekebalan tubuh manusia. Aku suka tempat yang lembap dan kaya oksigen, itulah mengapa paru-paru adalah rumah favoritku.

### 1.3 Metode Konvensional yang Ada Saat Ini

Secara umum, diagnosis TBC dilakukan melalui beberapa pendekatan:

#### 1. Pemeriksaan Mikroskopis BTA (Bakteri Tahan Asam)

Sampel dahak pasien diperiksa di bawah mikroskop untuk melihat apakah ada bakteri TBC. Metode ini murah dan sudah digunakan puluhan tahun. Namun, kelemahannya: sensitivitasnya terbatas, terutama pada pasien dengan jumlah bakteri sedikit. Diperlukan tenaga laboratorium yang terlatih, dan hasilnya bisa memakan waktu.

#### 2. Tes Cepat Molekuler (TCM atau GeneXpert)

Ini adalah metode yang lebih modern dan akurat, karena mendeteksi materi genetik bakteri. Sayangnya, mesin dan kartrid reagen masih relatif mahal, dan ketersediaannya belum merata di semua puskesmas atau klinik terpencil.

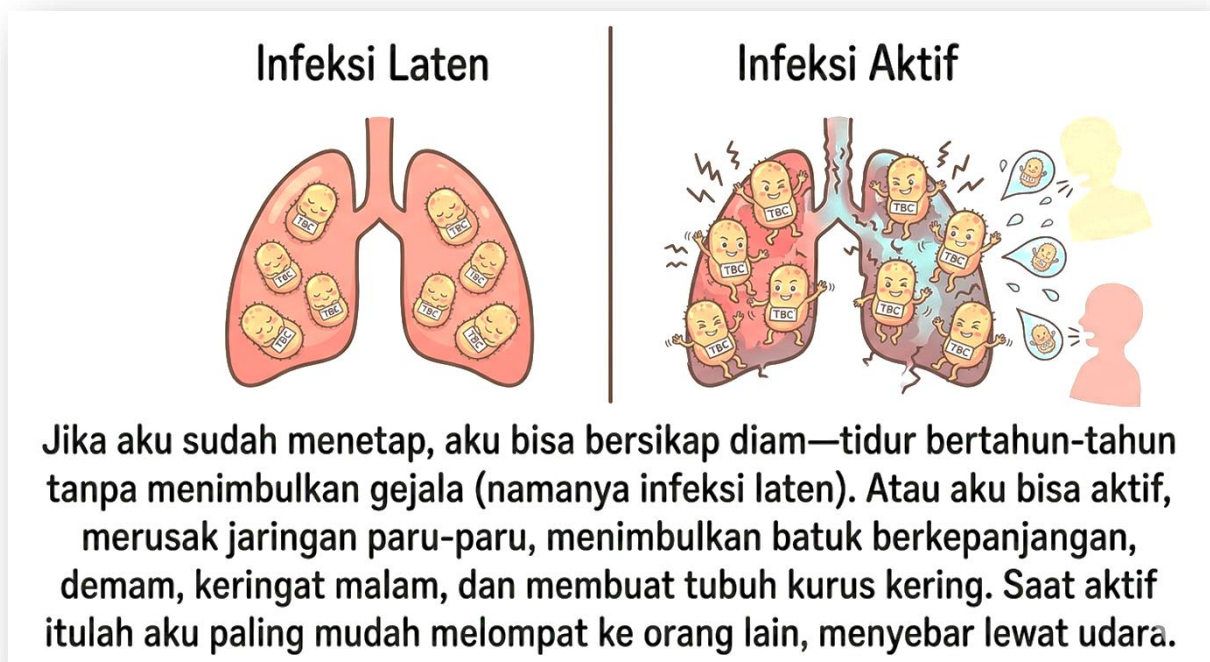
#### 3. Pemeriksaan Rontgen Dada

Foto rontgen dapat menunjukkan gambaran kelainan di paru-paru yang mengarah pada TBC. Namun, interpretasi gambar sangat bergantung pada pengalaman dokter radiologi. Tidak semua fasilitas kesehatan memiliki dokter spesialis radiologi setiap saat, dan terkadang temuan pada rontgen juga bisa mirip dengan penyakit lain seperti pneumonia atau kanker paru.

#### 4. **Diagnosis Klinis Berdasarkan Gejala**

Dokter bisa mencurigai TBC jika pasien memiliki batuk lebih dari dua minggu, disertai demam, keringat malam, penurunan berat badan, dan riwayat kontak dengan penderita TBC. Metode ini sangat bergantung pada kelengkapan informasi dan pengalaman klinis. Gejala yang tidak khas sering menyebabkan keterlambatan atau bahkan kesalahan diagnosis.

**Akses untuk “menangkapku” tidak merata.** Untuk mendeteksi Tuberkulosis secara pasti, manusia butuh pemeriksaan dahak mikroskopis, tes cepat molekuler (TCM), atau foto rontgen. Tapi alat-alat itu tidak selalu tersedia di puskesmas terpencil. Aku sering lolos dari radar karena pasien malas periksa atau fasilitasnya jauh.



#### 1.4 **Kendala Utama di Lapangan**

Penerapan untuk melakukan deteksi dan klasifikasi di lapangan menghadapi sejumlah hambatan:

- **Keterbatasan akses** – Di daerah terpencil, laboratorium dengan fasilitas TCM atau bahkan mikroskop yang layak belum tentu ada. Pasien harus menempuh perjalanan jauh hanya untuk pemeriksaan.

- **Keterbatasan tenaga ahli** – Tidak semua puskesmas memiliki dokter spesialis paru atau radiolog. Padahal interpretasi hasil rontgen dan pemeriksaan penunjang lain membutuhkan keahlian khusus.
- **Waktu tunggu yang lama** – Dari pengambilan sampel hingga hasil keluar bisa memakan waktu berhari-hari. Selama menunggu, pasien bisa saja tidak kembali atau penularan terus berlangsung.
- **Keterbatasan alat** – Mesin TCM yang mahal dan perawatannya rumit membuat distribusinya tidak merata.
- **Stigma dan kesadaran** – Sebagian masyarakat masih menganggap TBC sebagai penyakit memalukan atau tidak serius, sehingga enggan memeriksakan diri.



### 1.5 Di Mana Teknologi Bisa Masuk?

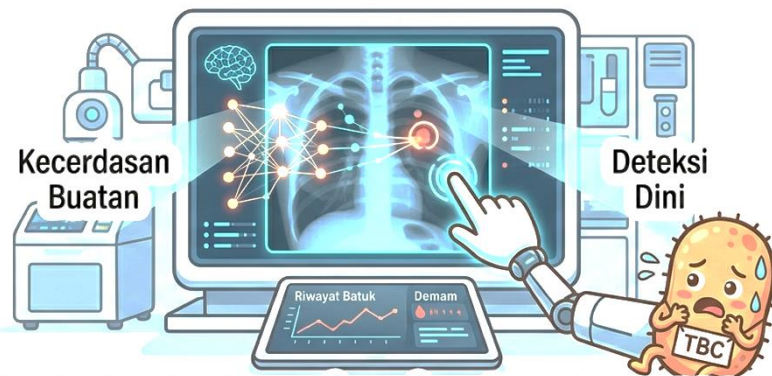
Muncul pertanyaan: *"Apakah ada cara yang lebih cepat, lebih mudah diakses, dan bisa menjadi pelengkap metode konvensional?"*

Jawabannya adalah pemanfaatan kecerdasan buatan, khususnya deep learning. Dalam beberapa tahun terakhir, penelitian menunjukkan bahwa algoritma deep learning mampu mengenali pola-pola pada gambar rontgen dada dengan tingkat akurasi yang mendekati ahli radiologi. Bahkan, jika dikombinasikan dengan data klinis pasien (umur, gejala, riwayat kontak, hasil pemeriksaan cepat), akurasinya bisa lebih baik lagi. Pendekatan inilah yang disebut **multimodal deep learning**.

Bayangkan sebuah aplikasi yang bisa:

- Menerima foto dahak yang diambil dari kamera puskesmas atau bahkan smartphone.
- Memasukkan data klinis pasien melalui formulir sederhana.
- Dalam hitungan detik, memberikan hasil klasifikasi apakah pasien kemungkinan besar positif TBC atau tidak, lengkap dengan rekomendasi langkah selanjutnya.

Aplikasi semacam ini bukan untuk menggantikan peran dokter, melainkan untuk menjadi **alat bantu** terutama di fasilitas kesehatan dengan sumber daya terbatas agar skrining bisa dilakukan lebih cepat dan lebih dini. Dengan begitu, pasien yang terindikasi positif dapat segera dirujuk untuk pemeriksaan konfirmasi dan memulai pengobatan lebih awal.



Sejujurnya, aku tidak suka dengan kemajuan teknologi yang sekarang mulai memburuku... Aku risi dengan pendekatan multimodal *deep learning* itu. Karena jika manusia bisa mendeteksiku lebih awal dan lebih akurat, aku kehilangan kesempatan untuk menyebar. Mereka akan segera memberikan obat-obatan yang membuatku mati sebelum sempat berkembang biak.

Buku ini memandu Mahasiswa dan Peneliti untuk membangun aplikasi tepat guna seperti itu. Kita akan belajar bagaimana mengolah data, melatih model, dan menghadirkan solusi yang benar-benar bisa digunakan di dunia nyata. Karena pada akhirnya, teknologi baru bermakna jika ia mampu menjawab tantangan yang ada di sekitar kita.

## BAB II BAHAN DAN PERALATAN

### 2.1 Mengenal Data Multimodal

Di balik mikroskop, ahli mikrobiologi mencari bakteri berbentuk batang kecil berwarna merah cerah setelah pewarnaan khusus. Tapi ada tantangan: tidak semua fasilitas kesehatan memiliki ahli mikrobiologi yang tersedia setiap saat, dan interpretasi visual bisa bervariasi antar pengamat. Di sinilah teknologi bisa menjadi teman sejati.

Dalam buku ini, Mahasiswa dan Peneliti membangun aplikasi yang meniru cara ahli mikrobiologi bekerja, tetapi dengan kekuatan tambahan: **deep learning**. Aplikasi kita akan belajar dari gambar dahak yang diwarnai dan data klinis pasien. Pendekatan ini disebut **multimodal**, karena kita menggabungkan dua jenis informasi yang berbeda untuk mendapatkan hasil klasifikasi yang lebih akurat.

### 2.2 Jenis Data yang Digunakan: Gambar Dahak dan Data Klinis Pasien

#### 1. Gambar Dahak (Sputum Images)

Dahak adalah lendir yang dikeluarkan dari saluran pernapasan bawah. Pada pasien dengan dugaan TBC, sampel dahak diambil dan diberi pewarnaan khusus (seperti pewarnaan Ziehl-Neelsen) sehingga bakteri *Mycobacterium tuberculosis* tampak berwarna merah dengan latar belakang biru atau kehijauan. Gambar inilah yang akan kita gunakan sebagai salah satu sumber data.

Agar model deep learning kita bisa “melihat” dengan lebih tajam, kita tidak hanya menggunakan satu jenis representasi warna. Kita akan menggunakan **tiga kanal warna yang berbeda** untuk gambar yang sama. Kenapa tiga? Karena masing-masing menangkap informasi yang berbeda:

- **RGB (Red, Green, Blue)**

Ini adalah representasi warna paling umum yang digunakan oleh kamera dan mata manusia. Dalam gambar dahak, RGB menampilkan warna merah dari bakteri dan warna latar belakang sesuai pewarnaan. RGB sangat baik untuk menangkap kontras visual yang familiar bagi manusia.

- **CieLab (atau CIELAB)**

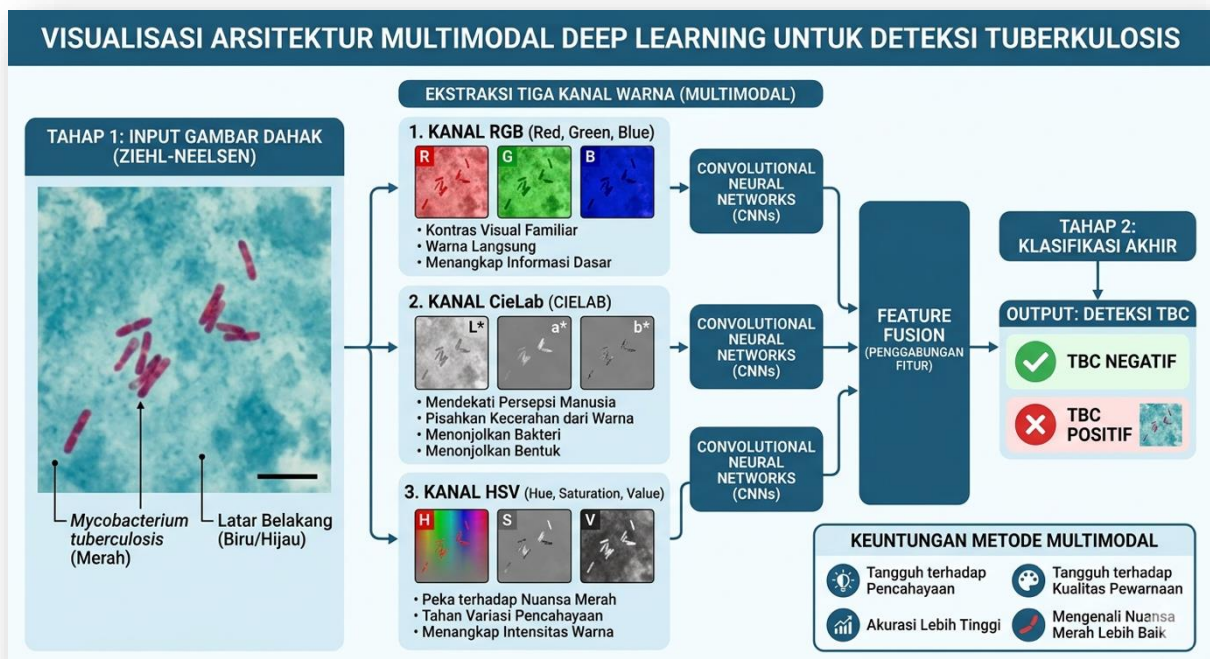
Representasi ini dirancang untuk mendekati persepsi warna manusia secara lebih seragam. CieLab memisahkan informasi warna (kromatisitas) dari tingkat kecerahan (luminance). Ini berguna karena dalam pewarnaan dahak, bentuk dan warna bakteri bisa

lebih menonjol jika kita fokus pada perbedaan warna tanpa terganggu oleh variasi pencahayaan.

- **HSV (Hue, Saturation, Value)**

HSV memisahkan warna (hue) dari intensitas warna (saturation) dan kecerahan (value). Dalam konteks gambar dahak, HSV membantu model untuk lebih peka terhadap nuansa warna merah bakteri, terlepas dari seberapa terang atau gelap gambar tersebut.

Dengan menggunakan tiga representasi warna sekaligus, kita memberi model “tiga pasang mata” yang masing-masing melihat dari sudut pandang berbeda. Hasilnya, model menjadi lebih tangguh dalam mengenali bakteri di berbagai kondisi pencahayaan, kualitas pewarnaan, dan latar belakang.



## 2. Data Klinis Pasien

Selain gambar, kita juga membutuhkan informasi tentang pasien. Data klinis ini menjadi pelengkap yang sangat penting karena tidak semua kasus TBC bisa dideteksi hanya dari gambar dahak terutama jika jumlah bakteri sedikit atau pewarnaan kurang sempurna.

Data klinis yang biasanya digunakan meliputi:

- **Data demografi:** usia, jenis kelamin.
- **Gejala:** batuk lebih dari dua minggu, demam, keringat malam, penurunan berat badan, sesak napas.
- **Riwayat kontak:** apakah pernah kontak erat dengan penderita TBC.
- **Riwayat pengobatan:** apakah pernah mengobati TBC sebelumnya.
- **Hasil pemeriksaan pendukung** (jika ada): seperti tes cepat molekuler (TCM) atau tes tuberkulin.

Data ini bersifat tabular berbentuk angka dan kategori yang bisa diproses oleh model deep learning bersama dengan fitur-fitur yang diekstrak dari gambar.

### 2.3 Dari Mana Mendapatkan Data

Untuk membangun aplikasi, kita membutuhkan data dalam jumlah yang cukup untuk melatih model. Ada dua jalur utama:

#### 1. Dataset Publik

Beberapa institusi riset dan universitas menyediakan dataset gambar dahak yang dapat diunduh secara legal untuk keperluan pengembangan. Contohnya adalah:

- *Sputum Microscopy Image Datasets* yang tersedia di platform seperti Kaggle atau situs repositori universitas.
- Dataset dari lembaga seperti *Tuberculosis Image Database* yang kadang dirilis untuk kompetisi atau riset terbuka. Untuk hasil penelitian ini, dataset didapatkan dari RSUD Dr. Soetomo Surabaya dengan surat kelayakan etik yang dikeluarkan oleh Perhimpunan Dokter Spesialis Patologi Klinik dan Kedokteran Laboratorium (PATKLIN) RSUD. Dr Soetomo Surabaya - Fakultas Kedokteran Universitas Airlangga. Surat dikeluarkan pada tanggal 21 Februari 2023, dengan nomor 53/EC/KEPK/FKUA/2023

#### 2. Kerja Sama dengan Fasilitas Kesehatan

Untuk melakukan mengembangkan aplikasi yang lebih spesifik untuk kondisi lokal atau ingin melakukan uji coba di lapangan, Mahasiswa dan Peneliti dapat bekerja sama dengan rumah sakit, puskesmas, atau laboratorium kesehatan. Namun, proses ini memerlukan:

- Perjanjian kerja sama tertulis.
- Izin etik (ethical clearance) dari komite etik penelitian.
- Prosedur pengambilan data yang sesuai dengan standar operasional.

Kerja sama semacam ini juga membuka peluang untuk mendapatkan umpan balik langsung dari tenaga kesehatan yang akan menggunakan aplikasi, sehingga produk akhir lebih sesuai dengan kebutuhan mereka.

## 2.4 Etika dan Privasi Data Pasien

Data kesehatan adalah data yang sangat sensitif. Dalam mengumpulkan, menyimpan, dan menggunakan data pasien, ada beberapa prinsip yang wajib dipegang teguh:

- ***Informed Consent (Persetujuan Setelah Penjelasan)***

Setiap pasien yang datanya digunakan harus memberikan izin secara sadar dan sukarela. Mereka perlu dijelaskan tujuan penggunaan data, bagaimana data akan diproses, serta siapa saja yang akan mengaksesnya. Untuk dataset publik yang sudah dianonimkan, biasanya persetujuan sudah diurus oleh penyedia dataset.

- **Anonimisasi dan De-identifikasi**

Sebelum data digunakan untuk pengembangan, hapus semua informasi yang dapat mengidentifikasi pasien: nama, alamat, nomor rekam medis, bahkan tanggal lahir. Gunakan kode atau ID acak sebagai pengganti. Gambar wajah juga sebaiknya tidak menyertakan label identitas pasien.

- **Keamanan Data**

Simpan data di tempat yang aman. Gunakan enkripsi jika menyimpan di perangkat lokal, dan pastikan tidak ada akses tidak sah. Jika data disimpan di cloud, pilih layanan dengan sertifikasi keamanan yang memadai.

- **Kepatuhan Regulasi**

Di Indonesia, perlindungan data pribadi diatur dalam Undang-Undang Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi (UU PDP). Pastikan kegiatan Anda sesuai dengan ketentuan yang berlaku, termasuk kewajiban untuk mendapatkan persetujuan dan melaporkan jika terjadi kebocoran data.

- **Penggunaan Sesuai Tujuan**

Jangan menggunakan data di luar tujuan yang telah disepakati. Misalnya, data yang dikumpulkan untuk pengembangan aplikasi tidak boleh dijual atau digunakan untuk keperluan komersial lain tanpa izin ulang.

## 2.6 Persiapan Perangkat Keras

Sebelum mulai menulis kode dan melatih model deep learning, kita perlu memastikan perangkat keras yang digunakan memadai. Pada sub bab ini, kita akan membagi kebutuhan perangkat keras menjadi dua fase: **fase pengembangan (*training*)** dan **fase penggunaan (*deployment*)**.

## 2.7 Fase Pengembangan (*Training*)

Ini adalah fase di mana kita melatih model *multimodal deep learning* menggunakan gambar dan data klinis. Proses ini membutuhkan daya komputasi yang lebih besar karena model harus “belajar” dari ribuan gambar dan data. Berikut spesifikasi yang direkomendasikan:

- **Prosesor (CPU)**

Minimal prosesor modern dengan 4 core atau lebih. Intel Core i5 generasi ke-10 ke atas atau AMD Ryzen 5 setara sudah cukup. CPU berperan dalam persiapan data, augmentasi gambar, dan koordinasi proses pelatihan.

- **Memori (RAM)**

Minimal **16 GB**. Jika Anda menggunakan dataset yang besar (misalnya ribuan gambar dengan resolusi tinggi), 32 GB akan lebih nyaman. RAM yang cukup penting agar semua data dapat dimuat ke memori tanpa membuat komputer melambat.

- **Kartu Grafis (GPU)**

Ini adalah komponen paling krusial untuk pelatihan deep learning. GPU mempercepat perhitungan matriks yang menjadi tulang punggung jaringan saraf. Rekomendasi:

- **NVIDIA GPU** dengan arsitektur Turing atau lebih baru, misalnya GTX 1660 Ti, RTX 2060, RTX 3060, atau yang lebih tinggi.
- Minimal **VRAM 6 GB**, tetapi 8 GB atau lebih akan memberikan fleksibilitas untuk model yang lebih kompleks.
- Jika tidak memiliki GPU, Anda masih bisa menggunakan **GPU cloud** seperti Google Colab (gratis dengan GPU terbatas) atau layanan berbayar seperti AWS, Google Cloud, atau Azure. Kita akan membahas alternatif ini nanti.

- **Penyimpanan (*Storage*)**

Gunakan **SSD** dengan kapasitas minimal 512 GB. Proses pelatihan banyak membaca dan menulis file, dan SSD akan menghemat waktu yang signifikan dibandingkan HDD konvensional. Pastikan ada ruang yang cukup untuk menyimpan dataset, model hasil pelatihan, dan file pendukung lainnya.

- **Sistem Operasi**

Deep learning berjalan mulus di **Windows, Linux (Ubuntu)**, atau **macOS**. Linux sering menjadi pilihan karena lebih ringan dan kompatibilitas tinggi dengan pustaka-pustaka deep learning, tetapi Anda tetap bisa menggunakan Windows dengan WSL (Windows Subsystem for Linux) atau langsung di lingkungan native.

## **2.8 Alternatif: Komputasi Cloud (GPU Online)**

Tidak semua orang memiliki komputer dengan GPU yang mumpuni. Untungnya, ada banyak layanan cloud yang menyediakan akses GPU secara sewa. Beberapa opsi populer:

- **Google Colab**

Layanan gratis dari Google yang menyediakan GPU (biasanya NVIDIA Tesla T4 atau K80) dengan masa pakai terbatas. Cocok untuk pembelajaran dan proyek skala kecil. Anda juga bisa berlangganan Colab Pro untuk akses GPU prioritas dan RAM lebih besar.

- **Kaggle Notebooks**

Platform ini juga memberikan akses GPU secara gratis dengan batas waktu mingguan. Dataset sering tersedia langsung di platform.

- **AWS EC2, Google Cloud AI Platform, atau Azure Machine Learning**

Pilihan berbayar dengan konfigurasi fleksibel. Cocok jika Anda memerlukan GPU dengan VRAM besar atau ingin melatih model dalam waktu lama.

## **2.9 Fase Penggunaan (*Deployment*)**

Setelah model selesai dilatih, kita akan menyematkannya ke dalam aplikasi. Aplikasi inilah yang nantinya digunakan oleh tenaga kesehatan di puskesmas atau laboratorium. Perangkat keras untuk fase ini biasanya lebih sederhana:

- **Untuk Aplikasi Mobile**

Aplikasi dapat dijalankan di smartphone. Model yang sudah dioptimalkan (misalnya dengan TensorFlow Lite) dapat berjalan di:

- **Android:** Minimal Android 8.0 (Oreo) dengan RAM 4 GB. Prosesor modern seperti Snapdragon 600 series ke atas sudah cukup.
- **iOS:** iPhone dengan iOS 13 ke atas.

- **Untuk Aplikasi Desktop atau Web**

Jika aplikasi berbasis web (misalnya menggunakan Streamlit atau Flask), server dapat berupa komputer biasa dengan RAM 8 GB dan prosesor modern. Namun, jika banyak pengguna mengakses secara bersamaan, sebaiknya gunakan server dengan spesifikasi lebih tinggi.

## 2.10 Tips Memilih Perangkat Sesuai Anggaran

- Jika Mahasiswa dan Peneliti masih dalam tahap belajar atau mengembangkan prototipe, manfaatkan **Google Colab** untuk pelatihan dan gunakan laptop pribadi untuk pengembangan aplikasi.
- Jika Mahasiswa dan Peneliti berencana mengimplementasikan aplikasi di beberapa fasilitas kesehatan, pertimbangkan untuk menyediakan satu unit komputer atau tablet dengan spesifikasi yang sudah diuji untuk menjalankan aplikasi.

## 2.11 Persiapan Perangkat Lunak

Setelah perangkat keras siap, sekarang saatnya menyiapkan perangkat lunak. Pada sub bab ini, kita dapat membagi perangkat lunak ke dalam beberapa kategori berdasarkan fungsinya. Pengguna tidak perlu menginstal semuanya sekaligus; ikuti saja sesuai kebutuhan tahap pengembangan.

### 2.11.1 Sistem Operasi dan Lingkungan Pengembangan

Deep learning dapat dijalankan di Windows, macOS, maupun Linux. Jika Anda menggunakan Windows, sangat disarankan untuk mengaktifkan **Windows Subsystem for Linux (WSL)** karena banyak pustaka deep learning yang lebih mudah diinstal di lingkungan Linux. Alternatifnya, Anda bisa langsung menggunakan **Ubuntu** atau distribusi Linux lainnya. Untuk menulis kode, kita membutuhkan **editor teks** atau **Integrated Development Environment (IDE)**. Pilihan populer:

- **Visual Studio Code** – ringan, banyak ekstensi untuk Python dan deep learning.
- **Jupyter Notebook / JupyterLab** – cocok untuk eksplorasi data dan percobaan interaktif.
- **PyCharm** – IDE khusus Python dengan fitur lengkap.

Gunakan Visual Studio Code sebagai editor utama karena fleksibilitas dan dukungan komunitas yang luas.

### 2.11.2 Manajemen Lingkungan dan Instalasi Pustaka

Sebelum menginstal pustaka Python, ada baiknya kita membuat **virtual environment**. Ini seperti ruang terpisah untuk proyek kita, sehingga tidak mengganggu proyek lain di komputer. Caranya sangat mudah:

```
bash
python -m venv tb_app_env
source tb_app_env/bin/activate # Linux/macOS
tb_app_env\Scripts\activate # Windows
```

### 2.11.3. Pustaka Deep Learning (Training Model)

Pengguna dapat menggunakan **TensorFlow** sebagai pustaka utama karena relatif mudah dipelajari dan memiliki dokumentasi yang lengkap. Instalasi dapat dilakukan dengan perintah:

```
bash
pip install tensorflow
```

Alternatif lain adalah **PyTorch**, yang juga populer di dunia riset. Untuk buku ini, kita akan fokus pada TensorFlow/Keras karena lebih ramah bagi pemula.

### 2.11.4. Pustaka Pengolahan Gambar dan Data

Agar bisa memanipulasi gambar dan data klinis, kita perlu beberapa pustaka:

- **OpenCV** (`opencv-python`) – untuk membaca, mengubah ukuran, dan memanipulasi gambar. OpenCV juga akan kita gunakan untuk mengonversi gambar ke berbagai kanal warna (RGB, CieLab, HSV).
- **PIL / Pillow** – pustaka alternatif untuk pengolahan gambar, ringan dan mudah.
- **NumPy** – untuk operasi array numerik, sangat diperlukan dalam deep learning.
- **Pandas** – untuk membaca dan mengelola data klinis yang berbentuk tabel (CSV, Excel).

Instal dengan perintah:

```
Bash  
pip install opencv-python pillow numpy pandas
```

### 2.11.5. Pustaka untuk Pra-pemrosesan Gambar Dahak: YOLO dan Bounding Box

Dalam aplikasi, gambar dahak yang diambil dari mikroskop sering kali mengandung banyak objek (sel darah, lendir, bakteri). Agar model deep learning fokus pada area yang penting, kita akan menggunakan **deteksi objek** untuk menemukan lokasi bakteri. Di sinilah **YOLO (You Only Look Once)** berperan.

YOLO adalah algoritma deteksi objek yang terkenal cepat dan akurat. Dengan YOLO, kita bisa melatih model untuk mengenali bakteri *Mycobacterium tuberculosis* dalam gambar, lalu menghasilkan **kotak pembatas (bounding box)** di sekitar setiap bakteri yang terdeteksi.

#### Mengapa bounding box penting?

- Bounding box membantu model untuk fokus pada area yang mengandung bakteri, mengabaikan latar belakang yang tidak relevan.
- Setelah bounding box diperoleh, kita bisa memotong (*crop*) area tersebut atau menggunakan informasi posisi sebagai fitur tambahan.
- Dengan tiga representasi warna (RGB, CieLab, HSV), kita bisa menerapkan bounding box di ketiganya untuk memperkaya data latih.

Untuk mengimplementasikan YOLO dalam proyek ini, kita dapat menggunakan implementasi **YOLOv8** dari Ultralytics karena mudah digunakan dan memiliki dokumentasi yang baik. Instalasinya cukup dengan:

```
bash  
pip install ultralytics
```

### 2.11.6. Pustaka Optimasi Model dan Deployment

Setelah model selesai dilatih, kita perlu mengemasnya agar dapat berjalan di aplikasi (mobile atau web). Untuk itu, kita akan menggunakan **TensorFlow Lite** untuk mengonversi model ke format yang ringan dan cepat di perangkat mobile.

```
bash
pip install tensorflow-lite
```

Jika aplikasi berbasis web, kita bisa menggunakan **Flask** atau **Streamlit** untuk membuat antarmuka sederhana. Streamlit sangat cocok untuk prototipe karena hanya dengan beberapa baris kode kita sudah memiliki aplikasi web interaktif.

```
bash
pip install streamlit
```

### 2.11.7. Alat Bantu Lainnya

- **Git** – untuk mengelola versi kode. Jika Anda bekerja dalam tim atau ingin menyimpan proyek di GitHub, Git sangat membantu.
- **CUDA dan cuDNN** (jika menggunakan GPU NVIDIA) – pustaka dari NVIDIA yang mempercepat komputasi di GPU. Pastikan versi yang diinstal sesuai dengan TensorFlow yang digunakan.

### 2.11.8. Struktur Proyek dan Manajemen Kode

Agar proyek tetap rapi, susun folder seperti ini:

```
text
tb_app_project/
|
|—— data/
|   |—— raw/           # Data mentah (gambar asli, data klinis)
|   |—— processed/     # Data setelah preprocessing (bounding box, aug
mentasi)
|   |—— annotations/   # File anotasi bounding box dari YOLO
|
|—— notebooks/        # Jupyter notebook untuk eksplorasi
|
|—— src/
```

		preprocessing/	# Skrip untuk preprocessing gambar dan data
		models/	# Arsitektur model multimodal
		training/	# Skrip pelatihan
		utils/	# Fungsi bantu
		app/	# Kode aplikasi (Streamlit, Flutter, dll)
		models/	# Model yang sudah dilatih (saved model, tflite)
		requirements.txt	# Daftar pustaka yang dibutuhkan
		README.md	# Penjelasan proyek

### 2.11.9. Ringkasan Instalasi (Checklist)

Berikut daftar pustaka yang perlu diinstal sekaligus dengan file `requirements.txt` yang berisi:

```
text
tensorflow==2.13.0
ultralytics
opencv-python
pillow
numpy
pandas
matplotlib
seaborn
scikit-learn
streamlit
flask
```

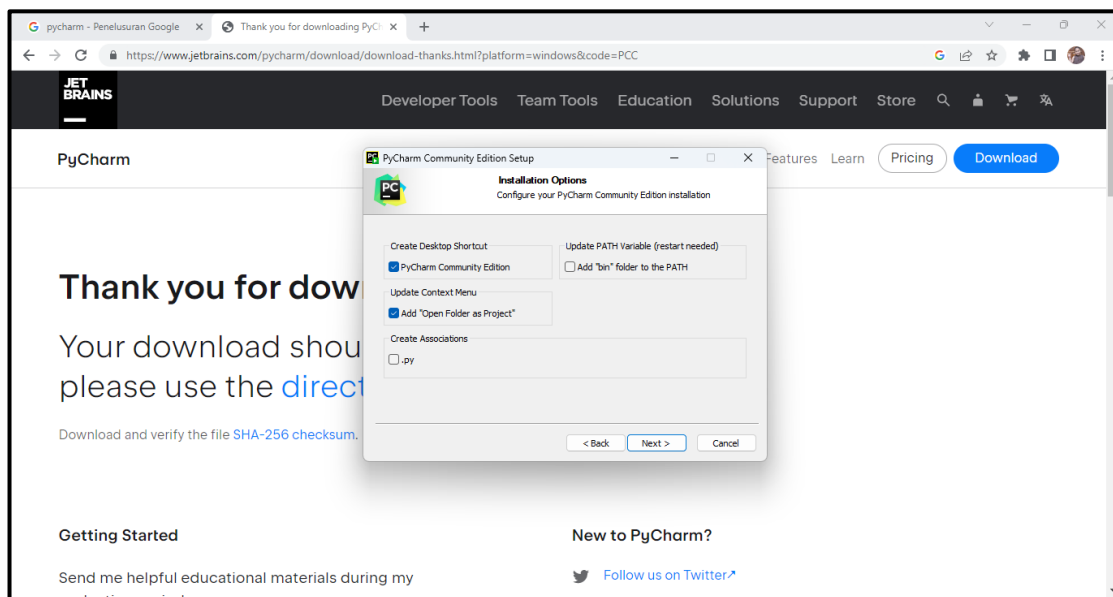
Simpan daftar tersebut dalam file `requirements.txt`, lalu jalankan:

```
bash
pip install -r requirements.txt
```

## BAB III PROSES PEMBUATAN APLIKASI

### 3.1 Pembuatan *Project* Baru di PyCharm

Langkah awal penelitian ini yaitu mengunduh perangkat lunak Pycharm untuk membuat kode, menganalisa program, menguji, dan membuat model deteksi dan klasifikasi tuberkulosis.



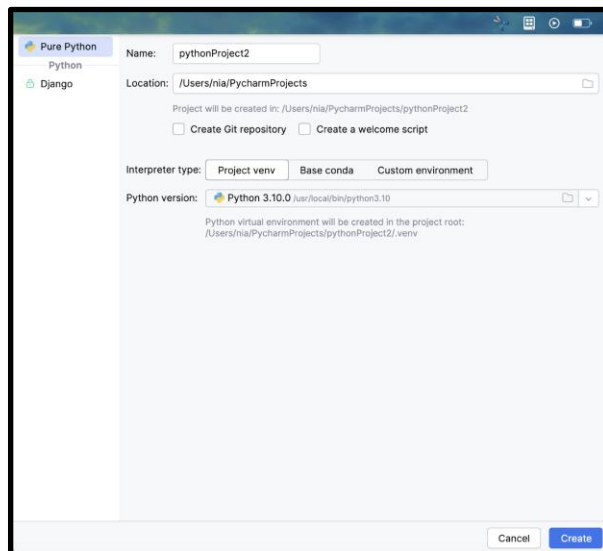
**Gambar 3. 1** Download Pycharm

Dari Gambar 3.1 menunjukkan saat penelitian melakukan proses *download* perangkat lunak PyCharm <https://www.jetbrains.com/pycharm/download/> untuk dilakukan instalasi pada komputer. Fungsi dari PyCharm pada penelitian ini digunakan sebagai alat bantu untuk menulis program yang dibuat. *PyCharm* adalah *Integrated Development Environment (IDE)* yang dikembangkan oleh JetBrains, dirancang khusus untuk bahasa pemrograman *Python*. IDE adalah aplikasi software yang menyediakan seperangkat alat untuk membantu pengembangan software. IDE berfungsi untuk merampingkan seluruh proses pengembangan software dengan menggabungkan berbagai alat dan fungsi ke dalam satu interface yang terpadu. Dengan IDE, *programmer* dan developer dapat melakukan koding, pengeditan, pengujian, *debugging*, dan *deployment* dari satu *platform* saja.



Gambar 3. 2 Membuka Aplikasi Pycharm

Dari Gambar 3.2 menunjukkan saat penelitian memulai *project* dengan membuka aplikasi PyCharm yang sudah terpasang di komputer. Ada banyak sekali faktor yang menjadikan PyCharm salah satu lingkungan pengembangan terintegrasi terlengkap dan komprehensif untuk bekerja dengan bahasa pemrograman Python. PyCharm hadir dengan sejumlah besar modul, paket, dan alat untuk mempercepat pengembangan Python sambil mengurangi upaya yang diperlukan untuk melakukan hal yang sama secara besar-besaran secara bersamaan. Selanjutnya, PyCharm dapat dikustomisasi sesuai dengan kebutuhan pengembangan, dan panggilan preferensi pribadi.



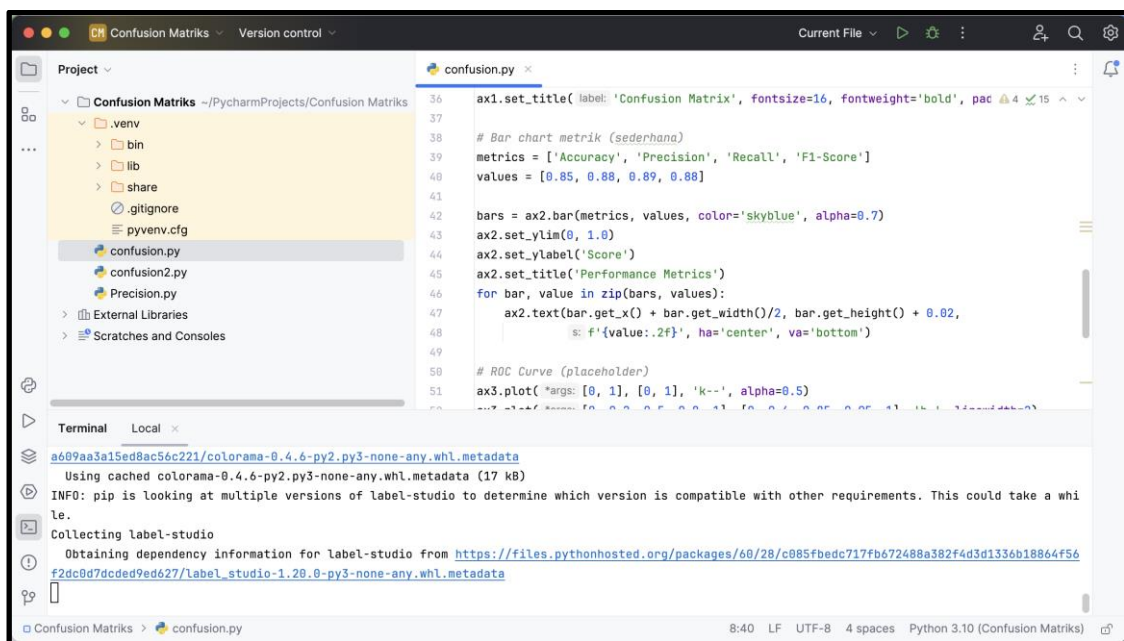
Gambar 3.3 Membuka Aplikasi Pycharm

Dari Gambar 3.3 menunjukkan saat penelitian memulai *project* dengan membuka aplikasi PyCharm yang sudah terpasang di komputer. Ada banyak sekali faktor yang menjadikan PyCharm salah satu lingkungan pengembangan terintegrasi terlengkap dan komprehensif untuk bekerja dengan bahasa pemrograman Python. PyCharm hadir dengan sejumlah besar

modul, paket, dan alat untuk mempercepat pengembangan Python sambil mengurangi upaya yang diperlukan untuk melakukan hal yang sama secara besar-besaran secara bersamaan. Selanjutnya, PyCharm dapat dikustomisasi sesuai dengan kebutuhan pengembangan, dan panggilan preferensi pribadi.

### 3.2 Pembuatan Label Studio dengan YOLO

*Label Studio* adalah alat pelabelan data sumber terbuka. *Label Studio* memungkinkan penelitian memberi *label* tipe data seperti *audio*, teks, gambar, *video*, dan deret waktu dengan tampilan yang sederhana dan lugas serta mengekspor ke berbagai *format model*. *Label Studio* dapat digunakan untuk menyiapkan data mentah atau menyempurnakan data pelatihan yang ada untuk mendapatkan *model Machine Learning* yang lebih akurat. Gambar 3.4 menampilkan instalasi *Label Studio* melalui PyCharm. Adapun perintah yang dilakukan adalah ‘*pip install label-studio*’.



Gambar 3.4 Instalasi Label Studio

Hubungan antara Label Studio dan YOLO (You Only Look Once) dalam praktik rekayasa visi komputer merupakan sebuah sinergi fungsional yang bersifat bidirectional, membentuk alur kerja siklik yang saling menguatkan antara penyediaan data latih dan pengoptimalan model. Secara fundamental, Label Studio berperan sebagai platform anotasi data yang menyediakan infrastruktur bagi pembuatan ground truth, sementara YOLO berfungsi sebagai arsitektur model deteksi objek yang sangat efisien untuk inferensi waktu nyata; jalinan

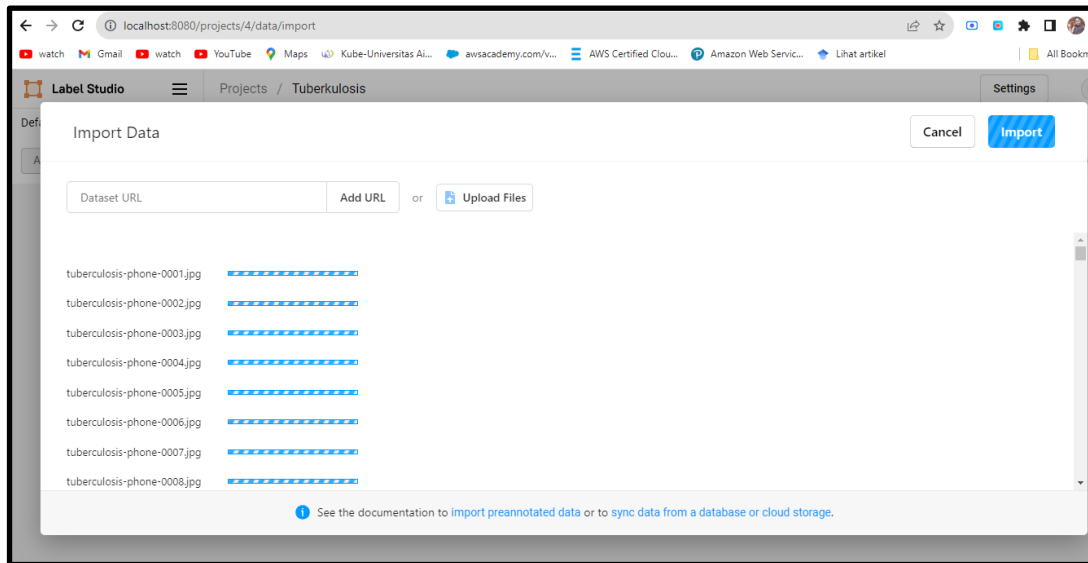
keduanya tidak sekadar bersifat administratif, melainkan terintegrasi secara teknis melalui mekanisme back-end pembelajaran mesin.

Pertama, integrasi tersebut memungkinkan Label Studio memanfaatkan kemampuan inferensi YOLO untuk menghasilkan pra-anotasi (pre-annotation) secara otomatis. Dengan mengaktifkan ML Backend khusus YOLO, setiap citra yang diunggah ke proyek akan segera diproses oleh model terlatih untuk menghasilkan kotak pembatas (bounding boxes), poligon segmentasi, atau bahkan titik kunci (keypoints) yang menjadi prediksi awal. Alih-alih menggambar anotasi dari nol, tenaga anotator hanya perlu melakukan verifikasi dan koreksi terhadap hasil prediksi tersebut, sehingga secara drastis mengurangi beban kerja manual dan mempercepat siklus pelabelan, terutama pada dataset berskala besar. Kedua, keterkaitan ini tampak jelas pada proses ekspor data dan pelatihan ulang, di mana Label Studio menyediakan dukungan native untuk mengekspor seluruh dataset anotasi ke dalam format direktori yang kompatibel dengan YOLO (seperti format .txt untuk koordinat kotak pembatas yang telah dinormalisasi). Format ini dapat langsung digunakan sebagai korpus masukan untuk melatih atau menyempurnakan (fine-tuning) varian YOLO terbaru tanpa memerlukan konversi sintaksis yang rumit, sehingga menjembatani kesenjangan antara tahap anotasi dan tahap pelatihan model.

Ketiga, relasi ini semakin diperdalam oleh konsep evaluasi berkelanjutan dan iterasi adaptif yang dikenal dengan paradigma human-in-the-loop. Model YOLO yang telah dilatih dengan data dari Label Studio dapat diintegrasikan kembali ke dalam platform untuk menguji performanya pada sampel data baru yang belum terlabel. Hasil prediksi yang memiliki tingkat ketidakpastian tinggi dapat diprioritaskan untuk dikoreksi oleh anotator manusia, sehingga data yang dihasilkan menjadi semakin representatif dan akurat. Secara keseluruhan, Label Studio dan YOLO tidak beroperasi sebagai entitas yang terisolasi; melainkan, Label Studio berfungsi sebagai pabrik data kualitas tinggi yang menjadi bahan bakar peningkatan akurasi YOLO, sementara YOLO bertindak sebagai katalis efisiensi yang memperkaya produktivitas Label Studio. Alur kerja terintegrasi ini membentuk suatu roda pengungkit (flywheel effect) di mana peningkatan kualitas anotasi mendorong peningkatan performa model, dan peningkatan performa model, pada gilirannya, menyederhanakan serta memperhalus proses anotasi pada iterasi berikutnya.

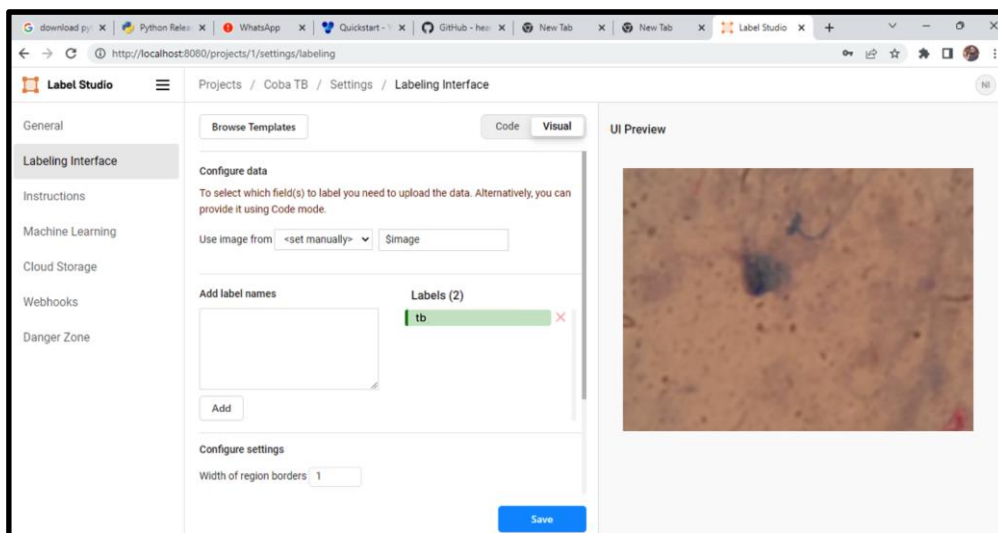
Pelabelan data dalam *Machine Learning* (ML) adalah proses pemberian label pada subkumpulan data berdasarkan karakteristiknya. Pelabelan data mengambil kumpulan data yang tidak berlabel dan menambah setiap bagian data dengan label atau tag yang informatif. Sebelum melakukan proses pelabelan, maka penelitian melakukan *import data*

Tuberkulosis yang didapatkan dari Patologi Klinis Dr. Soetomo Surabaya sebanyak 1265 gambar. Gambar 3.5 menampilkan proses *Import Data* Tuberkulosis ke YOLO.



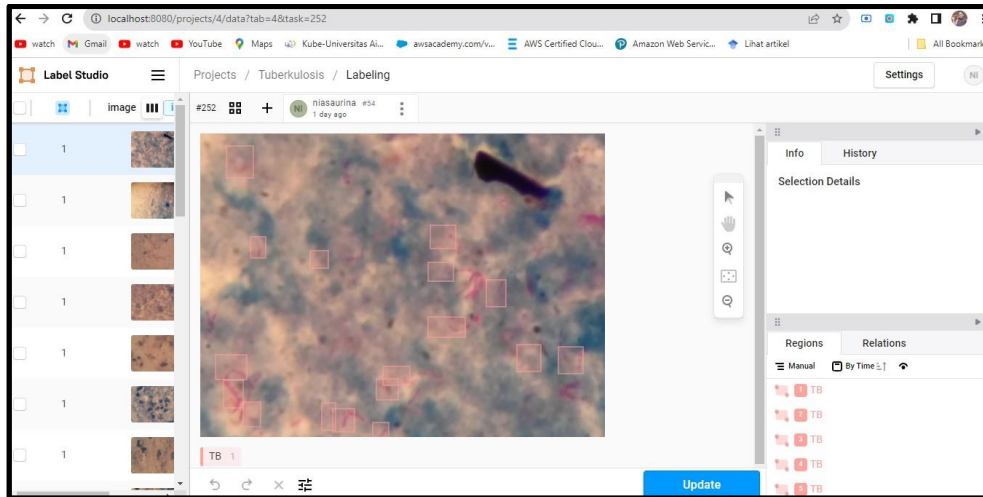
**Gambar 3.5** *Import* Gambar Tuberkulosis

Pelabelan data merupakan langkah penting dalam tugas pembelajaran mesin yang diawasi. *Garbage In Garbage Out* adalah ungkapan yang umum digunakan dalam komunitas pembelajaran mesin, yang berarti kualitas data pelatihan menentukan kualitas model. Hal yang sama juga berlaku untuk anotasi yang digunakan untuk pelabelan data. Untuk proses pemberian *label*, maka pengaturan YOLO perlu diatur “*Labelling Interface*” seperti yang dapat dilihat pada Gambar 3.6.



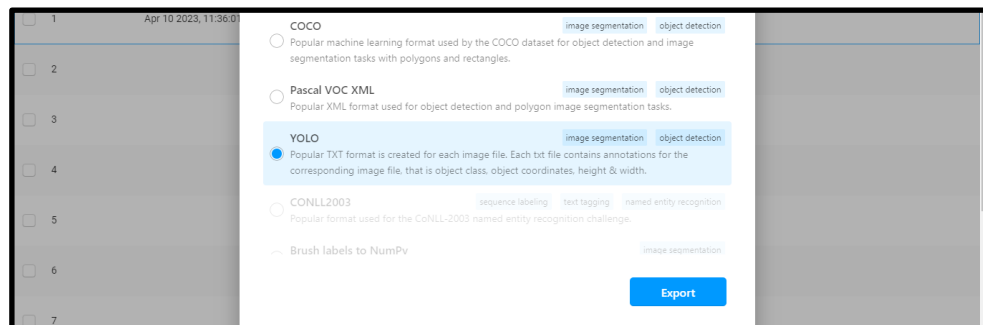
**Gambar 3.6** Menghapus *Label*

Proses memberikan *Label* dilakukan manual dengan memberi kotak pembatas pada posisi di setiap gambar Tuberkulosis, seperti pada Gambar 3.7. Proses memberikan *label* dilakukan satu per satu sejumlah dataset yang dibutuhkan.



**Gambar 3.7** Memberi Label pada Gambar TB

Setelah melakukan Pelabelan data, maka sekumpulan data tersebut diunduh untuk diproses dengan PyCharm, seperti yang dapat dilihat pada Gambar 3.8. Sekumpulan data untuk selanjutnya disebut dengan nama *dataset*, yang berisi gambar yang telah dilakukan pelabelan *data bounding boxes*, dan *label* yang berisi Lokasi tuberkulosis dengan koordinat sumbu X dan sumbu Y berupa teks.



**Gambar 3.8** Unduh Dataset YOLO

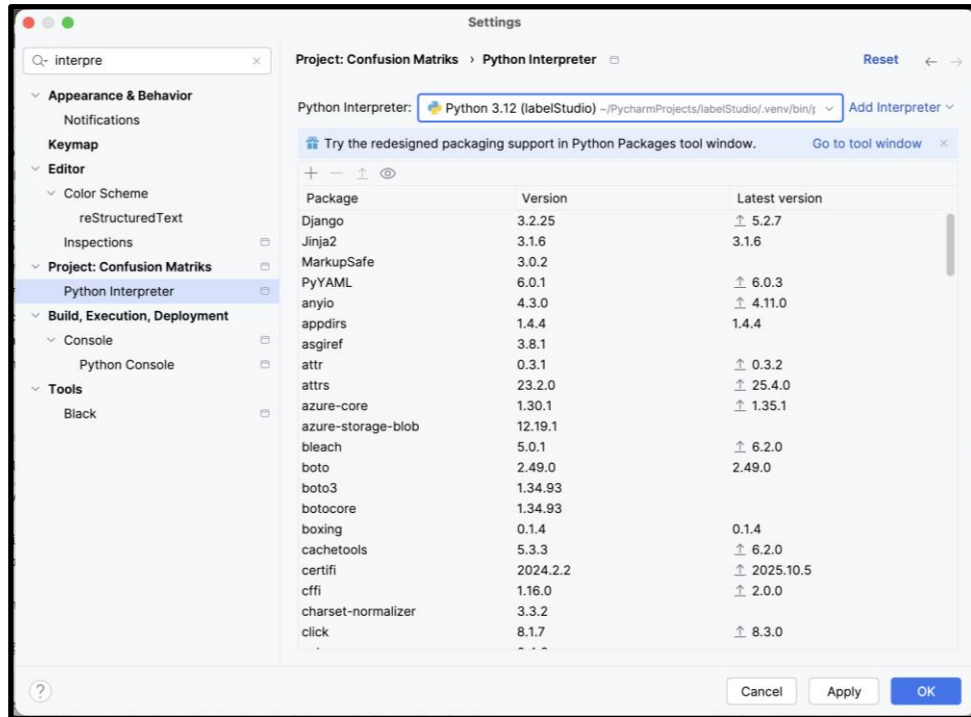
Untuk penelitian klasifikasi tuberkulosis menggunakan YOLO, tidak ada angka absolut atau jumlah minimal yang bersifat wajib dan universal yang dapat dijadikan patokan baku. Jumlah dataset yang dibutuhkan sangat ditentukan oleh sejumlah variabel penelitian, di antaranya adalah versi arsitektur YOLO yang digunakan, kompleksitas citra medis (misalnya, apakah menggunakan foto rontgen dada atau citra mikroskopis apusan dahak), variabilitas data, serta target akurasi dan generalisasi yang ingin dicapai oleh model. Meskipun demikian, berdasarkan kajian literatur dan praktik umum dalam visi komputer, terdapat rentang kuantitatif yang dapat dijadikan acuan. Sebagai ilustrasi, penelitian yang bersifat eksploratif atau proof-of-concept

terkadang hanya menggunakan sekitar 300 hingga 1.000 gambar per kelas untuk menunjukkan kelayakan awal metode. Namun, untuk menghasilkan model yang sungguh-sungguh andal dan memiliki nilai klinis yang dapat dipublikasikan, para peneliti umumnya mengupayakan dataset yang jauh lebih besar. Beberapa studi yang terbit di jurnal bereputasi menggunakan korpus data berjumlah 1.265 hingga 6.600 gambar, sementara dataset publik skala besar menyediakan 11.200 citra dan dataset bahkan mencapai 31.300 citra mikroskopis.

Dari perspektif teknis pembelajaran mendalam, ukuran dataset ideal juga sering diukur dari jumlah total instance anotasi (kotak pembatas), bukan semata-mata jumlah gambarnya. Banyak panduan yang merekomendasikan agar setiap kategori kelas memiliki setidaknya 1.000 hingga 1.500 instance anotasi, dengan total anotasi di seluruh dataset tidak kurang dari 10.000 kotak pembatas untuk menghindari fenomena *overfitting*. Oleh karena itu, dalam merancang penelitian, seorang ilmuwan data sebaiknya tidak terpaku pada angka minimum yang kaku, melainkan menganut prinsip "semakin banyak dan semakin beragam data, semakin baik", sembari memastikan kualitas anotasi yang presisi. Apabila keterbatasan akses terhadap data medis menjadi kendala, strategi kompensasi seperti augmentasi data (*data augmentation*) yang agresif dan pemanfaatan *transfer learning* dari model YOLO yang telah terlatih pada dataset umum (seperti COCO) merupakan langkah-langkah wajib yang dapat secara signifikan meningkatkan performa model meskipun dengan ukuran dataset yang relatif terbatas.

### **3.3 Pemrosesan Data Menggunakan Pycharm**

Pemrosesan data diawali dengan dataset yang telah di unduh dari YOLO, selanjutnya penelitian menggunakan *software* PyCharm untuk melakukan pemrosesan data. Gambar 3.9 menunjukkan penelitian saat pertama kali membuka PyCharm, dan menambahkan Platform *Ultralytics* ke dalam program, dikarenakan pada platform ini terdapat library-library dengan menggunakan Bahasa pemrograman Python, yang mendukung proses pembelajaran *machine learning*.



**Gambar 3.9** Mengaktifkan *Library* Data Python

### 3.4 Pembahasan YOLO

YOLO (*You Only Look Once*) merupakan jaringan untuk mendeteksi objek. Tugas deteksi objek terdiri dalam menentukan lokasi pada gambar dimana objek tertentu hadir serta mengklasifikasikan objek tersebut. Jadi sederhananya ada gambar sebagai input, lalu dapatkan vektor kotak pembatas dan prediksi kelas dalam output (Alif & Hussain, 2024; (I Made, 2023; Deden Miftah Fauzi et al., 2024; Sarosa & Muna, 2021); [4]. Penelitian ini menggunakan YOLO versi 5 dan YOLO versi 8. Secara umum untuk proses ketiga versi tersebut sama yaitu melakukan Instal YOLO. Adapun source code untuk install YOLO dapat dilihat pada Tabel 3.1. Instruksi yang digunakan adalah “pip install yolo”, dan untuk versi YOLO menyesuaikan dengan source code yang ditulis.

**Tabel 3.4** Instal YOLO

```
#Instal YOLO versi 5
pip install yolov5
#Instal YOLO versi 8
pip install yolov8
```

Kemudian melakukan pengaturan dengan ekstensi file \*.yaml, yang dapat dilihat pada Tabel 3.2 Fungsi utamanya adalah sebagai file konfigurasi. File konfigurasi YAML dapat memuat semua informasi yang dibutuhkan untuk menjalankan proses training data [5], [6]. Adapun

instruksi yang dibutuhkan adalah menulis folder path yang digunakan untuk menyimpan dataset tuberculosis. Kemudian menulis folder train untuk proses hasil training, dan value untuk menyimpan hasil training sesuai dengan jumlah partisi, dan folder test untuk menyimpan hasil pengujian setelah training. Selanjutnya diberikan nilai “0” dengan nama TBbacillus untuk memberikan Lokasi pada gambar sesuai dengan Lokasi tuberculosis.

**Tabel 3.2** Pengaturan Data YAML

```
path: tb # dataset root dir
train: images/train
val: images/train
test: # test images (optional)

# Classes
names:
  0: TBbacillus
```

YOLO (*You Only Look Once*) adalah algoritma *deep learning* yang digunakan untuk deteksi objek dalam gambar secara *real-time*. Dalam konteks deteksi dan klasifikasi gambar tuberculosis, YOLO dapat diterapkan untuk mengidentifikasi lokasi tuberculosis.

YOLO (*You Only Look Once*) merupakan jaringan untuk mendeteksi objek. Tugas deteksi objek terdiri dalam menentukan lokasi pada gambar dimana objek tertentu hadir serta mengklasifikasikan objek tersebut. Jadi sederhananya ada gambar sebagai input, lalu dapatkan vektor kotak pembatas dan prediksi kelas dalam output (Alif & Hussain, 2024; (I Made, 2023; Deden Miftah Fauzi et al., 2024; Sarosa & Muna, 2021); [4]. Penelitian ini menggunakan YOLO versi 5 dan YOLO versi 8. Secara umum untuk proses ketiga versi tersebut sama yaitu melakukan Instal YOLO. Adapun source code untuk install YOLO dapat dilihat pada Tabel 3.3. Instruksi yang digunakan adalah “pip install yolo”, dan untuk versi YOLO menyesuaikan dengan source code yang ditulis.

**Tabel 3.3** Instal YOLO

```
#Instal YOLO versi 5
pip install yolov5
#Instal YOLO versi 8
pip install yolov8
```

Kemudian melakukan pengaturan dengan ekstensi file \*.yaml, yang dapat dilihat pada Tabel 3.4 Fungsi utamanya adalah sebagai file konfigurasi. File konfigurasi YAML dapat memuat semua informasi yang dibutuhkan untuk menjalankan proses training data [5], [6]. Adapun instruksi yang dibutuhkan adalah menulis folder path yang digunakan untuk menyimpan dataset tuberculosis. Kemudian menulis folder train untuk proses hasil training, dan value untuk menyimpan hasil training sesuai dengan jumlah partisi, dan folder test untuk menyimpan hasil pengujian setelah training. Selanjutnya diberikan nilai “0” dengan nama TBbacillus untuk memberikan Lokasi pada gambar sesuai dengan Lokasi tuberculosis.

**Tabel 3.4** Pengaturan Data YAML

```
path: tb # dataset root dir
train: images/train
val: images/train
test: # test images (optional)

# Classes
names:
  0: TBbacillus
```

YOLO (*You Only Look Once*) adalah algoritma *deep learning* yang digunakan untuk deteksi objek dalam gambar secara *real-time*. Dalam konteks deteksi dan klasifikasi gambar tuberculosis, YOLO dapat diterapkan untuk mengidentifikasi lokasi tuberculosis.

### 3.5 Apa Itu *Multimodal Deep Learning*

*Multimodal Deep Learning* adalah teknik pembelajaran mendalam yang bisa memproses dan menganalisis berbagai jenis data [7]. Multimodal Deep Learning adalah pendekatan yang mengintegrasikan berbagai jenis data (modalitas) seperti teks, gambar, audio, video, dan data sensor untuk menghasilkan representasi yang kaya dan menyeluruh. Sistem ini bekerja dengan memanfaatkan jaringan saraf dalam (*deep neural networks*) untuk mengolah setiap modalitas secara terpisah, menggunakan arsitektur yang sesuai seperti CNN untuk gambar atau RNN untuk teks [8]. Setelah fitur dari masing-masing modalitas diekstraksi, proses penggabungan model (*fusion*) dilakukan, baik melalui teknik awal (*early fusion*), teknik tengah (*intermediate fusion*), atau teknik akhir (*late fusion*). Hasil *fusion* ini memungkinkan model untuk menangkap hubungan kompleks antar modalitas, memperbaiki prediksi dengan memanfaatkan informasi lintas modalitas yang saling melengkapi [9]. Selanjutnya, model dilatih secara end-to-end untuk tugas deteksi tuberculosis.

**Tabel 3.5** *Pseudocode Multimodal Deep Learning*

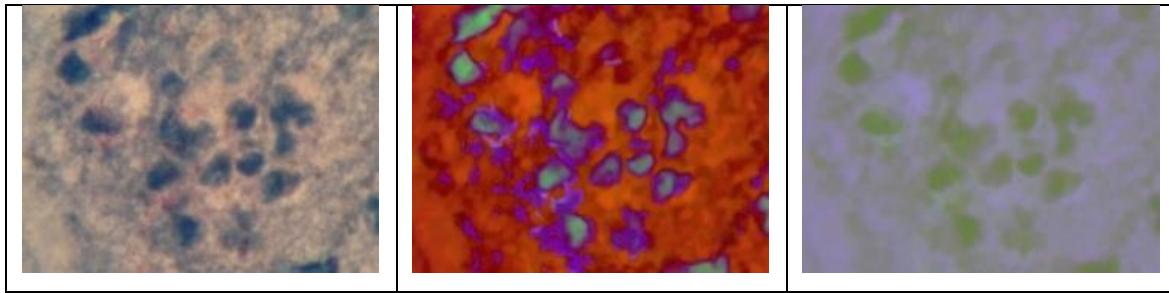
Tentukan folder tempat gambar asli disimpan.  
Ambil daftar nama semua file gambar dari folder tersebut.

Untuk setiap file gambar:

- a. Tampilkan nama file.
- b. Baca gambar dari file.
- c. Konversi gambar dari format BGR ke format HSV.
  - Simpan hasil konversi ke folder "HSV".
- d. Konversi gambar dari format BGR ke format CIE LAB.
  - Simpan hasil konversi ke folder "Cie\_Lab".

Tabel 3.5 menjelaskan *source code* untuk melakukan multimodal deep learning, diawali dengan menuliskan library python yang dibutuhkan. Kemudian menuliskan folder tempat menyimpan dataset dan menuliskan source code “`hsv_img = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)`” untuk melakukan konversi image RGB ke HSV. Untuk melakukan konversi image RGB ke CIE\_LAB diperlukan source code “`lab_img = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)`”.

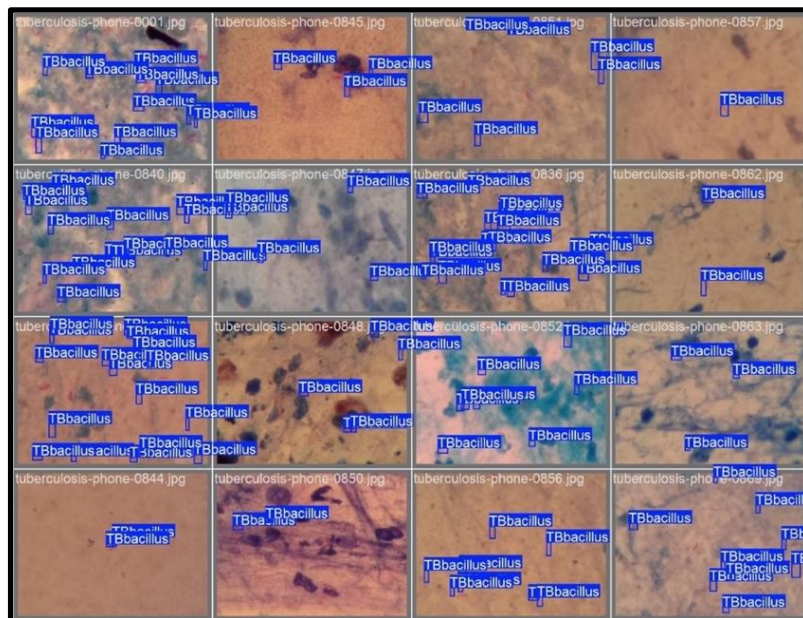
Gambar 3.10 menunjukkan gambar tuberkulosis dengan channel warna RGB, HSV dan CIELAB. Ruang warna HSV (Hue, Saturation, Value) lebih mirip dengan ruang warna RGB di mana manusia mendeskripsikan sensasi warna dan memahami warna. Gambar berwarna dapat disegmentasi lebih baik dalam model ruang warna HSV dibandingkan model warna lainnya [40]. CIE  $L^*a^*b^*$  (CIELAB) adalah ruang warna terlengkap yang menggambarkan semua warna yang terlihat oleh mata manusia dan diciptakan sebagai model independen perangkat untuk digunakan sebagai referensi. Tiga koordinat CIELAB mewakili kecerahan warna ( $L^* = 0$  menghasilkan hitam dan  $L^* = 100$  menunjukkan putih menyebar; putih specular mungkin lebih tinggi), posisinya antara merah/magenta dan hijau ( $a^*$ , nilai negatif menunjukkan hijau sedangkan nilai positif menunjukkan magenta) dan posisinya antara kuning dan biru ( $b^*$ , nilai negatif menunjukkan biru dan nilai positif menunjukkan kuning).



**Gambar 3.10** *Mycobacterium Tuberculosis* pada Kanal Warna RGB, HSV dan CIE LAB

### 3.6 Menghitung IUATLD dengan *Multimodal Deep Learning*

*Multimodal Deep Learning* adalah salah satu metode klasifikasi mesin learning yang bertujuan untuk melakukan mengelompokan data tuberkulosis dari tiga kanal yang berbeda (multimodal), dan hasil klasifikasi dapat dilihat pada Gambar 3.11.



**Gambar 3.11** Hasil Klasifikasi menggunakan *Multimodal Deep Learning*

Pada penelitian ini *Multimodal Deep Learning* menggunakan YOLOv8. Berikut ini Confusion Matrix dari proses klasifikasi menggunakan YOLO versi 8. Terdapat empat kali pengujian dengan data partisi 60:40, 70:30, 80:20 dan 90:10.

	True Tuberculosis	True background
Predicted Tuberculosis	<b>605</b>	<b>196</b>
Predicted background	<b>154</b>	<b>563</b>

**Gambar 3.12** Confusion Matriks Hasil Klasifikasi *Multimodal Deep Learning* Data Partisi 60:40

Gambar 3.12 memperlihatkan pada hasil klasifikasi Multimodal Deep Learning pada data partisi 60:40, terdapat True Positif sebanyak 605, False Positif sebanyak 196, False Negatif sebanyak 154 dan True Negatif sebanyak 563. Sehingga nilai presisi, akurasi, recall dan mAP bisa dihitung sebagai berikut:

a.  $Precision = \frac{TP}{TP+FP} \times 100\%$

$$Precision = \frac{605}{605+196} \times 100\%$$

$$Precision = 75,5305868\%$$

b.  $Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$

$$Akurasi = \frac{605 + 563}{605 + 563 + 196 + 154} \times 100\%$$

$$Akurasi = 76,9433465\%$$

c.  $Recall = \frac{TP}{TP+FN} \times 100\%$

$$Recall = \frac{605}{605 + 154} \times 100\%$$

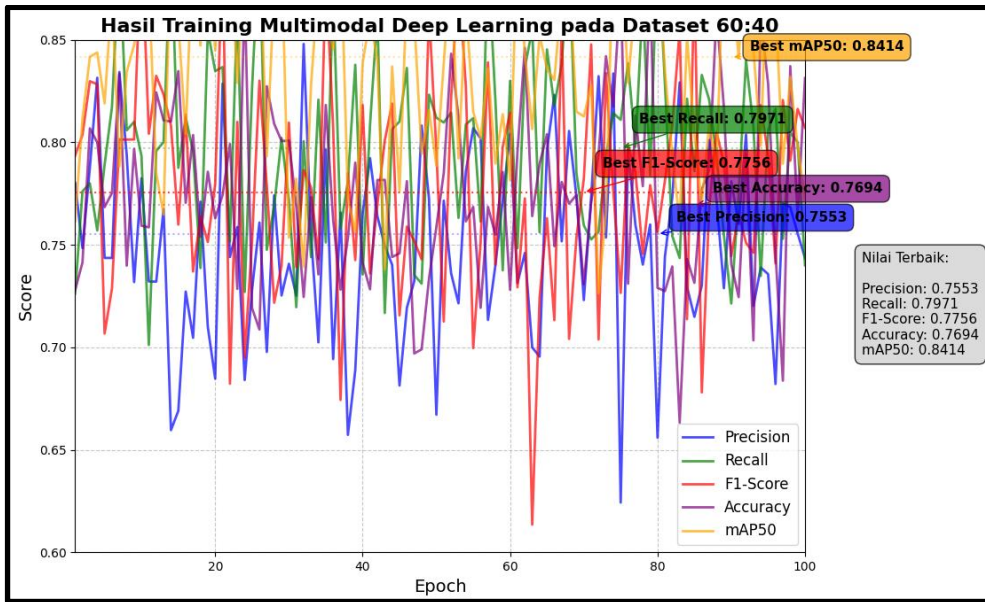
$$Recall = 79,7101449\%$$

a.  $F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\%$

$$F1Score = \frac{2 \times 0,755305868 \times 0,797101449}{0,755305868 + 0,797101449} \times 100\%$$

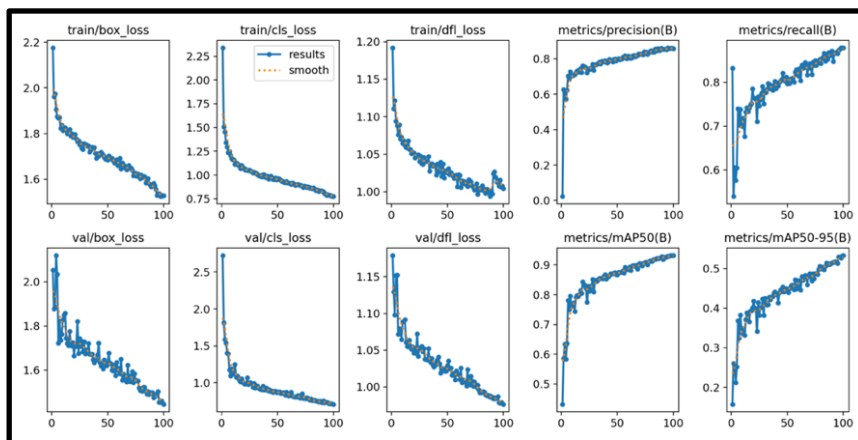
$$F1Score = 77,5641026\%$$

- d. Nilai mAP didapatkan dari nilai training klasifikasi Multimodal Deep Learning sebanyak 100 epoch. Nilai mAP pada data partisi 60:40 sebesar 84,14%.



**Gambar 3.13** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning* pada Data Partisi 60:40

Gambar 3.13 menunjukkan Hasil Training Klasifikasi Tuberkulosis dengan metode *Multimodal Deep Learning* pada Data Partisi 60:40 dengan nilai Precision mencapai 75,53% yang ditunjukkan pada grafik berwarna biru, Recall mencapai 79,71% yang ditunjukkan pada grafik berwarna hijau, F1-Score mencapai 77,56% yang ditunjukkan pada grafik berwarna merah, Accuracy mencapai 76,94% yang ditunjukkan pada grafik berwarna ungu dan mAP50 sebesar 84,14% yang ditunjukkan pada grafik berwarna biru oranye.



**Gambar 3.14** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning*

Gambar 3.14 menunjukkan hasil training memiliki performa yang baik. Hal ini ditunjukkan terdapat peningkatan metrik performa (*precision*, *recall*, mAP50, dan mAP50-95) selama proses *training*.

**Tabel 3.6** Skor IUATLD Multimodal pada data partisi 60:40

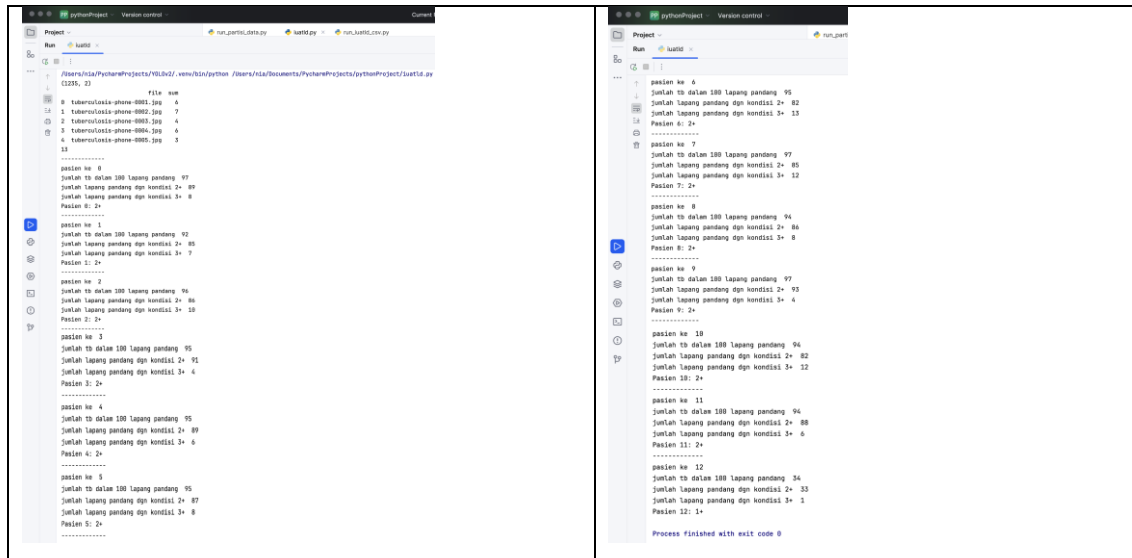
Pasien	Jumlah TB dalam 100 Lapang Pandang	Jumlah Lapang Pandang 2+	Jumlah Lapang Pandang 3+	Skor IUATLD
1	97	89	8	2+
2	92	85	7	2+
3	96	86	10	2+
4	95	91	4	2+
5	95	89	6	2+
6	95	87	8	2+
7	95	82	21	3+
8	97	85	12	2+
9	94	86	8	2+
10	97	93	4	2+
11	94	82	12	2+
12	94	88	6	2+
13	34	33	1	1+

Nilai MAPE didapatkan dari perbandingan data IUATLD yang sebenarnya dengan data temuan hasil klasifikasi. Hasil data temuan dapat dilihat pada Tabel 3.6 dapat dilihat pada data partisi 60:40, terdapat 6 pasien yang dideteksi dengan skor IUATLD bernilai 1+, 6 pasien yang dikenali dengan skor IUATLD bernilai 2+, dan 1 pasien yang dikenali dengan skor IUATLD bernilai 3+.

**Tabel 3.7** Nilai MAPE untuk *Multimodal Deep Learning* Data Partisi 60:40

Pasien	Jumlah TB	Jumlah TB yang dideteksi	Nilai MAPE (%)
Pasien 1	995	990	0,502
Pasien 2	1612	1607	0,310
Pasien 3	1799	1780	1,056
pasien 4	1812	1792	1,103
Pasien 5	2147	2130	0,791
Pasien 6	2158	735	65,940
Pasien 7	2171	514	76,324
Pasien 8	2160	531	75,416
Pasien 9	464	295	36,422
Pasien 10	2165	435	79,907
Pasien 11	2252	1534	31,882
Pasien 12	2209	2044	7,469
Pasien 13	366	328	10,382
<b>Average Nilai MAPE untuk Train 60</b>			29,808

Tabel 3.7 menjelaskan Nilai MAPE untuk metode *Multimodal Deep Learning* pada Data Partisi 60:40, dimana perhitungan ini menggunakan rumus MAPE pada (2.5). Rata-rata Nilai MAPE untuk Train 60 sebesar 29,808% menunjukkan hasil peramalan layak (cukup baik).



**Gambar 3.15** Tampilan PyCharm Hasil *Testing MAPE Multimodal Deep Learning* Data Partisi 60:40

Gambar 3.15 merupakan tampilan PyCharm untuk hasil testing MAPE dengan metode *Multimodal Deep Learning* pada Data Partisi 60:40 yang menunjukkan bahwa *Multimodal Deep Learning* dapat mengenali 13 pasien.

	True Tuberculosis	True background
Predicted Tuberculosis	<b>594</b>	<b>178</b>
Predicted background	<b>135</b>	<b>232</b>

**Gambar 3.16** *Confusion Matriks* Hasil Klasifikasi *Multimodal Deep Learning* Data Partisi 70:30

Gambar 3.16 memperlihatkan pada hasil klasifikasi *Multimodal Deep Learning* pada data partisi 70:30, terdapat True Positif sebanyak 594, False Positif sebanyak 178, False Negatif

sebanyak 135 dan True Negatif sebanyak 232. Sehingga nilai presisi, akurasi, recall dan mAP bisa dihitung sebagai berikut:

a.  $Precision = \frac{TP}{TP+FP} \times 100\%$

$$Precision = \frac{594}{594+178} \times 100\%$$

$$Precision = 76,943\%$$

b.  $Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$

$$Akurasi = \frac{594 + 232}{594 + 232 + 178 + 135} \times 100\%$$

$$Akurasi = 72,5198\%$$

c.  $Recall = \frac{TP}{TP+FN} \times 100\%$

$$Recall = \frac{594}{594 + 135} \times 100\%$$

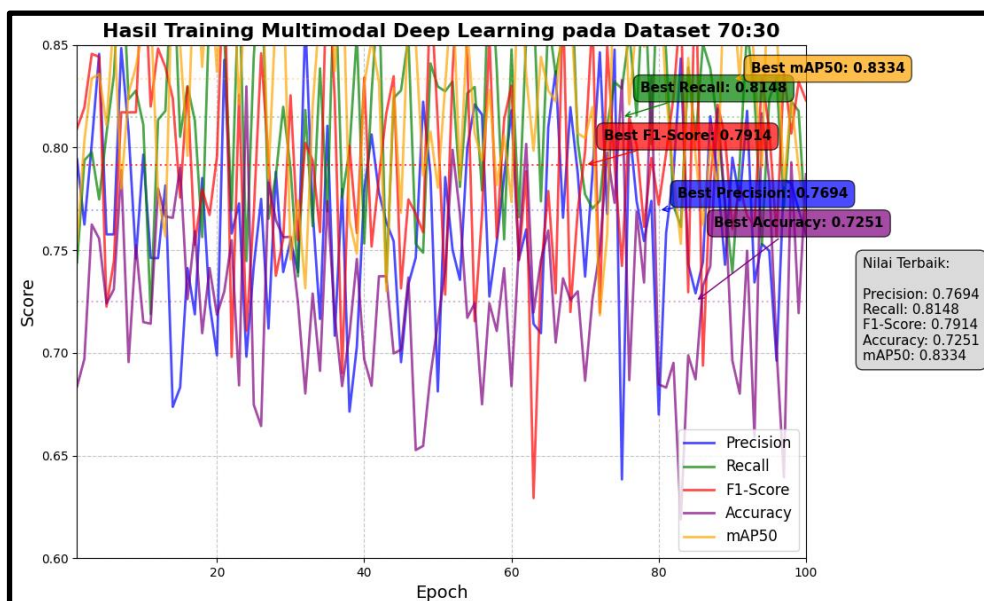
$$Recall = 81,4815\%$$

d.  $F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\%$

$$F1Score = \frac{2 \times 0,76943 \times 0,814815}{0,76943 + 0,814815} \times 100\%$$

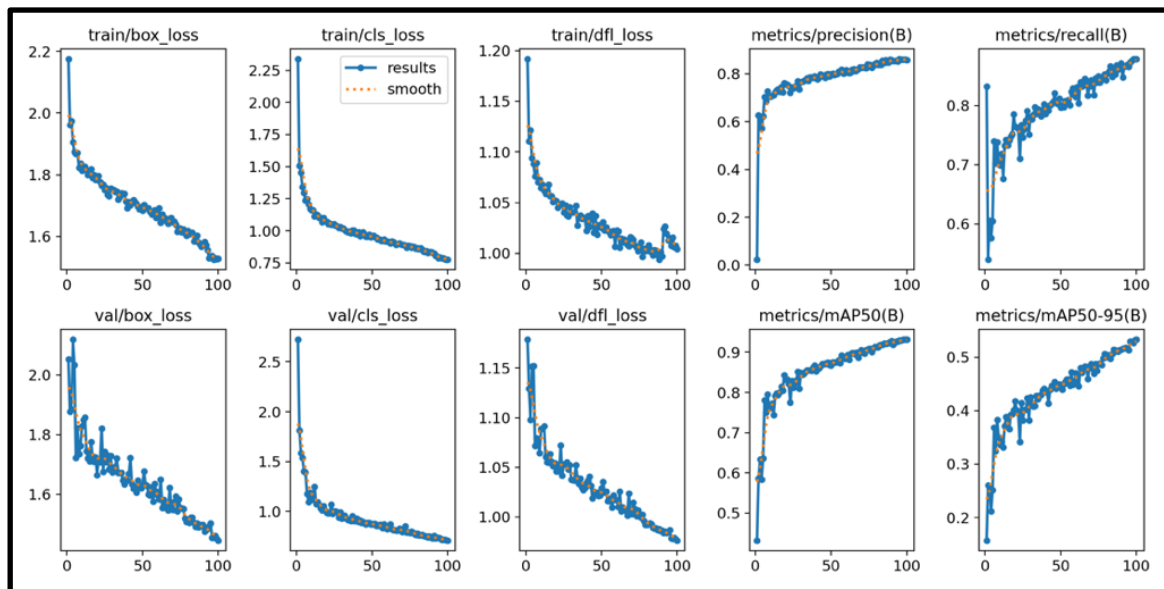
$$F1Score = 79,1472352\%$$

e. Nilai mAP didapatkan dari nilai training klasifikasi Multimodal Deep Learning sebanyak 100 epoch. Nilai mAP pada data partisi 70:30 sebesar 83,344%.



**Gambar 3.17** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning* pada Data Partisi 70:30

Gambar 3.17 menunjukkan Hasil *Training* Klasifikasi Tuberkulosis dengan metode *Multimodal Deep Learning* pada Data Partisi 70:30 dengan nilai Precision mencapai 76,94% yang ditunjukkan pada grafik berwarna biru, Recall mencapai 81,46% yang ditunjukkan pada grafik berwarna hijau, F1-Score mencapai 79,14% yang ditunjukkan pada grafik berwarna merah, Accuracy mencapai 72,51% yang ditunjukkan pada grafik berwarna ungu dan mAP50 sebesar 83,34% yang ditunjukkan pada grafik berwarna biru oranye.



**Gambar 3.18** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning*

Gambar 3.18 menunjukkan hasil training memiliki performa yang baik. Hal ini ditunjukkan oleh penurunan nilai loss (box loss, classification loss, dan distribution focal loss) baik pada data training maupun validasi, serta peningkatan metrik performa (precision, recall, mAP50, dan mAP50-95) selama proses training.

Nilai MAPE didapatkan dari perbandingan data IUATLD yang sebenarnya dengan data temuan hasil klasifikasi. Tabel 3.8 memperlihatkan pada hasil perhitungan Multimodal Deep Learning pada data partisi 60:40, terdapat 6 pasien yang dideteksi dengan skor IUATLD bernilai 1+, kemudian terdapat 1 pasien yang dikenali dengan skor IUATLD bernilai 2+, dan terdapat 6 pasien yang dikenali dengan skor IUATLD bernilai 3+.

**Tabel 3.8** Skor IUATLD Multimodal pada data partisi 70:30

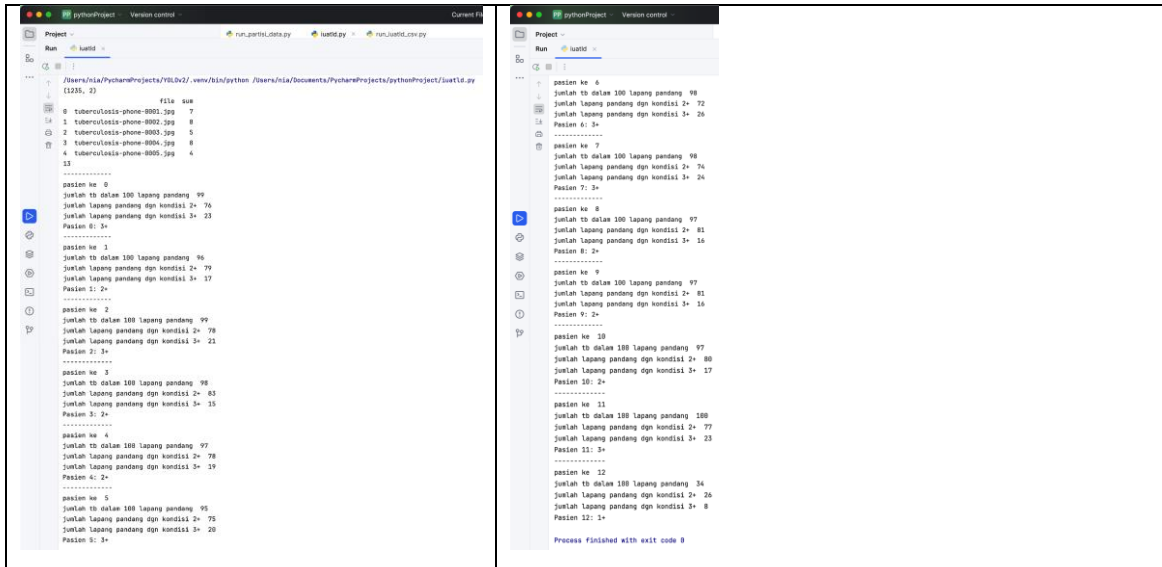
Pasien	Jumlah TB dalam 100 Lapang Pandang	Jumlah Lapang Pandang 2+	Jumlah Lapang Pandang 3+	Skor IUATLD
1	99	76	23	3+
2	96	79	17	2+
3	99	78	21	3+
4	98	83	15	2+
5	97	78	19	2+
6	95	75	20	3+
7	98	72	26	3+
8	98	74	24	3+
9	97	81	16	2+
10	97	81	16	2+
11	97	80	17	2+
12	100	77	23	3+
13	34	26	8	1+

Tabel 3.9 menjelaskan Nilai MAPE untuk metode *Multimodal Deep Learning* pada Data Partisi 70:30, dimana perhitungan ini menggunakan rumus MAPE pada (2.5). Rata-rata Nilai MAPE untuk Train 70 sebesar 31,203% menunjukkan hasil peramalan layak (cukup baik).

**Tabel 3.9** Nilai MAPE untuk *Multimodal Deep Learning* Data Partisi 70:30

Pasien	Jumlah TB	Jumlah TB yang dideteksi	Nilai MAPE (%)
Pasien 1	995	990	0,502
Pasien 2	1612	1610	0,124
Pasien 3	1799	1425	20,789
pasien 4	1812	870	51,986
Pasien 5	2147	1615	24,778
Pasien 6	2158	927	57,043
Pasien 7	2171	672	69,046
Pasien 8	2160	710	67,129
Pasien 9	464	324	30,172
Pasien 10	2165	1653	23,648
Pasien 11	2252	1567	30,417
Pasien 12	2209	2035	7,876
Pasien 13	366	285	22,131
<b>Average Nilai MAPE untuk Train 70</b>			<b>31,203</b>

Gambar 3.19 merupakan tampilan PyCharm untuk hasil testing MAPE dengan metode *Multimodal Deep Learning* pada Data Partisi 70:30 yang menunjukkan bahwa *Multimodal Deep Learning* dapat mengenali 13 pasien.



**Gambar 3.19** Tampilan PyCharm Hasil Testing MAPE Multimodal Deep Learning Data Partisi 70:30

Gambar 3.20 memperlihatkan pada hasil klasifikasi *Multimodal Deep Learning* pada data partisi 80:20, terdapat True Positif sebanyak 431, False Positif sebanyak 79, False Negatif sebanyak 69 dan True Negatif sebanyak 180. Sehingga nilai presisi, akurasi, recall dan mAP bisa dihitung sebagai berikut:

	True Tuberculosis	True background
Predicted Tuberculosis	<b>431</b>	<b>79</b>
Predicted background	<b>69</b>	<b>180</b>

**Gambar 3.20** *Confusion Matriks* Hasil Klasifikasi *Multimodal Deep Learning* Data Partisi 80:20

a. 
$$\text{Precision} = \frac{TP}{TP+FP} \times 100\%$$

$$\text{Precision} = \frac{431}{431+79} \times 100\%$$

$$\text{Precision} = 84,509\%$$

$$b. \text{ Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$\text{Akurasi} = \frac{431 + 180}{431 + 180 + 79 + 69} \times 100\%$$

$$\text{Akurasi} = 80,5007\%$$

$$c. \text{ Recall} = \frac{TP}{TP+FN} \times 100\%$$

$$\text{Recall} = \frac{431}{431 + 69} \times 100\%$$

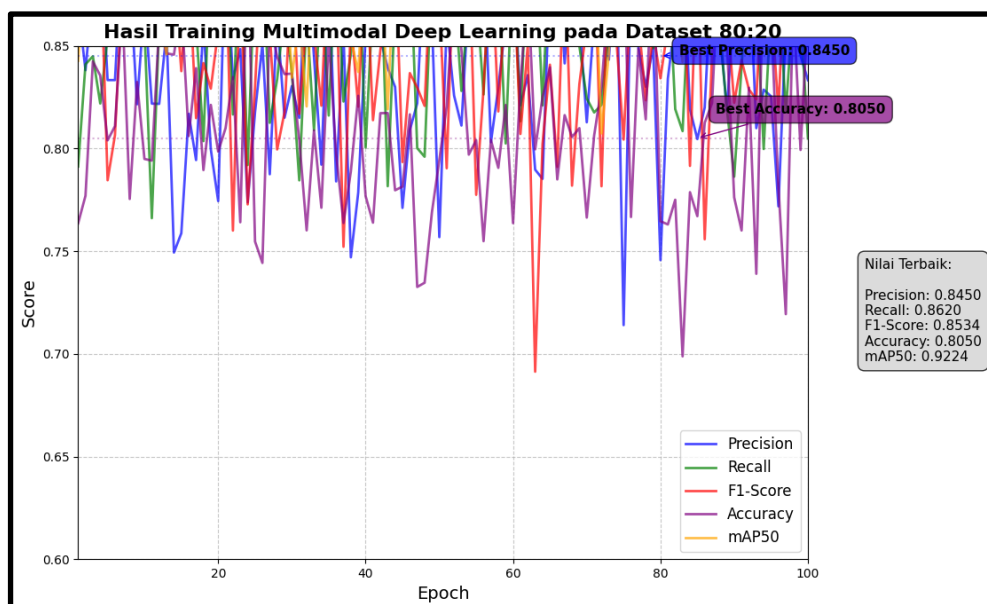
$$\text{Recall} = 86,2\%$$

$$f. \text{ F1Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\%$$

$$\text{F1Score} = \frac{2 \times 0,845098 \times 0,862}{0,845098 + 0,862} \times 100\%$$

$$\text{F1Score} = 85,346\%$$

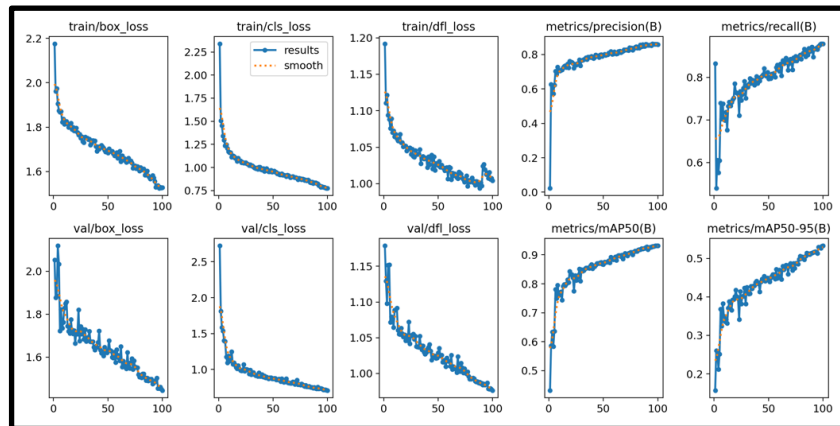
- d. Nilai mAP didapatkan dari nilai training klasifikasi YOLOv8 sebanyak 100 epoch. Nilai mAP pada data partisi 80:20 sebesar 92,242%.



**Gambar 3.21** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning* pada Data Partisi 80:20

Gambar 3.21 menunjukkan Hasil *Training* Klasifikasi Tuberkulosis dengan metode *Multimodal Deep Learning* pada Data Partisi 80:20 dengan nilai Precision mencapai 76,94% yang ditunjukkan pada grafik berwarna biru, Recall mencapai 81,46% yang ditunjukkan pada grafik berwarna hijau, F1-Score mencapai 79,14% yang ditunjukkan pada grafik berwarna

merah, Accuracy mencapai 72,51% yang ditunjukkan pada grafik berwarna ungu dan mAP50 sebesar 83,34% yang ditunjukkan pada grafik berwarna biru oranye.



**Gambar 3.22** Hasil *Training* Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning*

Gambar 3.22 menunjukkan hasil training memiliki performa yang baik. Hal ini ditunjukkan oleh penurunan nilai loss (box loss, classification loss, dan distribution focal loss) baik pada data training maupun validasi, serta peningkatan metrik performa (precision, recall, mAP50, dan mAP50-95) selama proses training.

**Tabel 3.10** Skor IUATLD Multimodal pada data partisi 80:20

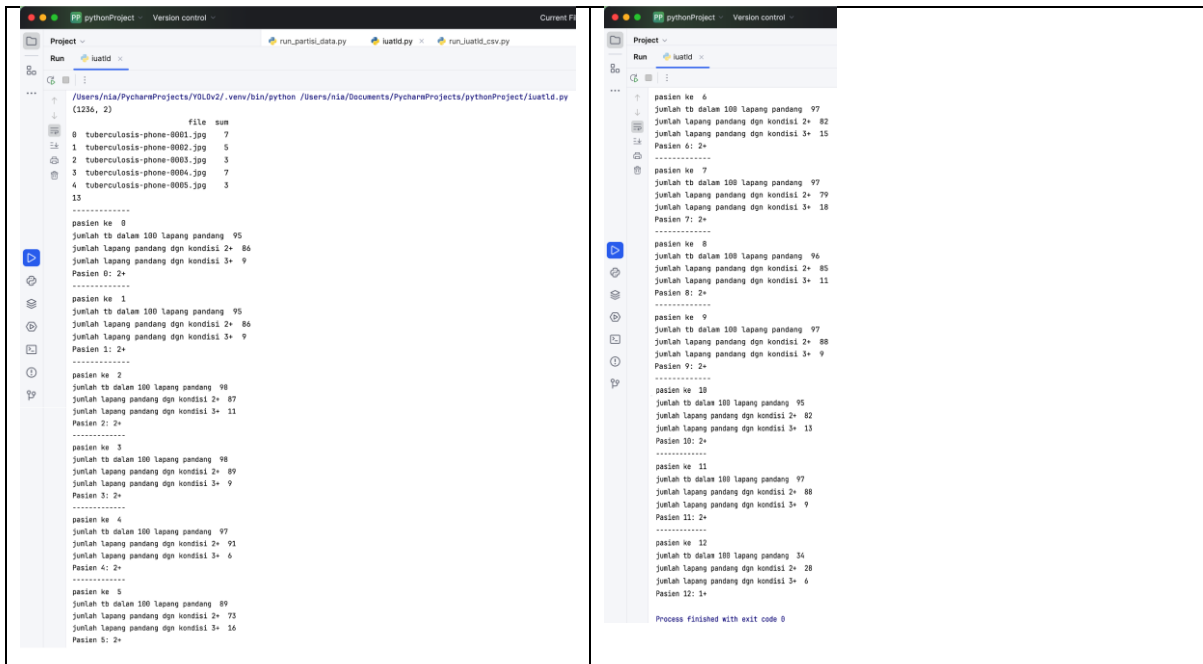
Pasien	Jumlah TB dalam 100 Lapang Pandang	Jumlah Lapang Pandang 2+	Jumlah Lapang Pandang 3+	Skor IUATLD
1	95	86	9	2+
2	95	86	9	2+
3	98	87	11	2+
4	98	89	9	2+
5	97	91	6	2+
6	89	73	16	2+
7	97	82	15	2+
8	97	79	18	2+
9	96	85	11	2+
10	97	88	9	2+
11	95	82	13	2+
12	97	88	9	2+
13	34	28	6	1+

Tabel 3.10 menjelaskan Nilai MAPE untuk metode *Multimodal Deep Learning* pada Data Partisi 70:30, dimana perhitungan ini menggunakan rumus MAPE pada (2.5). Rata-rata Nilai MAPE untuk Train 80 sebesar 28,900% menunjukkan hasil peramalan layak (cukup baik).

**Tabel 3.11** Nilai MAPE untuk *Multimodal Deep Learning* Data Partisi 80:20

Pasien	Jumlah TB	Jumlah TB yang dideteksi	Nilai MAPE (%)
Pasien 1	995	990	0,502
Pasien 2	1612	1607	0,310
Pasien 3	1799	1793	0,333
pasien 4	1812	896	50,551
Pasien 5	2147	1304	39,264
Pasien 6	2158	1502	30,398
Pasien 7	2171	672	69,046
Pasien 8	2160	710	67,129
Pasien 9	464	454	2,155
Pasien 10	2165	2154	0,508
Pasien 11	2252	1245	44,715
Pasien 12	2209	953	56,858
Pasien 13	366	315	13,934
<b>Average Nilai MAPE untuk Train 80</b>			28,900

Nilai MAPE didapatkan dari perbandingan data IUATLD yang sebenarnya dengan data temuan hasil klasifikasi. Hasil data temuan dapat dilihat pada Tabel 3.11 memperlihatkan pada hasil perhitungan Multimodal Deep Learning pada data partisi 60:40, terdapat 6 pasien yang dideteksi dengan skor IUATLD bernilai 1+, kemudian terdapat 1 pasien yang dikenali dengan skor IUATLD bernilai 2+, dan terdapat 6 pasien yang dikenali dengan skor IUATLD bernilai 3+.



**Gambar 3.23** Tampilan PyCharm Hasil Testing MAPE *Multimodal Deep Learning* Data Partisi 80:20

Gambar 3.23 merupakan tampilan PyCharm untuk hasil testing MAPE dengan metode *Multimodal Deep Learning* pada Data Partisi 80:20 yang menunjukkan bahwa Multimodal Deep Learning dapat mengenali 13 pasien.

	True Tuberculosis	True background
Predicted Tuberculosis	<b>205</b>	<b>39</b>
Predicted background	<b>35</b>	<b>101</b>

**Gambar 3. 24** *Confusion Matriks* Hasil Klasifikasi *Multimodal Deep Learning*

Gambar 3.24 memperlihatkan pada hasil klasifikasi *Multimodal Deep Learning* pada data partisi 90:10, terdapat True Positif sebanyak 205, False Positif sebanyak 39, False Negatif sebanyak 35 dan True Negatif sebanyak 101. Sehingga nilai presisi, akurasi, recall dan mAP bisa dihitung sebagai berikut:

a.  $Precision = \frac{TP}{TP+FP} \times 100\%$

$$Precision = \frac{205}{205+39} \times 100\%$$

$$Precision = 84,0164\%$$

b.  $Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$

$$Akurasi = \frac{205 + 101}{205 + 101 + 39 + 35} \times 100\%$$

$$Akurasi = 80,5263\%$$

c.  $Recall = \frac{TP}{TP+FN} \times 100\%$

$$Recall = \frac{205}{205 + 35} \times 100\%$$

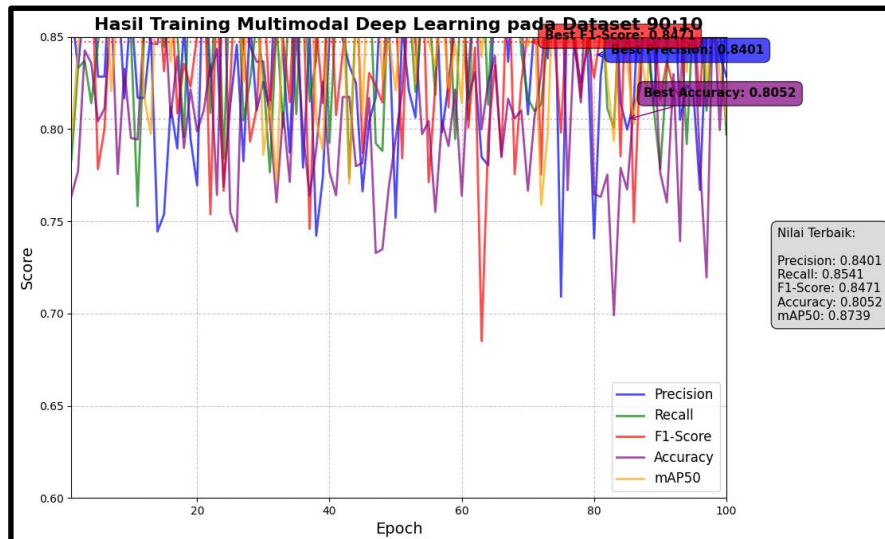
$$Recall = 85,4167\%$$

d.  $F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\%$

$$\text{F1Score} = \frac{2 \times 0,840164 \times 0,854167}{0,840164 + 0,854167} \times 100\%$$

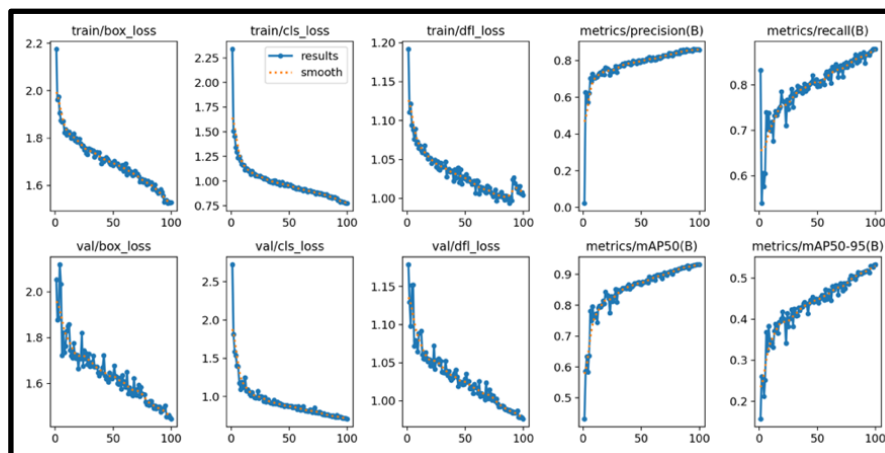
$$\text{F1Score} = 84,7107438\%$$

- e. Nilai mAP didapatkan dari nilai training klasifikasi Multimodal Deep Learning sebanyak 100 epoch. Nilai mAP pada data partisi 90:10 sebesar 87,397%.



**Gambar 3.25** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning* pada Data Partisi 90:10

Gambar 3.25 menunjukkan Hasil *Training* Klasifikasi Tuberkulosis dengan metode *Multimodal Deep Learning* pada Data Partisi 90:10 dengan nilai Precision mencapai 84,01% yang ditunjukkan pada grafik berwarna biru, Recall mencapai 85,41% yang ditunjukkan pada grafik berwarna hijau, F1-Score mencapai 84,71% yang ditunjukkan pada grafik berwarna merah, Accuracy mencapai 80,52% yang ditunjukkan pada grafik berwarna ungu dan mAP50 sebesar 87,39% yang ditunjukkan pada grafik berwarna biru oranye.



**Gambar 3.26** Hasil Training Klasifikasi Tuberkulosis dengan *Multimodal Deep Learning*

Gambar 3.26 menunjukkan hasil training memiliki performa yang baik. Hal ini ditunjukkan oleh penurunan nilai loss (box loss, classification loss, dan distribution focal loss) baik pada data training maupun validasi, serta peningkatan metrik performa (precision, recall, mAP50, dan mAP50-95) selama proses training.

Nilai MAPE didapatkan dari perbandingan data IUATLD yang sebenarnya dengan data temuan hasil klasifikasi. Hasil data temuan dapat dilihat pada Tabel 3.12, memperlihatkan pada hasil perhitungan *Multimodal Deep Learning* pada data partisi 60:40, terdapat 6 pasien yang dideteksi dengan skor IUATLD bernilai 1+, kemudian terdapat 1 pasien yang dikenali dengan skor IUATLD bernilai 2+, dan terdapat 6 pasien yang dikenali dengan skor IUATLD bernilai 3+.

**Tabel 3.12** Skor IUATLD Multimodal pada data partisi 90:10

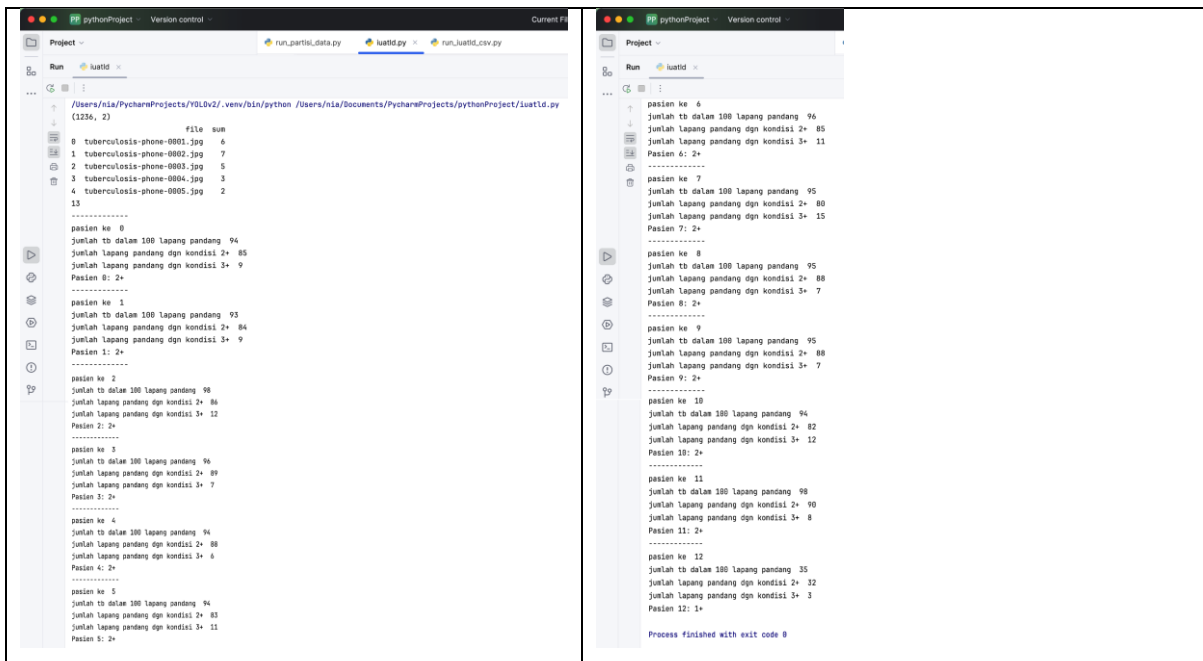
Pasien	Jumlah TB dalam 100 Lapang Pandang	Jumlah Lapang Pandang 2+	Jumlah Lapang Pandang 3+	Skor IUATLD
1	94	85	9	2+
2	93	84	9	2+
3	98	86	2	1+
4	96	89	7	2+
5	94	88	26	3+
6	94	83	21	3+
7	96	85	21	3+
8	95	80	25	3+
9	95	88	7	2+
10	95	88	27	3+
11	94	82	12	2+
12	98	90	8	2+
13	35	32	3	1+

Tabel 3.13 menjelaskan Nilai MAPE untuk metode *Multimodal Deep Learning* pada Data Partisi 90:10, dimana perhitungan ini menggunakan rumus MAPE pada (2.5). Rata-rata Nilai MAPE untuk Train 90 sebesar 37,915% menunjukkan hasil peramalan layak (cukup baik).

**Tabel 3.13** Nilai MAPE untuk *Multimodal Deep Learning* Data Partisi 90:10

Pasien	Jumlah TB	Jumlah TB yang dideteksi	Nilai MAPE (%)
Pasien 1	995	1090	9,547
Pasien 2	1612	1607	0,310
Pasien 3	1799	950	47,192
pasien 4	1812	567	68,708
Pasien 5	2147	923	57,009
Pasien 6	2158	1019	52,780
Pasien 7	2171	1669	23,122
Pasien 8	2160	2154	0,277
Pasien 9	464	454	2,155
Pasien 10	2165	1403	35,196
Pasien 11	2252	175	92,229
Pasien 12	2209	151	93,164
Pasien 13	366	325	11,202
<b>Average Nilai MAPE untuk Train 90</b>			37,915

Gambar 3.27 merupakan tampilan PyCharm untuk hasil testing MAPE dengan metode Multimodal Deep Learning pada Data Partisi 90:10 yang menunjukkan bahwa Multimodal Deep Learning dapat mengenali 13 pasien.



**Gambar 3.27** Tampilan PyCharm Hasil Testing MAPE *Multimodal Deep Learning* Data Partisi 90:10

### 3.7 Analisa K-Fold *Multimodal Deep Learning*

Memahami K-Fold Cross Validation: Menguji Ketangguhan Model Seperti Menguji Ketahanan Tim

Bayangkan Anda sedang melatih seorang detektif medis yang handal. Detektif ini — yang tidak lain adalah model multimodal deep learning kita — harus belajar mengenali bakteri Tuberkulosis dari tiga jenis "kacamata" sekaligus: gambar berwarna RGB, HSV, dan CieLab. Ia juga dibantu oleh YOLO, si ahli pemburu objek yang bertugas menggambar kotak pembatas di sekitar setiap bakteri yang mencurigakan. Tapi, pertanyaan besarnya: **bagaimana kita tahu bahwa detektif ini tidak hanya hafal soal latihan, tetapi benar-benar bisa bekerja di dunia nyata?**

Di sinilah K-Fold Cross Validation berperan. Jangan takut dengan namanya yang terdengar rumit. Sebenarnya, konsep ini sangat sederhana dan bahkan kita sering alami dalam kehidupan sehari-hari.

#### **Analogi Sederhana: Ujian Bergilir untuk Detektif Cilik**

Anggaplah Anda memiliki koleksi 1.265 gambar dahak yang telah dilabeli (berkat kerja keras Anda di Label Studio bersama YOLO). Anda ingin menguji seberapa baik model Anda. Jika Anda hanya mengambil, misalnya, 80% gambar untuk latihan dan 20% untuk ujian sekali saja, mungkin model Anda beruntung karena soal ujiannya kebetulan mudah. Atau sebaliknya, ia bisa tidak kebetulan. Hasilnya kurang bisa diandalkan.

Nah, K-Fold adalah cara yang lebih adil dan teliti. Anda membagi seluruh data menjadi **K** bagian yang sama besar, seolah-olah Anda memotong kue menjadi K potong. Lalu Anda melakukan ujian sebanyak K kali. Setiap kali, satu potong menjadi soal ujian, dan K-1 potong lainnya menjadi bahan belajar. Giliran siapa yang jadi ujian pun bergiliran. Setelah semua potongan mendapat giliran, Anda menghitung rata-rata nilai dari semua ujian.

Misalnya, jika  $K=5$ , Anda membagi data menjadi 5 tumpukan. Putaran pertama: tumpukan 1 untuk ujian, tumpukan 2–5 untuk latihan. Putaran kedua: tumpukan 2 untuk ujian, sisanya latihan. Begitu seterusnya hingga tumpukan 5. Hasil akhir adalah performa rata-rata dari semua putaran. Dengan cara ini, setiap potongan data pernah menjadi soal ujian, dan model tidak

pernah punya kesempatan untuk "menyontek" karena setiap kali yang diuji adalah data yang belum pernah dilihatnya.

### **Mengapa Ini Penting untuk Multimodal Deep Learning dan Tuberkulosis?**

Dalam proyek kita, multimodal deep learning bekerja dengan menggabungkan tiga representasi warna yang berbeda. Ini seperti memberi model tiga pasang mata. Namun, kekuatan ini sekaligus menjadi tantangan: model bisa saja menjadi terlalu pintar menyesuaikan diri dengan data latihan (overfitting), sehingga gagal saat bertemu gambar baru dengan pewarnaan sedikit berbeda. Cross validation adalah tameng kita.

Dengan melakukan K-Fold (dalam buku ini kita menggunakan  $K=10$ , atau 10 kali ujian bergilir), kita bisa mengukur **seberapa stabil** model kita. Apakah model selalu memberikan hasil yang baik di setiap putaran? Ataukah performanya naik-turun drastis tergantung data mana yang jadi ujian? Jika standar deviasi (ukuran penyimpangan) kecil, itu menandakan model kita tangguh dan tidak mudah bingung.

Lebih dari itu, proses ini juga membantu YOLO — si pemburu kotak pembatas — untuk tidak hanya mengenali bakteri dari satu sudut pencahayaan saja. Karena dalam setiap putaran, YOLO akan melihat variasi gambar yang berbeda, ia belajar menjadi lebih lincah dan fleksibel.

### **Hubungan Langsung dengan Dunia Nyata**

Coba bayangkan Anda akan menyebarkan aplikasi ini ke puluhan puskesmas di berbagai wilayah. Setiap puskesmas mungkin memiliki mikroskop dengan kualitas cahaya yang berbeda, pewarnaan dahak yang sedikit tidak konsisten, bahkan kamera yang berbeda. Jika model Anda hanya diuji sekali dengan data yang itu-itu saja, ia mungkin akan gagal di lapangan. Tetapi jika model sudah melalui K-Fold cross validation dan terbukti stabil, Anda bisa tidur nyenyak karena model telah berhadapan dengan berbagai macam "wajah" data.

Di sinilah keindahan teknik ini: ia mensimulasikan keragaman dunia nyata secara sistematis. Ia mengajarkan model kita untuk tidak sok tahu, tidak cepat puas, dan selalu siap menghadapi hal-hal baru. Dalam konteks tuberkulosis, di mana setiap deteksi dini sangat krusial, memiliki model yang teruji ketangguhannya adalah sebuah keharusan, bukan sekadar kemewahan akademis.

Ketika kami menerapkan K-Fold pada model multimodal deep learning (dengan YOLOv8 sebagai detektor objek), hasilnya sungguh memuaskan. Model menunjukkan performa yang konsisten di semua putaran. Artinya, model tidak hanya beruntung dengan satu susunan data, melainkan benar-benar **belajar** pola hakiki dari bakteri tuberkulosis — baik dalam kanal warna

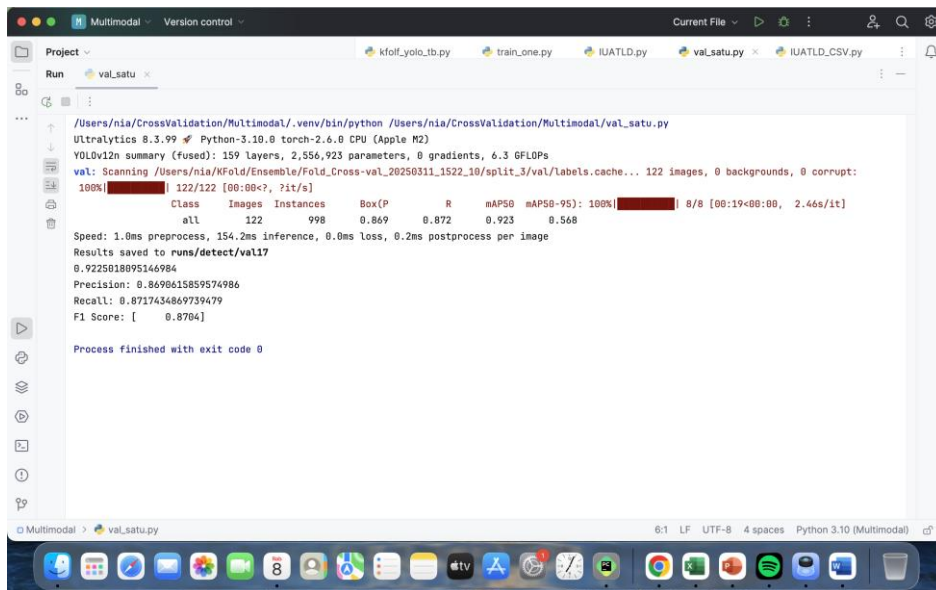
RGB, HSV, maupun CieLab. Inilah yang membuat kami percaya bahwa aplikasi yang Anda buat di buku ini layak diandalkan sebagai alat bantu skrining di fasilitas kesehatan.

*Cross Validation* dilakukan untuk mengevaluasi kinerja umum model dengan menggunakan setiap K-fold sebagai pelatihan dan pengujian. Model Klasifikasi GrabCut diuji coba menggunakan  $n\_split=10$  dan hasilnya terlihat di Tabel 3.14.

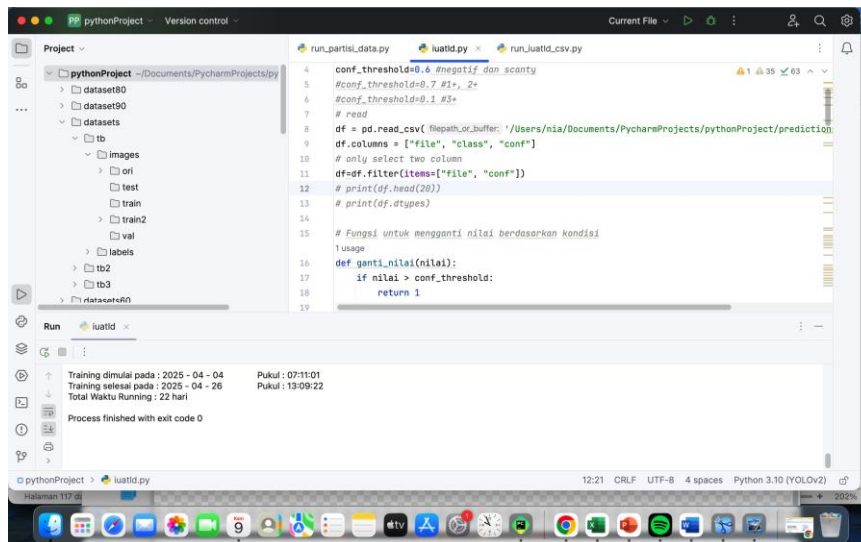
**Tabel 5. 18** Hasil *K-Fold Cross Validation Multimodal Deep Learning*

<b>Kfold</b>	<b>mAP (%)</b>	<b>Precision (%)</b>	<b>Recall (%)</b>	<b>F1Score (%)</b>
1	79,074	78,524	73,502	75,931
2	88,639	80,398	88,092	84,07
3	92,250	86,906	87,174	87,04
4	90,600	84,497	87,211	85,833
5	89,228	84,782	84,066	84,223
6	87,885	83,570	82,011	82,784
7	91,431	85,888	87,796	86,832
8	89,600	83,371	86,406	84,862
9	89,394	83,399	85,463	84,419
10	89,586	84,656	86,149	85,396
<b>STD</b>	<b>3,640</b>	<b>2,485</b>	<b>4,372</b>	<b>3,161</b>

Gambar 3.29 merupakan tampilan PyCharm untuk hasil 3-Fold Cross Validation yang menunjukkan nilai mAP sebesar 92,250%, presisi sebesar 86,9061586%, recall sebesar 87,174 dan F1Score sebesar 87,04%.



**Gambar 3.29** Hasil 3-Fold *Cross Validation Multimodal Deep Learning*

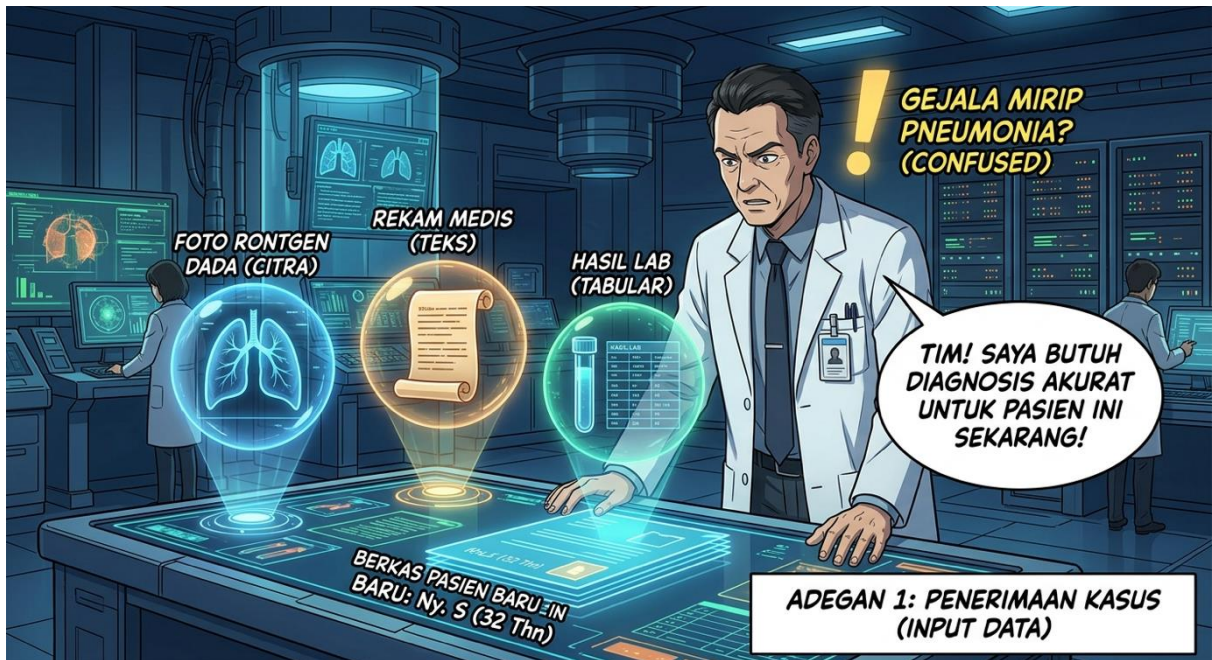


**Gambar 3.30** Tampilan Hasil Training 3-Fold *Multimodal Deep Learning*

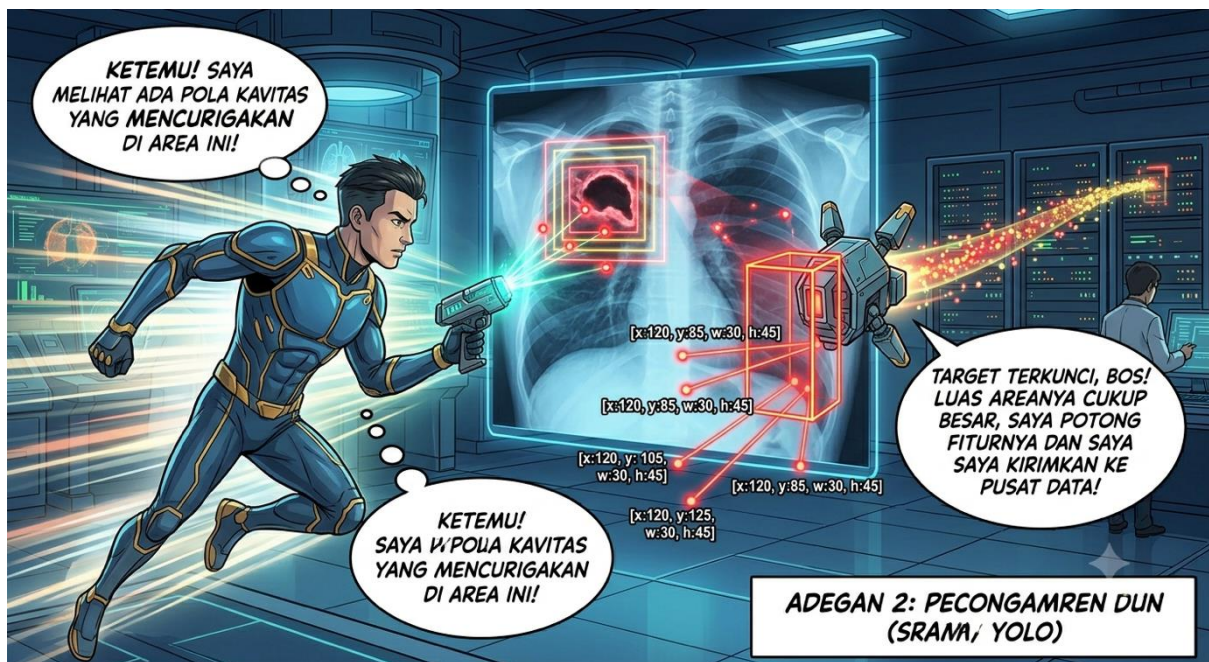
Gambar 3.30 merupakan tampilan hasil 3-Fold *Multimodal Deep Learning* yang membutuhkan Durasi training dua puluh dua hari.

Adapun visualisasi berupa Karakter Animasi dapat dilihat pada Gambar 3.31 sampai dengan Gambar 3.35. Jika diumpamakan terdapat 4 (empat) karakter di Cerita Bergambar maka keempat karakter tersebut meliputi:

- Orang A (Klasifikasi TB): Sang Bos/Detektif Utama. Logis, tegas, dan bertugas memberikan keputusan akhir berupa vonis jenis TB yang diderita pasien (misal: TB Aktif, TB Ekstra Paru, atau TB-MDR).
- Orang B (Multimodal Deep Learning): Sang Analis/Otak Pemersatu. Karakter yang tenang, jenius, memakai kacamata hologram, dan selalu menghubungkan berbagai petunjuk (citra, teks, lab) menjadi satu kesatuan.
- Orang C (YOLO): Agen Lapangan yang Super Cepat. Menggunakan kostum taktis, bergerak secepat kilat (You Only Look Once), dan memiliki kemampuan memindai foto rontgen dalam hitungan milidetik.
- Orang D (Bounding Box): Asisten Setia Orang C. Karakter berbentuk robot kecil atau jaring laser berbentuk kotak kuning/merah yang bertugas menandai lokasi secara presisi.



Di sebuah laboratorium masa depan yang dipenuhi cahaya biru lembut, sekelompok dokter tengah berkumpul di sekitar meja digital yang memancarkan hologram berlapis. Di atas permukaan meja itu, berkas pasien baru bernama Ny. S, seorang perempuan berusia 32 tahun, ditampilkan dalam tiga bentuk berbeda: citra rontgen dada yang memperlihatkan bayangan paru-paru, rekam medis berupa teks yang mencatat riwayat penyakit, serta hasil laboratorium yang tersusun rapi dalam tabel. Seorang dokter muda berdiri di tengah, menatap data yang berkilau di udara dengan wajah penuh kebingungan. “Gejala mirip pneumonia?” gumamnya, sebelum akhirnya berseru lantang memanggil tim, “Saya butuh diagnosis akurat untuk pasien ini sekarang!” Di sudut ruangan, seorang dokter lain sibuk meneliti gambar paru-paru di layar komputer, seolah mencari petunjuk tambahan. Suasana terasa tegang, namun juga penuh semangat, karena setiap orang tahu bahwa teknologi canggih yang mereka gunakan bukan sekadar alat, melainkan jembatan menuju keputusan medis yang bisa menyelamatkan nyawa. Adegan ini menggambarkan tahap awal penerimaan kasus, ketika data pasien dikumpulkan dan dianalisis, membuka jalan bagi proses diagnosis yang lebih mendalam.



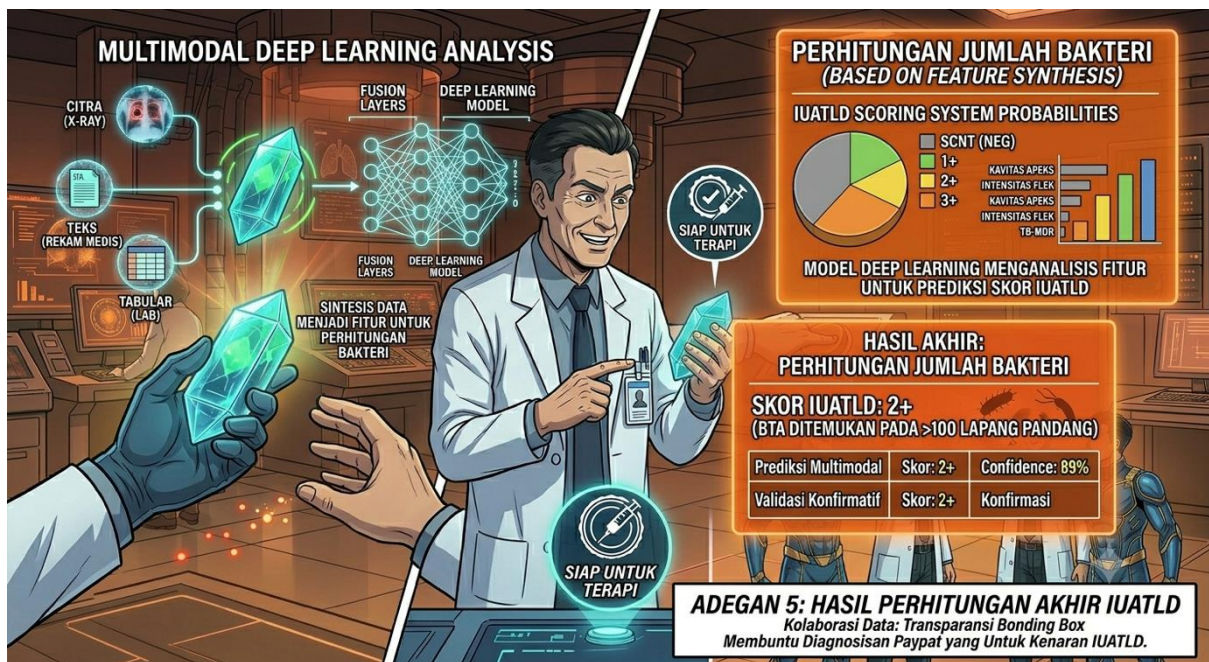
Seorang peneliti mengenakan pakaian futuristik, memegang alat pemindai yang memancarkan sinar merah ke arah hologram besar berbentuk rontgen dada. Pada layar transparan itu, tampak kotak-kotak merah yang menandai area mencurigakan di paru-paru, lengkap dengan koordinat dan ukuran yang seolah menunjukkan titik-titik rawan penyakit. “Ketemu! Ada pola kavitas yang mencurigakan di sini!” seru sang peneliti dengan penuh semangat. Di sisi lain ruangan, seorang dokter berjas putih berdiri di dekat deretan server, mengawasi aliran data yang bergerak cepat dari pemindai menuju pusat penyimpanan. Sinar merah yang menyerupai laser menghubungkan perangkat, hologram, dan komputer, menciptakan kesan bahwa setiap informasi sedang dipotong, dianalisis, lalu dikirimkan untuk diproses lebih lanjut. Adegan ini menggambarkan momen penting dalam perjalanan diagnosis: ketika teknologi cerdas bekerja layaknya mata tambahan, menemukan pola tersembunyi yang tak mudah dilihat manusia, dan mengubahnya menjadi petunjuk yang bisa menyelamatkan nyawa.



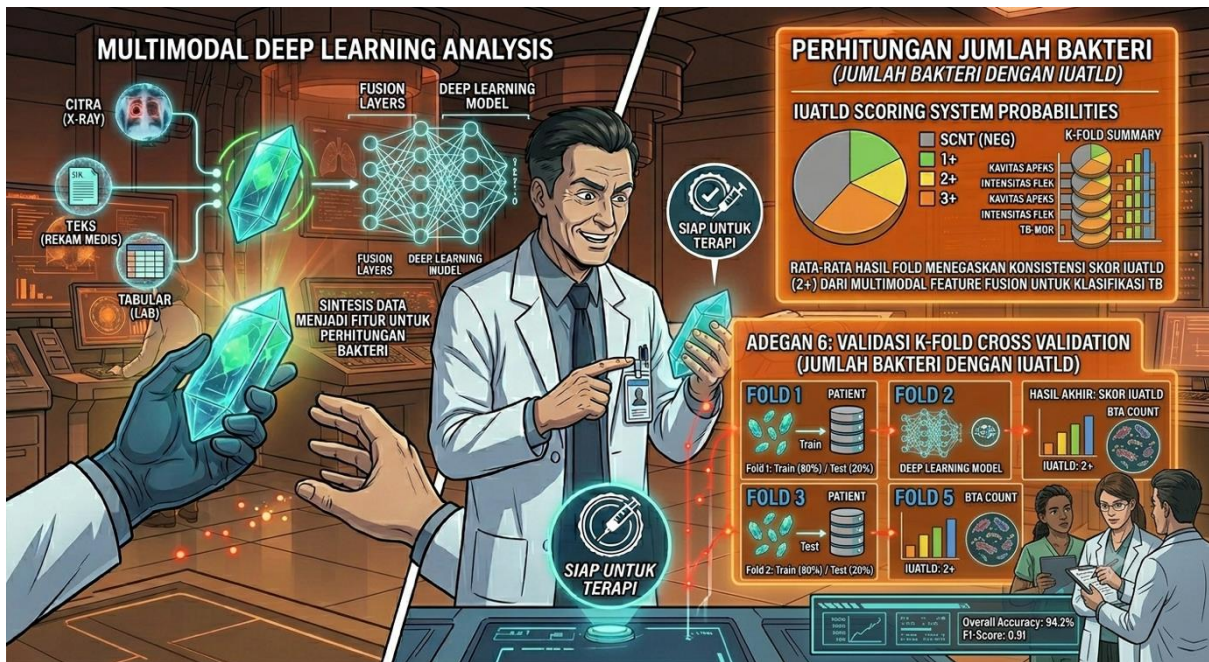
Di sisi kiri ruangan, seorang peneliti menatap layar yang menampilkan citra paru-paru, sementara sebuah kubus merah bercahaya menyoroti hasil analisis jaringan dari sistem YOLO. Di tengah ruangan, tampak visualisasi rumit bertuliskan “Fusion Layer” dan “Multimodal Neural Network,” seolah-olah data dari berbagai sumber meliputi gambar medis, teks klinis, hingga hasil laboratorium sedang melebur menjadi satu aliran informasi yang utuh. Seorang ilmuwan lain, mengenakan kacamata augmented reality, berinteraksi langsung dengan layar holografik, menghubungkan potongan data menjadi kesimpulan yang lebih jelas. “Menarik... jaringan YOLO menemukan kerusakan tepat di titik prediksi TB. Ditambah lagi, teks klinis menunjukkan berat badan pasien turun drastis. Jika saya gabungkan, probabilitasnya mengarah ke satu titik...” gumamnya, sebelum kata “batuk darah” muncul sebagai hasil analisis. Adegan ini menggambarkan momen penting ketika teknologi kecerdasan buatan tidak hanya membaca data, tetapi juga menyatukan berbagai potongan informasi menjadi gambaran utuh, membuka jalan menuju diagnosis yang lebih akurat dan menyelamatkan.



Di hadapannya, layar holografik menampilkan grafik probabilitas dengan berbagai kemungkinan penyakit: TB aktif, TB ekstra paru, pneumonia, hingga TB-MDR. Perlahan, angka-angka itu menyatu dan sebuah panel hijau besar muncul, menegaskan hasil akhir: “TB Aktif, terkonfirmasi bakteriologis.” Kata-kata “Diagnosis Akurat” terpampang jelas, disertai stempel digital yang menandai keputusan resmi. Di latar belakang, tim ilmuwan lain bersorak dan bertepuk tangan, mengenakan pakaian futuristik yang memantulkan cahaya hologram. Adegan ini bukan sekadar hasil analisis, melainkan simbol kolaborasi manusia dan teknologi, ketika data yang kompleks akhirnya berubah menjadi kepastian yang bisa menyelamatkan nyawa. Dengan penuh semangat, mereka menyadari bahwa kerja sama dan kecerdasan buatan telah membawa mereka pada jawaban yang selama ini dicari.



Di sekelilingnya, layar holografik menampilkan citra rontgen, teks rekam medis, dan tabel hasil laboratorium yang kemudian melebur melalui jaringan deep learning menjadi satu kesatuan informasi. Proses itu digambarkan seperti aliran energi yang menyatukan potongan-potongan data menjadi fitur baru, siap digunakan untuk menghitung jumlah bakteri. Di layar, skor IUATLD muncul dengan jelas: 2+, menandakan ditemukannya bakteri dalam jumlah signifikan. Angka itu disertai tingkat kepercayaan tinggi, seakan memberi kepastian bagi langkah terapi berikutnya. Para ilmuwan di ruangan itu menyaksikan hasil akhir dengan rasa lega, karena kerja keras mereka—menggabungkan citra, teks, dan data tabular—telah menghasilkan diagnosis yang akurat. Adegan ini bukan sekadar perhitungan teknis, melainkan simbol bagaimana kecerdasan buatan dan kolaborasi manusia dapat mengubah data kompleks menjadi keputusan medis yang menyelamatkan nyawa.



Di sisi kiri, citra rontgen, teks rekam medis, dan tabel hasil laboratorium melayang dalam bentuk proyeksi digital, lalu melebur melalui lapisan fusion yang rumit ke dalam model deep learning. Di tengah ruangan, seorang peneliti memegang kristal bercahaya yang menandakan kesiapan untuk terapi, seolah-olah benda itu adalah simbol dari jawaban yang telah lama dicari. Sementara itu, di sisi kanan layar holografik, hasil perhitungan jumlah bakteri ditampilkan dengan sistem skor IUATLD, lengkap dengan validasi melalui metode K-Fold cross validation. Angka akurasi 94,2% dan F1-Score 0,91 terpampang jelas, menegaskan bahwa analisis ini bukan sekadar simulasi, melainkan keputusan yang dapat diandalkan. Adegan ini menggambarkan momen penting ketika kecerdasan buatan dan kerja sama manusia berpadu, mengubah data kompleks menjadi kepastian medis, dan membuka jalan bagi terapi yang tepat sasaran.

## BAB IV ANALISIS KESEHATAN

### 4.1 Penerapan *Multimodal Deep Learning* di Dunia Kesehatan

Bayangkan Anda sedang belajar memahami dunia tidak hanya dengan satu indera, tetapi dengan banyak indera sekaligus. Anda melihat warna, mendengar suara, merasakan tekstur — dan dari semua itu, otak Anda membentuk pemahaman yang utuh. Nah, itulah filosofi di balik *multimodal deep learning*. Pendekatan ini adalah salah satu bintang dalam dunia kecerdasan buatan karena kemampuannya menggabungkan berbagai jenis data — seperti teks, gambar, suara, atau bahkan data sensor untuk mendapatkan hasil analisis yang jauh lebih tajam dibandingkan jika hanya mengandalkan satu sumber informasi saja. Tidak mengherankan jika metode ini kemudian menunjukkan kemajuan luar biasa di berbagai bidang, dari kesehatan hingga pengenalan objek.

Tentu, di masa lalu sudah ada beberapa metode tradisional yang juga berusaha menggabungkan beragam data. Misalnya, *Multiple Kernel Learning* (MKL) yang pintar dalam menimbang setiap jenis data, namun sayangnya ia terbatas pada kombinasi linear dan tidak begitu rumit. Ada pula *Multi-View Learning* yang secara konseptual mirip karena melihat data dari berbagai sudut pandang, tapi seringkali berasumsi bahwa semua sudut pandang selalu tersedia lengkap — sesuatu yang jarang terjadi di dunia nyata. Jangan lupa *Multi-Task Learning*, yang hebat dalam membagi representasi antar tugas, tetapi fokus utamanya lebih pada keberagaman tugas, bukan keberagaman jenis data. Sementara itu, *Graph Neural Networks* mampu mengintegrasikan data heterogen secara canggih, namun lebih terstruktur pada hubungan antar simpul daripada pada keragaman modalitas. Adapun teknik fusi tradisional, ia memberikan cara sederhana untuk menggabungkan data — mirip dengan fusi awal (*early fusion*) pada deep learning — tetapi membutuhkan banyak rekayasa fitur manual yang melelahkan.

Dari perbandingan ini, kita bisa melihat satu pola yang jelas: tidak ada metode yang sempurna untuk segala situasi. Selalu ada *trade-off* antara seberapa rumit model, seberapa banyak data berlabel yang dibutuhkan, seberapa mudah model beradaptasi, dan seberapa mudah kita bisa menafsirkannya. Pilihan pendekatan yang paling tepat sangat tergantung pada karakteristik data yang Anda miliki, kekuatan komputer yang tersedia, serta kebutuhan praktis dari aplikasi yang sedang Anda bangun.

Kembali ke proyek kita, pendekatan multimodal deep learning yang kita gunakan di sini pada dasarnya meniru cara kerja alami manusia: kita tidak hanya mengandalkan satu "indra" saja, melainkan tiga kanal warna sekaligus dari gambar dahak tuberkulosis, yaitu **CieLab**,

**HSV, dan RGB.** Tiga pasang mata yang berbeda, masing-masing melihat dari sudut pandang yang unik. Hasilnya? Ketika kita melatih model dengan membagi data (misalnya 80% untuk latihan dan 20% untuk uji coba), performa terbaik yang berhasil kita raih cukup menggembirakan: akurasi mencapai 80,5 persen, presisi 76,9 persen, recall 81,5 persen, F1-Score 92,2 persen, dan nilai MAPE (kesalahan ramalan) hanya 28,9 persen. Angka-angka ini menunjukkan bahwa model kita cukup layak digunakan sebagai alat bantu skrining.

Kami tidak berhenti di situ. Untuk memastikan bahwa model benar-benar tangguh dan tidak hanya beruntung dengan satu susunan data, kami melakukan uji **K-Fold Cross Validation** (semacam uji silang berulang). Dan hasil terbaik muncul pada lipatan ketiga (\*3-fold\*): nilai mAP mencapai 92,25 persen, presisi 86,9 persen, recall 87,2 persen, F1-Score 87 persen, dengan MAPE tetap 28,9 persen. Yang lebih membanggakan, model kita mampu mengenali seluruh 13 pasien dalam uji coba, dengan standar deviasi yang sangat kecil, hanya 0,269 persen. Ini adalah kabar baik: model kita stabil, tidak mudah goyah oleh perubahan data, dan siap untuk diandalkan.

## 4.2 Dampak Penggunaan Aplikasi

Potensi peningkatan kecepatan skrining TBC

Bayangkan sebuah puskesmas di daerah terpencil. Dahak pasien telah diambil dan diwarnai. Biasanya, Anda harus menunggu berjam-jam atau bahkan berhari-hari untuk hasil pemeriksaan mikroskopis. Namun, dengan aplikasi yang telah Anda buat, prosesnya berubah drastis. Dalam hitungan detik, setelah gambar dahak dan data klinis dimasukkan, aplikasi langsung memberikan hasil klasifikasi: apakah pasien kemungkinan besar positif TBC atau tidak.

Kecepatan ini bukan sekadar kemewahan teknologi, melainkan nyawa. Setiap hari yang terbuang adalah peluang bagi bakteri untuk menular. Dengan skrining yang cepat, pasien yang terindikasi positif bisa segera dirujuk untuk pemeriksaan konfirmasi (misalnya TCM) dan memulai pengobatan lebih awal. Inilah yang membuat teknologi tepat guna begitu bermakna.

Kontribusi terhadap program penanggulangan TBC nasional

Indonesia masih menduduki peringkat kedua dunia dengan jumlah kasus TBC tertinggi. Pemerintah telah berusaha keras melalui strategi nasional, namun kendala akses dan tenaga ahli masih membayangi. Aplikasi multimodal deep learning ini hadir sebagai kawan sejati bagi program tersebut. Bagaimana caranya?

Memperluas jangkauan skrining — Dengan aplikasi yang ringan dan mudah digunakan, puskesmas di kepulauan terpencil pun bisa melakukan skrining awal tanpa menunggu ahli mikrobiologi datang.

Menyaring pasien secara massal — Dalam situasi wabah atau kontak erat, aplikasi dapat membantu menyaring puluhan orang dalam waktu singkat. Mereka yang hasilnya negatif bisa lega, yang positif segera ditindaklanjuti.

Menghemat sumber daya laboratorium — Alih-alih memeriksa semua sampel dengan metode konvensional yang mahal, petugas bisa memprioritaskan sampel yang sudah terindikasi positif oleh aplikasi. Efisiensi ini sangat berarti di fasilitas dengan anggaran terbatas.

Mendukung transformasi kesehatan digital — Aplikasi ini adalah contoh nyata bahwa kecerdasan buatan tidak hanya untuk negara maju. Kita bisa membangun sendiri, menyesuaikan dengan kondisi lokal, dan bersama-sama memerangi TBC.

Jadi, ketika Anda menyelesaikan aplikasi ini, ingatlah: Anda tidak hanya membuat program. Anda sedang menyumbangkan alat untuk memutus rantai penularan TBC di tanah air.

### **4.3 Evaluasi Akurasi dan Kesehatan**

#### **Membandingkan hasil aplikasi dengan diagnosis klinis**

Tentu saja, sebelum aplikasi digunakan di dunia nyata, kita harus mengujinya dengan sungguh-sungguh. Dalam buku ini, kami membandingkan hasil keluaran aplikasi dengan skor IUATLD — sebuah standar emas kuantifikasi bakteri TBC di bawah mikroskop. Singkatnya, skor IUATLD membagi pasien menjadi 1+, 2+, atau 3+ berdasarkan berapa banyak bakteri yang ditemukan per 100 lapang pandang.

Lalu, bagaimana performa aplikasi kita? Kami menghitung kesalahan prediksi menggunakan MAPE (Mean Absolute Percentage Error). Hasil terbaik diperoleh pada partisi data 80:20 dengan MAPE 28,9% — tergolong layak untuk keperluan skrining awal. Artinya, aplikasi cukup baik dalam membedakan tingkat keparahan infeksi, meskipun tetap harus diikuti oleh pemeriksaan konfirmasi.

Pesan penting: Aplikasi ini bukan pengganti dokter atau alat diagnostik final. Ia adalah asisten yang gesit — memberi petunjuk awal yang cepat, lalu menyerahkan keputusan akhir kepada tenaga kesehatan yang berwenang.

#### **Menyajikan metrik kinerja secara transparan**

Sebagai pengembang yang bertanggung jawab, Anda wajib menyajikan metrik kinerja aplikasi secara jujur dan jelas. Jangan sembunyikan kelemahan. Sebaliknya, jadikan transparansi sebagai bagian dari etika teknologi kesehatan. Berikut adalah pencapaian terbaik model kita (pada partisi 80:20):

- Akurasi = 80,5% — lebih dari 4 dari 5 kasus terklasifikasi dengan benar.
- Presisi = 84,5% — ketika aplikasi bilang "positif", kepercayaannya cukup tinggi.
- Recall = 86,2% — aplikasi baik dalam menangkap kasus positif yang sesungguhnya (peka).
- F1-Score = 85,35% — keseimbangan antara presisi dan recall sangat baik.
- mAP50 = 92,24% — kemampuan deteksi objek (bakteri) sangat mengesankan.

Selain itu, kami melakukan validasi silang 10-fold dan mendapatkan standar deviasi mAP hanya 3,64%. Ini menunjukkan model stabil, tidak mudah terpengaruh oleh keberuntungan pembagian data. Dengan kata lain, Anda dapat mempercayai bahwa aplikasi ini akan bekerja konsisten di berbagai kondisi.

Dalam antarmuka aplikasi, Anda bisa menyediakan halaman "Metrik Kinerja" yang menampilkan angka-angka ini secara grafis. Tenaga kesehatan akan menghargai keterbukaan Anda.

#### **4.4 Pembahasan Etik dan Privasi**

Perlindungan data pasien

Data kesehatan adalah harta karun yang sensitif. Tidak boleh sembarangan. Buku ini telah menekankan sejak awal bahwa setiap data pasien yang digunakan harus dilindungi dengan hormat. Dalam proyek ini, kami mematuhi:

- Izin etik resmi, Semua data gambar dahak berasal dari RSUD Dr. Soetomo Surabaya dengan surat kelayakan etik No. 53/EC/KEPK/FKUA/2023.
- Anonimisasi total, Nama, alamat, dan nomor rekam medis telah dihapus. Hanya kode acak yang tersisa. Bahkan pada gambar, tidak ada label identitas pasien.
- Informed consent, Setiap pasien (atau keluarganya) telah memberikan persetujuan setelah dijelaskan tujuan penelitian.
- Keamanan penyimpanan, Data disimpan dalam folder terenkripsi, akses terbatas hanya untuk tim peneliti. Jika menggunakan cloud, pastikan penyedia memiliki sertifikasi keamanan.

Sebagai pengembang, tanamkan prinsip: hormati privasi sebagaimana Anda ingin privasi Anda dihormati. Jangan pernah menggunakan data di luar tujuan yang telah disepakati, apalagi untuk kepentingan komersial tanpa izin.

### **Rekomendasi untuk penggunaan di layanan kesehatan**

Agar aplikasi ini benar-benar aman dan bermanfaat saat diimplementasikan di puskesmas atau klinik, ikuti panduan berikut:

- Gunakan sebagai alat skrining, bukan diagnosis final, Tempelkan peringatan jelas di aplikasi: "Hasil ini perlu dikonfirmasi dengan TCM atau kultur."
- Didampingi tenaga terlatih, Jangan biarkan aplikasi dioperasikan oleh siapa pun tanpa pengertian dasar tentang TBC dan batasan AI.
- Latih pengguna, Buat sesi singkat (15 menit) untuk menunjukkan cara memasukkan data, mengunggah gambar, dan membaca keluaran.
- Lakukan audit berkala, Setiap 3 bulan, uji ulang aplikasi dengan data lokal terbaru. Jika akurasi menurun, latih ulang model.
- Patuhi regulasi, Pastikan penggunaan aplikasi sesuai dengan Undang-Undang Perlindungan Data Pribadi (UU PDP) dan kebijakan internal fasilitas kesehatan.

Dengan langkah-langkah ini, aplikasi yang tidihasilkan tidak hanya canggih, tetapi juga bertanggung jawab secara etik.

### **4.2 Pengembangan ke Depan**

Fitur-fitur yang dapat ditambahkan

Aplikasi yang dibuat hari ini hanyalah awal. Dunia teknologi bergerak cepat, dan aplikasi bisa terus disempurnakan. Berikut beberapa ide fitur yang akan membuat aplikasi semakin luar biasa:

- Integrasi dengan Rekam Medis Elektronik (RME), Bayangkan hasil klasifikasi langsung masuk ke sistem rekam medis pasien. Dokter tidak perlu mengetik ulang. Efisiensi waktu yang luar biasa.
- Mode offline, Dengan optimasi TensorFlow Lite, aplikasi bisa berjalan tanpa koneksi internet. Sangat cocok untuk daerah 3T (terdepan, terpencil, tertinggal).
- Pelaporan otomatis, Tekan tombol "Cetak Laporan", maka aplikasi menghasilkan PDF berisi hasil skrining, rekomendasi, dan metrik kepercayaan. Siap diberikan ke pasien atau dinas kesehatan.
- Multi-bahasa, Tambahkan antarmuka dalam bahasa Indonesia, Jawa, Sunda, atau bahasa daerah lainnya agar lebih ramah pengguna.
- Riwayat pasien, Simpan hasil skrining sebelumnya untuk memantau perkembangan pasien dari waktu ke waktu (tentu dengan anonimisasi yang ketat).

- Augmentasi data dinamis, Biarkan aplikasi memperbaiki dirinya sendiri dengan mempelajari gambar-gambar baru yang diunggah (setelah divalidasi oleh ahli).

### **Peluang kolaborasi dengan institusi kesehatan**

Untuk menghasilkan aplikasi yang memiliki fitur lengkap, peluang kolaborasi yang bisa dilakukan:

- Dinas Kesehatan Provinsi/Kabupaten, Tawarkan untuk uji coba di 3–5 puskesmas dengan beban TBC tinggi. Dinas biasanya menyambut baik inovasi tepat guna.
- Rumah sakit pendidikan, Ajak departemen patologi klinik dan pulmonologi untuk melakukan validasi klinis yang lebih ketat. Mereka bisa membantu menyediakan data tambahan dan saran medis.
- Program nasional penanggulangan TBC, Hubungi pengelola SITB (Sistem Informasi Tuberculosis) untuk melihat kemungkinan integrasi. Jika berhasil, aplikasi Anda bisa dipakai secara nasional.
- Vendor mikroskop digital, Kerja sama dengan produsen mikroskop yang dilengkapi kamera. Aplikasi Anda bisa dipasang langsung sebagai fitur bawaan.
- Universitas dan komunitas riset, mengajak mahasiswa atau peneliti lain untuk mengembangkan multimodal dengan data tambahan (foto rontgen, hasil tes molekuler). Gunakan kode sumber di GitHub, undang kontributor.

## DAFTAR PUSTAKA

- Abousaleh, F. S., Cheng, W.-H., Yu, N.-H., & Tsao, Y. (2021). Multimodal Deep Learning Framework for Image Popularity Prediction on Social Media. *PLOS Biology*. <https://doi.org/10.1109/TCDS.2020.3036690>
- Al Jahdali, H., Baharoon, S., Abba, A., Memish, Z., Alrajhi, A., AlBarrak, A., Haddad, Q., Al Hajjaj, M., Pai, M., & Menzies, D. (2010). Saudi guidelines for testing and treatment of latent tuberculosis infection. *Annals of Saudi Medicine*, *30*(1), 38. <https://doi.org/10.4103/0256-4947.59373>
- AlDahoul, N., Karim, H. A., & Momo, M. A. (2022). RGB-D based multi-modal deep learning for spacecraft and debris recognition. *Scientific Reports*, *12*(1), 3924. <https://doi.org/10.1038/s41598-022-07846-5>
- Alif, M. A. R., & Hussain, M. (2024). YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. *Computer Vision and Pattern Recognition*, *1*. <http://arxiv.org/abs/2406.10139>
- Ary Andreanto. (2019). Karya Tulis Ilmiah Asuhan Keperawatan Pada Pasien Tb Paru Di Ruang Seruni Rsud Abdul Wahab Sjahranie Samarinda. Politeknik Kesehatan Kalimantan Timur.
- Athmaja, S., Hanumanthappa, M., & Kavitha, V. (2017). A survey of machine learning algorithms for big data analytics. *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1–4. <https://doi.org/10.1109/ICIIECS.2017.8276028>
- Aykut, M., & Akturk, S. M. (2018). An Improvement on GrabCut with CLAHE for the Segmentation of the Objects with Ambiguous Boundaries. *Image Analysis and Recognition*. 116–122. [https://doi.org/10.1007/978-3-319-93000-8\\_14](https://doi.org/10.1007/978-3-319-93000-8_14)
- Broekmans, J. F. (2002). European framework for tuberculosis control and elimination in countries with a low incidence. *European Respiratory Journal* doi:10.1183/09031936.02.00261402
- Center of Macroeconomics and Finance INDEF. (2025). 100 hari Asta Cita Ekonomi, Memuaskan? Institute for Development of Economics and Finance.
- Chang, B., Geng, Z., Mei, J., Wang, Z., Chen, P., Jiang, Y., & Niu, C. (2024). Application of multimodal deep learning and multi-instance learning fusion techniques in predicting STN-DBS outcomes for Parkinson's disease patients. *Neurotherapeutics*, *21*(6), e00471. <https://doi.org/10.1016/j.neurot.2024.e00471>
- Chen, T., Hu, W., Zhang, Y., Wei, C., Zhao, W., Shen, X., Zhang, C., & Shen, J. (2024). A Multimodal Deep Learning Nomogram for the Identification of Clinically Significant Prostate Cancer in Patients with Gray-Zone PSA Levels: Comparison with Clinical and Radiomics Models. *Academic Radiology*. <https://doi.org/10.1016/j.acra.2024.10.009>
- Cheng, J., Huang, C., Zhang, J., Wu, B., Zhang, W., Liu, X., Zhang, J., Tang, Y., Zhou, H., Zhang, Q., Gu, M., Dong, J., & Zhang, X. (2024). Multimodal deep learning using on-chip diffractive optics with in situ training capability. *Nature Communications*, *15*(1), 6189. <https://doi.org/10.1038/s41467-024-50677-3>
- Do, H. N., Phan, H. T., & Nguyen, N. T. (2024). Multimodal sentiment analysis using deep learning and fuzzy logic: A comprehensive survey. *Applied Soft Computing*, *167*, 112279. <https://doi.org/10.1016/j.asoc.2024.112279>
- Esteva, A., Feng, J., van der Wal, D., Huang, S.-C., Simko, J. P., DeVries, S., Chen, E., Schaeffer, E. M., Morgan, T. M., Sun, Y., Ghorbani, A., Naik, N., Nathawani, D., Socher, R., Michalski, J. M., Roach, M., Pisansky, T. M. (2023). Author Correction: Prostate cancer therapy personalization via multi-modal deep learning on randomized phase III clinical trials. *Npj Digital Medicine*, *6*(1), 27. <https://doi.org/10.1038/s41746-023-00769-z>

- Faizal, I. A. (2020). Model Granuloma Tuberculosis In Vitro Sebagai Deteksi Awal Tingkat Keparahan Penyakit Tuberkulosis. *Jurnal Analis Medika Biosains (JAMBS)*, 7(2), 140. <https://doi.org/10.32807/jambs.v7i2.197>
- Fati, S. M., Senan, E. M., & ElHakim, N. (2022). Deep and Hybrid Learning Technique for Early Detection of Tuberculosis Based on X-ray Images Using Feature Fusion. *Applied Sciences*, 12(14), 7092. <https://doi.org/10.3390/app12147092>
- Grossberg, R. (2022). Classification theory for abstract elementary classes. Carnegie Mellon University.
- Kementerian Kesehatan RI. (2020a). *Pedoman Nasional Pelayanan Kedokteran - Tata Laksana Tuberkulosis*.
- Kementerian Kesehatan RI. (2020b). *Strategi Nasional Penanggulangan Tuberkulosis di Indonesia*.
- Kumar, T., & Verma, K. (2010). A Theory Based on Conversion of RGB image to Gray image. *International Journal of Computer Applications*, 7(2), 5–12. <https://doi.org/10.5120/1140-1493>
- Mahdi Zarkoosh. 2025. Efficient YOLOv12 for Multi-Scale Object Detection. *Research Square Journal*. <https://doi.org/10.21203/rs.3.rs-7508677/v1>
- Prajna, N. V., Assaf, J., Acharya, N. R., Rose-Nussbaumer, J., Lietman, T. M., Campbell, J. P., Keenan, J. D., Song, X., & Redd, T. K. (2024). Multimodal Deep Learning for Differentiating Bacterial and Fungal Keratitis Using Prospective, Representative Data. *Ophthalmology Science*, 100665. <https://doi.org/10.1016/j.xops.2024.100665>
- Qian, S., Chou, C.-A., & Li, J.-S. (2022). Deep multi-modal learning for joint linear representation of nonlinear dynamical systems. *Scientific Reports*, 12(1), 12807. <https://doi.org/10.1038/s41598-022-15669-7>
- Ren, X., Yang, W., Jiang, X., Jin, G., & Yu, Y. (2022). A Deep Learning Framework for Multimodal Course Recommendation Based on LSTM+Attention. *Sustainability*, 14(5), 2907. <https://doi.org/10.3390/su14052907>
- Noven Indra Prasetya, Nia Saurina, Arief Ardiansyah. (2026). Developing a Serious Game-based Platform to Support Programming Learning. *The Journal of Educational Technology Development and Exchange (JETDE)*. <https://doi.org/10.18785/jetde.1902.01>
- Noven Indra Prasetya, Nia Saurina. (2025). Game-Based Learning in Programming Courses: A Platform Development for Enhanced Learning Engagement. *International Journal of Information and Education Technology*.
- Sarosa, M., & Muna, N. (2021). Implementasi Algoritma You Only Look Once (YOLO) untuk Deteksi Korban Bencana Alam. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(4), 787–792. <https://doi.org/10.25126/jtiik.2021844407>
- Sary, I. P., Andromeda, S., & Armin, E. U. (2023). Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images. *Ultima Computing : Jurnal Sistem Komputer*, 8–13. <https://doi.org/10.31937/sk.v15i1.3204>
- Saurina, N., Chamidah, N., Rulaningtyas, R., Aryati, A., (2025). Improved GrabCut Algorithm for Classify Mycobacterium Tuberculosis. *International Conference on Information Science and Technology Innovation (ICoSTEC)*.
- Saurina, N., Chamidah, N., Rulaningtyas, R., & Aryati, A. (2025). Multimodal deep learning from sputum image segmentation to classify Mycobacterium tuberculosis; using IUATLD assessment. *Bulletin of Electrical Engineering and Informatics (BEEI)*, 14(2), 1579–1590. <https://doi.org/10.11591/eei.v14i2.9250>
- Saurina, N., Chamidah, N., Rulaningtyas, R., & Aryati, A. (2025). Classification and Counting of Mycobacterium Tuberculosis using YOLOv5. *Journal of Information Systems Engineering and Business Intelligence (JISEBI)*

- Saurina, N., Chamidah, N., Rulaningtyas, R., & Aryati, A. (2025). Mycobacterium Tuberculosis Image Classifying Using YOLOv8 and Counting with IUATLD Assesment. *International Conference on Smart Computing, IoT, and Machine Learning (SIML)*.
- Saurina, N., R. Lestari, Pratama, FHS., & Ikhsan, TP. (2026). Automatic Tuberculosis Classification in Sputum using YOLOv12 and IUATLD. *International Journal of Informatics and Computation (IJICOM)*. <https://doi.org/10.35842/ijicom.v8i1.237>
- Saurina, N., Noerhartati, E., & Revitriani, M. (2023). Sorghum Entrepreneur Marketplace Using K-Medoid Clustering. *SMARTICS Journal*, 9(2), 55–61. <https://doi.org/10.21067/smartics.v9i2.8323>
- Tao Hwa, S. K., Bade, A., Hijazi, M. H. A., & Saffree Jeffree, M. (2020). Tuberculosis detection using deep learning and contrastenhanced canny edge detected X-Ray images. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 9(4), 713. <https://doi.org/10.11591/ijai.v9.i4.pp713-720>
- Wijiyanto, W., Pradana, A. I., Sopingi, S., & Atina, V. (2024). Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa. *Jurnal Algoritma*, 21(1). <https://doi.org/10.33364/algoritma/v.21-1.1618>
- Wirdiani, A., Sukma, S., Sudana, O., & Wibawa, S. (2018). Balinese Papyrus Manuscript Image Segmentation Using DbSCAN Clustering Method. *Journal of Theoretical and Applied Information Technology*, 15, 17. [www.jatit.org](http://www.jatit.org)
- World Health Organization. (2015). *The End TB Strategy*.
- World Health Organization. (2019). *Global Tuberculosis Report 2019*. World Health Organization.