1. Ambil Gambar dari gallery

* Atur permission nya di mainactivity
* private val requestPermissionLauncher =  
   registerForActivityResult(  
   ActivityResultContracts.RequestPermission()  
   ) **{** isGranted: Boolean **->** if (isGranted) {  
   Toast.makeText(this, "Permission request granted", Toast.*LENGTH\_LONG*).show()  
   } else {  
   Toast.makeText(this, "Permission request denied", Toast.*LENGTH\_LONG*).show()  
   }  
   **}**private fun allPermissionsGranted() =  
   ContextCompat.checkSelfPermission(  
   this,  
   REQUIRED\_PERMISSION  
   ) == PackageManager.*PERMISSION\_GRANTED*

companion object {  
 private const val REQUIRED\_PERMISSION = Manifest.permission.*CAMERA*}

* on create, tampilin untuk minta request, lalu set btn gallery nya
* if (!allPermissionsGranted()) {  
   requestPermissionLauncher.launch(REQUIRED\_PERMISSION)  
  }  
    
  binding.galleryButton.setOnClickListener **{** startGallery() **}**
* jangan lupa tambahkan ke manifest utk ambil permission nya

<uses-feature android:name="android.hardware.camera" />  
<uses-permission android:name="android.permission.CAMERA" />

* fungsi untuk start gallery dan ambil uri nya

private fun startGallery() {  
 // *TODO: Mendapatkan gambar dari Gallery.* launcherGallery.launch(*PickVisualMediaRequest*(ActivityResultContracts.PickVisualMedia.ImageOnly))  
}  
  
private val launcherGallery = registerForActivityResult(  
 ActivityResultContracts.PickVisualMedia()  
) **{** uri: Uri? **->** if (uri != null) {  
 currentImageUri = uri  
 showImage()  
 } else {  
 Log.d("Photo Picker", "No media selected")  
 }  
**}**

1. Tampilkan preview gambar di main activity
2. private fun showImage() {  
    // *TODO: Menampilkan gambar sesuai Gallery yang dipilih.* currentImageUri?.*let* **{** Log.d("Image URI", "showImage: $**it**")  
    binding.previewImageView.setImageURI(**it**)  
    **}**}
3. Mulai analyze (Btn analyze) gunain ml nya

* On create

binding.analyzeButton.setOnClickListener **{** analyzeImage() **}**

* Masukkan model ml tf lite ke project
* Tambahkan dependensi nya

***implementation*("org.tensorflow:tensorflow-lite-task-vision:0.4.4")**

* Atur class helper nya

1. Tampilkan gambar dari gallery ke result activity

* Putextra image uri
* Get extra image uri untuk ditampilkan

1. Tampilkan cancer / non cancer dengan persentase nya
2. Menambahkan Fitur Crop dan Rotate Sebelum Gambar Diproses

* Tambah di dependensi dulu