LAMPIRAN

AndroidManifest.xml

```
<?xml version="1.0"
encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.
android.com/apk/res/android"
xmlns:tools="http://schemas.an
droid.com/tools">
    <uses-feature
android:name="android.hardware
.camera"
      android:required="false"
/>
    <uses-permission
android:name="android.permissi
on.CAMERA" />
    <uses-permission
android:name="android.permissi
on.INTERNET" />
    <application
    android:allowBackup="true"
android:dataExtractionRules="@
xml/data_extraction_rules"
android:fullBackupContent="@xm
l/backup_rules"
android:icon="@mipmap/ic_launc
her"
android:label="@string/app_nam
e"
android:roundIcon="@mipmap/ic_
launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.Tu
gasAkhir"
android:usesCleartextTraffic="
true"
        tools:targetApi="31">
        <activity
android:name=".ScanProdukActiv
ity"
android:exported="false" />
        <activity
android:name=".ProdukActivity"
      android:exported="false"
/>
        <activity
android:name=".MerekActivity"
android:exported="false" />
        <activity
android:name=".SplashScreenAct
ivity"
      android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.a
ction.MAIN" />
<category
android:name="android.intent.c
ategory.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

APIClient.kt

```
package
com.b221280037.alifia.tugasakh
ir
import retrofit2.Retrofit
import
retrofit2.converter.gson.GsonC
onverterFactory
object APIClient {
    // Konstanta yg menyimpan
URL dasar (base URL) dari API
server
    const val BASE_URL =
"http://nothis.pustikomumpar.x
yz/"
    private var retrofit:
Retrofit? = null
    // Fungsi untuk
mengembalikan instance
Retrofit
    fun getApiClient():
Retrofit {
        if (retrofit == null)
{
            retrofit =
Retrofit.Builder()
.baseUrl(BASE_URL)
.addConverterFactory(GsonConve
rterFactory.create())
                .build()
        } return retrofit!!
    }
}
```

APIInterface.kt

```
package
com.b221280037.alifia.tugasakh
ir
import retrofit2.Call
import retrofit2.http.GET
import retrofit2.http.Query
```

```
interface APIInterface {
@GET("apiboikot.php?laman=daft
armerk")
    fun daftarmerk():
Call<List<DataValue>>
@GET("apiboikot.php?laman=daft
arproduk")
    fun daftarproduk(
        @Query("IdMerk")
idmerk: String
    ): Call<List<DataValue>>
@GET("apiboikot.php?laman=cari
produk")
    fun cariproduk(
        @Query("id") id:
String
    ): Call<List<DataValue>>
}
```

ScanProdukActivity.kt

```
package
com.b221280037.alifia.tugasakh
ir
import android.Manifest
import
android.content.pm.PackageMana
ger
import android.graphics.Color
import android.graphics.RectF
import
android.graphics.drawable.Grad
ientDrawable
import android.os.Bundle
import android.text.Html
import android.util.Size
import android.view.Gravity
import
android.widget.ImageView
import
android.widget.LinearLayout
import
android.widget.RelativeLayout
import android.widget.TextView
import
androidx.annotation.OptIn
import
androidx.appcompat.app.ActionB
ar
import
androidx.appcompat.app.AppComp
atActivity
import
androidx.appcompat.widget.Tool
bar
import
androidx.camera.core.CameraSel
ector
import
androidx.camera.core.Experimen
talGetImage
import
androidx.camera.core.ImageAnal
ysis
import
androidx.camera.core.ImageProx
y
import
androidx.camera.core.Preview
import
androidx.camera.lifecycle.Proc
essCameraProvider
import
androidx.camera.view.PreviewVi
ew
import
androidx.core.app.ActivityComp
at
import
androidx.core.content.ContextC
ompat
import
com.android.volley.toolbox.Str
ingRequest
import
com.android.volley.toolbox.Vol
ley
import
com.bumptech.glide.Glide
import
com.google.android.material.bo
ttomsheet.BottomSheetDialog
import
com.google.android.material.fl
oatingactionbutton.FloatingAct
ionButton
import
com.google.mlkit.vision.common
.InputImage
import
com.google.mlkit.vision.text.T
extRecognition
import
com.google.mlkit.vision.text.l
atin.TextRecognizerOptions
import
```

```
com.b221280037.alifia.tugasakhir.APIClient.BASE_URL
import org.json.JSONObject
@OptIn(ExperimentalGetImage::class)
class ScanProdukActivity : AppCompatActivity() {
    // komponen ui dan logika pemrosesan
    private lateinit var previewView: PreviewView //viewfinder kamera
    private lateinit var tvResult: TextView // manmpilkan hasil deteksi text
    private lateinit var btnReset: FloatingActionButton // resert scan
    private lateinit var scannerOverlay: ScannerOverlayView // menandai area scan
    private val textRecognizer = TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS) // objek MLKit u/ mengenali teks
    private var isProcessing = false
    private var lastDetectedText: String = "" // menyimpan teks yg terakhir dikenali
    override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_scan_produk
        previewView = findViewById(R.id.previewView)
        tvResult = findViewById(R.id.tvResult)
        scannerOverlay = findViewById(R.id.scannerOverlay)
        btnReset = findViewById(R.id.btnReset)
        tvResult.text = "Arahkan kamera ke dalam box" // petunjuk awal ke user
        // jika tombol reset di tekan (kembali ke kondisi awal)

btnReset.setOnClickListener {
            tvResult.text = "Arahkan kamera ke dalam box"
            lastDetectedText = ""
            isProcessing = false
            startCamera()
        }
        // cek izin kamera, jika belum diizinkan, minta izin
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
            == PackageManager.PERMISSION_GRANTED) {
            startCamera()
        } else {
ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.CAMERA), 100)
        }
        setupToolbar()
    }
    // stelah user merespon permintaan izin
    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == 100 && grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            startCamera()
        } else {
            tvResult.text = "Izin kamera ditolak"
```

```
        }
    }
    // u/ menginisialisasi
kamera & analisis gambar
secara real-time
    private fun startCamera()
{
        val
cameraProviderFuture =
ProcessCameraProvider.getInsta
nce(this)
cameraProviderFuture.addListen
er({
            val cameraProvider
= cameraProviderFuture.get()
            // preview kamera
            val preview =
Preview.Builder().build().also
{
it.surfaceProvider =
previewView.surfaceProvider
            }
            // konfigurasi
ImageAnalysis
            val imageAnalysis
= ImageAnalysis.Builder()
.setTargetResolution(Size(1280
, 720))
.setBackpressureStrategy(Image
Analysis.STRATEGY_KEEP_ONLY_LA
TEST)
                .build()
            // proses frame
dengan analyzer
imageAnalysis.setAnalyzer(Cont
extCompat.getMainExecutor(this
)) { imageProxy ->
                if
(!isProcessing) {
isProcessing = true
processImageProxy(imageProxy)
                }
            }
            // gunakan kamera
belakang
            val cameraSelector
=
CameraSelector.DEFAULT_BACK_CA
MERA
cameraProvider.unbindAll()
cameraProvider.bindToLifecycle
(this, cameraSelector,
preview, imageAnalysis)
        },
ContextCompat.getMainExecutor(
this))
    }
    // Deteksi OCR (proses
image frame untuk mengenali
teks)
    private fun
processImageProxy(imageProxy:
ImageProxy) {
        val mediaImage =
imageProxy.image
        if (mediaImage !=
null) {
            val image =
InputImage.fromMediaImage(medi
aImage,
imageProxy.imageInfo.rotationD
egrees)
            tvResult.text =
"Mendeteksi..."
textRecognizer.process(image)
.addOnSuccessListener {
visionText ->
                    var
foundInBox = false
                    for (block
in visionText.textBlocks) {
                        val
box = block.boundingBox
                        if
(box != null) {
val rectF = RectF(box)
val mappedRect =
mapRectToViewCoordinates(rectF
, imageProxy)
                            //
hanya proses teks yg berada
dalam bounding box
                            if
(scannerOverlay.scannerRect.co
ntains(
mappedRect.centerX(),
mappedRect.centerY()
)                            )
{
val detectedText =
block.text.trim()
if (detectedText.isNotBlank()
&& detectedText !=
lastDetectedText) {
```

```
lastDetectedText =
detectedText
checkWithServer(detectedText)
foundInBox = true
break
}
                            }
                        }
                    }
                    if
(!foundInBox) {
isProcessing = false // siap
memproses frame berikutnya
                    }
                }
.addOnFailureListener {

isProcessing = false // gagal
proses
                }
.addOnCompleteListener {
imageProxy.close()
                }
        } else {
            imageProxy.close()
            isProcessing =
false
        }
    }
    // Mengecek Produk di
Database (teks dikirim ke
server u/ validasi)
    private fun
checkWithServer(detectedText:
String) {
        val url = BASE_URL +
"cek_produk.php"
        val requestQueue =
Volley.newRequestQueue(this)
        val postRequest =
object :
StringRequest(Method.POST,
url,
            { response ->
                try {
                    val json =
JSONObject(response)
                    if
(json.getBoolean("status")) {
showBottomSheet(
json.getString("statusproduk")
,
json.getString("gambarproduk")
,
json.getString("namamerk"),
json.getString("namaproduk"),
json.getString("keteranganprod
uk")
                        )
                    } else {
tvResult.text = "Produk tidak
ditemukan"
                    }
                } catch (e:
Exception) {
tvResult.text = "Parsing
error"
                }
                isProcessing =
false
            },
            { error ->
                tvResult.text
= "Error: ${error.message}"
                isProcessing =
false
            }) {
            override fun
getParams():
MutableMap<String, String> {
                return
hashMapOf("nama_produk" to
detectedText)
            }
        }
requestQueue.add(postRequest)
    }
    // mapping koordinat dari
bounding box hasil deteksi ke
koordinat layar
    private fun
mapRectToViewCoordinates(rect:
RectF, imageProxy:
ImageProxy): RectF {
        val scaleX =
previewView.width.toFloat() /
imageProxy.height.toFloat()
        val scaleY =
previewView.height.toFloat() /
imageProxy.width.toFloat()
        val mapped = RectF(
            rect.left *
scaleX,
            rect.top * scaleY,
```

```
            rect.right *
scaleX,
            rect.bottom *
scaleY
        )
        return mapped
    }
    // menampolkan Informasi
Produk
    private fun
showBottomSheet(
        statusproduk: String,
        gambarproduk: String,
        namamerk: String,
        namaproduk: String,
        keteranganproduk:
String
    ) {
        val bottomSheetView =
layoutInflater.inflate(R.layou
t.bottom_sheet_detailproduk,
null)
        val relColor =
bottomSheetView.findViewById<R
elativeLayout>(R.id.relColor)
        val ivClose =
bottomSheetView.findViewById<I
mageView>(R.id.ivClose)
        val backgroundDrawable
= relColor.background.mutate()
as GradientDrawable
        if (statusproduk ==
"Y") {
backgroundDrawable.setColor(Co
ntextCompat.getColor(this,
R.color.merah))
        } else {
backgroundDrawable.setColor(Co
ntextCompat.getColor(this,
R.color.hijau))
        }
        val ivFoto =
bottomSheetView.findViewById<I
mageView>(R.id.ivproduk)
        Glide.with(this)
            .load(BASE_URL +
"produk/$gambarproduk")

.placeholder(R.drawable.notfou
nd)
.error(R.drawable.notfound)
            .into(ivFoto)
        val tvStatus =
bottomSheetView.findViewById<T
extView>(R.id.tvStatus)
        if (statusproduk ==
"Y") {
            tvStatus.text =
"Di Boikot".uppercase()
tvStatus.setTextColor(ContextC
ompat.getColor(this,
R.color.merah))
        } else {
            tvStatus.text =
"Tidak Di Boikot".uppercase()
tvStatus.setTextColor(ContextC
ompat.getColor(this,
R.color.hijau))
        }
bottomSheetView.findViewById<T
extView>(R.id.tvMerek).text =
namamerk
bottomSheetView.findViewById<T
extView>(R.id.tvProduk).text =
namaproduk
bottomSheetView.findViewById<T
extView>(R.id.tvKeterangan).te
xt =
Html.fromHtml(keteranganproduk
, Html.FROM_HTML_MODE_LEGACY)
        val bottomSheetDialog
= BottomSheetDialog(this)
bottomSheetDialog.setCancelabl
e(false)

bottomSheetDialog.setContentVi
ew(bottomSheetView)
ivClose.setOnClickListener {
bottomSheetDialog.dismiss()
            startCamera()
        }
bottomSheetDialog.show()
    }
    // setup tampilan toolbar
aplikasi
    private fun setupToolbar()
{
        window.statusBarColor
= ContextCompat.getColor(this,
R.color.warna2)
        val toolbar: Toolbar =
findViewById(R.id.toolbar)
toolbar.setBackgroundResource(
R.color.warna2)
```

```
setSupportActionBar(toolbar)
        supportActionBar?.let
{ actionBar ->
            val titleTextView
=
TextView(applicationContext).a
pply {
                text = "Scan
Produk".uppercase()
                textSize = 17f
setTextColor(Color.WHITE)
                gravity =
Gravity.CENTER
            }
            val titleLayout =
LinearLayout(applicationContex
t).apply {
                orientation =
LinearLayout.VERTICAL
            }
            val params =
LinearLayout.LayoutParams(

LinearLayout.LayoutParams.WRAP
_CONTENT,
LinearLayout.LayoutParams.WRAP
_CONTENT
            ).apply {
                gravity =
Gravity.CENTER
            }
titleLayout.addView(titleTextV
iew, params)
            actionBar.apply {
                displayOptions
=
ActionBar.DISPLAY_SHOW_CUSTOM
                setCustomView(
titleLayout,
ActionBar.LayoutParams(
ActionBar.LayoutParams.WRAP_CO
NTENT,
ActionBar.LayoutParams.WRAP_CO
NTENT,
Gravity.CENTER
                    ))

setDisplayHomeAsUpEnabled(true
)
            }
        }
toolbar.setNavigationIcon(R.dr
awable.ic_baseline_close) //
ikon close
toolbar.setNavigationOnClickLi
stener {
            onBackPressed() //
kembali ke halaman sebelumnya
        }
    }
}
```

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : A. Alifia Fitrah Maharani	**Pembimbing I : Masnur, ST., M.Kom**
NIM : 221280037	**Pembimbing II : Sudirman Sahidin, ST., M.Kom**
Judul Skripsi : Perancangan Aplikasi Pendeteksi Boikot Pada Produk Konsumen Berbasis OCR	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Pendahuluan. Manfaat dalam penelitian		Konsultasi 1 - buatkan sumber referensi utama - buatkan flow chart	
Konsultasi 2 Metode. Manfaat rumus yg sesuai dgn penelitian		Konsultasi 2 Acc Proposal	
Konsultasi 3 Acc. 22/11-2024.		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : A. ALIFIA FITRAH MAHARANI	Pembimbing I : MASNUR, ST., M.Kom
NIM : 221280037	Pembimbing II : SUDIRMAN SAHIDIN, ST., M.Kom
Judul Skripsi : PERANCANGAN APLIKASI PENDETEKSI BOIKOT PADA PRODUK KONSUMEN BERBASIS OPTICAL CHARACTER RECOGNITION	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Tambah bab revisi + tabel + Grafik		Konsultasi 1 Tambahkan beberapa referensi	
Konsultasi 2 Acc		Konsultasi 2 Kerjakan Aplikasi Sesuai arahan Penguji	
Konsultasi 3 Acc. Ujian Tutup		Konsultasi 3 Tambahkan beberapa fitur	
Konsultasi 4		Konsultasi 4 Lakukan Pengujian berulang.	
Konsultasi 5		Konsultasi 5 Acc	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik