

DAFTAR PUSTAKA

- Agustina, R., Diswandi N., Windy A., Rika S. 2019. Pengaruh Suhu dan Lama Penyangraian Terhadap Sifat Fisik-Kimia Kopi Arabika dan Kopi Robusta. *Prosiding Seminar Nasional Inovasi Teknologi untuk Masyarakat* : 285 – 299.
<https://doi.org/10.17605/OSF.IO/2MCPK>
- Alwanda, M.R., Raden PKR., Derry A. 2020. Implementasi Metode *Convolutional Neural Network* Menggunakan Arsitektur LeNet-5 untuk Pengenalan *Doodle*. *Jurnal Algoritme* 1 (1) : 45 – 56.
<https://doi.org/10.35957/algoritme.v1i1.434>
- Ambarwati, L., Bella ST., Agrifa DS., Eko PJP., Amir MH. 2019. *Message Insertion Using the Convolutional Neural Network Model Approach*. *Jurnal dan Penelitian Teknik Informatika* 4 (1) : 215 – 220.
<https://doi.org/10.33395/sinkron.v4i1.10159>
- Anhar dan Rahma AP. 2023. Perancangan dan Implementasi *Self-Checkout System* pada Toko Ritel Menggunakan *Convolutional Neural Network (CNN)*. *Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, dan Teknik Elektronika* 11 (2) : 466 – 478.
<http://dx.doi.org/10.26760/elkomika.v11i2.466>
- Azmi, K., Sarjon D., Sumijan. 2023. Implementasi *Convolutional Neural Network (CNN)* untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek* 16 (1) : 28 – 40.
<https://doi.org/10.52072/unitek.v16i1.504>
- Bahrumi, P., Ratna, Rahmat F. 2022. Levelisasi Penyangraian Kopi: Suatu Kajian. *Jurnal Ilmiah Mahasiswa Pertanian* 7 (1) : 522 – 525.
<https://doi.org/10.17969/jimfp.v7i1.19022>
- Dalimunthe, AK., Priyambada, Gani S. 2023. Pengaruh Suhu Pemanasan Terhadap Waktu Roasting Biji Kopi Arabika Jenis Gayo Varietas Abbysinia di Tilasawa Coffee Roaster Yogyakarta. *Agricultural Engineering Innovation Journal* 1 (2) : 77 – 93.
<https://doi.org/10.55180/aei.v1i2.716>
- Ibrahim, K., Sofia S., Bambang H., Sjafril D. 2022. Klasifikasi *Grade* Telur Ayam Negeri secara *non-Invasive* Menggunakan *Convolutional Neural Network (CNN)*. *Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, dan Teknik Elektronika* 10 (2) : 297 – 308.
<https://dx.doi.org/10.26760/elkomika.v10i2.297>
- Firmansyach, WA., Umi H., Yudhistira AW. 2023. Analisa Terjadinya *Overfitting* dan *Underfitting* pada Algoritma *Naive Bayes* dan *Decision Tree* dengan Teknik *Cross Validation*. *Jurnal Mahasiswa Teknik Informatika* 7 (1) : 262 – 269.
<https://doi.org/10.36040/jati.v7i1.6329>
- Harahap, M., Jefferson, Surya B., Suprianto S., Christi AT. 2021. *Implementation of Convolutional Neural Network in the Classification of red blood cells have affected of malaria*. *Jurnal dan Penelitian Teknik Informatika* 5 (2) : 199 – 207.

- <https://doi.org/10.33395/sinkron.v5i2.10713>
- Intyanto, GW. 2021. Klasifikasi Citra Bunga dengan Menggunakan *Deep Learning: CNN (Convolutional Neural Network)*. Jurnal Arus Elektro Indonesia 7 (3) : 80 – 83.
<https://doi.org/10.19184/jaei.v7i3.28141>
- Khasanah, U., Bayu S., Farikhin. 2023. Dilated Convolutional Neural Network for Skin Cancer Classification Based on Image Data. Jurnal Teori dan Aplikasi Matematika 7 (1) : 196 – 207.
<https://doi.org/10.31764/jtam.v7i1.11667>
- Maulana, F.F. dan Naim R. 2019. Klasifikasi Citra Buah Menggunakan Convolutional Neural Network. Journal of Informatics and Computer Science 1 (2) : 104 – 108.
<https://doi.org/10.26740/jinacs.v1n02.p104-108>
- Prabowo, D.A., Dedy A. dan Ari M. 2018. Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan *Color Object Tracking*. Jurnal Pseudocode 5 (2) : 85 – 91.
<https://doi.org/10.33369/pseudocode.5.2.85-91>
- Prakosa, A.B., Hendry, Radius T. 2023. Implementasi Model *Deep Learning Convolutional Neural Network (CNN)* pada Citra Penyakit Daun Jagung untuk Klasifikasi Penyakit Tanaman. Jurnal Pendidikan Teknologi Informasi 6 (1) : 107 – 116.
<https://repository.uksw.edu/handle/123456789/31076>
- Prastyaningsih, Y. dan Wiwik K. 2021. Sistem Temu Kembali Citra Pada Level Penyangraian Biji Kopi Menggunakan Ekstraksi Fitur Warna. Jurnal Inovtek Polbeng 6 (2) : 222 – 233.
<https://doi.org/10.35314/isi.v6i2.2086>
- Riska, S.Y. dan Puji S. 2016. Klasifikasi Level Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Multi-SVM. Jurnal Ilmiah Informatika 1 (1) : 39 – 45.
<https://doi.org/10.35316/jimi.v1i1.442>
- Rochmawati, N., Hanik BH., Yuni Y., Hapsari PAT., Wiyli Y., Agus P. 2021. Analisa *Learning Rate* dan *Batch Size* pada Klasifikasi Covid Menggunakan *Deep Learning* dan *Optimizer Adam*. Journal Information Engineering and Educational Technology 5 (2) : 44 – 48.
<https://doi.org/10.26740/jieet.v5n2.p44-48>
- Qudsi, NK., Rosa AA., Arie RS. 2020. Identifikasi Citra Tulisan Tangan Digital Menggunakan *Convolutional Neural Network (CNN)*. Prosiding Seminar Informatika Aplikatif Polinema : 44 – 53.
<http://jurnalti.polinema.ac.id/index.php/SIAP/article/view/344>
- Soekarta, R, Muhammad Y., Fadli H., Nurul AB. 2023. Implementasi Deep Learning untuk Deteksi Jenis Obat Menggunakan Algoritma CNN Berbasis Website. Jurnal Informatika 7 (4) : 455 – 464.
<http://dx.doi.org/10.31000/jika.v7i4.9751>

- Suartika, I.W., Arya Y.W., Rully S. 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS* 5 (1) : A65 – A69.
<https://doi.org/10.12962/j23373539.v5i1.15696>
- Sutarsi, Elisa R., Iwan T. 2016. Penentuan Tingkat Sangrai Kopi Berdasarkan Sifat Fisik Kimia Menggunakan Mesin Penyangrai Tipe Rotari. *Prosiding Seminar Nasional APTA* : 306 – 312.
<http://repository.unej.ac.id/handle/123456789/79885>
- Suud HM., Dyah AS., Safa RI. 2021. Perubahan Sifat Fisik dan Citra Rasa Kopi Arabika Asal Bondowoso pada Berbagai Tingkat Penyangraian. *Jurnal Agrotek* 8 (2) : 70 – 75.
<https://doi.org/10.31764/jau.v8i2.5238>

LAMPIRAN

Lampiran 1. Source Code Python

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import transforms
from torch.utils.data import DataLoader
from torchvision.datasets import ImageFolder
import coremltools as ct

# Definisikan transformasi yang diperlukan
transform = transforms.Compose([
    transforms.Resize((32, 32)),      # Ubah ukuran gambar menjadi 32x32 piksel
    transforms.ToTensor(),           # Konversi gambar menjadi tensor
    transforms.Normalize((0.5, 0.5, 0.5), # Normalisasi nilai piksel
                        (0.5, 0.5, 0.5))
])

# Muat dataset pelatihan
train_dataset =
ImageFolder("/Users/aainayyahm/Documents/Xcode/Cofi/coffee_dataset/train",
transform=transform)
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)

# Muat dataset validasi
val_dataset =
ImageFolder("/Users/aainayyahm/Documents/Xcode/Cofi/coffee_dataset/test",
transform=transform)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False)

# Tentukan arsitektur model LeNet-5
class LeNet5(nn.Module):
    def __init__(self):
        super(LeNet5, self).__init__()
        self.conv1 = nn.Conv2d(3, 6, 5) #
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 22 * 22, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 3) # Output 3 kelas (light, medium, dark roast)

    def forward(self, x):
        x = torch.relu(self.conv1(x))
        x = torch.max_pool2d(x, 2, stride=1)
        x = torch.relu(self.conv2(x))
        x = torch.max_pool2d(x, 2, stride=1)
        x = torch.flatten(x, 1)
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
```

```

    x = self.fc3(x)
    return x

# Inisialisasi model
model = LeNet5()

# Tentukan loss function dan optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Latih model
num_epochs = 10
for epoch in range(num_epochs):
    model.train() # Set model ke mode pelatihan
    for images, labels in train_loader:
        optimizer.zero_grad() # Reset gradien
        outputs = model(images) # Lakukan forward pass
        loss = criterion(outputs, labels) # Hitung loss
        loss.backward() # Hitung gradien
        optimizer.step() # Update parameter

# Evaluasi model pada set pelatihan
correct_train = 0
total_train = 0
total_loss_train = 0
with torch.no_grad():
    for images_train, labels_train in train_loader:
        outputs_train = model(images_train)
        _, predicted_train = torch.max(outputs_train.data, 1)
        total_train += labels_train.size(0)
        correct_train += (predicted_train == labels_train).sum().item()
        loss_train = criterion(outputs_train, labels_train)
        total_loss_train += loss_train.item()

accuracy_train = correct_train / total_train
loss_avg_train = total_loss_train / len(train_loader)

# Evaluasi model pada set validasi
model.eval() # Set model ke mode evaluasi
correct_val = 0
total_val = 0
total_loss_val = 0
with torch.no_grad():
    for images_val, labels_val in val_loader:
        outputs_val = model(images_val)
        _, predicted_val = torch.max(outputs_val.data, 1)
        total_val += labels_val.size(0)
        correct_val += (predicted_val == labels_val).sum().item()
        loss_val = criterion(outputs_val, labels_val)
        total_loss_val += loss_val.item()

```

```

accuracy_val = correct_val / total_val
loss_avg_val = total_loss_val / len(val_loader)

# Print hasil
print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {loss_avg_train:.4f}, Train
Accuracy: {100*accuracy_train:.2f}%, Val Loss: {loss_avg_val:.4f}, Val Accuracy:
{100*accuracy_val:.2f}%')

# Simpan model yang telah dilatih
torch.save(model.state_dict(),
"/Users/aainayyahm/Documents/Xcode/Cofi/coffee_dataset/coffee_roast_classifier.
pth")

model_path =
"/Users/aainayyahm/Documents/Xcode/Cofi/coffee_dataset/coffee_roast_classifier.
pth" # Ganti dengan path model Anda di Google Drive
model.load_state_dict(torch.load(model_path))
model.eval()

# Konversi model PyTorch menjadi model TorchScript
example_input = torch.randn(1, 3, 32, 32)
traced_model = torch.jit.trace(model, example_input)

# Konversi model TorchScript menjadi model Core ML
mlmodel = ct.convert(
    traced_model,
    inputs=[ct.ImageType(name="image", shape=(1, 3, 32, 32), scale=1/255.0)], #
Menyesuaikan konfigurasi untuk menggunakan gambar sebagai input
    classifier_config=ct.ClassifierConfig(class_labels=['Dark', 'Medium', 'Light']), #
Sesuaikan dengan label kelas Anda
    minimum_deployment_target=ct.target.iOS14
)

# Simpan model Core ML
mlmodel.save("/Users/aainayyahm/Documents/Xcode/Cofi/coffee_dataset/LeNet5.
mlmodel")

print("Model Core ML telah disimpan.")

```

Lampiran 2. Source Code Swift

1. ContentView

```

import SwiftUI

struct ContentView: View {
    var body: some View {
        NavigationStack{

```

```

ZStack {
    Color("Primary")
        .frame(maxWidth: .infinity, maxHeight: .infinity)
        .edgesIgnoringSafeArea(.all)
    Text("Choose Your Perfectly Coffee!")
        .font(.system(size: 34, weight: .bold))
        .frame(width: 434)
        .multilineTextAlignment(.center)
        .foregroundColor(Color("Secondary"))
        .offset(y: -330)
    Image("Welcome")
        .scaleEffect(0.3)
    NavigationLink(destination:
MenuView().navigationBarBackButtonHidden()){
        RoundedRectangle(cornerRadius: 100)
            .frame(width: 268, height: 48)
            .foregroundColor(Color("Secondary"))
            .shadow(color: Color.black.opacity(0.25), radius: 1
                , x: 2, y: 0)
            .shadow(color: Color.black.opacity(0.25), radius: 1
                , x: -2, y: 0)
            .shadow(color: Color.black.opacity(0.25), radius: 1
                , x: 0, y: 2)
            .shadow(color: Color.black.opacity(0.25), radius: 1
                , x: 0, y: -2)
            .overlay(
                Text("Get Started")
                    .foregroundColor(.white)
                    .font(.system(size: 17, weight: .semibold))
            )
        }.offset(y: 330)
    }
}.preferredColorScheme(.light)
}
}

```

```

#Preview {
    ContentView()
}

```

2. MenuView

```

import SwiftUI
import CoreML
import Vision
import UIKit

```

```

struct ClassificationResult {
    let image: UIImage
    let result: String
}

```



```

        Image(systemName: "chevron.right")
            .font(.system(size: 17, weight: .bold))
            .foregroundColor(Color("Secondary"))
            .offset(x: 160)
    )
}
Text("ARTICLE FOR YOU")
    .frame(width: 306, alignment: .center)
    .font(.system(size: 17, weight: .bold))
    .foregroundColor(Color("Secondary"))
    .kerning(4.42)
    .offset(x: -60, y: 10)
ArticleView()
    .frame(height: 228)
    .offset(x:10)

Text("Let's start to scan your Coffee!")
    .font(.system(size: 17, weight: .regular))
    .foregroundColor(Color("Secondary"))
    .offset(y: 10)

RoundedRectangle(cornerRadius: 100)
    .frame(width: 268, height: 48)
    .foregroundColor(Color("Secondary"))
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 2, y: 0)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: -2, y: 0)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: 2)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: -2)
    .overlay(
        Text("Start")
            .foregroundColor(.white)
            .font(.system(size: 17, weight: .semibold))
    )
    .onTapGesture {
        isShowingModal = true
    }
}
Button(action: {
    self.showingAlert = true
}) {
    Image("information_button")
}
.offset(x:152, y:-370)
.alert(isPresented: $showingAlert) {
    Alert(title: Text("Informasi"), message: Text("Untuk memaksimalkan
hasil klasifikasi, pengambilan gambar harus dilakukan di tempat yang terang (tidak
gelap atau remang."), dismissButton: .default(Text("OK")))
}
if let result = classificationResult {

```

```

        ResultView(classificationResult: .constant(result), image:
selectedImage, typeOfCoffee: $typeOfCoffee, origin: $origin, process: $process)
            .frame(maxWidth: .infinity, maxHeight: .infinity)
            .background(Color.white)
            .onAppear {
                isShowingModal = false
            }
        }
    }
    .sheet(isPresented: $isShowingModal) {
        LevelSettingView(image: $selectedImage, classificationResult:
$classificationResult, typeOfCoffee: $typeOfCoffee, origin: $origin, process:
$process)
            .presentationDetents([.height(400)])
    }
}
}
}
}

```

```

struct History1View: View {
    var body: some View {
        ZStack{
            Color("Primary")
                .edgesIgnoringSafeArea(.all)
            VStack{
                Text("HISTORY")
                    .font(.system(size: 28, weight: .bold))
                    .foregroundColor(Color("Secondary"))
                    .kerning(4.42)
                    .offset(x:-90, y:0)
            }
            ScrollView {
                VStack(spacing: 10) {
                    ForEach(classificationHistory.indices, id: \.self) { index in
                        let result = classificationHistory[index]
                        RoundedRectangle(cornerRadius: 20)
                            .foregroundColor(Color("Tertiary"))
                            .frame(width: 337, height: 128)
                            .overlay(
                                RoundedRectangle(cornerRadius: 20)
                                    .frame(width: 110, height: 110)
                                    .overlay(
                                        Image(uiImage: result.image)
                                            .resizable()
                                            .aspectRatio(contentMode: .fill)
                                            .frame(width: 110, height: 110)
                                            .cornerRadius(20)
                                            .clipped()
                                    )
                                )
                            .offset(x: -105)
                    }
                }
            }
        }
    }
}

```

```
.overlay(  
  HStack(spacing: 0) {  
    VStack(alignment: .leading, spacing: 10) {  
      Text("Type of Coffee")  
        .font(.system(size: 10, weight: .regular))  
        .multilineTextAlignment(.leading)  
      Text("Origin")  
        .font(.system(size: 10, weight: .regular))  
        .multilineTextAlignment(.leading)  
      Text("Process")  
        .font(.system(size: 10, weight: .regular))  
        .multilineTextAlignment(.leading)  
      Text("Roasting Profil")  
        .font(.system(size: 10, weight: .regular))  
        .multilineTextAlignment(.leading)  
    }  
    .frame(width: 80)  
    VStack(alignment: .leading, spacing: 10) {  
      Text(": \(result.typeOfCoffee)")  
        .font(.system(size: 10, weight: .bold))  
        .multilineTextAlignment(.leading)  
      Text(": \(result.origin)")  
        .font(.system(size: 10, weight: .bold))  
        .multilineTextAlignment(.leading)  
      Text(": \(result.process)")  
        .font(.system(size: 10, weight: .bold))  
        .multilineTextAlignment(.leading)  
      Text(": \(result.roastingProfile)")  
        .font(.system(size: 10, weight: .bold))  
        .multilineTextAlignment(.leading)  
    }  
    .frame(width: 120)  
  }  
  .offset(x: 60)  
)  
}  
}  
}.padding(.horizontal)  
}  
}  
}.navigationBarItems(leading: BackButton())  
}  
}
```

```
struct LevelSettingView: View {  
  @Binding var image: UIImage?  
  @Binding var classificationResult: String?  
  @State private var showingImagePicker = false  
  @State private var sourceType: UIImagePickerController.SourceType?
```

```

@Binding var typeOfCoffee: String
@Binding var origin: String
@Binding var process: String
@State private var showingActionSheet = false

var body: some View {
    VStack(spacing: 15) {
        Text("Detail of Coffee Bean")
            .font(.system(size: 20, weight: .semibold))
            // .offset(y: 25)

//        CoffeeRoastLevelPicker(selectedRoastLevel: $selectedRoastLevel)
//            .offset(y: 5)
        Text("Type of Coffee")
            .font(.system(size: 17, weight: .regular))
            .offset(x: -112)
        TextField("Text...", text: $typeOfCoffee)
            .frame(width: 337, height: 35)
            .font(.system(size: 17))
            .multilineTextAlignment(.leading)
            .textFieldStyle(RoundedBorderTextFieldStyle())
        Text("Origin")
            .font(.system(size: 17, weight: .regular))
            .offset(x: -145)
        TextField("Text...", text: $origin)
            .frame(width: 337, height: 35)
            .font(.system(size: 17))
            .multilineTextAlignment(.leading)
            .textFieldStyle(RoundedBorderTextFieldStyle())
        Text("Process")
            .font(.system(size: 17, weight: .regular))
            .offset(x: -137)
        TextField("Text...", text: $process)
            .frame(width: 337, height: 35)
            .font(.system(size: 17))
            .multilineTextAlignment(.leading)
            .textFieldStyle(RoundedBorderTextFieldStyle())
        RoundedRectangle(cornerRadius: 100)
            .frame(width: 268, height: 48)
            .foregroundColor(Color("Secondary"))
            .shadow(color: Color.black.opacity(0.25), radius: 1, x: 2, y: 0)
            .shadow(color: Color.black.opacity(0.25), radius: 1, x: -2, y: 0)
            .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: 2)
            .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: -2)
            .overlay(
                Button(action: {
                    self.showingActionSheet = true
                }) {
                    Text("Take a Picture")
                        .foregroundColor(.white)
                }
            )
    }
}

```

```

        .font(.system(size: 17, weight: .semibold))
    }
    .actionSheet(isPresented: $showingActionSheet) {
        ActionSheet(title: Text("Take a Picture"), buttons: [
            .default(Text("Camera")) {
                self.sourceType = .camera
                self.showingImagePicker = true
            },
            .default(Text("Choose from Gallery")) {
                self.sourceType = .photoLibrary
                self.showingImagePicker = true
            },
            .cancel()
        ])
    }
)
.offset(y:8)
}
.padding()
.sheet(isPresented: $showingImagePicker, onDismiss: {
    if let image = self.image {
        classifyImage(image)
    }
}) {
    if let sourceType = self.sourceType {
        ImagePicker(image: self.$image, sourceType: sourceType)
    }
}
}

func classifyImage(_ image: UIImage) {
    guard let cilimage = CUIImage(image: image) else { return }

    // Load Core ML model
    guard let model = try? MLModel(contentsOf:
LeNet5_CreateML_100.urlOfModelInThisBundle) else {
        fatalError("Failed to load Core ML model")
    }

    // Create a Vision request
    guard let coreMLModel = try? VNCoreMLModel(for: model) else {
        fatalError("Failed to create Vision Core ML model")
    }

    let request = VNCoreMLRequest(model: coreMLModel) { request, error in
        if let error = error {
            print("Error: \(error)")
            return
        }
    }
}

```

```

guard let results = request.results as? [VNClassificationObservation],
    let topResult = results.first else {
    fatalError("Unexpected result type from VNCoreMLRequest")
}

// Get the top classification result
self.classificationResult = topResult.identifier

// Create a ClassificationResult instance
let classificationResult = ClassificationResult(image: image, result:
topResult.identifier, typeOfCoffee: self.typeOfCoffee, origin: self.origin, process:
self.process, roastingProfile: topResult.identifier) // Menggunakan hasil prediksi
sebagai profil pemanggangan kopi

// Add the classification result to history
classificationHistory.append(classificationResult)
}

// Perform the request
let handler = VNImageRequestHandler(cilmage: cilmage)
do {
    try handler.perform([request])
} catch {
    print("Failed to perform classification: \(error.localizedDescription)")
}
}

}

#Preview {
    MenuView()
}

```

3. GreetingView

```

import SwiftUI

struct GreetingView: View {
    @State private var greetingText: String = ""
    @State private var currentDate: String = ""

    var body: some View {
        VStack(alignment: .center, spacing: 5) {
            Text(greetingText)
                .frame(width: 306, alignment: .center)
                .font(.system(size: 17, weight: .bold))
                .foregroundColor(Color("Secondary"))
                .kerning(4.42)
            Text(currentDate)
                .font(.system(size: 17, weight: .regular))

```

```

        .foregroundColor(Color("Secondary"))
    }
    .edgesIgnoringSafeArea(.all)
    .onAppear {
        updateGreetingAndDate()
    }
}

private func updateGreetingAndDate() {
    let calendar = Calendar.current
    let hour = calendar.component(.hour, from: Date())
    let formatter = DateFormatter()
    formatter.dateFormat = "EEEE, d MMMM yyyy"
    currentDate = formatter.string(from: Date())

    switch hour {
    case 0..<12:
        greetingText = "GOOD MORNING"
    case 12..<18:
        greetingText = "GOOD AFTERNOON"
    default:
        greetingText = "GOOD NIGHT"
    }
}
}

```

4. ArticleView

```

import SwiftUI

struct ArticleView: View {
    var body: some View {
        NavigationView {
            HStack(spacing: 16) {
                ScrollView(.horizontal){
                    LazyHStack(spacing: 18) {
                        ForEach(0..

```

```

@State private var isShowingModal = false
var article: Article

var body: some View {
    ZStack(alignment: .bottomLeading) {
        Image(article.imageName)
            .resizable()
            .aspectRatio(contentMode: .fill)
            .frame(width: 205, height: 227)
            .cornerRadius(10)
            .overlay(
                Rectangle()
                    .foregroundColor(.clear)
                    .frame(width: 205, height: 300)
                    .overlay(
                        Image("ArticleBg")
                            .scaleEffect(2.0)
                            .offset(y: -140)
                            .overlay(
                                Text(article.title)
                                    .font(.headline)
                                    .foregroundColor(.black)
                                    .multilineTextAlignment(.center)
                                    .shadow(color: Color.black.opacity(0.25), radius: 1
                                        , x: 2, y: 0)
                                    .offset(y: 10)
                            )
                    )
            )
    }
    .preferredColorScheme(.light)
    .frame(maxWidth: .infinity)
    .cornerRadius(8)
    .onTapGesture {
        isShowingModal = true
    }
    .sheet(isPresented: $isShowingModal) {
        ArticleDetailView(article: article)
    }
}
}

```

```

struct ArticleDetailView: View {
    var article: Article

    var body: some View {
        VStack(alignment: .leading) {

```



```
ScrollView{
    Image(article.imageName)
        .resizable()
        .aspectRatio(contentMode: .fill)
        .frame(width: 336, height: 250)
        .cornerRadius(10)
        .padding(.top, 20)

    HStack{
        Text("Sumber: \ \(article.imageSource)")
            .font(.caption)
            .foregroundColor(.gray)

        Spacer()

    }.padding(.leading)
    Spacer()
    VStack(spacing: 15){
        Text(article.title)
            .font(.system(size: 20))
            .bold()

        Text(article.para1)
            .font(.body)

        Text(article.para2)
            .font(.body)

        Text(article.para3)
            .font(.body)

        Text(article.para4)
            .font(.body)

        Text(article.para5)
            .font(.body)

        Text(article.para6)
            .font(.body)

        Text(article.para7)
            .font(.body)

        Text(article.para8)
            .font(.body)

        Text(article.para9)
            .font(.body)
    }.frame(width: 336)
```

```

    }
  }
  .padding(.horizontal, 16)
  .foregroundColor(.black)
  .background(Color("Tertiary"))
  .preferredColorScheme(.light)
}
}

```

```

#Preview {
  ArticleView()
}

```

5. ResultView

```
import SwiftUI
```

```

struct ResultView: View {
  @Binding var classificationResult: String
  var image: UIImage?
  @Binding var typeOfCoffee: String
  @Binding var origin: String
  @Binding var process: String
  var body: some View {
    NavigationStack{
      ZStack {
        Color("Primary")
          .frame(maxWidth: .infinity, maxHeight: .infinity)
          .edgesIgnoringSafeArea(.all)
        VStack {
          Text("RESULT")
            .frame(width: 306, alignment: .center)
            .font(.system(size: 28, weight: .bold))
            .kerning(4.42)
            .foregroundColor(Color("Secondary"))
            .offset(x: -100, y: -50)
          if let image = image {
            Image(uiImage: image)
              .resizable()
              // .aspectRatio(contentMode: .fit)
              .frame(width: 324, height: 369)
              .cornerRadius(20)
              .offset(y:-20)
          } else {
            Spacer()
              .frame(height: 369)
          }
        }
      }
    }
  }
}

```

```

HStack(spacing:30){
  VStack(alignment: .leading, spacing: 25){
    Text("Type of Coffee")
      .font(.system(size: 17, weight: .regular))
      .multilineTextAlignment(.leading)
    Text("Origin")
      .font(.system(size: 17, weight: .regular))
      .multilineTextAlignment(.leading)
    Text("Process")
      .font(.system(size: 17, weight: .regular))
      .multilineTextAlignment(.leading)
    Text("Roasting Profil")
      .font(.system(size: 17, weight: .regular))
      .multilineTextAlignment(.leading)
  }
  .frame(width: 140)
  VStack(alignment: .leading, spacing: 25){
    Text(": \ (typeOfCoffee)")
      .font(.system(size: 17, weight: .bold))
      .multilineTextAlignment(.leading)
    Text(": \ (origin)")
      .font(.system(size: 17, weight: .bold))
      .multilineTextAlignment(.leading)
    Text(": \ (process)")
      .font(.system(size: 17, weight: .bold))
      .multilineTextAlignment(.leading)
    Text(": \ (classificationResult)")
      .font(.system(size: 17, weight: .bold))
      .multilineTextAlignment(.leading)
  }
  .frame(width: 150)
}.offset(x:-5, y: 20)

```

```

NavigationLink(destination:
MenuView().navigationBarBackButtonHidden()) {
  RoundedRectangle(cornerRadius: 100)
    .frame(width: 268, height: 48)
    .foregroundColor(Color("Secondary"))
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 2, y: 0)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: -2, y: 0)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: 2)
    .shadow(color: Color.black.opacity(0.25), radius: 1, x: 0, y: -2)
    .overlay(
      Text("Done")
        .foregroundColor(.white)
        .font(.system(size: 17, weight: .semibold))
    )
  }.offset(y:50)
}
}

```

```

    }
  }
}

```

```

struct ResultView_Previews: PreviewProvider {
    static var previews: some View {
        ResultView(classificationResult: .constant("Light"), typeOfCoffee:
        .constant("Type"), origin: .constant("Origin"), process: .constant("Process"))
    }
}

```

6. ImagePicker

```

import Foundation
import SwiftUI
import Vision

```

```

struct ImagePicker: UIViewControllerRepresentable {
    @Binding var image: UIImage?
    @Environment(\.presentationMode) var presentationMode
    var sourceType: UIImagePickerController.SourceType

    class Coordinator: NSObject, UINavigationControllerDelegate,
    UIImagePickerControllerDelegate {
        let parent: ImagePicker

        init(_ parent: ImagePicker) {
            self.parent = parent
        }

        func imagePickerController(_ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {
            if let uimage = info[.originalImage] as? UIImage {
                parent.image = uimage
                parent.classifyImage(uimage) // Call classifyImage here
            }

            parent.presentationMode.wrappedValue.dismiss()
        }
    }

    func makeCoordinator() -> Coordinator {
        Coordinator(self)
    }

    func makeUIViewController(context: Context) -> UIImagePickerController {
        let picker = UIImagePickerController()
        picker.delegate = context.coordinator
        picker.sourceType = sourceType
        return picker
    }
}

```

```

    }

    func updateUIViewController(_ viewController: UIImagePickerController,
context: Context) {}

    func classifyImage(_ image: UIImage) {

    }
}

```

7. ArticleData

```
import Foundation
```

```

struct Article: Identifiable {
    var id = UUID()
    var imageName: String
    var imageSource: String
    var title: String
    var para1: String
    var para2: String
    var para3: String
    var para4: String
    var para5: String
    var para6: String
    var para7: String
    var para8: String
    var para9: String
}

```

```
let articles: [Article] = [
```

```

    Article(imageName: "article_image1", imageSource: "ottencoffee.co.id", title:
"Cara Membuat Kopi yang Ideal", para1:"Rasa nikmat kopi memang bisa
membangkitkan suasana. Kopi juga mampu membangun semangat di pagi hari.
Setelah bangun pagi, kopi akan membuat badan terasa segar dan bertenaga.
Namun semua itu takkan berarti tanpa rasa yang istimewa. Kenikmatan dan
kesegaran kopi memang sangat dipengaruhi oleh jenis kopi. Namun teknik
menyeduh yang tepat juga mampu memberikan pengaruh besar pada rasa yang
dihasilkan. Untuk itu, penting bagi para penikmat kopi untuk mengetahui
bagaimana cara membuat kopi dengan benar.",para2: "Sebelum mulai menyeduh
kopi, pertama kamu harus mengetahui bahwa membeli kopi dan menyeduhnya
sesegera mungkin adalah tindakan yang paling tepat. Hal ini memungkinkan kamu
memperoleh rasa terbaik dari kopi yang ingin Anda nikmati. Selain itu, pastikan
juga bahwa perlengkapan yang akan digunakan memang benar-benar bersih.
Pastikan tidak ada minyak kopi ataupun sisa seduhan kopi sebelumnya.", para3:
"Jika Anda membeli kopi, pastikan untuk membeli kopi dalam bentuk biji, dan
menggilingnya ketika mau diseduh. Alat penggiling kopi atau grinder juga ada
beberapa jenis. Burr ataupun mill grinder lebih dianjurkan karena penggiling
tersebut mampu menghasilkan hasil gilingan yang konsisten. Sebaiknya jangan

```

gunakan blade grinder karena penggiling kopi ini tidak mampu memberikan hasil gilingan yang konsisten.", para4: "Rasa yang terlalu pahit atau rasa yang terlalu datar semua itu disebabkan karena ekstraksi kopi yang dihasilkan. Jika biji kopi digiling dengan cara yang terlalu halus, maka rasa pahitnya akan sangat terasa. Sebaliknya jika rasa kopi terlalu datar, semua itu dikarenakan hasil penggilingannya yang kurang halus. Penggunaan air juga mempengaruhi rasa yang dihasilkan.", para5: "Untuk memperoleh rasa kopi yang berkualitas, gunakanlah air yang telah disaring atau air minum botolan. Untuk menjaga rasa, hindari penggunaan air yang telah disuling atau diringankan kadar mineralnya. Rasio air yang digunakan juga perlu diperhatikan, dianjurkan 1 gram kopi 15 ml air. Perhitungan ini juga berlaku untuk kelipatannya. Meski setiap orang punya selera rasionya masing-masing.", para6: "Suhu air paling ideal untuk menyeduh kopi adalah antara 90 hingga 96 derajat Celcius. Pada suhu ini, kopi akan terekstrak secara optimal. Suhu yang lebih dingin akan menyebabkan rasa kopi terasa datar. Kopi yang dimasak pada suhu yang terlalu panas juga akan menghilangkan kualitas rasa dari kopi tersebut.", para7: "Lama menyeduh kopi juga mempengaruhi rasa kopi. Rasa terbaik dari kopi akan terasa setelah 4 menit terjadi kontak antara kopi dengan air panas. Jika Anda menggunakan french press, waktu kontak idealnya adalah antara 2 hingga 4 menit. Untuk jenis kopi espresso, lama kontak hanya berkisar antara 20 hingga 30 detik saja. Jika rasa yang dihasilkan kurang optimal, kemungkinan waktu kontak antara kopi dengan air panas masih terlalu cepat atau bahkan terlalu lama.", para8: "Salah satu cara membuat kopi yang harus kamu perhatikan adalah suhu. Rasa kopi terbaik hanya bisa dinikmati segera setelah waktu kontak idealnya tercapai. Karena itulah, segera nikmati kopi yang telah dimasak. Untuk menjaga temperatur agar mampu bertahan lebih lama, gunakan cangkir kopi atau mug. ", para9: "Tapi pastikan untuk menjaga temperatur kopi pada suhu antara 80 hingga 85 derajat Celcius. Jika Anda menggunakan pemanas elektrik, pastikan jangan pernah menyeduh atau memanaskan kopi lebih dari 15 menit. Satu hal lagi yang perlu diperhatikan adalah jangan pernah memanaskan ulang kopi kamu. Jika memang sudah dingin, akan lebih baik jika menikmati kopi dalam kondisi tersebut. Meski rasanya sudah memudar, ini jauh lebih baik dibandingkan dengan memanaskannya ulang."),

Article(imageName: "article_image2", imageSource: "twentytwentycoffee.com", title: "Mengenal Profil Roasting Kopi", para1: "Ketika kita berbicara tentang profil sangrai dari kopi, kita sebenarnya mendiskusikan proses penyangraian yang dilakukan pada biji kopi tersebut. Sederhananya, profil kopi roasting adalah nama/jenis sangrai yang di berikan setelah biji kopi mentah selesai disangrai. Profil roasting mempunyai metode sangrai dan waktu yang berbeda-beda yang menyebabkan cita rasa kopi semakin kompleks dan patut untuk di coba.", para2: "Pemanggang kopi atau yang di kenal sebagai Roaster harus terus-menerus bereksperimen dengan berbagai cara untuk meningkatkan kualitas biji kopi mereka secara keseluruhan. Ha ini banyak terdapat kreativitas, kemauan untuk mencoba dan siap gagal, dan banyak hal yang di catat oleh seorang roaster saat menyangrai biji kopi. Dibutuhkan disiplin, ketelitian yang dilakukan selama proses pemanggangan sehingga dapat di buktikan pada hasil akhir proses sangrai tersebut", para3: "Berbagai macam jenis biji kopi mempunyai kelembaban yang berbeda-beda. Sangatlah penting untuk mengetahui seberapa banyak kadar air tersebut terkandung pada biji kopi mentah, karena hal itu sangat mempengaruhi proses roasting dalam mengeluarkan cita rasa, aroma, body, pada biji kopi. Untuk mengetahui jumlah kadar air yang hilang pada proses roasting, dapat di hitung

dengan cara menimbang biji kopi sebelum dan sesudah proses roasting.", para4: "Pro-Tip : Untuk Roaster/pemanggang, menemukan persentase ini secara teratur adalah cara cepat dan efektif untuk memastikan konsistensi batch yang berbeda dari waktu ke waktu.", para5: "Setiap jenis Profil Roasting mempunyai tingkatan warna yang berbeda-beda pada biji kopi. Tingkatan warna itu dapat memastikan hasil cita rasa yang di inginkan. Misalnya, tingkatan warna seperti Coklat-Muda (Light-Roast) cenderung memiliki rasa asam yang menonjol dari biji kopi tersebut, tetapi profil (Light-Roast) juga menonjolkan aroma yang lebih Fruity. Seangkan tingkatan warna coklat-gelap (Dark-Roast) cenderung memiliki rasa yang lebih pahit, dan lebih smoky. Jenis profil Dark-Roast ini mempunyai rasa coklat yang dominan, tetapi semakin lama biji kopi di sangrai, semakin banyak senyawa organik alami pada biji kopi tersebut terurai yang mengurai cita rasa yang kompleks.", para6: "Seorang Roaster harus selalu mengukur berbagai tahapan proses pemangangan itu sendiri untuk lebih memahami bagaimana setiap batch akan berubah. Setiap perbedaan dari batch ke batch dapat mengindikasikan perubahan pada biji kopi yang di sangrai dan dapat membantu menunjukkan potensi kekurangan atau masalah kualitas sebelum itu menimbulkan masalah.", para7: "Ada beberapa hal umum diperhatikan oleh roaster, seperti start temperature, turning point, cracking dan suhu akhir", para8: "", para9: ""),

Article(imageName: "article_image3", imageSource: "ottencoffee.co.id", title: "Jenis-Jenis Kopi Nusantara", para1:"Indonesia adalah negara yang diberkahi dengan letak geografis dan struktur tanah baik tempat bertumbuhnya kopi. Beberapa wilayah di Indonesia tersebar perkebunan kopi yang masing-masing hasil kopinya memiliki karakteristik unik dan berbeda-beda. Apa saja kopi nusantara dan bagaimana karakteristik setiap biji kopinya?", para2: "Kopi Sumatera - Kopi Sumatera adalah salah satu kopi paling terkenal di dunia. Kopi Sumatera yang paling terkenal berasal dari Sumatera Utara dengan kopi Sidikalang, Lintong dan Mandheling. Kopi Sumatera memiliki cita rasa yang berat. Bisa dikatakan paling berat dan kompleks di antara jenis-jenis kopi yang ada di dunia. Beberapa ahli kopi mengatakan kopi Sumatera memiliki cita rasa unik karena dengan karakteristik dengan aroma rempah dan juga earthy. Kopi Sumatera memiliki tekstur halus dan berbau tajam. Inilah yang menyebabkan kopi Sumatera menjadi salah satu kopi paling laris. Kopi Suamtera diproses dalam dua cara yaitu proses semi-washed dan dry-processed. Ditanam di ketinggian dan kontur tanah ideal menjadikan kopi Sumatera berkualitas terbaik bahkan di mata Internasional.", para3: "Kopi Sulawesi - Tana Toraja adalah daerah yang diberkahi tanah tempat kopi tumbuh subur dengan kualitas yang tak kalah baik dari kopi dari daerah lain. Rasa yang kuat dan kadar asam yang tinggi menjadikan kopi Toraja diminati pasar yang memang menyukai kopi dengan keasaman tinggi. Meskipun sering disebut-sebut bercita rasa mirip dengan kopi Sumatera, tapi kopi Toraja memiliki ciri sendiri yang tentunya berbeda. Kopi Toraja memiliki bentuk biji yang lebih kecil dan lebih mengkilap dan licin pada kulit bijinya. Meskipun memiliki cita rasa asam, kopi Toraja memiliki aroma earthy yang khas. Dan menurut ahli kopi aroma itulah yang menjadikan kopi Toraja berbeda dengan karakteristik yang unik pula.", para4: "Kopi Aceh Gayo - Siapa yang tidak kenal dengan kopi Aceh Gayo yang sudah tersohor kenikmatannya? Kopi yang berasal dari daerah Tanah Gayo Aceh Tengah ini menjadi salah satu jenis kopi yang paling banyak dikonsumsi masyarakat maupun yang diekspor ke luar negeri. Kopi Gayo memiliki ciri unik dengan kekhasan aroma yang berbeda dengan kopi-kopi lain di Indonesia. Kopi Gayo menghasilkan sebagian besar jenis kopi Arabika terbaik. Cita rasa kopi Gayo sendiri terasa lebih

pahit dengan tingkat keasaman rendah. Aromanya yang sangat tajam menjadikan jenis kopi ini disukai. Tak heran kopi ini menjadi penghasil kopi terbesar di Asia. Meskipun rasanya pahit, kopi Gayo memberi aroma gurih pada setiap tegukan.", para5: "Kopi Bali Kintamani - Kopi yang berasal dari daerah Kintamani Bali nan sejuk ini memang memiliki keunikan cita rasa yang berbeda dari kopi di daerah lain di nusantara. Kopi Bali Kintamani memiliki cita rasa buah-buahan yang asam dan segar. Hal tersebut terjadi dikarenakan tanaman kopi di Bali Kintamani ditanam bersamaan dengan tanaman lain seperti aneka sayuran dan buah jeruk. Kopi jenis ini menggunakan sistem 'tumpang sari' bersama dengan jenis tanaman lain. Itu kenapa biji kopinya meresap rasa buah-buahan seperti jeruk.", para6: "Kopi Papua Wamena - Kopi yang berasal dari wilayah Timur Indonesia ini tumbuh pada ketinggian 1.500 m dengan suhu 20 derajat. Menjadikannya kopi dengan cita rasa ringan dan memiliki keharuman tajam yang nikmat. Kopi Papua Wamena memiliki tingkat keasaman yang rendah, mungkin dikarenakan letak geografisnya dan juga struktur tanah tempat kopi ini bertumbuh. Yang membuat kopi ini berkualitas tinggi adalah para petani menanam kopi ini secara organik karena tidak menggunakan bahan-bahan kimia yang tentu bisa memengaruhi kopi yang akan dihasilkan. Untuk kamu yang menyukai kopi dengan rasa ringan dan lembut, aroma tajam yang nikmat serta tekstur yang nyaris tanpa ampas, kopi Papua Wamena adalah pilihan yang tak mungkin salah.", para7: "Kopi Flores Bajawa - Flores terkenal dengan keindahan alamnya yang menakjubkan, siapa sangka daerah ini ternyata juga menghasilkan kopi yang tak kalah nikmatnya. Kopi Flores Bajawa adalah kopi yang berasal dari Kabupaten Ngada. Kopi ini tumbuh di dataran Flores yang subur meskipun di kelilingi oleh pegunungan yang masih aktif maupun tidak. Tanah tempat kopi ini dihasilkan ternyata mengandung andosols subur dari abu gunung berapi yang ternyata sangat baik untuk menanam kopi. Dan jadilah kopi Flores Bajawa yang nikmatnya tak kalah dengan kopi nusantara lainnya. Kopi Flores Bajawa biasanya melalui proses giling basah. Kopi ini memiliki sedikit aroma fruity dan sedikit bau tembakau pada after taste-nya. Sebuah keunikan yang mungkin tak didapatkan dari biji kopi yang berasal dari daerah lain.", para8: "Kopi Jawa - Kopi yang berasal dari Pulau Jawa ini ternyata memiliki keunikan cita rasa sendiri. Aroma rempah yang lahir secara alami menjadikan kopi jenis ini dinikmati karena memiliki karakteristik yang berbeda. Meskipun kopi Jawa tidak sekuat kopi Sumatera dan Sulawesi dari segi cita rasa dan aroma, tetapi dia tetap memiliki penikmat sendiri karena aroma tipis rempah yang dihasilkan. Menjadikan pengalaman minum kopi terasa lebih unik dan berbeda. Sebagian besar kopi Jawa melalui proses giling basah. Itu jugalah yang membuat cita rasanya tidak terlalu kuat. Meski begitu kopi Jawa tetap diminati karena menurut beberapa ahli, tidak semua kopi nusantara mampu menghasilkan kopi yang beraroma 'rempah'.", para9: "Selain kopi-kopi yang dipaparkan tadi, ada banyak sekali jenis kopi di nusantara yang jarang dibicarakan. Sebut saja di Pulau Jawa sendiri ada kopi-kopi dari Jawa Barat, ada pula Kopi Temanggung dan masih banyak lagi. Dari Pulau Sumatera sendiri ada Kopi Solok, Kopi Lampung, dan beberapa kopi yang tumbuh di Palembang, Kerinci dan sekitarnya. Hebatnya, ada banyak kopi yang tumbuh di hampir seluruh pulau di nusantara. Kopi-kopi ini memiliki ciri yang berbeda dan memberi kenikmatan yang rupa-rupa. Seperti Indonesia yang berbhineka, begitu pulalah kopi-kopi yang ada di sana."),

Article(imageName: "article_image4",imageSource: "ottencoffee.co.id", title: "Mengenal Berbagai Macam Proses Pengolahan Kopi", para1:"Bagaimana kopi diproses dan diolah setelah panen akan memengaruhi bahkan bisa memberikan efek

dramatis kepada hasil akhir kopi yang diseduh. Sehingga proses pengolahan termasuk faktor penting yang tidak boleh dilewatkan dalam industri kopi.", para2: "SEBELUM mengetahui bagaimana kopi diproses, ada baiknya jika kita tahu dulu berbagai struktur dan lapisan dari buah/ceri kopi. Pada dasarnya, struktur buah kopi (coffee cherry) terdiri dari pericarp (kulit daging terluar) dan biji kopi. Pericarp sendiri terdiri lagi dari beberapa lapisan seperti kulit, daging kulit, layer getah (yang biasanya terdiri dari gula alami dan semacam kandungan alkohol) dan perkamen. Lapisan pericarp adalah yang paling sering dibersihkan, namun lapisan ini juga berpengaruh dalam menambah rasa pada kopi.", para3: "Setelah dipanen, buah-buah kopi yang dipetik kemudian dibawa ke tempat pengolahan/penggilingan untuk memisahkan biji dengan kulit dagingnya. Biji inilah yang kemudian akan dikeringkan agar tetap aman disimpan sebelum dijual ke pasar. Idealnya, kopi memiliki tingkat kelembaban alami sekitar 60%, namun ia dikeringkan sampai kelembabannya hanya berkisar sekitar 11-12 % saja. Tujuannya supaya biji kopi itu tidak terlalu lembab lalu membusuk ketika "menunggu" dijual.", para4: "Setelah dipetik, ceri kopi akan segera diolah. Proses pengolahannya pun bermacam-macam. Beberapa diantaranya adalah seperti berikut.", para5: "Natural Process - Proses natural ini juga dikenal dengan dry process. Proses ini termasuk teknik paling tua yang ada dalam sejarah proses pengolahan kopi. Setelah dipanen, ceri kopi akan diletakkan di atas permukaan alas-alas plastik dan dijemur di bawah sinar matahari. Beberapa produsen kopi kadang menjemurnya di teras bata atau di meja-meja pengering khusus yang memiliki airflow (pengalir udara) di bagian bawah. Ketika dijemur di bawah matahari, biji-biji kopi ini harus dibolak-balik secara berkala agar biji kopi mengering secara merata, dan untuk menghindari jamur/pembusukan. Pada proses natural, buah kopi yang dikeringkan masih dalam berbentuk buah/ceri, lengkap dengan semua lapisan-lapisannya. Prosesnya yang natural dan alami ini akan membuat ceri terfermentasi secara natural pula karena kulit luar ceri akan terkelupas dengan sendirinya. Proses natural ini dianggap mampu memberi notes ala buah-buahan pada kopi, dengan hints umum seperti blueberry, strawberry atau buah-buahan tropis. Kopi pun cenderung memiliki keasaman (acidity) rendah, rasa-rasa yang eksotis dan body yang lebih banyak.", para6: "Washed Process - Atau yang juga dikenal dengan sebutan wet process. Umumnya, proses ini bertujuan untuk menghilangkan semua kulit-kulit daging yang melekat pada biji kopi sebelum dikeringkan. Setelah dipanen, ceri-ceri kopi biasanya 'diseleksi' terlebih dahulu dengan merendamnya di dalam air. Ceri yang mengapung akan dibuang, sementara yang tenggelam akan tetap dibiarkan untuk proses lanjutan karena ceri-ceri demikian dianggap telah matang. Selanjutnya kulit luar dan kulit daging ceri kopi akan dibuang dengan menggunakan mesin khusus yang disebut depulper (pengupas). Biji kopi yang sudah terlepas dari kulitnya ini kemudian dibersihkan lagi dengan memasukkannya ke dalam bejana khusus berisi air agar sisa-sisa kulit yang masih melekat bisa luruh sepenuhnya akibat proses fermentasi. Durasi, atau lamanya kopi difermentasi ini berbeda-beda pada setiap produsen. Namun umumnya berkisar antara 24-36 jam tergantung temperatur, ketebalan layer getah pada ceri kopi, dan konsentrasi enzimnya. Jika suhu di sekitarnya semakin hangat, maka prosesnya akan semakin cepat pula. Kopi-kopi hasil washed process umumnya memiliki karakter yang lebih bersih, light, sedikit berasa buah, body cenderung ringan dan lembut dengan tingkat keasaman (acidity) lebih banyak.", para7: "Pulped Natural Process - Proses ini sering digunakan di Brazil. Setelah dipanen, buah kopi dikupas dengan mesin mekanik untuk membuang kulit dan sebagian besar daging buahnya. Dari sini, biji kopi kemudian dijemur di meja-meja pengering. Sisa-sisa daging buah yang masih lengket biasanya akan luruh pada

proses ini. (Konon sisa-sisa daging buah yang turut dijemur itu memberi tambahan sweetness dan body pada kopi).", para8: "Honey (Miel) Process - Proses ini agak mirip dengan pulped natural dan umumnya digunakan di banyak negara-negara Amerika Tengah seperti Costa Rica dan El Salvador. Belakangan proses ini juga semakin populer di Indonesia. Pada honey process, ceri kopi akan dikupas dengan mesin mekanis, tapi metode ini menggunakan lebih sedikit air jika dibandingkan pulped natural process. Mesin depulper akan dikendalikan untuk menentukan seberapa banyak daging buah yang mau tetap ditinggalkan melekat dengan biji sebelum dijemur. Kulit daging yang tersisa ini dalam Bahasa Spanyol diistilahkan dengan miel yang berarti madu (honey). Sederhananya, pada honey process ada sedikit lendir—atau mucilage dalam istilah Bahasa Inggris—yang tampak lengket pada biji kopi. Dari sinilah proses ini kemudian dinamakan honey process. Jadi bukan karena menggunakan madu, ya.", para9: "Semi-Washed - Proses ini sangat umum ditemui di Indonesia dan sering kita kenal dengan istilah 'giling basah'. Proses semi washed melibatkan dua kali proses pengeringan. Setelah dipetik, kulit terluar ceri kopi dikupas dengan menggunakan depulper dan dikeringkan sebentar. Jika umumnya kelembaban kopi disisakan hingga 11-12 % ketika proses pengeringan, maka pada proses semi-washed, kelembaban kopi disisakan hingga 30-35 % sebelum dikupas lagi hingga bentuknya benar-benar biji/green bean. Nah, green bean inilah yang kemudian dikeringkan lagi sampai ia benar-benar cukup kering untuk disimpan. Kopi-kopi dengan proses semi-washed cenderung memiliki tingkat sweetness yang intens, body lebih penuh, dengan tingkat keasaman lebih rendah jika dibandingkan kopi-kopi washed processed. Plus, konon kopi dengan proses ini juga memiliki rasa-rasa yang lebih beragam.")

]

8. BackButton

```
import SwiftUI
```

```
struct BackButton: View {
    @Environment(\.presentationMode) var presentationMode
    var body: some View {
        Button(action: {
            presentationMode.wrappedValue.dismiss()
        }) {
            Rectangle()
                .fill(.clear)
                .frame(width: 100, height: 50)
                .overlay(
                    Image(systemName: "chevron.left")
                        .foregroundColor(Color("Secondary"))
                        .font(.system(size: 17, weight: .bold))
                        .imageScale(.large)
                        .offset(x: -27))
                }
        }
    }
}
```

Lampiran 3. Asset Penelitian



Gambar 21. Logo aplikasi.

Lampiran 4. Dokumentasi Penelitian



Gambar 22. Dokumentasi pengumpulan *dataset*.



Gambar 23. Dokumentasi pengumpulan *dataset*.



Gambar 24. Merek kopi yang digunakan untuk *dataset*.



Gambar 25. Toarco Toraja Coffe untuk evaluasi aplikasi.



Gambar 26. The Cold Crafters untuk evaluasi aplikasi.

Lampiran 5. Daftar Riwayat Hidup***CURRICULUM VITAE*****A. Data Pribadi**

1. Nama : Ayla Ainayyah. M
2. Tempat, tanggal lahir : Beru, 16 Mei 1999
3. Alamat : Pinrang, Sulawesi Selatan
4. Kewarganegaraan : Indonesia

B. Riwayat Pendidikan

1. Tamat SLTA tahun 2016 di SMAN 11 Unggulan Pinrang
2. Sarjana (S1) tahun 2021 di Universitas Hasanuddin

C. Pekerjaan dan Riwayat Pekerjaan

1. Mahasiswa S2 – Universitas Hasanuddin (Agustus 2022 – Juli 2024)
2. *Learner* – Apple Developer Academy (Maret 2023 – Desember 2023)
3. *Project Intern* – PT. Koltiva (Juli 2022 – Desember 2022)
4. Pendamping Lapangan – Pendampingan Program Pengembangan Budidaya Kedelai (P3BK) Universitas Hasanuddin (September 2021 – Desember 2021)
5. *Branch Consumer Funding Intern* – PT. Bank Tabungan Negara Tbk. (Maret 2021 – Agustus 2021)
6. *Student Volunteer* – Direktorat Komunikasi Universitas Hasanuddin (November 2020 – Februari 2021)