

# **Tugas 4**

## **IF4073 Interpretasi dan Pengolahan Citra**



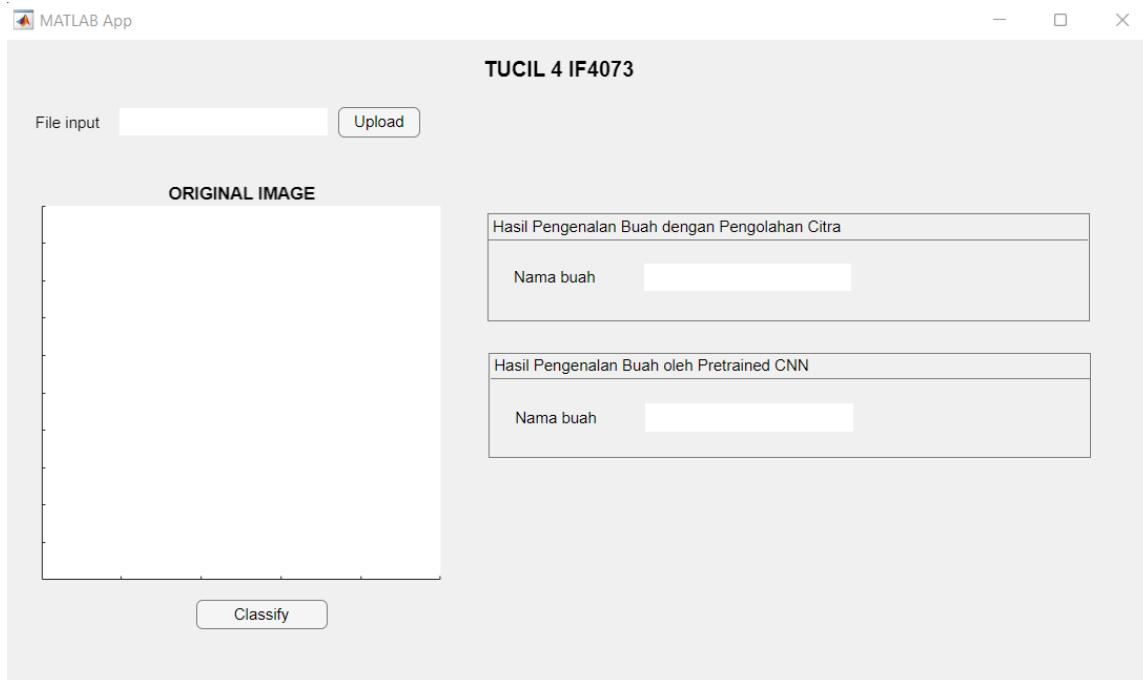
**Oleh:**

**13519034      Ruhiyah Faradishi Widiaputri**  
**13519042      Haning Nanda Hapsari**

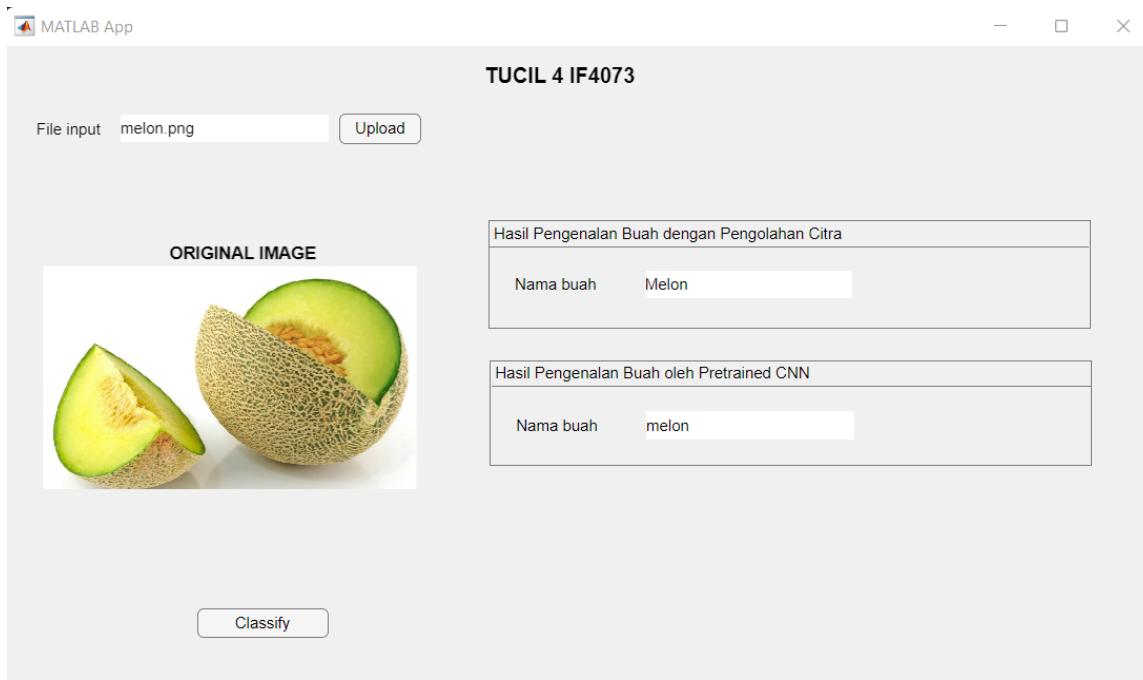
**Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2022/2023**

## A. Screenshot GUI

Berikut ini adalah tampilan GUI dari aplikasi yang telah dibuat.



Gambar: Tampilan awal



Gambar: Fruit Recognition

## B. Kode Program

Kode program dapat diakses di tautan ini [https://github.com/ruhiyahfw/IF4073\\_tucil4](https://github.com/ruhiyahfw/IF4073_tucil4)

1. Pengenalan buah-buahan dengan menggunakan teknik-teknik pengolahan citra  
Untuk mengklasifikasi buah-buahan yang diinputkan, dilakukan *training* menggunakan dataset beberapa buah yaitu apel (*apple*), pisang (*banana*), jeruk (*orange*), melon, dan pir (*pear*). Adapun kode program yang dilakukan untuk *training* disimpan dalam file *training\_image\_processing.m*. Isi dari program tersebut adalah sebagai berikut.

```
% Mencari lokasi dataset
cd('FruitsData');
% Nama masing-masing folder
fruitdata = {'apple';'banana';'orange';'melon';'pear'};
% Menghitung kategori buah yang di training
countfruit = length(fruitdata);
for i = 1:countfruit
    cd(char(fruitdata(i)));
    citradata = dir('*jpg');
    countdata = length(citradata);
    % Menghitung jumlah buah pada masing-masing kategori
    for j=1:countdata
        filename = citradata(j).name;
        % Mengubah citra menjadi grey scale
        image = imread(filename);
        if size(imread(filename),3) == 1
            continue
        else
            xcitra = rgb2gray(imread(filename));
        end
        % Menggunakan pendekatan GLCM dan warna
        features = graycoprops(graycomatrix(xcitra));
        featuresmat(j+countdata*(i-1),1) = features.Contrast;
        featuresmat(j+countdata*(i-1),2) = features.Correlation;
        featuresmat(j+countdata*(i-1),3) = features.Energy;
        featuresmat(j+countdata*(i-1),4) = features.Homogeneity;
        R = image(:,:,:1);
        G = image(:,:,:2);
        B = image(:,:,:3);
        featuresmat(j+countdata*(i-1),5) = R(22,50);
        featuresmat(j+countdata*(i-1),6) = G(22,50);
        featuresmat(j+countdata*(i-1),7) = B(22,50);

        kelas(j+countdata*(i-1)) = i;
    end
    cd('..');
end
cd('..');
% Menyimpan matrix GLCM dan kelas agar tidak dilakukan training
berulang
% kali
save ('features_mat.mat', 'featuresmat');
save ('kelas.mat','kelas');
```

Pada training dataset dilakukan operasi dengan pendekatan GLCM yaitu Grey Level Co-occurrence Matrix yaitu suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra. Pada training ini digunakan parameter berupa kontras, korelasi, entropi, dan *homogeneity* dari citra. Selain itu, digunakan juga pendekatan warna. Warna tersebut diubah dalam nilai spesifik bukan dalam bentuk matriks.

Setelah dilakukan training citra diklasifikasi dengan menggunakan program seperti berikut ini.

```
classdef ImageProcessingClassifier
methods (Static)
    function [klasifikasi] = predict(img)
        % Load matrix training
        filefeatures = load("features_mat.mat");
        featuresmat = filefeatures.featuresmat;
        filekelas = load('kelas.mat');
        kelas = filekelas.kelas;
        % Menggunakan metode KNN dengan machine learning
        model = fitcknn(featuresmat,kelas');
        if Helper.isGrayscale(img)
            m = graycomatrix(img);
        else
            m = graycomatrix(rgb2gray(img));
        end
        g = graycoprops(m);
        uji(1,1) = g.Contrast;
        uji(1,2) = g.Correlation;
        uji(1,3) = g.Energy;
        uji(1,4) = g.Homogeneity;
        R = img(:,:,:1);
        G = img(:,:,:2);
        B = img(:,:,:3);
        uji(1,5) = R(22,50);
        uji(1,6) = G(22,50);
        uji(1,7) = B(22,50);
        % Melakukan prediksi berdasarkan kesamaan kontras,
korelasi,
        % entropi, homogeneity, warna
        klasifikasi = predict(model,uji(1,:));
    end
end
end
```

Pada program tersebut digunakan *library machine learning* berupa KNN (K-Nearest Neighbor). Algoritma ini merupakan algoritma klasifikasi yang menitikberatkan pada jarak euclidean. Parameter yang digunakan dalam klasifikasi ini sama yaitu kontras, korelasi, entropi, *homogeneity*, dan warna dari citra.

## 2. Pengenalan buah-buahan dengan *pretrained* CNN

Pengenalan buah-buahan dengan *pretrained* CNN dilakukan dengan *fine tuning* model *pre-trained* AlexNet. Berikut ini adalah kode program dari *fine tuning* *pretrained* AlexNet untuk pengenalan buah-buahan.

```
% LOAD DATA
% first download FruitsData.zip here:
https://drive.google.com/file/d/102ID\_qzmz\_T\_Uimo4cRqUoGWm9w6m\_qB/view?usp=sharing
% then unzip it inside src folder
imds = imageDatastore('FruitsData', ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');

% SPLIT TRAIN AND TEST DATA
[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7, 'randomized');

% LOAD PRETRAINED NETWORK
net = alexnet;      % we will use pretrained AlexNet

% REPLACE FINAL LAYERS
% Extract all layers, except the last three, from the pretrained
% network.
layersTransfer = net.Layers(1:end-3);

% get number of labels from train data
numClasses = numel(categories(imdsTrain.Labels));

% create 3 new layers to be appended to layersTransfer
layers = [
    layersTransfer

    fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20, 'BiasLearnRateFactor',20)
        softmaxLayer
        classificationLayer];

% DATA AUGMENTATION
% reflection and translation: prevents the network from overfitting
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);

% resize training data so that it can be feed into the model
inputSize = net.Layers(1).InputSize;
augimdsTrain = augmentedImageDatastore([227 227],imdsTrain, ...
    'ColorPreprocessing', 'gray2rgb', ...
    'DataAugmentation',imageAugmenter);
```

```

% resize the validation data automatically
augimdsValidation = augmentedImageDatastore([227 227],imdsValidation,
'ColorPreprocessing', 'gray2rgb');

% SPECIFY THE TRAINING OPTIONS
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',10, ...
    'InitialLearnRate',1e-4, ... % set the initial learning rate to
a small value to slow down learning in the transferred layers
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'Verbose',false, ...
    'Plots','training-progress');

% TRAIN THE NETWORK
netTransfer = trainNetwork(augimdsTrain,layers,options);

% CLASSIFY VALIDATION IMAGES
[YPred,scores] = classify(netTransfer,augimdsValidation);

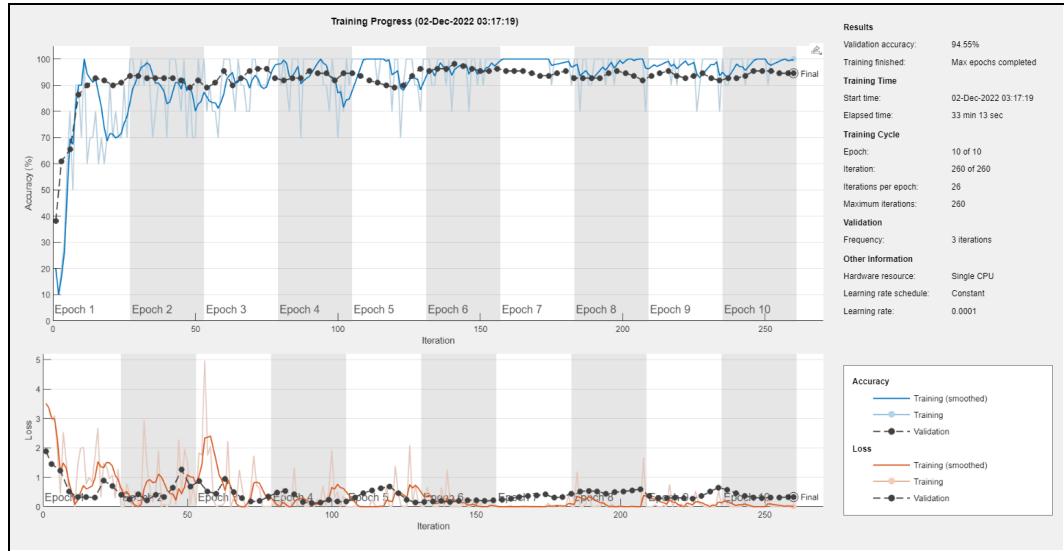
% CALCULATE CLASSIFICATION ACCURACY ON VALIDATION SET
YValidation = imdsValidation.Labels;
accuracy = mean(YPred == YValidation)

% SAVE TRAINED MODEL
save ('trained_fruit_classifier_034_042.mat', 'netTransfer')

```

*Fine tuning* model *pre-trained* AlexNet dilakukan dengan menggunakan citra-citra latih dari 5 buah yaitu apel (*apple*), pisang (*banana*), jeruk (*orange*), melon, dan pir (*pear*). Setiap jenis buah dikumpulkan 75 citra. Citra-citra ini dapat diakses di [https://drive.google.com/file/d/1O2ID\\_qzmz\\_T\\_Uimo4cRqUoGWm9w6m\\_qB/view?usp=sharing](https://drive.google.com/file/d/1O2ID_qzmz_T_Uimo4cRqUoGWm9w6m_qB/view?usp=sharing). Citra-citra ini kemudian dibagi menjadi data latih dan data validasi dengan perbandingan 7:3.

Model *pretrained* CNN yang digunakan pada tugas klasifikasi buah ini adalah model *pretrained* AlexNet. Untuk melakukan *fine tuning*, 3 *layer* terakhir dari *pretrained* AlexNet diganti dengan 3 *layer* baru untuk menyesuaikan keluaran dari yang aslinya 1000 label menjadi 5 label saja sesuai dengan banyak jenis buah yang ingin diklasifikasikan. Data latih sebelum digunakan untuk pelatihan model terlebih dahulu di augmentasi yaitu dengan ditranslasi dan dirotasi secara acak. Augmentasi citra ini bertujuan untuk mencegah *overfitting*. Data latih dan data validasi sebelum masuk ke dalam model juga dikenakan preproses yaitu dengan memastikan ukurannya adalah [227 227 3] sesuai dengan ukuran *input layer* AlexNet. Pelatihan dilakukan dengan ukuran *mini batch* = 10, maksimum *epoch* = 10 dengan pengacakan di setiap *epoch*, dan *learning rate* awal = 1e-4. Berikut ini adalah plot *fine tuning* model.



Dapat dilihat dari plot di atas model memiliki akurasi validasi mencapai 94.55%. Pelatihan berjalan selama 33 menit dan 13 sekon pada *single CPU*.

Model yang sudah dilatih kemudian disimpan ke file `trained_fruit_classifier_034_042.mat` yang dapat diakses di [https://drive.google.com/file/d/1yK3l8uxnsqQM-pRQ5zPnbLEcUcSa9\\_NS/view?usp=sharing](https://drive.google.com/file/d/1yK3l8uxnsqQM-pRQ5zPnbLEcUcSa9_NS/view?usp=sharing).

Kode program klasifikasi dengan model yang sudah di *fine tune* ini adalah sebagai berikut.

```

classdef PretrainedCNNClassifier
methods (Static)
    function [YPred, score] = predict(image)
        % load fine-tuned model
        file = load("trained_fruit_classifier_034_042.mat");
        newnet = file.netTransfer;

        % resize image
        im = imresize(image, [227 227]);

        % classify image
        [YPred,scores] = classify(newnet, im);

        % get highest score
        score = int8(max(scores))*100;
        score = int2str(score);
    end
end
end

```

Model yang telah disimpan pertama-tama dimuat (perhatikan bahwa file model yang telah disimpan berada dalam folder yang sama dengan `PretrainedCNNClassifier.m` yaitu di folder `src`). Kemudian citra yang akan

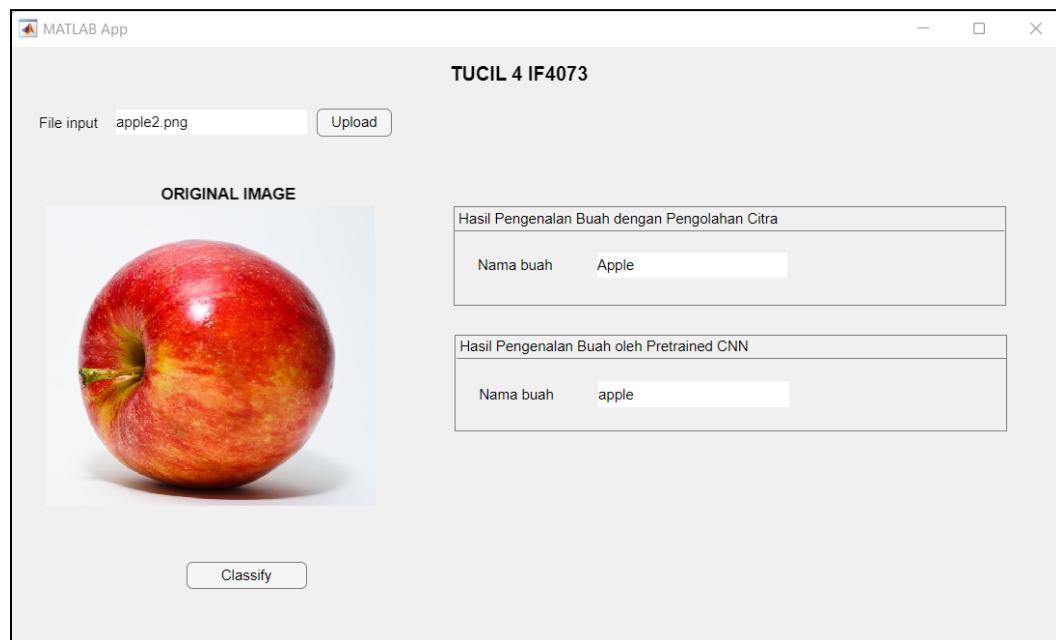
diprediksi disesuaikan ukurannya menjadi [227 227] agar bisa masuk ke dalam model. Selanjutnya dilakukan prediksi kelas citra dengan menggunakan fungsi *classify*.

C. Contoh hasil eksekusi program dan analisis hasil

Berikut ini merupakan contoh hasil dari eksekusi program.

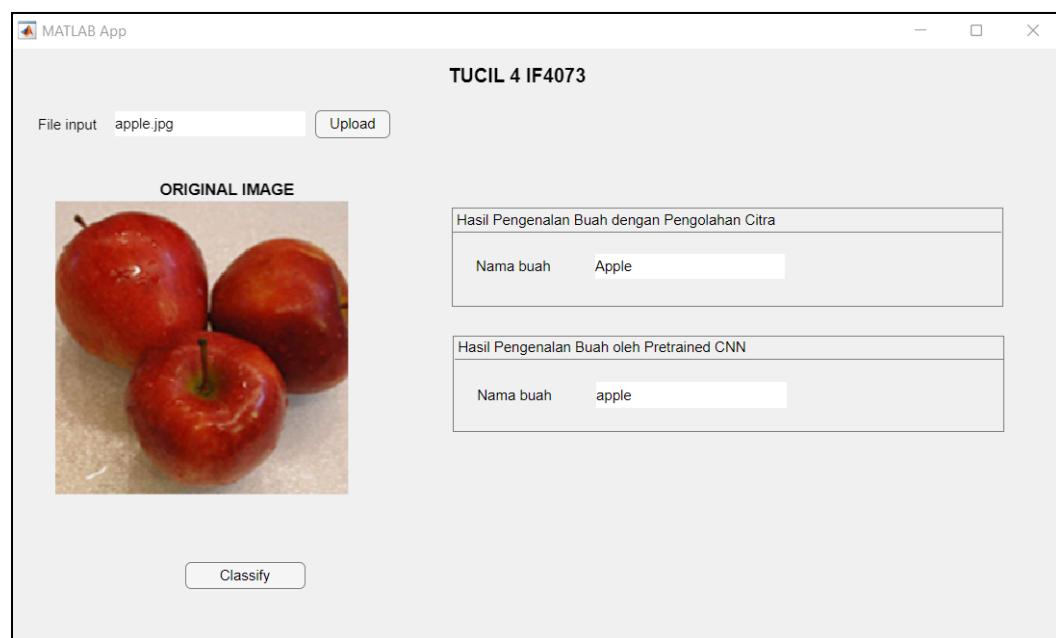
1. Pengenalan citra apel

a. Citra 1

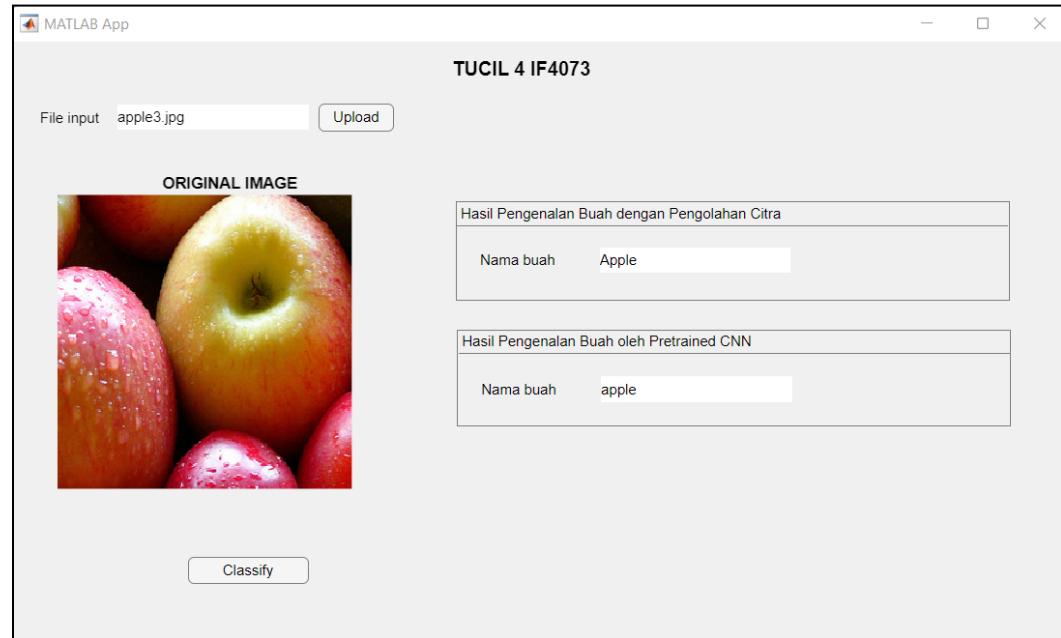


Kedua metode sudah bisa mengenali apel di atas dengan benar.

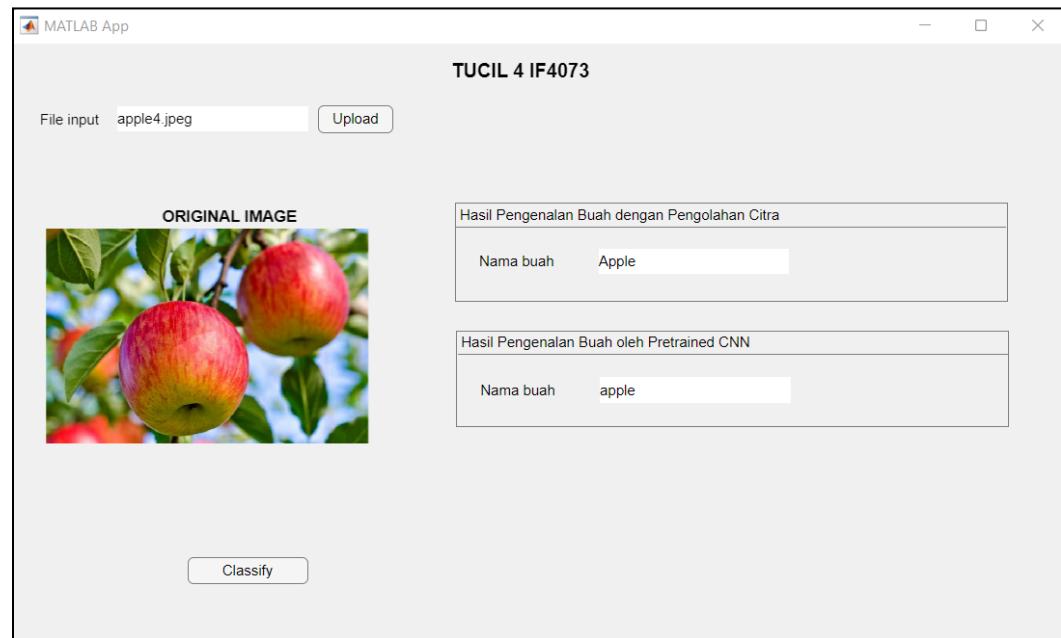
b. Citra 2



Kedua metode sudah bisa mengenali apel di atas dengan benar.  
c. Citra 3

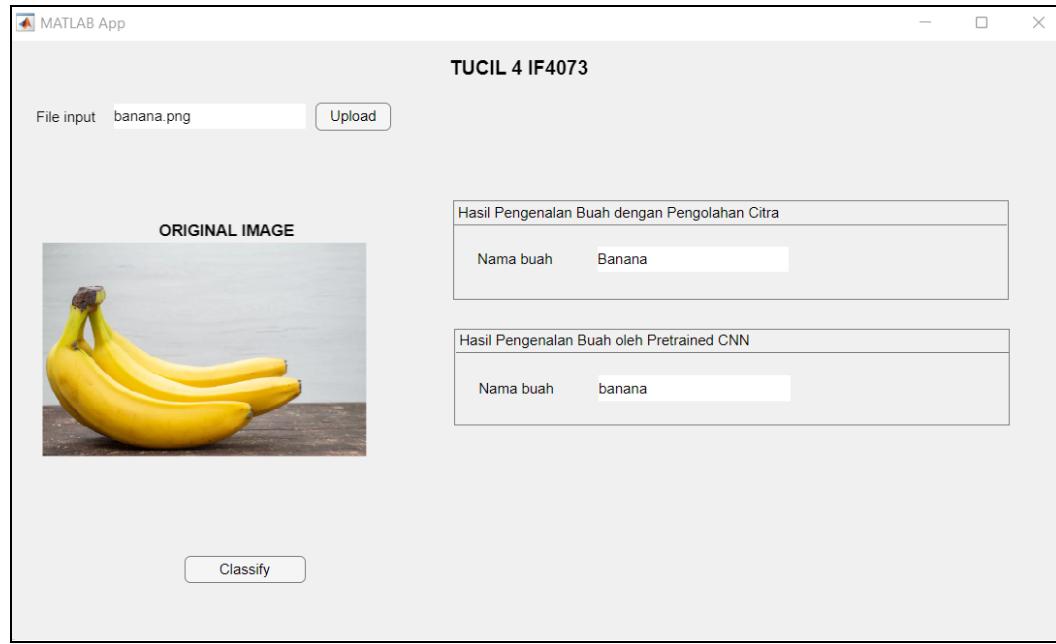


Kedua metode sudah bisa mengenali apel di atas dengan benar.  
d. Citra 4

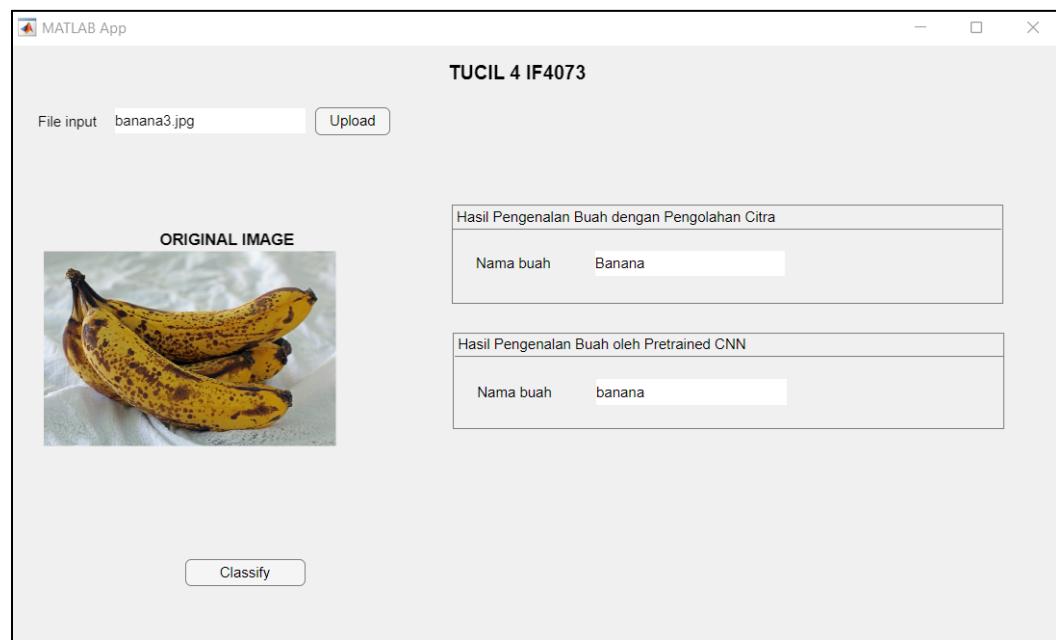


Kedua metode sudah bisa mengenali apel di atas dengan benar.

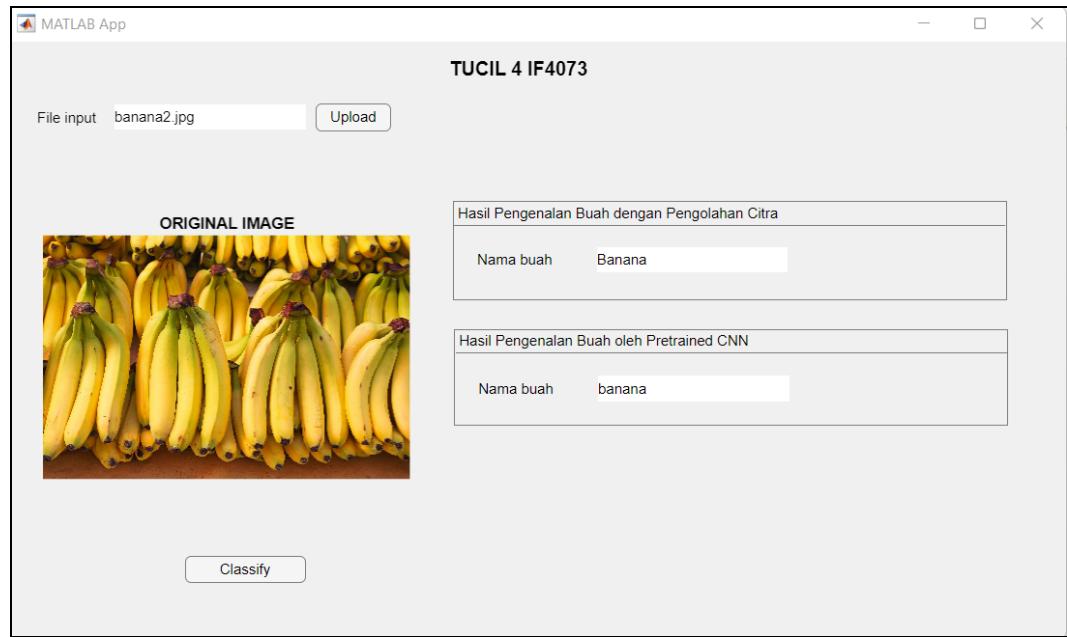
2. Pengenalan citra pisang
  - a. Citra 1



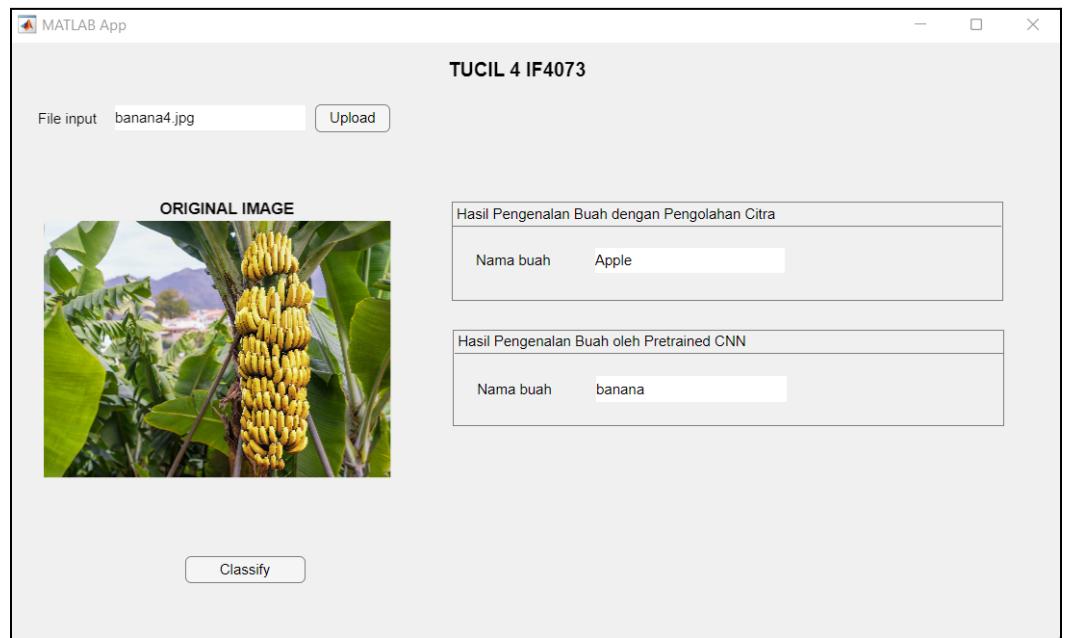
Kedua metode sudah bisa mengenali pisang di atas dengan benar.  
b. Citra 2



Kedua metode sudah bisa mengenali pisang di atas dengan benar.  
c. Citra 3

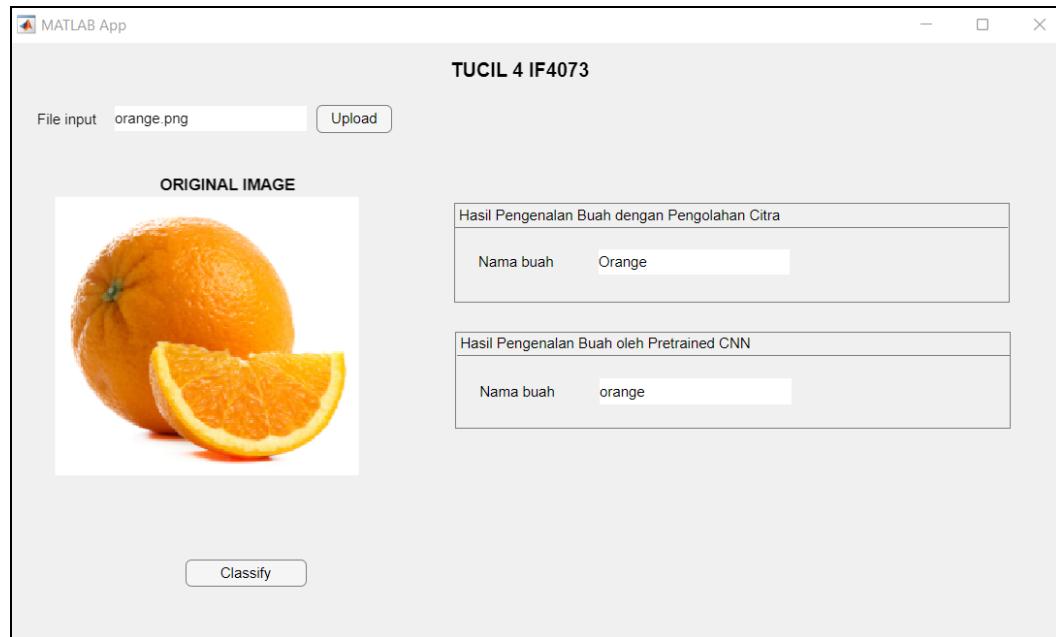


Kedua metode sudah bisa mengenali pisang di atas dengan benar.  
d. Citra 4

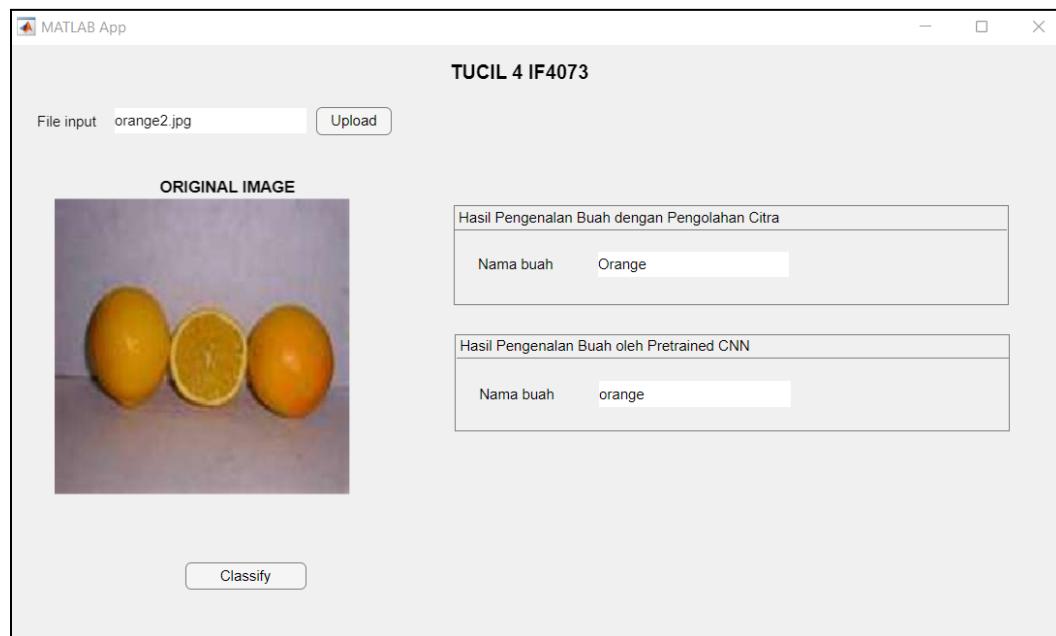


Model *finetuned* CNN berhasil mengenali citra pisang di atas sedangkan metode pengenalan citra gagal mengenali citra pisang di atas (hasil prediksinya adalah apel).

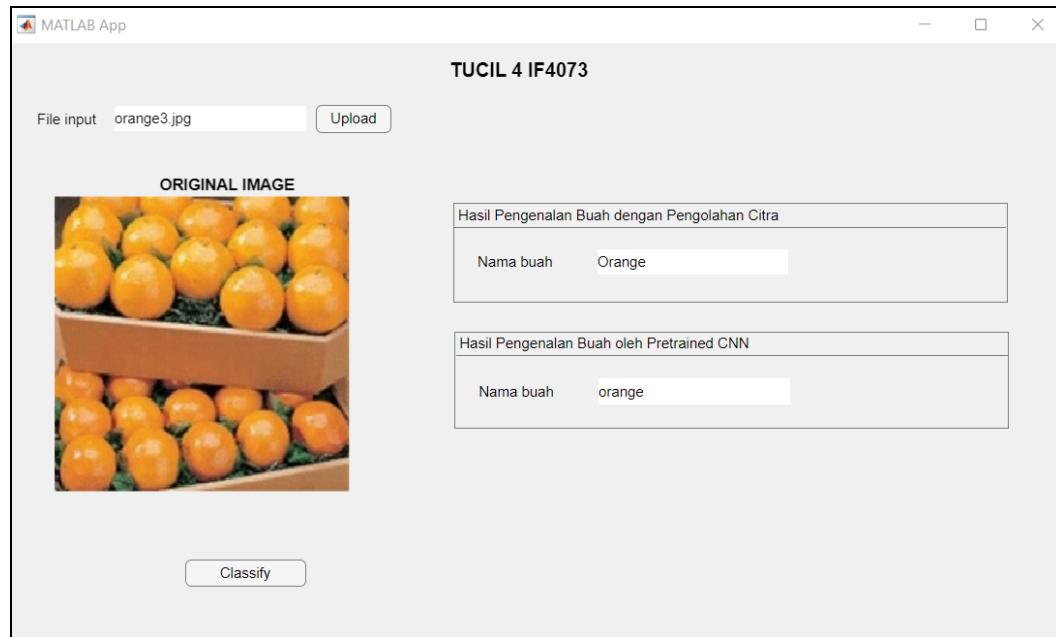
3. Pengenalan citra jeruk
  - a. Citra 1



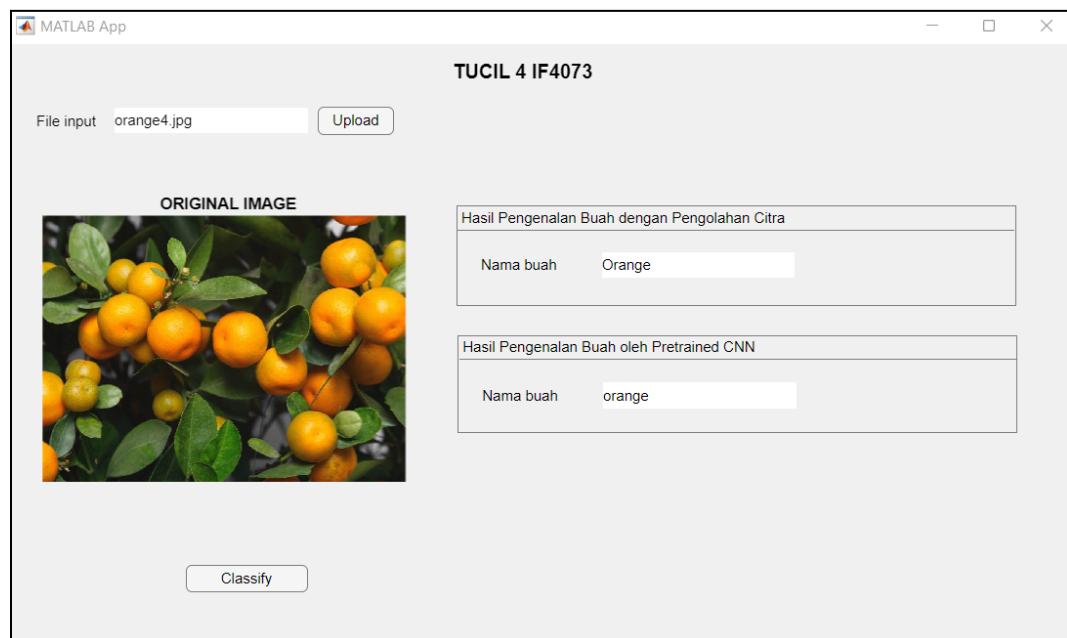
Kedua metode sudah bisa mengenali jeruk di atas dengan benar.  
b. Citra 2



Kedua metode sudah bisa mengenali jeruk di atas dengan benar.  
c. Citra 3

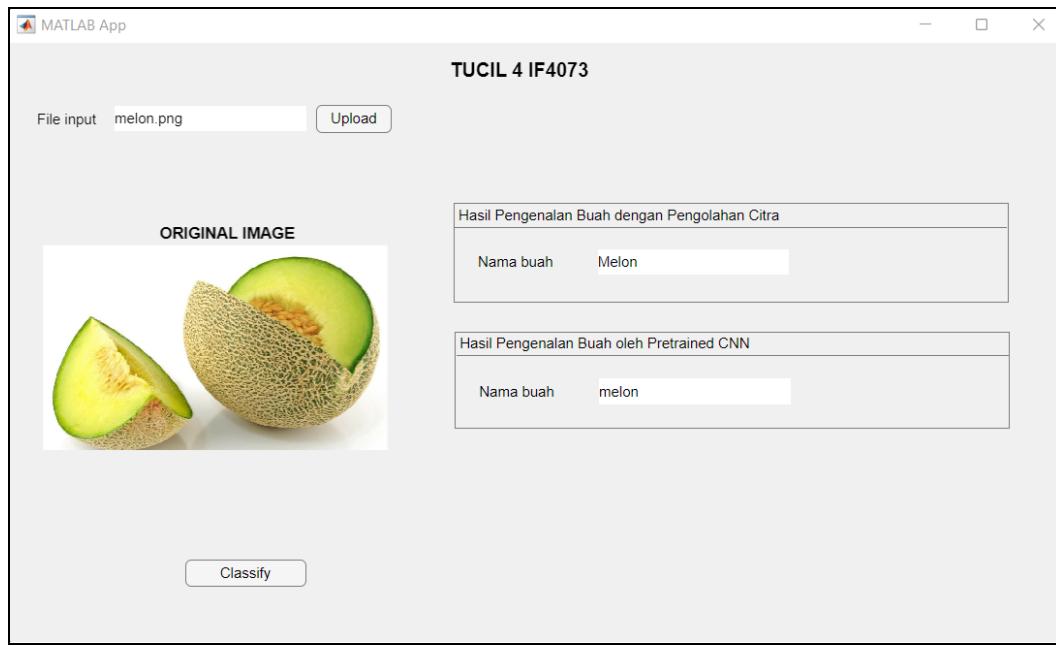


Kedua metode sudah bisa mengenali jeruk di atas dengan benar.  
d. Citra 4

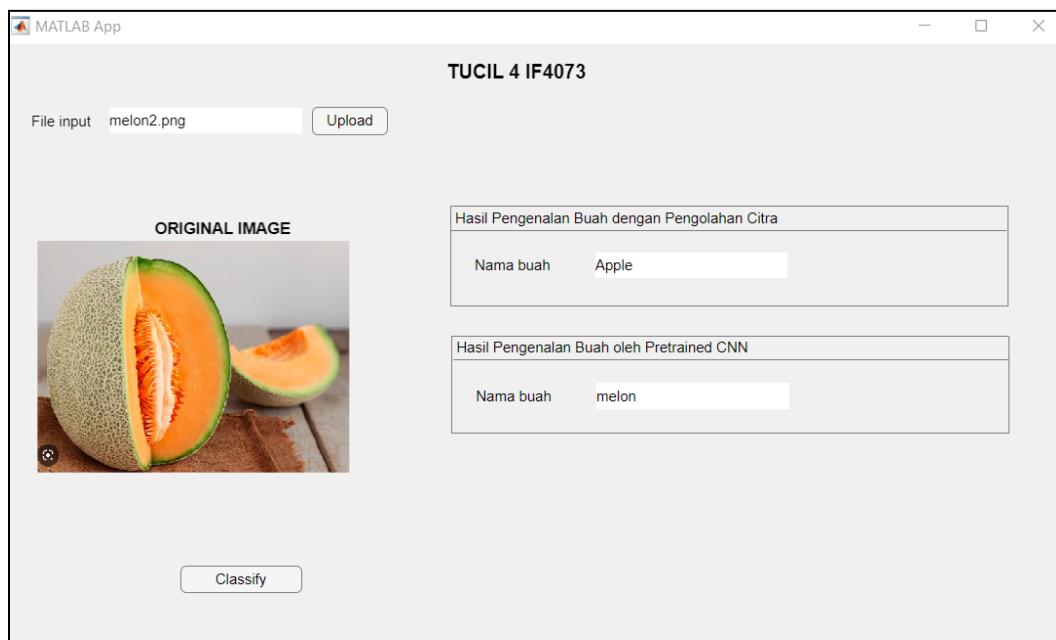


Kedua metode sudah bisa mengenali jeruk di atas dengan benar.

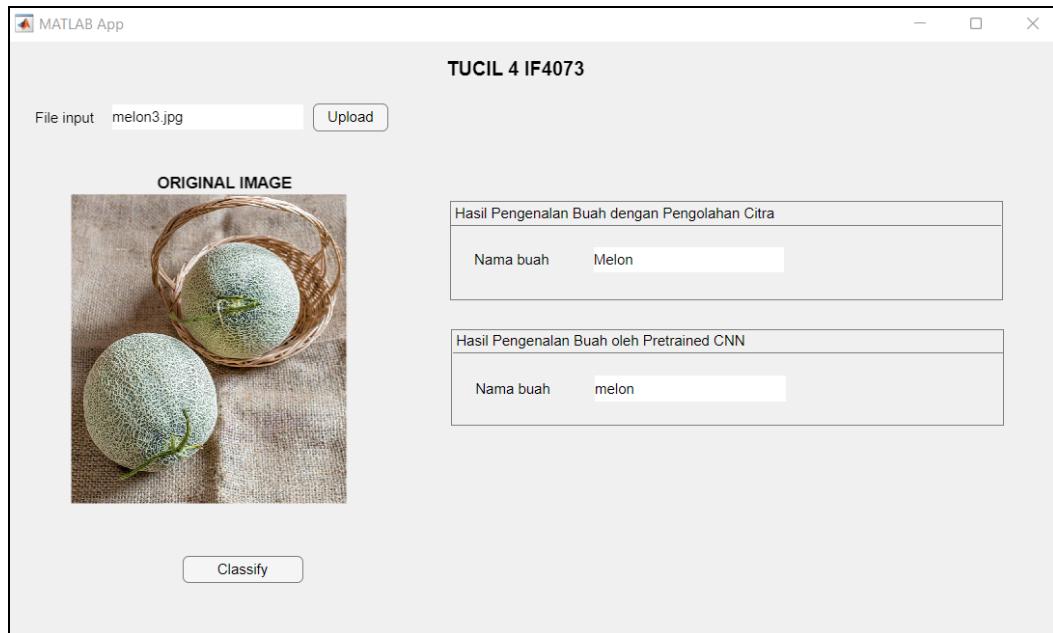
4. Pengenalan citra melon
  - a. Citra 1



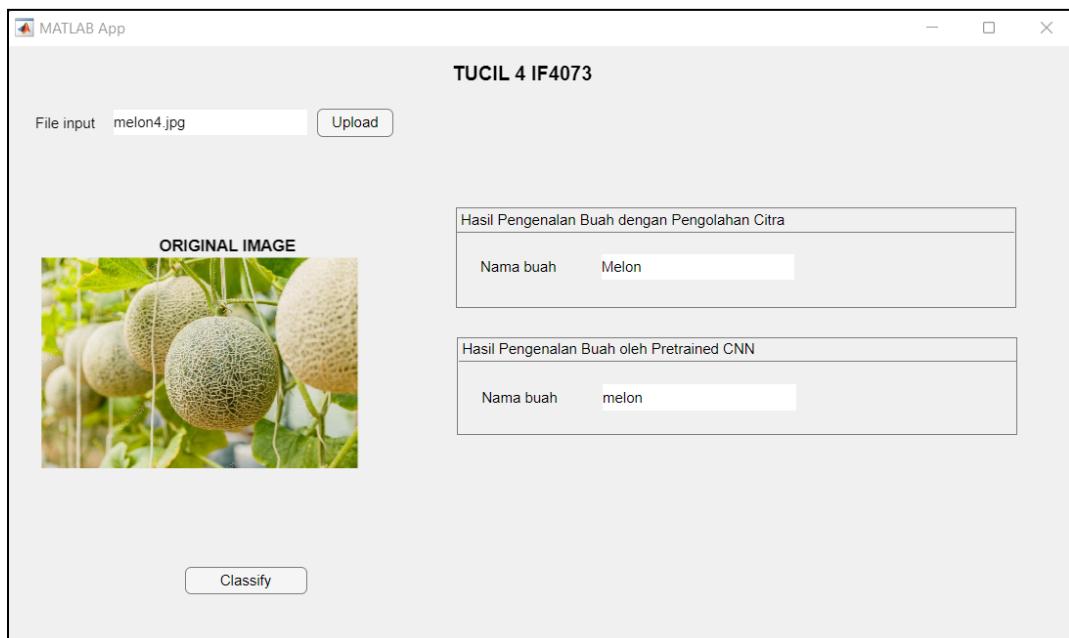
Kedua metode sudah bisa mengenali melon di atas dengan benar.  
b. Citra 2



Model *finetuned* CNN berhasil mengenali citra melon di atas sedangkan metode pengenalan citra gagal mengenali citra melon di atas (hasil prediksinya adalah apel).  
c. Citra 3

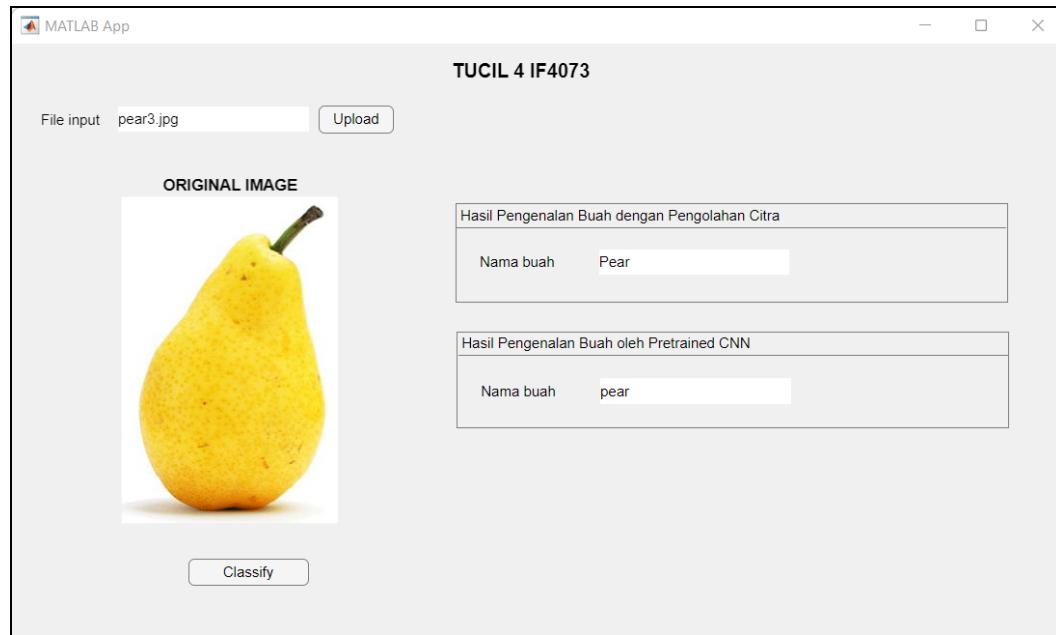


Kedua metode sudah bisa mengenali melon di atas dengan benar.  
d. Citra 4

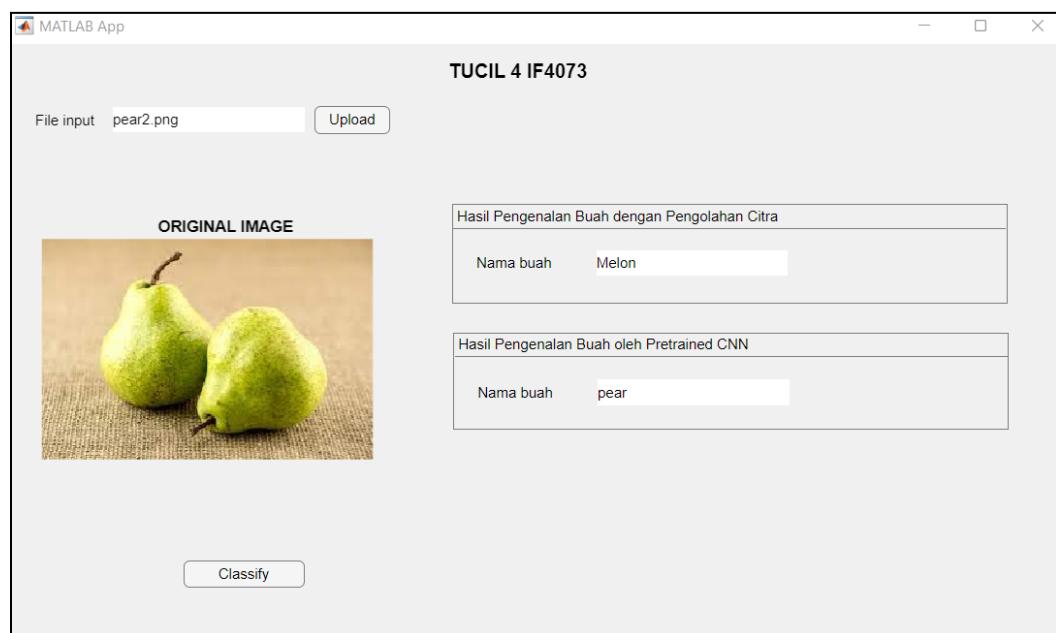


Kedua metode sudah bisa mengenali melon di atas dengan benar.

5. Pengenalan citra pir
- a. Citra 1

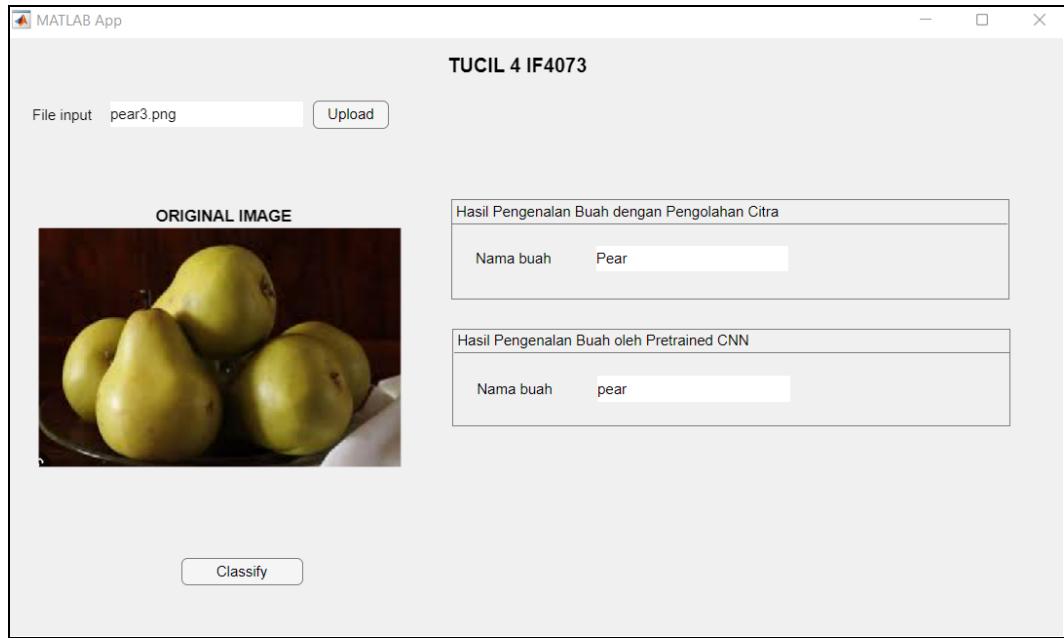


Kedua metode sudah bisa mengenali pear di atas dengan benar.  
b. Citra 2

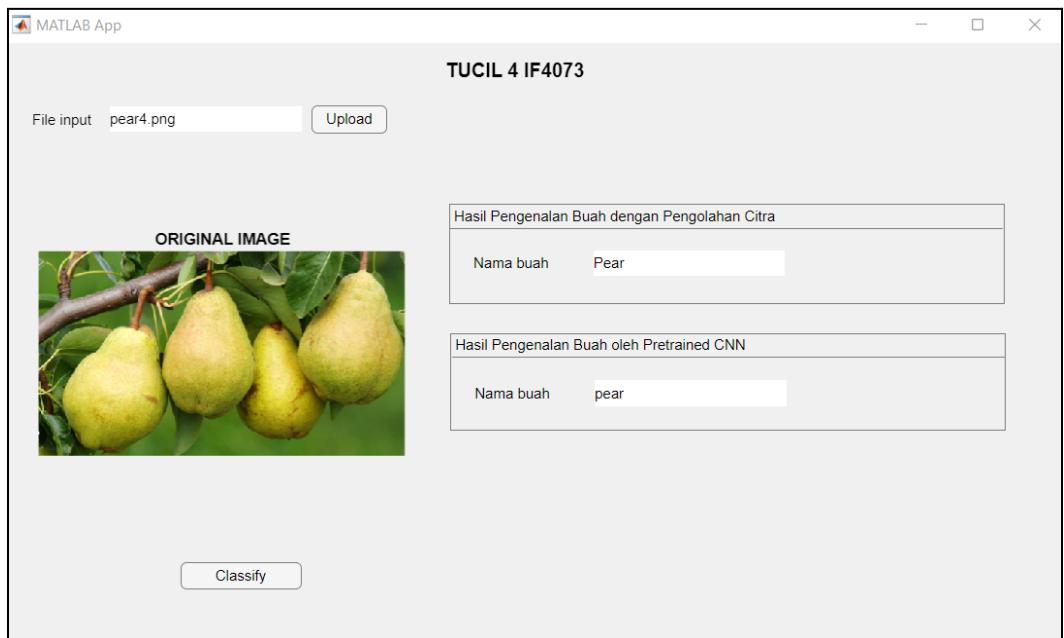


Model *finetuned* CNN berhasil mengenali citra pear di atas sedangkan metode pengenalan citra gagal mengenali citra pear di atas (hasil prediksinya adalah melon).

c. Citra 3



Kedua metode sudah bisa mengenali pear di atas dengan benar.  
d. Citra 4



Kedua metode sudah bisa mengenali pear di atas dengan benar.

Pada hasil pengenalan buah menggunakan *deep learning*, hampir semua buah dapat diklasifikasikan dengan benar. Sedangkan pengenalan menggunakan metode pengolahan citra tidak semua buah dapat diklasifikasikan dengan benar. Hal ini dapat terjadi karena pendekatan yang digunakan yaitu GLCM dan pendekatan warna disertai dengan algoritma *machine learning* KNN dengan menggunakan metode *euclidean distance*. Kesalahan pengklasifikasian ini dapat disebabkan oleh beberapa faktor yaitu parameter yang kurang spesifik dan jumlah data training yang kurang

banyak sehingga cara meminimalisasi kekurangan ini yaitu dengan menambahkan parameter uji yang lebih spesifik dan memperbanyak dataset.

D. Kesimpulan

Berdasarkan hasil dari kedua metode yaitu menggunakan pengolahan citra (*machine learning*) dan *deep learning*, klasifikasi menggunakan *deep learning* memiliki akurasi yang lebih baik jika dibandingkan dengan *machine learning*. Meskipun demikian kedua metode pengenalan ini sudah baik dalam mengklasifikasikan citra buah. Dapat dilihat kedua metode dapat mengklasifikasikan buah baik pada citra dengan latar belakang polos, citra dengan latar belakang tidak polos, citra kumpulan atau seonggok buah, dan bahkan citra buah di pohon.

E. Komentar dan refleksi

Tugas 4 IF4073 sangat menarik karena sangat berbeda dengan tugas-tugas sebelumnya. Pada tugas sebelumnya mahasiswa hanya diminta untuk melakukan berbagai macam manipulasi citra, sedangkan pada tugas kali ini kemampuan untuk manipulasi citra tersebut digunakan untuk menyelesaikan masalah sehari-hari.