

# RANCANG BANGUN APLIKASI KLASIFIKASI TINGKAT KEMATANGAN SANGRAI KOPI MELALUI CITRA DIGITAL MENGGUNAKAN CNN (CONVOLUTIONAL NEURAL NETWORK ) BERBASIS ANDROID

Edward Hendryawan Michael<sup>1</sup>, Kestrilia Rega Prilianti<sup>2</sup>, Mochamad Subianto<sup>3</sup>

Teknik Informatika Universitas Machung Villa Puncak Tidar N-1 Malang email

korespondensi: [311610005@student.machung.ac.id](mailto:311610005@student.machung.ac.id)

## Abstrak

Proses sangrai kopi merupakan proses yang penting, bahkan secara persentase proses sangrai berpengaruh sebesar hingga 30% dalam sumbangan aroma dan rasa. Namun selama ini proses sangrai kopi hanya dapat dilakukan oleh seseorang yang sudah ahli dalam menyangrai kopi sehingga dapat menentukan tingkat kematangan kopi yang benar dan tepat. Selain itu dalam menentukan tingkat kematangan sangrai kopi masih banyak berdasarkan pendapat subjektif seseorang. Di Indonesia sendiri tingkat kematangan sangrai kopi dikategorikan ke dalam tiga tingkatan yaitu *light roast*, *medium roast*, dan *dark roast*. Maka dari itu penulis, merancang sebuah aplikasi yang mampu mengklasifikasikan tingkat kematangan sangrai kopi. Aplikasi ini bertujuan untuk dapat membantu standarisasi dalam menentukan tingkat kematangan sangrai kopi bagi orang-orang yang mencoba melakukan sangrai kopi secara mandiri. Aplikasi ini dibentuk dalam bentuk aplikasi pada perangkat *mobile* android untuk memudahkan pengguna sehingga dapat digunakan kapanpun disaat yang diinginkan. Pada penelitian ini dilakukan percobaan menggunakan kombinasi arsitektur CNN antara LeNet5, AlexNet, MiniVGGNet dengan fungsi optimasi ADAM, SGD, dan NADAM. Penelitian ini berhasil menghasilkan model dengan akurasi tertinggi hingga 98% menggunakan kombinasi antara LeNet5 dengan fungsi optimasi ADAM. Model inilah yang digunakan sebagai opsi standar pada aplikasi ketika melakukan klasifikasi sangrai kopi.

**Kata kunci:** sangrai, kopi, android, LeNet5, AlexNet, MiniVGGNet, ADAM, SGD, NADAM

## Abstract

*The process of roasting coffee is an important process, even the percentage of roasting process influences up to 30% in the contribution of aroma and taste. But so far the process of roasting coffee can only be done by someone who is an expert at roasting coffee so that it can determine the correct and appropriate level of roasted coffee. In determining the level of roasted coffee is still a lot based on someone's subjective opinion. In Indonesia the level of roasted coffee is categorized into three levels, light roast, medium roast, and dark roast. Therefore the author, designed an application that is able to classify the level of roasted coffee. This application aims to be able to help standardize in determining the level of roasted coffee for people who are trying to do coffee roasting independently. This application was formed in form of an application on an android mobile device to facilitate the use so that it can be used whenever desired. In this research was conducted using combination of CNN architecture between LeNet5, AlexNet, MiniVGGNet with ADAM, SGD, and NADAM optimization functions. This research succeeded in producing a model with highest accuracy 98% using a combination of LeNet5 with ADAM optimization functions. This model is used as a default option in application when classifying roasted coffee.*

**Keywords:** roast, coffee, android, LeNet5, AlexNet, MiniVGGNet, ADAM, SGD, NADAM

## 1. Pendahuluan

Kopi adalah jenis minuman seduh yang sudah dinikmati sejak lama. Dulunya kopi diminati oleh para orang tua, namun saat ini kopi juga menjadi minuman favorit bagi para remaja dengan munculnya tren kafe-kafe yang digunakan sebagai tempat nongkrong. Dalam menghasilkan kopi yang nikmat biji kopi melalui proses yang namanya *coffee roasting* atau sangrai biji kopi. Proses sangrai ini dapat mempengaruhi kualitas, dan bahkan secara persentase berpengaruh hingga 30% dalam menyumbangkan rasa dan aroma pada biji kopi. Dalam tingkat kematangan sangrai kopi, di Indonesia pada umumnya menggunakan tiga jenis tingkatan yaitu *light roast*, *medium roast*, dan *dark roast*. Sedangkan secara internasional terdapat 8 tingkatan yang lebih detail yaitu *light*, *cinnamon*, *medium*, *high*, *city*, *full city*, *french*, dan *italian* untuk merepresentasikan tingkat kematangan sangrai kopi. Seluruh tingkat kematangan kopi ini sudah tepat tergantung selera peminumnya.

Selama ini untuk dapat menentukan sebuah tingkat kematangan sangrai kopi dilakukan secara manual dan diperlukan seorang pakar atau ahli pada bidang ini. Berdasarkan latar belakang masalah tersebut maka diperoleh rumusan masalah, yaitu selama ini penentuan tingkat kematangan sangrai kopi masih berdasarkan pendapat dari berbagai orang yang bisa sangat subjektif. Hal ini dapat menimbulkan masalah ketidakpuasan pelanggan, atau ketika melakukan proses sangrai kopi secara mandiri hanya dapat menentukan berdasarkan perkiraan sendiri. Oleh karena itu dikembangkan ide untuk membuat sebuah aplikasi yang secara otomatis dapat mengklasifikasikan tingkat kematangan sangrai kopi memanfaatkan *smartphone* android, sehingga kapanpun dan dimanapun dapat digunakan dengan praktis. Dengan memanfaatkan *convolutional neural network* sebagai *machine learning* untuk melakukan proses klasifikasi tingkat kematangan sangrai kopi ini diharapkan mampu membuat sistem yang lebih efektif, efisien, serta *realtime*.

## 2. Metode / Algoritma

### 2.1. Analisis Kebutuhan

Untuk mendukung pengerjaan penelitian ini diperlukan analisis apa saja kebutuhan yang perlu dipersiapkan. Hal-hal yang perlu dipersiapkan adalah seperti *hardware* yang digunakan, *software* apa saja yang diperlukan, kebutuhan sebuah pakar untuk dapat digunakan sebagai referensi, serta analisis kebutuhan apa saja yang diperlukan dari sisi pengguna.

## 2.2. Studi Pustaka

Studi pustaka dilakukan dengan tujuan untuk mencari dan mempelajari berbagai referensi sebagai dasar untuk langkah penelitian selanjutnya. Pada tahap ini dilakukan pencarian informasi yang berkaitan dengan pengolahan citra *digital*, bagaimana struktur *convolutional neural network* yang dapat digunakan agar memiliki pemrosesan citra dengan efektif dan akurat, dan berbagai referensi yang dapat membantu dalam membuat aplikasi. Tujuan dilakukan studi pustaka adalah agar semakin baik pemahaman pengolahan citra sehingga menghasilkan sistem yang dapat mengenali warna dari tingkat kematangan sangrai kopi dengan baik. Selain itu dengan membaca berbagai referensi dapat mengetahui perkembangan teknologi yang ada sehingga menghindari kasus plagiarisme.

## 2.3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang berupa citra biji kopi mentah serta ke-3 jenis sangrai kopi yaitu *light-roast*, *medium-roast*, *dark-roast*. Jumlah citra yang diambil sebanyak 240 citra kopi. Dari hasil seluruh sangrai, biji kopi dibagi menjadi 60 citra mentah, 60 citra *light roast*, 60 citra *medium roast*, dan 60 citra *dark roast*. Selesai mengambil citra digital kopi tersebut, maka dilakukan pergantian biji kopi yang diambil dari hasil sangrai atau biji mentah yang tersisa. Pada saat pengambilan citra akan diperlakukan pencahayaan yang berbeda-beda yaitu ketika pencahayaan ruangan cukup terang dan ketika ruangan tidak terlalu terang.

## 2.4 Pre-processing Citra

Pada penelitian ini digunakan 2 jenis *preprocessing* citra yang digunakan yaitu *histogram equalization* pada citra digital dan *resize*. Untuk penjelasan dari masing-masing *pre-processing* tersebut sebagai berikut:

### 1) Histogram Equalization

*Histogram equalization* pada citra warna (RGB) dan *resize*. *Histogram equalization* digunakan untuk menangani masalah pencahayaan yang tidak merata pada citra. *Histogram equalization* dilakukan dengan membaca masing-masing *pixel* dari citra dengan nilai rentang 0 sampai 255 dan dijadikan *input* pada *histogram*. Setelah membaca seluruh nilai *pixel* maka dilakukan distribusi kumulatif.

### 2) Resize

*Resize* atau mengubah sebuah ukuran, berguna untuk menyesuaikan ukuran citra agar menjadi ukuran *input* yang pas dengan arsitektur CNN yang digunakan dalam penelitian ini.]

## 2.5. Desain Jaringan Syaraf Tiruan

Dalam tahap ini akan dibuat desain dari CNN yang akan digunakan dalam klasifikasi tingkat kematangan sangrai kopi. Arsitektur yang akan digunakan dalam penelitian ini adalah LeNet5, AlexNet, dan MiniVGGNet. Dipilihnya ketiga arsitektur tersebut karena berdasarkan kedalaman

dan kekompleksan strukturnya. Kemudian alasan lainnya dipilihnya menggunakan 3 jenis arsitektur dalam penelitian ini adalah untuk menemukan arsitektur mana yang dapat menghasilkan tingkat akurasi yang baik dalam mengklasifikasikan tingkat kematangan sangrai kopi.

### 1) LeNet5

Arsitektur LeNet yang digunakan adalah LeNet5 yang terdiri dari 2 *convolution layer*, 2 *pooling layer*, dan 1 *fully connected layer*. Komponen kuncinya adalah lapisan konvolusi yang dimilikinya. Lapisan-lapisan ini mengeksplorasi struktur data gambar. Lapisan *convolutional* menghubungkan setiap *output* kepada beberapa *input* terdekat. Secara intuitif, ini berarti layer akan mempelajari fitur-fitur lokal. Lapisan *pooling* kemudian menggabungkan *input* terdekat [1].

### 2) AlexNet

Arsitektur berikutnya yang akan digunakan dalam penelitian adalah AlexNet dibuat oleh Alex Krizhevsky pada tahun 2012. Arsitektur dari AlexNet yang akan digunakan tersusun dari 5 *convoluitonal layers* dan 3 *fully connected layer*. Dua lapisan *convolutional* pertama diikuti oleh lapisan *Max Pooling* yang saling tumpang tindih. Lapisan *convolutional* ketiga, keempat dan kelima terhubung secara langsung. Lapisan *convolutional* kelima diikuti oleh lapisan *Max Pooling* yang saling tumpang tindih, yang hasilnya menjadi serangkaian dua lapisan yang terhubung sepenuhnya. Lapisan kedua yang terhubung sepenuhnya dimasukkan ke dalam sebuah *classifier softmax* dengan sejumlah 1000 label kelas, namun pada penelitian ini hanya terdiri 4 *class* maka nilai 1000 tersebut diganti menjadi 4. Lapisan *max pooling* biasanya digunakan untuk mengecilkan lebar dan tinggi tensor, menjaga kedalamannya tetap sama. Lapisan *max pool* yang tumpang tindih mirip dengan layer *max pool*, kecuali jendela yang berdekatan tempat *max* dihitung saling tumpang tindih.. Sifat

*pooling* yang tumpang tindih ini membantu mengurangi tingkat kesalahan top-1 sebesar 0,4% dan tingkat kesalahan top-5 masing-masing sebesar 0,3% bila dibandingkan dengan menggunakan *windows pooling* non-tumpang tindih ukuran  $2 \times 2$  dengan langkah 2 yang akan menghasilkan *output* berukuran sama [2].

### 3) MiniVGGNet

MiniVGGNet merupakan sebuah arsitektur VGGNet atau biasa disebut juga sebagai VGG yang disederhanakan sedemikian rupa. Pada umumnya VGG16 memiliki struktur yang terdiri dari 12 *convolution layer* yang terkadang diikuti oleh *maximum pooling layer* dan 4 *fully connected layer*. Arsitektur VGG sendiri sangatlah kompleks sehingga tidak semua perangkat dapat menjalankan arsitektur tersebut, oleh karena itu dengan MiniVGGNet ini dapat menyederhanakan VGG yang ada namun dengan performa yang tidak jauh berbeda. MiniVGGNet sendiri terdiri dari 4 *convolution layer*, 2 *pooling layer*, dan 2 *fully connected layer*. Ciri unik dari MiniVGGNet adalah setiap 2 *convolution layer* diikuti oleh *pooling layer*.

## 2.6. Metode Optimasi

Dalam penelitian ini akan digunakan 3 jenis metode optimasi yaitu *stochastic gradient descent*(SGD), *adaptive momentum* (ADAM), dan *nesterov adaptive momentum*(NADAM). Seluruh fungsi optimasi akan diuji pada setiap arsitektur yang digunakan dalam penelitian. Dari pengujian kombinasi antara arsitektur dengan fungsi optimasi ini akan ditemukan kombinasi mana yang dapat menghasilkan model tercerdas dengan akurasi yang terbaik. Untuk penjelasan masing-masing fungsi optimasinya sebagai berikut:]

### 1) Stochastic gradient descent (SGD)

*Stochastic gradient descent* memperkirakan gradien hanya dengan menggunakan satu titik data. Dalam mengevaluasi gradien hanya memerlukan waktu yang singkat dibandingkan dengan menjumlahkan semua data. Gradien yang dihitung untuk elemen tertentu diambil sebagai perkiraan gradien nyata. Bobot dalam model dihitung ulang sesuai dengan gradien yang dihitung untuk satu elemen, yang mengarah pada fakta bahwa model tersebut disesuaikan ketika bergerak dari setiap elemen berturut-turut sampel ke yang berikutnya. SGD dirumuskan sebagai berikut [3]:

$$(1) \theta_{t+1} = \theta_t - \eta \cdot \nabla J(\theta_t)$$

Pembaruan parameter pada waktu t +1 dihitung menggunakan persamaan,  $\eta$  adalah tingkat belajar dan  $J$  adalah fungsi kesalahan. Pembaruan sering di SGD dapat memberikan kemungkinan untuk menemukan lokal minum baru dan lebih baik. Namun dampaknya sering pembaruan adalah kesulitan dalam memperoleh konvergensi. 2) *Adaptive Momentum* (ADAM)

ADAM menghitung tingkat pembelajaran adaptif untuk setiap parameter. ADAM menyimpan gradien kuadrat dari *epoch* sebelumnya yang telah dikoreksi oleh bias. Sebagai tambahan, ADAM juga menyimpan gradien rata-rata masa lalu yang dikoreksi ( $\hat{m}_t$ ). Kedua nilai ( $\hat{v}_t$  dan  $\hat{m}_t$ ) memperkirakan momen pertama yang membusuk dan momen kedua dari gradien yang dihitung oleh persamaan (2) - (5) [4],  $\beta$  adalah konstanta peluruhan (5). Itu pembaruan parameter pada langkah waktu t + 1 dihitung menggunakan persamaan (6) [5]

$$(2) \hat{m}_t = \beta \hat{m}_{t-1} + (1 - \beta) g_t$$

$$(3) \hat{v}_t = \beta^2 \hat{v}_{t-1} + (1 - \beta^2) g_t^2$$

$$(4) \hat{m}_t = \beta_1 \hat{m}_{t-1} + (1 - \beta_1) g_t$$

$$(5) \hat{v}_t = \beta_2 \hat{v}_{t-1} + (1 - \beta_2) g_t^2$$

$$(6) \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t$$

### 3) Nesterov Adaptive Momentum (NADAM)

NADAM adalah kombinasi dari ADAM dan *Nesterov Accelerated Gradient* (NAG). Ini menggabungkan momentum Nesterov ke dalam ADAM [5]. Karena momentum Nesterov dapat dicapai hasil yang lebih baik daripada momentum klasik, dihipotesiskan bahwa NADAM akan lebih baik daripada ADAM. Parameter pembaruan pada langkah waktu t + 1 dihitung menggunakan persamaan sebagai berikut)

$$(7) \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot (\beta_1 \cdot \hat{m}_t + \frac{(1 - \beta_1)}{1 - \beta_1^t} g_t)$$

## 2.7. Alur dan Desain Aplikasi

Dalam penelitian ini, aplikasi dibuat untuk perangkat *smartphone* berbasis android yang bersifat dapat dibawa kemana saja. Dengan menggunakan perangkat *smartphone* harapannya adalah agar memudahkan *user*, serta dapat digunakan secara cepat dan praktis. Alasan lainnya dipilih perangkat *smartphone* adalah karena di era teknologi yang berkembang pesat saat ini, hampir semua orang memiliki perangkat *smartphone*, sehingga dapat digunakan oleh siapa saja. Dalam pengembangan aplikasi *mobile* pada penelitian ini menggunakan *framework* cordova. Aplikasi yang dihasilkan oleh *framework* cordova hampir memiliki cara kerja yang sama dengan *native apps* yang dibuat menggunakan bahasa Java bagi android, ataupun swift pada *smartphone* berbasis IOS. Dengan *framework* cordova ini memungkinkan pengembang membuat aplikasi menggunakan HTML, JavaScript dan CSS untuk membentuk aplikasi tersebut dan dapat dijalankan pada berbagai *platform* dengan sedikit ataupun tanpa perubahan sama sekali. Sebutan aplikasi yang dihasilkan oleh cordova adalah *hybrid apps* karena seperti gabungan antara *native apps* dengan *progressive web apps* (PWA).



**Gambar 1. Alur Kerja Aplikasi**

Pada Gambar 1 pengguna mengambil citra *digital* dari biji kopi yang ingin diklasifikasikan, kemudian citra tersebut dikirimkan ke *server* dan akan dilakukan *preprocessing* kemudian proses klasifikasi. Setelah citra selesai diklasifikasikan maka hasilnya akan dikirimkan pada pengguna dan hasil biji kopi tersebut tergolong ke dalam jenis dengan tingkat sangrai tertentu ditampilkan pada *apps* pengguna. Dengan alur yang cukup simpel ini diharapkan dapat mempermudah pengguna dalam mengklasifikasikan sangrai biji kopi.

**2.8. Uji Coba Sistem dan Evaluasi**

Setelah aplikasi selesai dibuat, maka perlu dilakukan uji coba dan evaluasi apakah aplikasi tersebut sudah dapat digunakan dengan baik dan menghasilkan klasifikasi yang baik. Dalam penelitian ini dilakukan 3 pengujian statistik menggunakan *confusion matrix*, tes ANOVA, dan tes PostHoc.

1) *Confusion Matrix*

Pada umumnya *confusion matrix* digunakan pada klasifikasi yang memiliki 2 *class* saja, namun dalam penelitian ini menggunakan 4 *class* yaitu mentah, *light roast*, *medium roast*, serta *dark roast*. Dalam penelitian ini maka dilakukan penyesuaian dalam membentuk *confusion matrix* seperti pada Tabel 1 Hingga Tabel 4 yang menggambarkan bagaimana kondisi *confusion matrix* ketika memprediksi tiap *class* ketika nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN):

**Tabel 1. Confusion Matrix Untuk Klasifikasi Class Mentah**

T Class a Asal	Class Klasifikasi			
	Mentah	Light Roast	Medium Roast	Dark Roast
b Mentah	TP	FN	FN	FN
c Light Roast	FP	TN	TN	TN
d Medium Roast	FP	TN	TN	TN
e Dark Roast	FP	TN	TN	TN

**Tabel 2. Confusion Matrix Untuk Klasifikasi Class Light Roas**

Class Asal T	Class Klasifikasi			
	Mentah	Light Roast	Medium Roast	Dark Roast
a Mentah	TN	FP	TN	TN
b Light Roast	FN	TP	FN	FN
c Medium Roast	TN	FP	TN	TN
d Dark Roast	TN	FP	TN	TN

**Tabel 3. Confusion Matrix Untuk Klasifikasi Class Medium Roast**

Class Asal	Class Klasifikasi			
	Mentah	Light Roast	Medium Roast	Dark Roast
Mentah	TN	TN	TN	FP
Light Roast	TN	TN	TN	FP
Medium Roast	TN	TN	TN	FP
Dark Roast	FN	FN	FN	TP

**Tabel 4. Confusion Matrix Untuk Klasifikasi Class Dark Roast**

Class Asal	Class Klasifikasi			
	Mentah	Light Roast	Medium Roast	Dark Roast
Mentah	TN	TN	FP	TN
Light Roast	TN	TN	FP	TN
Medium Roast	FN	FN	TP	FN
Dark Roast	TN	TN	FP	TN

Dari ke empat tabel di atas menggambarkan bagaimana keadaan setiap *confusion matrix* ketika melakukan klasifikasi masing-masing *class*. Untuk penelitian ini ke empat tabel tersebut digabungkan menjadi 1 *confusion matrix* seperti Tabel 5.

**Tabel 5. Confusion Matrix Untuk Klasifikasi Tingkat Kematangan Sangrai Kopi**

Class Asal	Class Klasifikasi			
	Mentah	Light Roast	Medium Roast	Dark Roast
Mentah	EE	LE	ME	DE
Light Roast	EL	LL	ML	DL
Medium Roast	EM	LM	MM	DM
Dark Roast	ED	LD	MD	DD

Penjelasan dari Tabel 5 adalah EE, LL, MM, dan DD melambangkan keadaan ketika terjadi *True Positive* dari setiap *class*, sedangkan sisanya ketika prediksi tidak tepat. Maka untuk menghitung akurasi digunakan rumus sebagai berikut:

$$(8) \text{ Akurasi} = \frac{(EE + LL + MM + DD)}{(\text{Total Jumlah Seluruh Data})}$$

Dari rumus akurasi tersebut dilakukan penjumlahan seluruh klasifikasi yang benar dibagi dengan total jumlah keseluruhan data. Untuk spesifisitas dan sensitivitas dilakukan perhitungan masing-masing bagi *tiap* klasifikasi.



$$(9) \text{ Spesifisitas Mentah} = \frac{(LL+ML+DL+LM+MM+DM+LD+MD+DD)}{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}$$

$$(10) \text{ Spesifisitas Light Roast} = \frac{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}$$

$$(11) \text{ Spesifisitas Medium Roast} = \frac{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}$$

$$(12) \text{ Spesifisitas Dark Roast} = \frac{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}{(EE+ME+DE+EM+MM+DM+ED+MD+DD)}$$

Dari rumus spesifisitas di atas dapat dibandingkan pada Tabel 1, hingga Tabel 4 untuk posisi *true negative* dan *false positive* dengan posisi yang ada pada tabel 5, sehingga digunakanlah rumus tersebut. Selanjutnya untuk rumus sensitivitas digunakan rumus sebagai berikut:

$$(13) \text{ Sensitivitas Mentah} = \frac{EE}{(EE + (LE + ME + DE))}$$

$$(14) \text{ Sensitivitas Light Roast} = \frac{LL}{(LL + (EL + ML + DL))}$$

$$(15) \text{ Sensitivitas Medium Roast} = \frac{MM}{(MM + (EM + LM + DM))}$$

$$(16) \text{ Sensitivitas Dark Roast} = \frac{DD}{(DD + (ED + LD + MD))}$$

Dari rumus sensitivitas di atas sama seperti rumus spesifisitas dapat dibandingkan pada Tabel 1 hingga Tabel 4 untuk posisi *true positive* dan *false negative* dengan posisi yang ada pada tabel 5, sehingga digunakanlah rumus tersebut.

## 2) ANOVA

ANOVA merupakan singkatan dari “*Analysis of Variance*” yang merupakan tes omnibus, yang berarti menguji perbedaan secara keseluruhan antara kelompok. *One way* ANOVA adalah tes parametrik yang digunakan untuk menguji perbedaan yang signifikan antara 3 kelompok atau lebih. Alasan mengapa disebut sebagai *one way* ANOVA atau ANOVA satu arah meskipun ada 3 kelompok atau lebih adalah karena kelompok-kelompok tersebut masih dalam satu kategori variabel. Jika ada dua variabel maka akan disebut sebagai *two way* atau *two factor* ANOVA.

Pada dasarnya tes ANOVA ini dilakukan untuk menerima ataupun menolak sebuah hipotesis apakah ada perbedaan yang signifikan antar kelompok. Untuk menerima hipotesis tersebut maka nilai P-value (PR) > 0,005, sedangkan apabila nilai Pvalue < 0,005 berarti menolak hipotesis tersebut dan terdapat perbedaan yang cukup signifikan pada tiap kelompok yang diuji.

## 3) POST-HOC

Tes Post-Hoc dilakukan apabila tidak dapat membedakan kelompok mana saja yang berbeda berdasarkan tes ANOVA terlebih dahulu. Dengan melakukan tes post-hoc atau tes perbandingan yang direncanakan, memungkinkan untuk melihat kelompok mana saja yang berbeda secara signifikan satu sama lainnya. ketika bernilai “*False*” berarti menunjukkan tidak adanya perbedaan yang cukup signifikan antara kelompok yang dibandingkan, namun apabila bernilai “*True*” berarti terdapat perbedaan yang signifikan antara kelompok yang dibandingkan.

## 3. Hasil dan Pembahasan

### 3.1. Pengumpulan Data

Proses pengumpulan data ini merupakan tahap mengumpulkan seluruh data yang dibutuhkan yaitu data hasil sangrai kopi yang dilakukan secara mandiri dengan berbagai perlakuan. Seluruh data yang terkumpulkan selanjutnya akan diolah untuk digunakan pada pelatihan dan pengujian model CNN yang akan dibuat. Tahapan yang dilalui untuk pengumpulan data citra biji kopi sebagai berikut:

#### 1) Proses Sangrai Kopi

Variabel yang perlu diperhatikan dalam proses sangrai biji kopi ini adalah waktu, suhu, dan seberapa banyak biji kopi yang disangrai. Namun karena dalam proses sangrai yang dilakukan secara mandiri menggunakan peralatan yang sederhana yaitu wajan teflon dan kompor maka sulit untuk mengatur suhu secara stabil, maka variabel yang diatur adalah jumlah banyak biji kopi yang disangrai dan waktu yang diperlukan untuk proses sangrai tersebut. Selain itu ketika proses sangrai dilakukan perlu diperhatikan apakah sudah terjadi letupan (*Crack*) yaitu *first crack* dan *second crack*. Ukuran api yang digunakan ketika menyangrai seluruh sampel kopi tersebut adalah maksimum.

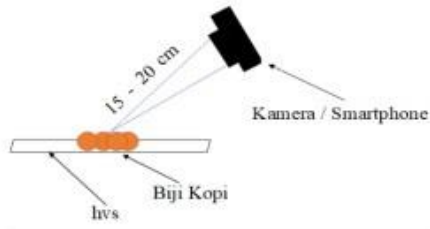
Tabel 6. Tabel Variabel Proses Sangrai

Sampel Ke-	Banyak Biji Kopi (Gram)	Waktu (Menit)	First Crack	Second Crack	Kategori
1	100	9	Sudah	Sudah	Dark Roast
2	50	2	Belum	Belum	Light Roast
3	50	4	Sudah	Sudah	Dark Roast
4	50	3	Sudah	Belum	Medium Roast
5	50	2.5	Belum	Belum	Light Roast
6	50	3.5	Sudah	Sebagian	Medium Roast

#### 2) Hasil Sangrai Kopi

Setelah seluruh proses sangrai kopi selesai dilakukan, maka selanjutnya adalah mengambil gambar dari biji kopi hasil sangrai tersebut untuk digunakan sebagai *dataset*. Dari setiap sampel diambil beberapa citra biji kopi sebanyak ±2 sendok teh untuk setiap citranya. Pengambilan citra ini dilakukan menggunakan kamera dari *smartphone* dengan resolusi 4000×3000 *piksel*, dan biji kopi beralaskan kertas

HVS agar *background* berwarna putih, dan memastikan bahwa ketika mengambil citra tidak ada bayangan yang masuk ke dalam sampel. Jarak kamera dengan sampel antara rentang 15-20 cm dengan sudut  $\pm 60^\circ$ . Untuk fokus pada kamera diarahkan pada bagian tengah dari sampel yang akan diambil citranya.



Gambar 2. Posisi Pengambilan Citra Sampel Biji Kopi

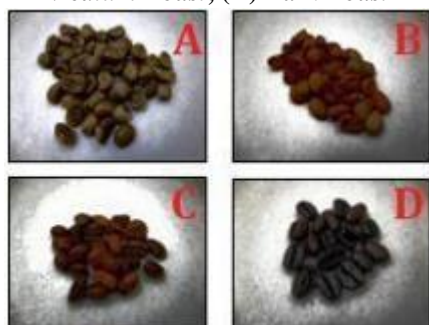
Untuk total citra yang diambil ada sebanyak 240 citra, dengan dibagi menjadi 60 citra biji kopi mentah, 60 citra biji kopi *light roast*, 60 citra biji kopi *medium roast*, dan 60 citra biji kopi *dark roast*.

### 3) Hasil *Histogram Equalization* Pada Citra RGB

Proses *histogram equalization* dilakukan pada citra RGB *dataset* sebanyak 240 buah berukuran  $4000 \times 3000$  piksel. Tujuan dari proses *histogram equalization* ini adalah untuk meningkatkan kontras pada citra sehingga detail warna dapat diperoleh, serta menyiapkan citra ke dalam kondisi siap diekstraksi ciri oleh arsitektur CNN. Pada Gambar 4 dapat dilihat hasil *histogram equalization* pada bagian biji kopi terjadi peningkatan kontras pada warnanya, serta warna sekitarnya yang tidak terkena titik fokus kamera warnanya menjadi sedikit keabu-abuan warnanya. Dari hasil ini *histogram equalization* sudah dapat melakukan peningkatan kontras dengan cukup baik. Citra hasil *histogram equalization* inilah yang nantinya digunakan sebagai data latih dan uji pada model.



Gambar 3. Sampel (A) Mentah, (B) *Light Roast*, (C) *Medium Roast*, (D) *Dark Roast*



Gambar 4. Sampel *Histogram Equalization* (A) Mentah, (B) *Light Roast*, (C) *Medium Roast*, (D) *Dark Roast*

### 3.2. Analisis Hasil Pelatihan dan Pengujian Arsitektur

Pada penelitian ini untuk 1 kali pelatihan model menggunakan *epoch* sebanyak 20 kali dan dilakukan sebanyak 10 kali untuk setiap kombinasi arsitektur CNN yaitu LeNet5, AlexNet dan MiniVGGNet dengan fungsi optimasi yang digunakan yaitu ADAM, NADAM, dan SGD. Kemudian untuk pembagian antara data latih dan data uji yang digunakan adalah 75% data latih atau 140 citra, dan 25% data uji atau 60 citra yang diambil secara acak. Untuk proses pelatihan dan pengujian seluruhnya menggunakan CPU. Kemudian ukuran *batch size* yang digunakan adalah 32. Selanjutnya untuk setiap arsitektur pada *convolution layer* digunakan *zero padding* yang berarti penambahan nilai 0 pada setiap tepian nilai pada citra.

Setelah memperoleh rata-rata akurasi dari 10 kali pelatihan masing-masing kombinasi. Kemudian rata-rata akurasi tertinggi pada masing-masing arsitektur dengan kombinasi salah satu fungsi optimasi akan di tes menggunakan ANOVA dan Post-Hoc untuk mengetahui apakah ada perbedaan yang cukup signifikan. Berikut ini pembahasan dari masing-masing arsitektur yang digunakan:

#### 1) LeNet5

Pada penelitian ini menggunakan arsitektur LeNet5 hasil modifikasi oleh Rosebrock [6].

Tabel 7. Arsitektur LeNet5

Layer	Output Size	Filter Size (F) / Strides (S) / Jumlah Filter (JF)
<i>Input Image</i>	32×32×3	
<i>Convolution 1</i>	32×32×20	F=5×5, S=1×1, JF = 20
Aktivasi (ReLU) 1	32×32×20	
<i>Pooling (Max Pooling) 1</i>	16×16×20	F=2×2, S=2×2
<i>Convolution 2</i>	16×16×50	F=5×5, S=1×1, JF = 50
Aktivasi (ReLU) 2	16×16×50	
<i>Pooling (Max Pooling) 2</i>	8×8×50	F=2×2, S=2×2
<i>Fully Connected Layer 1</i>	500	
Aktivasi (ReLU) 3	500	
<i>Fully Connected Layer 2</i>	4	
<i>SoftMax</i>	4	

Berikut ini adalah hasil akurasi, spesifitas dan sensitivitas berdasarkan hasil akurasi terendah dan tertinggi dari pelatihan arsitektur ini.

**Tabel 8. Hasil Akurasi LeNet5**

LeNet5			
Optimasi	Hasil Test Accuracy (Akurasi Pada Data Uji)		Rata-rata
	Tertinggi	Terendah	
ADAM	98%	67%	85%
SGD	40%	23%	32.1%
NADAM	92%	67%	82.5%

**Tabel 9. Hasil Sensitivitas LeNet5**

LeNet5				
Sensitivitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	0,93	0,67	0,85	0,88
Light Roast (ADAM)	1,00	1,00	0,99	
Medium Roast (ADAM)	1,00	0,25	0,67	
Dark Roast (ADAM)	1,00	1,00	0,99	
Mentah (SGD)	0,33	0,00	0,31	0,38
Light Roast (SGD)	0,92	1,00	0,90	
Medium Roast (SGD)	0,00	0,00	0,00	
Dark Roast (SGD)	0,62	0,15	0,29	
Mentah (NADAM)	0,93	0,87	0,81	0,86
Light Roast (NADAM)	1,00	1,00	0,98	
Medium Roast (NADAM)	0,80	0,10	0,64	
Dark Roast (NADAM)	1,00	1,00	1,00	

**Tabel 9. Hasil Spesifisitas LeNet5**

LeNet5				
Spesifisitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	1,00	1,00	0,99	0,95
Light Roast (ADAM)	1,00	0,69	0,87	
Medium Roast (ADAM)	0,98	0,91	0,95	
Dark Roast (ADAM)	1,00	1,00	1,00	
Mentah (SGD)	0,93	1,00	0,87	0,79
Light Roast (SGD)	0,66	0,04	0,35	
Medium Roast (SGD)	1,00	1,00	1,00	
Dark Roast (SGD)	0,83	1,00	0,95	
Mentah (NADAM)	0,96	1,00	1,00	0,95
Light Roast (NADAM)	0,96	0,63	0,88	
Medium Roast (NADAM)	0,98	0,95	0,92	
Dark Roast (NADAM)	1,00	1,00	1,00	

Berdasarkan Tabel 8 dan Tabel 9 diperoleh model LeNet5 dengan fungsi optimasi ADAM memiliki

sensitivitas overall sebesar 0.88 (88%) dengan spesifisitas overall sebesar 0.95 (95%). Jika dilihat berdasarkan rata-rata sensitivitas masing-masing kelas maka model ini mampu mengenali data uji *class dark roast* dan *class light roast* dengan sempurna yaitu 0.99 (99%). Model LeNet5 dengan fungsi SGD memiliki nilai sensitivitas overall hanya sebesar 0.38 (38%) dan nilai spesifisitas overall sebesar 0.79 (79%) menjadi yang terendah dalam pengujian arsitektur LeNet5. Bahkan berdasarkan rata-rata sensitivitas 0,00 dan spesifisitas sebesar 1,00 pada *class medium roast* terlihat bahwa data uji tidak ada yang terklasifikasi sebagai *medium roast* baik yang benar maupun yang salah terklasifikasikan. Performa model LeNet5 dengan fungsi optimasi NADAM dengan nilai overall sensitivitas sebesar 0.86 (86%) sedikit berada di bawah model yang menggunakan fungsi optimasi ADAM, dengan nilai spesivitas overall sebesar 0.95 (95%). Ratarata sensitivitas masing-masing *class* yang dimiliki juga tidak berbeda jauh bahkan pada *class dark roast* memiliki sensitivitas yang lebih tinggi dengan nilai 1,00 (100%) dan spesifisitas sebesar 1,00 (100%) yang berarti data uji dari *class dark roast* mampu dikenali dengan baik dan tidak ada data uji dari *class* lainnya yang salah terklasifikasikan sebagai *class dark roast*.

2) AlexNet

Pada penelitian ini menggunakan arsitektur AlexNet oleh Alex Krizhevsky sebagai berikut:

**Tabel 10. Arsitektur AlexNet**

Layer	Output Size	Filter Size (F) / Strides (S) / Jumlah Filter (JF)
Input Image	227×227×3	
Convolution 1	55×55×96	F = 11×11, S=4×4, JF = 96
Aktivasi (ReLU) 1	55×55×96	
Pooling (Max Pooling) 1	27×27×96	F=3×3, S=2×2
Dropout (0.5) 1	27×27×96	
Convolution 2	27×27×256	F = 5×5, S=1×1, JF = 256
Aktivasi (ReLU) 2	27×27×256	
Pooling (Max Pooling) 2	13×13×256	F=3×3, S=2×2
Dropout (0.5) 2	13×13×256	
Convolution 3	13×13×384	F = 3×3, S=1×1, JF = 384
Convolution 4	13×13×384	F = 3×3, S=1×1, JF = 384
Convolution 5	13×13×256	F = 3×3, S=1×1, JF = 256
Aktivasi (ReLU) 3	13×13×256	
Pooling (Max Pooling) 3	6×6×256	F=3×3, S=2×2
Dropout (0.5) 3	6×6×256	
Fully Connected Layer 1	4096	
Aktivasi (ReLU) 4	4096	
Dropout (0.5) 4	4096	
Fully Connected Layer 2	4096	
Aktivasi (ReLU) 5	4096	
Dropout (0.5) 5	4096	
Fully Connected Layer 3	4	
SoftMax	4	



Berikut ini adalah hasil akurasi, spesifitas dan sensitivitas berdasarkan hasil akurasi terendah dan tertinggi dari pelatihan arsitektur ini.

Tabel 11. Hasil Akurasi AlexNet

AlexNet			
Optimasi	Hasil Test Accuracy (Akurasi Pada Data Uji)		Rata-rata
	Tertinggi	Terendah	
ADAM	95%	20%	39,4%
SGD	35%	20%	22,8%
NADAM	22%	20%	21,4%

Tabel 12. Hasil Sensitivitas AlexNet

AlexNet				
Sensitivitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	0,93	0,00	0,44	0,42
Light Roast (ADAM)	1,00	1,00	0,67	
Medium Roast (ADAM)	0,90	0,00	0,20	
Dark Roast (ADAM)	1,00	0,00	0,38	
Mentah (SGD)	0,00	0,00	0,00	0,28
Light Roast (SGD)	1,00	1,00	0,73	
Medium Roast (SGD)	0,00	0,00	0,00	
Dark Roast (SGD)	0,69	0,00	0,38	
Mentah (NADAM)	0,00	0,00	0,00	0,25
Light Roast (NADAM)	0,00	1,00	0,30	
Medium Roast (NADAM)	0,00	0,00	0,00	
Dark Roast (NADAM)	1,00	0,00	0,70	

Tabel 13. Hasil Spesifisitas AlexNet

AlexNet				
Spesifisitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	0,96	1,00	0,76	0,77
Light Roast (ADAM)	1,00	0,00	0,66	
Medium Roast (ADAM)	0,98	1,00	0,87	
Dark Roast (ADAM)	1,00	0,00	0,80	
Mentah (SGD)	1,00	0,98	1,00	0,76
Light Roast (SGD)	0,19	0,02	0,32	
Medium Roast (SGD)	1,00	1,00	1,00	
Dark Roast (SGD)	1,00	1,00	0,71	
Mentah (NADAM)	1,00	1,00	1,00	0,75
Light Roast (NADAM)	1,00	0,00	0,70	
Medium Roast (NADAM)	1,00	1,00	1,00	
Dark Roast (NADAM)	0,00	1,00	0,30	

Berdasarkan Tabel 12 dan Tabel 13 dapat dilihat performa dari setiap model yang sangat rendah, dan model AlexNet

dengan fungsi optimasi ADAM menjadi yang terbaik dengan nilai overall sensitivitas sebesar 0,42 (42%) serta overall spesifisitas 0,77 (77%). Model ini masih mampu mengklasifikasikan *dataset* uji ke dalam semua *class*. Untuk model AlexNet dengan fungsi optimasi SGD dihasilkan nilai overall sensitivitas sebesar 0,28 (28%) serta overall spesifisitas 0,76 (76%). Model ini tidak dapat mengklasifikasikan *dataset* uji pada *class* mentah dan *dark roast* dengan benar, bahkan 9 hasil pelatihan menghasilkan nilai sensitivitas 0,00 dengan nilai spesifisitas 1,00 pada kedua *class* tersebut yang berarti tidak ada *dataset* uji yang terklasifikasikan pada *class* tersebut baik yang benar maupun salah.

Untuk hasil model AlexNet dengan fungsi optimasi NADAM memiliki performa yang paling rendah dengan nilai overall sensitivitas hanya 0,25 (25%) dan nilai overall spesifisitas sebesar 0,75 (75%). Model ini hanya mampu mengklasifikasikan ke dalam 2 pola saja yaitu, seluruh *dataset* uji diklasifikasikan sebagai *class light roast* atau seluruh data uji diklasifikasikan sebagai *class dark roast*. Hal ini ditunjukkan jika *class light roast* ataupun *dark roast* memiliki sebuah nilai maka seluruh *class* lainnya akan menghasilkan nilai sensitivitas 0,00 dan dengan nilai spesifisitas 1,00 yang berarti tidak ada *dataset* uji yang terklasifikasikan pada *class* tersebut baik yang benar maupun salah.

### 3) MiniVGGNet

Pada penelitian ini menggunakan arsitektur MiniVGGNet dengan arsitektur sebagai berikut:

Tabel 14. Arsitektur MiniVGGNet

Layer	Output Size	Filter Size (F) / Strides (S) / Jumlah Filter (JF)
Input Image	32×32×3	
Convolution 1	32×32×32	F=3×3, S=1×1, JF=32
Aktivasi (ReLU) 1	32×32×32	
Convolution 2	32×32×32	F=3×3, S=1×1, JF=32
Aktivasi (ReLU) 2	32×32×32	
Pooling (Max Pooling) 1	16×16×32	F=2×2, S=2×2
Dropout (0.5) 1	16×16×32	
Convolution 2	16×16×64	F=3×3, S=1×1, JF=64
Aktivasi (ReLU) 2	16×16×64	
Convolution 3	16×16×64	F=3×3, S=1×1, JF=64
Aktivasi (ReLU) 3	16×16×64	
Pooling (Max Pooling) 2	8×8×64	F=2×2, S=2×2
Dropout (0.5) 2	8×8×64	
Fully Connected Layer 1	512	
Aktivasi (ReLU) 3	512	
Dropout (0.5) 2	512	
Fully Connected Layer 2	4	
SoftMax	4	



Berikut ini hasil akurasi, spesifitas dan sensitivitas berdasarkan hasil akurasi terendah dan tertinggi

Tabel 15. Hasil Akurasi MiniVGGNet

MiniVGGNet			
Optimasi	Hasil Test Accuracy (Akurasi Pada Data Uji)		Rata-rata
	Tertinggi	Terendah	
ADAM	88%	48%	69,6%
SGD	35%	20%	24,6%
NADAM	95%	40%	75,1%

Tabel 16. Hasil Sensitivitas MiniVGGNet

MiniVGGNet				
Sensitivitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	0,80	0,13	0,55	0,74
Light Roast (ADAM)	1,00	1,00	1,00	
Medium Roast (ADAM)	0,80	0,10	0,43	
Dark Roast (ADAM)	1,00	1,00	0,99	
Mentah (SGD)	0,73	0,00	0,29	0,28
Light Roast (SGD)	0,75	1,00	0,55	
Medium Roast (SGD)	0,00	0,00	0,00	
Dark Roast (SGD)	0,00	0,00	0,28	
Mentah (NADAM)	1,00	0,00	0,69	0,79
Light Roast (NADAM)	0,92	1,00	0,98	
Medium Roast (NADAM)	0,98	0,00	0,51	
Dark Roast (NADAM)	1,00	0,92	0,99	

Tabel 17. Hasil Spesifisitas MiniVGGNet

MiniVGGNet				
Spesifisitas	Hasil Sensitivitas Berdasarkan Akurasi		Rata-rata	Overall
	Tertinggi	Terendah		
Mentah (ADAM)	1,00	1,00	0,96	0,89
Light Roast (ADAM)	0,92	0,54	0,78	
Medium Roast (ADAM)	0,93	0,78	0,80	
Dark Roast (ADAM)	1,00	1,00	1,00	
Mentah (SGD)	0,56	1,00	0,77	0,76
Light Roast (SGD)	0,65	0,00	0,50	
Medium Roast (SGD)	1,00	1,00	1,00	
Dark Roast (SGD)	0,96	1,00	0,76	
Mentah (NADAM)	0,96	0,98	0,98	0,91
Light Roast (NADAM)	1,00	0,31	0,79	
Medium Roast (NADAM)	0,98	0,95	0,91	
Dark Roast (NADAM)	1,00	1,00	1,00	

Berdasarkan Tabel 16 dan Tabel 17 dapat dilihat bahwa model MiniVGGNet dengan fungsi optimasi NADAM

menjadi yang terbaik dengan nilai overall sensitivitas sebesar 0,79 (79%) dan overall spesifisitas sebesar 0,91 (0,91%). Dilihat dari rata-rata sensitivitas yang dihasilkan pada *class dark roast* mencapai hingga 1,00 (100%) dengan nilai spesifisitasnya mencapai 1,00 (100%) yang menunjukkan bahwa model ini sangat baik mengklasifikasikan *dataset* uji *class dark roast* dan tidak salah mengklasifikasikan *class* lainnya sebagai *class dark roast*. Sedangkan model MiniVGGNet dengan fungsi optimasi SGD menjadi yang terburuk di antara 3 model yang ada. Model ini tidak mampu mengklasifikasikan data uji yang berasal *class medium roast* dengan benar yang ditunjukkan dengan rata-rata sensitivitasnya yang 0,00 (0%) dan dari *class* lainnya juga tidak ada yang salah diklasifikasikan sebagai *class medium roast* dengan ditunjukkan dari nilai rata-rata spesifisitas yang mencapai 1,00(100%).

Terakhir adalah model MiniVGGNet dengan fungsi optimasi ADAM, yang performanya sedikit berada dibawah model yang menggunakan fungsi optimasi NADAM. Overall sensitivitas yang dihasilkan sebesar 0.74(74%), dan nilai overall spesifisitasnya 0.89 (89%). Performa terbaiknya dihasilkan pada *class dark roast* dengan rata-rata sensitivitas mencapai 0,99 (99%) dan spesifisitasnya mencapai 1,00 (100%). Model ini mampu mengenali dan membedakan data uji pada *class dark roast* dengan sangat baik. Pada *class light roast* pun rata-rata sensitivitasnya mencapai nilai 1,00 (100%) namun nilai rata-rata spesivitasnya hanya 0,78 (78%) yang berarti model mampu mengenali data uji yang berasal dari *class light roast* dengan sangat baik, namun masih belum mampu membedakannya dengan *class* lainnya sehingga ada beberapa data uji dari *class* lainnya diklasifikasikan sebagai *class light roast*. Model ini cukup lemah dalam mengklasifikasikan biji kopi pada *class medium roast* yang dibuktikan hanya menghasilkan nilai rata-rata sensitivitasnya sebesar 0,48 48%) dengan spesivitasnya sebesar (80%).

### 3.3. Rangkuman Perfoma Model

Tabel 18. Tabel Akurasi Rata-Rata Seluruh Model

Fungsi Optimasi	Hasil Akurasi Rata-Rata Arsitektur		
	LeNet5	AlexNet	MiniVGGNet
ADAM	85%	39,4%	69,6%
SGD	32,1%	22,8%	24,6%
NADAM	82,5%	21,4%	75,1%

Berdasarkan Tabel 18 dapat dilihat bahwa LeNet5 dengan fungsi optimasi ADAM memiliki rata-rata akurasi tertinggi. Untuk model yang menggunakan arsitektur AlexNet memiliki rata-rata akurasi yang cukup rendah dengan menggunakan ke 3 fungsi optimasi tersebut. Untuk AlexNet sering dihasilkan model yang *unrepresentative validation dataset* jika menggunakan fungsi optimasi ADAM, dan pada seluruh pelatihan NADAM. Kemudian untuk MiniVGGNet pada kombinasi fungsi ADAM atau NADAM masih ada kemungkinan kecil dihasilkan model yang *overfitting* pada beberapa hasil pelatiahannya. Terakhir untuk seluruh arsitektur yang menggunakan fungsi

optimasi SGD dihasilkan model yang *underfitting* pada setiap pelatihannya.

Selanjutnya dilakukan tes ANOVA menggunakan 10 data hasil latih dari model LeNet5 dengan fungsi optimasi ADAM, AlexNet dengan fungsi optimasi NADAM, dan MiniVGGNet dengan fungsi optimasi NADAM. Ketiga model yang dipilih adalah yang memiliki rata-rata akurasi tertinggi pada masing-masing arsitektur dengan kombinasi fungsi optimasinya. Pada pengujian ANOVA ini memiliki hipotesis sebagai berikut ini:

- $H_0$  = Tidak ada perbedaan yang signifikan dari ke 3 model tersebut.
- $H_1$  = Ada perbedaan yang signifikan dari ke 3 model tersebut.

Tabel 19. Tabel Hasil Tes ANOVA

	df	sum sq	mean sq	F	PR(>F)
model	2,0	1,15062	0,575310	13,44962	0,000089
Residual	27,0	1,15493	0,042775	NaN	NaN

Pada Tabel 4.16 yang merupakan hasil tes ANOVA ditunjukkan bahwa nilai Pvalue atau PR(>F) memiliki nilai < dari 0,05, yang berarti menolak  $H_0$ . Hal ini enunjukkan adanya perbedaan yang cukup signifikan dari ke 3 model tersebut. Namun dari tes ANOVA ini belum menunjukkan model manakah yang memiliki perbedaan yang signifikan, oleh karena itu dilanjutkan pada tes PostHoc. Hipotesis yang ada pada tes PostHoc adalah sebagai berikut

- $H_0$  = Group 1 dengan Group 2 Tidak Memiliki Perbedaan Yang signifikan.
- $H_1$  = Group 1 dengan Group 2 Memiliki Perbedaan Yang signifikan

Tabel 20. Tabel Hasil Tes Post-Hoc

Group1	Group2	Meandiff	p-adj	Lower	Upper	Reject
AlexNet (ADAM)	LeNet5 (ADAM)	0,456	0,001	0,0267	0,6853	True
AlexNet (ADAM)	MiniVGGNet (NADAM)	0,357	0,0018	0,1277	0,5863	True
LeNet5 (ADAM)	MiniVGGNet (NADAM)	-0,099	0,5393	-0,3283	0,1303	False

Berdasarkan Tabel 20 dapat dilihat bahwa AlexNet fungsi optimasi ADAM memiliki perbedaan yang signifikan dibandingkan dengan LeNet5 dengan fungsi optimasi ADAM serta MiniVGGNet dengan fungsi optimasi NADAM. Hal ini ditunjukkan dengan nilai *reject* adalah "True" yang berarti menerima hipotesis  $H_0$ . Sedangkan untuk LeNet5 menggunakan fungsi optimasi ADAM tidak memiliki perbedaan yang cukup signifikan dengan model MiniVGGNet yang menggunakan fungsi optimasi NADAM yang ditunjukkan dengan *reject* bernilai "False" diartikan menerima hipotesis  $H_0$ .

### 3.4. Aplikasi

Untuk melakukan klasifikasi pertama pengguna dapat memilih model yang ingin digunakan seperti model menggunakan arsitektur LeNet5, AlexNet, atau MiniVGGNet dengan kombinasi fungsi optimasi ADAM, SGD, atau NADAM. Jika pengguna tidak memilih model

apa yang akan digunakan, maka secara otomatis akan dipilhkan model LeNet5 dengan fungsi optimasi ADAM.



Gambar 3. Tampilan Aplikasi ketika Klasifikasi

Kemudian pengguna dapat mengakses kamera *smartphone* dengan menekan tombol yang disediakan untuk mengambil citra digital kopi yang ingin diklasifikasikan. Kemudian gambar tersebut akan dikirimkan ke *server* untuk dilakukan proses klasifikasi, kemudian hasil klasifikasi dikembalikan ke pengguna. Pengguna memperoleh hasil *class* klasifikasi biji kopi tersebut beserta tingkat keyakinan model mengklasifikasikan sebagai *class* tersebut. Pada bagian *server* dapat dilakukan penambahan *dataset* citra menggunakan citra digital yang dikirimkan dari pengguna ketika melakukan klasifikasi, jika dibutuhkan untuk pelatihan model selanjutnya.



Gambar 4. Tampilan Halaman Penambahan Data Pada Server

### 4. Kesimpulan

Berdasarkan pada penelitian, hasil dan pembahasan yang telah dilakukan dan dipaparkan di atas, maka dapat diambil kesimpulan sebagai berikut:

1. Model LeNet5 dengan menggunakan optimasi ADAM mampu menghasilkan akurasi hingga 98% pada salah satu pelatihannya serta memiliki rata-rata akurasi

tertinggi dibandingkan seluruh model lainnya dengan nilai 85%.

2. Seluruh model yang menggunakan fungsi optimasi SGD baik dengan arsitektur LeNet5, AlexNet, dan MiniVGG Net tidak mampu mengklasifikasikan citra biji kopi yang berasal dari *class medium roast* dengan benar. Hal ini dibuktikan dengan seluruh nilai rata-rata spesivitasnya adalah 0,00 (0%).
3. Arsitektur AlexNet tidak cocok dalam penelitian ini dikarenakan meskipun menggunakan kombinasi fungsi optimasi ADAM, SGD, dan NADAM semuanya memiliki rata-rata akurasi <40%.
4. Secara uji performa menggunakan Post-Hoc model LeNet5 dengan optimasi ADAM memiliki performa yang sama dengan model MiniVGGNet dengan optimasi NADAM, sehingga dapat dipilih menggunakan salah satu dari model tersebut. Namun model yang dihasilkan dari LeNet5 dengan optimasi ADAM berukuran lebih kecil dan cukup ringan digunakan pada aplikasi

## 5. Referensi

- [1] Dwyer, H. (17 Agustus 2015). *Citing Internet sources* URL <https://blog.dataiku.com/deep-learning-with-dss>.
- [2] Nayak, S. (13 Juni 2018). *Citing Internet sources* URL <https://www.learnopencv.com/understanding-alexnet/>
- [3] Toulis, P., Horel, T., Airoidi, E.M. (4 Oktober 2015) *Citing Internet sources* URL <https://arxiv.org/abs/1510.00967v3>
- [4] Kingma, D.P., Ba, J.L., (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, San Diego, pp. 2015- 2017 [5] Dozat, T., (2016). Incorporating nesterov momentum into adam. *International Conference on Learning Representations Workshop*, San Juan, Puerto Rico, pp. 2013– 2016.
- [6] Rosebrock, A. (11 Desember 2017). *Citing Internet sources* URL <https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deeplearning/>