

**KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK  
PADA ARSITEKTUR MOBILENET**

**SKRIPSI**

Oleh :  
**A. FUAD JAUHARI**  
**NIM. 15650067**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2022**

**KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK  
PADA ARSITEKTUR MOBILENET**

**SKRIPSI**

Oleh :  
**A. FUAD JAUHARI**  
**NIM. 15650067**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2022**

HALAMAN PERSETUJUAN

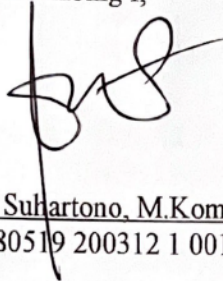
KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK  
PADA ARSITEKTUR MOBILENET

SKRIPSI

Oleh :  
A. FUAD JAUHARI  
NIM. 15650067

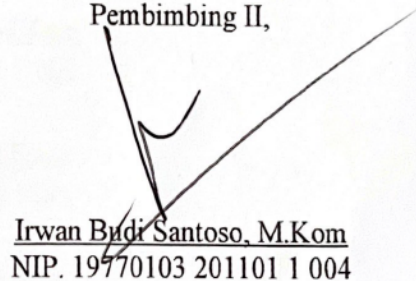
Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 9 Juni 2022

Pembimbing I,



Prof. Dr. Suhartono, M.Kom  
NIP. 19680519 200312 1 001


Pembimbing II,



Irwan Budi Santoso, M.Kom  
NIP. 19770103 201101 1 004

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrud Kurniawan ST., M.MT., IPM  
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK  
PADA ARSITEKTUR MOBILENET

SKRIPSI

Oleh :  
**A. FUAD JAUHARI**  
NIM. 15650067

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 23 Juni 2022

**Susunan Dewan Penguji**

Penguji Utama : Dr. Totok Chamidy, M.Kom  
NIP. 19691222 200604 1 001

Ketua Penguji : Fajar Rohman Hariri, M.Kom  
NIP. 19890515 201801 1 001

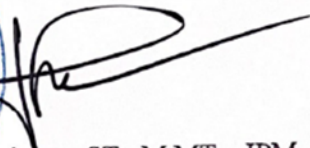
Sekretaris Penguji : Prof. Dr. Suhartono, M.Kom  
NIP. 19680519 200312 1 001

Anggota Penguji : Irwan Budi Santoso, M.Kom  
NIP. 19770103 201101 1 004

(  )  
(  )  
(  )  
(  )

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrul Kurniawan ST., M.MT., IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : A. Fuad Jauhari

NIM : 15650067

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Klasifikasi Jenis Beras Menggunakan Metode Convolutional Neural Network pada Arsitektur MobileNet

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Juni 2022  
Yang membuat pernyataan,



A. Fuad Jauhari  
NIM.15650067

## **MOTTO**

*... Bersyukur, Ikhlas, Sabar, dan Berprasangka Baik ...*

## HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Dengan menyebut nama Allah Yang Maha Pengasih dan Maha Penyayang.*

Alhamdulillah puji syukur selalu kami panjatkan kehadiran Allah SWT atas limpahan rahmat, taufiq, hidayah dan kemudahan sehingga skripsi dapat terselesaikan.

Sholawat dan salam selalu terlimpahkan kepada Nabi Muhammad SAW yang kita harapkan syafaatnya pada hari kiamat. Amin ya Rabbal ‘Alamin.

### **Penulis mempersembahkan karya ini kepada:**

Orang tua penulis secara jasmani, Bapak A. Harianto dan Ibu Siti Khoiriyah yang selalu mendukung, mendoakan dan memberikan semangat dalam menyelesaikan perkuliahan S1 Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Orang tua penulis secara rohani, Kyai Abdul Karim dan Ibu Nyai Siti Maslakah yang juga selalu membimbing, mendidik dan mendoakan penulis mulai dari tahun 2016.

Dosen pembimbing penulis, Bapak Prof. Dr. Suhartono, M.Kom dan Bapak Irwan Budi Santoso, M.Kom yang telah membimbing dan membantu dengan sabar dalam proses penulisan tugas akhir/skripsi.

Semua Keluarga Besar Teknik Informatika UIN Malang, Pondok Pesantren Syai-Urrifa’, Rukuru Studio, dan semua orang yang penulis sayangi yang tidak bisa penulis sebutkan satu per satu.

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah puji syukur selalu kami panjatkan kehadirat Allah SWT atas limpahan rahmat, taufiq, hidayah dan kemudahan sehingga penyusunan skripsi yang berjudul “Klasifikasi Jenis Beras Menggunakan Metode Convolutional Neural Network pada Arsitektur MobileNet” dapat terselesaikan. Tujuan dari penyusunan skripsi ini untuk memenuhi salah satu syarat menempuh ujian sarjana Teknik informatika di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penyusunan skripsi ini telah melibatkan banyak pihak yang telah membantu dalam banyak hal. Oleh karena itu, penulis sampaikan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A. selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan ST., M.MT., IPM selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Suhartono, M.Kom. selaku Dosen Pembimbing I yang telah membimbing dalam penyusunan skripsi ini.
5. Irwan Budi Santoso, M.Kom. selaku Dosen Pembimbing II yang telah membimbing dalam penyusunan skripsi ini.
6. Dr. M. Ainul Yaqin, M.Kom. selaku Dosen Wali yang selalu memberikan dukungan dan motivasi kepada penulis.
7. Orang tua yang telah memberikan doa dan dukungan kepada penulis sehingga penyusunan skripsi ini dapat terselesaikan.
8. Semua pihak yang telah membantu dan memberikan dukungan dalam penyusunan skripsi yang tidak bisa penulis sebutkan satu per satu.

Penulis menyadari bahwa penyusunan skripsi ini masih terdapat kekurangan. Oleh karena itu, penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca, khususnya untuk penulis secara pribadi.

Malang, 21 Juni 2022

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGAJUAN .....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN .....	iv
PERNYATAAN KEASLIAN TULISAN .....	v
MOTTO .....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR SIMBOL .....	xiv
ABSTRAK .....	xv
ABSTRACT .....	xvi
ملخص البحث.....	xvii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Pernyataan Masalah.....	7
1.3 Tujuan Penelitian.....	7
1.4 Manfaat Penelitian.....	7
1.5 Batasan Masalah.....	7
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1 <i>Deep Learning</i> .....	9
2.2 <i>Convolutional Neural Network (CNN)</i> .....	10
2.2.1 Arsitektur CNN.....	12
2.2.2 Arsitektur <i>MobileNet</i> .....	19
2.2.3 <i>Feature Extraction</i> (Ekstraksi Fitur) .....	21
2.3 <i>Google Colaboratory</i> .....	23
2.4 <i>Keras dan TensorFlow</i> .....	24
2.5 <i>TensorFlow Lite</i> .....	26
2.6 <i>Android Studio</i> .....	28
2.6.1 Struktur Proyek.....	28
2.6.2 Tampilan Antarmuka <i>Android Studio</i> .....	29
2.6.3 <i>Gradle System</i> .....	30
2.7 Beras.....	31
2.7.1 Beras IR 64 .....	32
2.7.2 Beras Ketan.....	32
2.7.3 Beras Basmathi .....	33
2.7.4 Beras Merah.....	34
2.7.5 Beras Hitam .....	35
2.7.6 Penelitian Terkait.....	36
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>39</b>

3.1 Desain sistem.....	39
3.1.1 <i>Input Image</i> .....	42
3.1.2 <i>Image Processing</i> menggunakan <i>Convolutional Neural Network (CNN)</i> <i>MobileNetV1</i> .....	44
3.1.3 <i>Training</i> .....	48
3.1.4 <i>Testing</i> .....	49
3.2 Desain Pengujian.....	49
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>51</b>
4.1 Perancangan Sistem dan <i>Dataset</i> Penelitian .....	51
4.1.1 <i>Dataset</i> .....	51
4.1.2 Pengambilan <i>Dataset</i> .....	52
4.1.3 Penyimpanan <i>Dataset</i> .....	53
4.1.4 Perancangan Program pada <i>Google Colaboratory</i> .....	54
4.1.5 Perancangan Program pada <i>Android Studio</i> .....	56
4.2 Pelatihan pada <i>Google Colaboratory</i> .....	58
4.2.1 Varietas Beras.....	59
4.2.2 Variasi Kualitas Gambar <i>Dataset</i> .....	59
4.2.3 Arsitektur CNN.....	60
4.3 Pengujian pada Perangkat <i>Android</i> .....	60
4.3.1 Hasil Pengujian pada Arsitektur <i>MobileNetV1</i> dengan Cahaya Putih.....	61
4.3.2 Hasil Pengujian pada Arsitektur <i>MobileNetV1</i> dengan Cahaya Kuning .	63
4.3.3 Hasil Pengujian pada Arsitektur <i>MobileNetV1</i> dengan Cahaya Putih dan Cahaya Kuning.....	65
4.3.4 Hasil Pengujian pada Arsitektur <i>MobileNetV1</i> per Varietas Beras .....	67
4.3.5 Perbandingan Hasil Pelatihan Semua Varietas Beras pada Semua Kondisi Cahaya.....	77
4.4 Pembahasan dan Integrasi Islam .....	80
<b>BAB V PENUTUP .....</b>	<b>83</b>
5.1 Kesimpulan.....	83
5.2 Saran.....	84
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 2.1 Jaringan Saraf Mendalam untuk Klasifikasi Digit Gambar .....	10
Gambar 2.2 Representasi Mendalam yang Dipelajari Model Klasifikasi Digit....	10
Gambar 2.3 Alur CNN pada Pengolahan Citra.....	11
Gambar 2.4 Contoh Jaringan CNN .....	12
Gambar 2.5 Alur Lapisan Konvolusi .....	13
Gambar 2.6 <i>Pooling Layer</i> .....	18
Gambar 2.7 <i>Fully Connected Layer</i> .....	19
Gambar 2.8 <i>Depthwise Separable Convolution</i> .....	20
Gambar 2.9 Pertukaran Klasifikasi Baru dengan Mempertahankan Basis Konvolusi yang Sama.....	23
Gambar 2.10 Tampilan <i>Google Colaboratory</i> .....	24
Gambar 2.11 Minat Pencarian pada <i>Search Engine Google</i> tentang Beberapa <i>Framework Deep Learning</i> dari Waktu ke Waktu .....	25
Gambar 2.12 Susunan Perangkat Lunak, Perangkat Keras pada <i>Deep Learning</i> .	26
Gambar 2.13 Arsitektur Model <i>TensorFlow Lite</i> .....	27
Gambar 2.14 Tampilan Antarmuka <i>Android Studio</i> .....	29
Gambar 2.15 Beras IR 64 .....	32
Gambar 2.16 Beras Ketan .....	33
Gambar 2.17 Beras Basmathi .....	34
Gambar 2.18 Beras Merah .....	35
Gambar 2.19 Beras Hitam .....	36
Gambar 3.1 Desain Arsitektur Sistem.....	40
Gambar 3.2 Perancangan Sistem .....	42
Gambar 3.3 Arsitektur <i>MobileNet</i> .....	45
Gambar 3.4 Arsitektur CNN .....	48
Gambar 4.1 Tampilan Direktori <i>Dataset Train</i> pada <i>Github</i> .....	53
Gambar 4.2 Tampilan Direktori <i>Data Test</i> pada <i>Github</i> .....	54
Gambar 4.3 Tampilan Program Klasifikasi Varietas Beras pada <i>Google Colaboratory</i> .....	55
Gambar 4.4 Tampilan Project Aplikasi Klasifikasi Varietas Beras di <i>Android Studio</i> .....	57
Gambar 4.5 <i>File Model (.tflite)</i> Beserta Labelnya (.txt) pada Folder <i>Assets</i> .....	58
Gambar 4.6 Hasil Pelatihan Beras Basmathi .....	69
Gambar 4.7 Hasil Pelatihan Beras Hitam .....	71
Gambar 4.8 Hasil Pelatihan Beras IR 64 .....	73
Gambar 4.9 Hasil Pelatihan Beras Ketan .....	75
Gambar 4.10 Hasil Pelatihan Beras Merah .....	77
Gambar 4.11 Kurva Hasil Pelatihan.....	77

## DAFTAR TABEL

Tabel 2.1 Arsitektur <i>MobileNet</i> .....	21
Tabel 3.1 <i>FlowChart Model</i> .....	47
Tabel 4.1 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Cahaya Putih .....	61
Tabel 4.2 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Cahaya Kuning .....	63
Tabel 4.3 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Cahaya Putih dan Kuning .....	65
Tabel 4.4 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Beras Basmathi .....	67
Tabel 4.5 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Beras Hitam .....	69
Tabel 4.6 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Beras IR 64 .....	71
Tabel 4.7 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Beras Ketan .....	73
Tabel 4.8 Hasil Pengujian Arsitektur <i>MobileNetV1</i> dengan Beras Merah .....	75
Tabel 4.9 Rata-rata Akurasi pada Masing-masing Varietas Beras .....	78
Tabel 4.10 Persentase Akurasi Benar dan Salah pada Masing-masing Varietas Beras .....	79

## DAFTAR SIMBOL

$W$	: Panjang atau tinggi <i>input</i>
$F$	: Panjang atau tinggi filter
$P$	: <i>Zero Padding</i>
$S$	: <i>Stride</i>

## ABSTRAK

Jauhari, A. Fuad. 2022. **Klasifikasi Jenis Beras Menggunakan Metode *Convolutional Neural Network* pada Arsitektur *MobileNet***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, M.Kom. (II) Irwan Budi Santoso, M.Kom.

---

**Kata Kunci** : *Convolutional Neural Network, Google Colaboratory, Klasifikasi Jenis Beras, Android.*

Metode *Convolutional Neural Network* (CNN) memiliki macam-macam model arsitektur yang dapat digunakan pada proses klasifikasi jenis beras berdasarkan citranya. Klasifikasi dapat lebih mudah digunakan jika arsitektur CNN dapat diterapkan pada perangkat *Android* yang umum digunakan.

Model arsitektur yang digunakan pada proses klasifikasi ini adalah arsitektur *MobileNet* dengan metode *Feature Extraction* (ekstraksi fitur). Arsitektur tersebut digunakan pada proses pelatihan dan pengujian pada *Google Colaboratory*. Hasil pelatihan tersebut disimpan dalam format *.tflite* untuk diimpor sebagai *Assets* pada *Android Studio* agar dapat diimplementasikan pada program *Android*.

Hasil pelatihan dan pengujian pada *Google Colaboratory* memiliki *dataset* yang memiliki ukuran 224x224 piksel. Hasil pelatihan dari arsitektur *MobileNet* pada *dataset* baik bernilai 1.0 dan nilai akurasi validasi sekitar 0.9333. Sedangkan pada *dataset* buruk akurasi pelatihan bernilai 1.0 dan nilai akurasi validasinya menurun menjadi 0.6889. Kemudian hasil pelatihan pada perangkat *Android* untuk masing-masing beras dalam 5 kali pengujian pada 3 kondisi cahaya yang berbeda adalah Beras Basmathi, Beras Hitam dan Beras Merah mencapai akurasi 100% pada semua kondisi cahaya. Beras IR 64 memiliki hasil akurasi mencapai 80% untuk cahaya putih, 60% untuk cahaya kuning, dan 100% untuk campuran cahaya putih dan kuning. Beras Ketan memiliki hasil akurasi mencapai 80% untuk cahaya putih, 100% untuk cahaya kuning, dan 80% untuk campuran cahaya putih dan kuning. Faktor yang dapat mempengaruhi perbedaan hasil pengujian pada perangkat *Android* adalah jarak obyek, kondisi cahaya, dan kualitas gambar.

## ABSTRACT

Jauhari, A. Fuad. 2022. **Rice Type Classification Using Convolutional Neural Network Method on MobileNet Architecture.** Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Prof. Dr. Suhartono, M.Kom. Supervisor (II) Irwan Budi Santoso, M.Kom.

---

**Keywords** : *Convolutional Neural Network, Google Colaboratory, Rice Type Classification, Android.*

The Convolutional Neural Network (CNN) method has various architectural models that can be used in the process of classifying rice types based on their image. Classification can be easier to use if the CNN architecture can be applied to commonly used Android devices.

The architectural model used in this classification process is the MobileNet architecture with the Feature Extraction method. The architecture is used in the training and testing process at Google Collaboratory. The training results are saved in .tflite format to be imported as Assets in Android Studio so that they can be implemented in Android programs.

The results of training and testing on Google Collaboratory have a dataset that has a size of 224x224 pixels. The training results from the MobileNet architecture on the dataset are both 1.0 and the validation accuracy value is around 0.9333. Meanwhile, in the bad dataset, the training accuracy is 1.0 and the validation accuracy value decreases to 0.6889. Then the results of the training on Android devices for each rice in 5 tests in 3 different light conditions are Basmathi Rice, Black Rice and Red Rice achieving 100% accuracy in all light conditions. Rice IR 64 has an accuracy of 80% for white light, 60% for yellow light, and 100% for a mixture of white and yellow light. Glutinous Rice has an accuracy of 80% for white light, 100% for yellow light, and 80% for a mixture of white and yellow light. Factors that can affect the difference in test results on Android devices are object distance, light conditions, and image quality.

## ملخص البحث

جوهري ,احمد فؤاد. ٢٠٢٢ . تصنيف نوع الأرز باستخدام طريقة الشبكة العصبية التلافيفية في بنية شبكة المحمول. أطروحة . قسم المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الحكومية الإسلامية في مالانج. المشرف : الأستاذ الدكتور سوهرتونو الماجستير و إروان بودى سنتوسو الماجستير.

---

**الكلمات الرئيسية :** الشبكة العصبية التلافيفية ، غوغل التعاونية ، تصنيف أنواع الأرز ، أندرويد.

تحتوي الشبكة العصبية التلافيفية (CNN) على مجموعة متنوعة من البنى التي يمكن استخدامها في عملية تصنيف أنواع الأرز بناءً على صورتها. يمكن أن يكون التصنيف أسهل في الاستخدام إذا كان من الممكن تطبيق بنية CNN على أجهزة أندرويد شائعة الاستخدام.

الهندسة المستخدمة في هذا التصنيف هي بنية شبكة المحمول مع طريقة استخراج الميزة. يتم استخدام البنية في عملية التدريب والاختبار في غوغل التعاونية. يتم حفظ نتائج التدريب بتنسيق . تفلانيت ليتم استيرادها كأصول في الاستوديو أندرويد بحيث يمكن تنفيذها في برامج أندرويد. تحتوي نتائج التدريب والاختبار على غوغل التعاونية على مجموعة بيانات بمقاس  $224 \times 224$  بكسل. وصلت نتائج التدريب من بنية شبكة المحمول على مجموعتي البيانات إلى ١.٠٠ ووصلت قيمة دقة التحقق إلى حوالي ٠.٩٣٣٣. وفي الوقت نفسه ، في مجموعة البيانات السيئة ، تصل دقة التدريب إلى ١.٠٠ وتنخفض قيمة دقة التحقق إلى ٠.٦٨٨٩. ثم نتائج التدريب على جهاز أندرويد لكل أرز في 5 اختبارات في 3 ظروف إضاءة مختلفة هي أرز بسمتي ، أرز أسود وأرز أحمر محققة دقة ١.٠٠٪ في جميع ظروف الإضاءة. تتميز أرز IR ٦٤ بدقة ٨٠٪ للضوء الأبيض و ٦٠٪ للضوء الأصفر و ١٠٠٪ لمزيج من الضوء الأبيض والأصفر. الأرز الدبق لديه دقة ٨٠٪ للضوء الأبيض و ١٠٠٪ للضوء الأصفر و ٨٠٪ لمزيج من الضوء الأبيض والأصفر. العوامل التي يمكن أن تؤثر على الاختلاف في نتائج الاختبار على أجهزة أندرويد هي مسافة الكائن وظروف الإضاءة وجودة الصورة.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada kehidupan sehari-hari, mayoritas penduduk Indonesia mengonsumsi beras sebagai bahan pokok makanan. Badan Standardisasi Nasional (BSN), menentukan mutu beras terbagi menjadi dua, yaitu premium dan medium. Beras terbagi atas 4 klasifikasi Standar Nasional Indonesia (SNI) mutu beras, yakni premium, medium I, medium II, dan medium III . Adapun syarat umum pada beras, yakni bebas hama dan penyakit, bebas bau apek, asam atau bau asing lainnya, bebas dari campuran dedak dan bekatul, dan bebas dari bahan kimia yang membahayakan konsumen. Untuk menentukan kualitas dan membedakan jenis beras tersebut maka proses pemeriksaan kualitas beras selalu dilakukan oleh para ahli dibidang pertanian dengan pengawasan dan pemeriksaan yang rutin (Badan Standardisasi Nasional, 2008). Kelemahan dari pemeriksaan diatas memerlukan waktu yang lama dan menghasilkan produk dengan kualitas yang tidak merata karena keterbatasan visual, faktor kelelahan, dan perbedaan persepsi masing-masing pengamat. Oleh karena itu, salah satu alternatif yang dibutuhkan adalah pengolahan citra agar pendeteksian lebih cepat dan akurat. Proses pengolahan citra ini memungkinkan digunakan secara luas terlebih dengan berkembangnya teknologi smartphone yang menawarkan fitur kamera yang semakin baik dan murah. Beberapa penelitian terdahulu mengembangkan algoritma pemrosesan citra dengan menilai menilai beras berdasarkan panjang, lebar, luas dan juga bekerja pada deteksi warna pada

bulir beras untuk menentukan kualitas beras (E. Z., 2017). Perlindungan konsumen mengenai kualitas beras sangatlah perlu, dikarenakan beras adalah komoditas makanan utama masyarakat Indonesia yang dikonsumsi setiap hari. Banyak terjadi penipuan dan pemalsuan kualitas beras oleh sebagian orang. Dengan menyampur atau mengoplos beras berkualitas baik dengan beras berkualitas rendah untuk dapat keuntungan yang banyak. Untuk melindungi masyarakat dari pemalsuan tersebut maka diperlukan pendeteksi kualitas beras yang nantinya bisa membantu masyarakat dan juga membantu pemerintah untuk mengontrol kualitas beras yang baik dan yang buruk yang sesuai Badan Standardisasi Nasional (BSN).

Dengan pentingnya hal tersebut, beras merupakan salah satu amanah Allah SWT yang dititipkan kepada manusia untuk dikonsumsi dengan sebaik-baiknya. Sebagai salah satu obyek konsumsi, pemilihan makanan yang berupa beras juga sangat penting. Al-Quran telah menjelaskan bahwa Allah telah menumbuhkan butir-butir tumbuhan dan biji-bijian yang sangat banyak jenisnya. Sebagaimana Firman Allah SWT dalam Surat Al-Baqoroh ayat 168 dan Surat Al-A'raf ayat 31 :

إِنَّ اللَّهَ فَالِكُ الْحَبِّ وَالنَّوَىٰ ۖ يُخْرِجُ الْحَيَّ مِنَ الْمَيِّتِ وَمُخْرِجُ الْمَيِّتِ مِنَ الْحَيِّ ۚ ذَٰلِكُمْ اللَّهُ فَالَىٰ تُوَفَّكُونَ (الانعام : ٩٥)

*“Sesungguhnya Allah menumbuhkan butir tumbuh-tumbuhan dan biji buah-buahan. Dia mengeluarkan yang hidup dari yang mati dan mengeluarkan yang mati dari yang hidup. (Yang memiliki sifat-sifat) demikian ialah Allah, maka mengapa kamu masih berpaling?.”* (Q.S. Al-An'am (6) : 95).

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً ۖ فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا نُخْرِجُ مِنْهُ حَبًّا مُتَرَاكِبًا ۖ وَمِنَ النَّخْلِ مِنْ طَلْعِهَا قِنْوَانٌ دَانِيَةٌ وَجَنَّاتٍ مِّنْ أَعْنَابٍ وَالزَّيْتُونَ وَالرُّمَّانَ مُشْتَبِهًا وَغَيْرَ مُتَشَابِهٍ ۗ انظُرُوا إِلَى ثَمَرِهِ إِذَا أَثْمَرَ وَيَنْعِهِ ۗ إِنَّ فِي ذَٰلِكُمْ لَآيَاتٍ لِّقَوْمٍ يُؤْمِنُونَ (الانعام : ٩٩)

*“Dan Dialah yang menurunkan air dari langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan, maka kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau itu butir yang banyak; dan dari mayang kurma, mengurai tangkai-tangkai yang menjulai, dan kebun-kebun anggur, dan (Kami keluarkan pula) zaitun dan delima yang serupa dan yang tidak serupa, perhatikanlah buahnya pada waktu berbuah, dan menjadi masak. Sungguh, pada yang demikian itu ada tanda-tanda (kekuasaan Allah) bagi orang-orang yang beriman.” (Q.S. Al-An’am (6) : 99).*

وَفِي الْأَرْضِ قِطْعٌ مُّتَجَوِّرَاتٌ وَجَنَّاتٌ مِّنْ أَعْنَابٍ وَزُرْعٌ وَنَخِيلٌ صِنَوَانٌ وَغَيْرُ صِنَوَانٍ يُسْقَى بِمَاءٍ وَاحِدٍ وَنُفِصَلٌ  
بَعْضَهَا عَلَى بَعْضٍ فِي الْأُكُلِ , إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّقَوْمٍ يَعْمَلُونَ (الرعد : ٤)

*“Dan di bumi terdapat bagian-bagian yang berdampingan, kebun-kebun anggur, tanaman-tanaman, pohon kurma yang bercabang, dan yang tidak bercabang; disirami dengan air yang sama, tetapi Kami lebihkan tanaman yang satu dari yang lainnya dalam hal rasa. Sungguh, pada yang demikian itu terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang mengerti.” (Q.S. Ar-Ra’d (13) : 4).*

Tiga ayat tersebut merupakan ayat yang memiliki keterkaitan dengan bahan pangan atau hasil bumi. Ayat tersebut dirasa memiliki isi tentang makanan di bumi yang bermacam-macam jenisnya, yang salah satunya adalah padi atau beras. Selain itu ayat tersebut juga berisi tentang Allah telah menumbuhkan macam-macam butir tumbuhan dan itu merupakan salah satu dari sekian banyaknya kekuasaan Allah. Salah satu jenis butir tumbuhan yang bermacam-macam itu adalah beras. Oleh karena itu, penulis memilih beras sebagai objek untuk proses klasifikasi.

Pemilihan beras dapat dilakukan berdasarkan jenisnya. Beras memiliki banyak jenis, seperti beras putih, coklat, merah, basmathi, hitam, ketan, dan lain-lain. Jenis beras yang paling banyak dikonsumsi oleh masyarakat adalah beras putih. Beberapa jenis beras yang beredar di pasaran, seperti beras IR 64, IR 42, basmathi, pandanwangi, ketan, rojo lele dan lain-lain (Trisnawan, dkk. 2019:16). Jenis-jenis beras dapat diklasifikasikan berdasarkan ukuran, bentuk, warna, tekstur, dan lain-lain.

Klasifikasi ini sangat penting untuk dilakukan karena masing-masing beras memiliki kandungan nutrisi yang berbeda. Teknik yang dilakukan pada proses klasifikasi jenis beras dapat dilakukan dengan melihat bentuk maupun warna beras. Jenis atau varietas beras yang digunakan dalam penelitian ini adalah lima jenis, yaitu beras IR 64, beras ketan, beras basmathi, beras merah, dan beras hitam.

Klasifikasi jenis beras dapat dilakukan dengan cara manual, yaitu dengan menggunakan visual dari mata manusia. Namun, proses tersebut dapat memberikan efek klasifikasi yang kurang akurat. Hal tersebut dikarenakan mata manusia sangat terbatas oleh waktu. Selain itu, cara tersebut dapat memungkinkan adanya kesalahan yang disebabkan oleh kelalaian, kelelahan mata manusia, dan faktor-faktor yang lain.

Selain dengan cara manual, klasifikasi jenis beras juga dapat dilakukan dengan memanfaatkan teknologi informasi, yaitu dengan menggunakan teknologi pengolahan citra digital. Teknologi tersebut diharapkan dapat mengatasi kelemahan dari cara manual seperti yang telah diuraikan sebelumnya. Proses klasifikasi dengan teknologi pengolahan citra digital dapat mendeteksi jenis beras melalui warna maupun bentuknya. Namun, proses klasifikasi yang masih umum dilakukan adalah proses klasifikasi yang masih menggunakan cara manual melalui visual mata manusia.

Rifayani Fadhilah (2019) melakukan penelitian dengan memanfaatkan pengolahan citra digital untuk melakukan perancangan program klasifikasi kopi beras. Penelitian tersebut dilakukan dengan metode rekayasa. Obyek rekayasa yang digunakan adalah citra yang dapat ditangkap oleh sensor penangkap citra. Metode

rekayasa dilakukan dengan menggunakan program pengolah citra yang dapat mengklasifikasikan kopi beras sesuai dengan karakteristik citranya.

Mas'ud Effendi, dkk (2017) membuat penelitian dengan memanfaatkan metode jaringan syaraf tiruan pada pengolahan citra digital untuk mengidentifikasi jenis dan mutu kopi. Penelitian tersebut menggunakan data citra sebanyak 570 gambar dengan 30 gambar pada masing-masing mutu. Perbandingan data yang digunakan adalah 70% untuk training dan 30% untuk testing. Beberapa metode pengolahan citra digital yang dapat digunakan untuk klasifikasi jenis beras adalah Jaringan Syaraf Tiruan, *K-Means Clustering*, *Convolutional Neural Network*, dan lain-lain. Beberapa metode tersebut memiliki karakter, kelebihan dan kekurangan masing-masing. Metode yang digunakan pada penelitian ini adalah metode *Convolutional Neural Network*.

Metode *Convolutional Neural Network* (CNN) merupakan jenis *deep learning* yang merupakan cabang dari *machine learning*. Metode ini dapat membuat komputer melakukan pekerjaan seperti manusia, maksudnya komputer dapat belajar dari proses training yang telah disediakan atau dibuat (Deng & Yu, 2013). Metode *Convolutional Neural Network* (CNN) merupakan metode deep learning yang mampu melakukan proses pembelajaran secara mandiri untuk melakukan beberapa proses, seperti pengenalan objek, ekstraksi objek, dan klasifikasi. Metode ini juga dapat diterapkan pada citra yang memiliki resolusi tinggi. Metode *Convolutional Neural Network* (CNN) merupakan operasi citra digital yang menggabungkan beberapa lapisan pemrosesan. CNN merepresentasikan dalam bentuk 2 dimensi, sehingga metode ini dapat digunakan dalam pemrosesan yang

menggunakan input berupa citra. Struktur CNN terdiri atas *input*, ekstraksi fitur, klasifikasi dan *output*.

Arrofiqoh & Harintaka (2018) melakukan penelitian dengan mengimplementasikan metode *convolutional neural network* untuk klasifikasi tanaman pada citra resolusi tinggi. Penelitian tersebut bermula karena adanya kendala pada proses interpretasi secara manual yaitu ketika melakukan pengenalan objek secara visual. Pada penelitian tersebut, *Convolutional Neural Network (CNN)* digunakan untuk menentukan perbedaan jenis tanaman dengan pelabelan semantik dari jenis tanaman. Terdapat 5 jenis tanaman yang digunakan pada penelitian. Hasil dari penelitian tersebut menyatakan bahwa metode *Convolutional Neural Network (CNN)* memiliki potensi dalam pendekatan untuk mengenali objek otomatis dalam menentukan perbedaan jenis tanaman sebagai bahan pertimbangan bagi interpreter penentuan objek citra.

Berdasarkan pemaparan diatas maka, Penelitian ini akan mengaplikasikan metode *Convolutional Neural Network (CNN)* dan menggunakan arsitektur *mobilenet* dalam melakukan klasifikasi jenis beras. Kelas yang digunakan pada proses klasifikasi beras terdiri dari beberapa jenis beras, yaitu beras IR 64, beras ketan, beras basmathi, beras merah, dan beras hitam. Implementasi *Convolutional Neural Network (CNN)* dilakukan pada perangkat *Android*. Hal ini dikarenakan *Android* merupakan salah satu smartphone yang banyak dipakai oleh masyarakat. Penggunaan *Convolutional Neural Network (CNN)* pada perangkat *Android* ini diharapkan dapat membantu dalam proses klasifikasi jenis beras. Peneliti memilih arsitektur *mobilenet* karena klasifikasi yang akan dibangun berbasis *android app*.

Jadi, arsitektur *mobilenet* yang memiliki kemungkinan paling cocok untuk digunakan.

## **1.2 Pernyataan Masalah**

Berdasarkan penjelasan pada latar belakang maka didapati pertanyaan yang akan diangkat dalam penelitian ini yaitu, :

Bagaimana menggunakan metode *Convolutional Neural Network (CNN)* untuk klasifikasi jenis beras pada Arsitektur *MobileNet*?

## **1.3 Tujuan Penelitian**

Tujuan utama dari penelitian ini adalah untuk proses klasifikasi jenis beras. Adapun maksud dari tujuan ini yaitu :

Menggunakan metode *Convolutional Neural Network (CNN)* klasifikasi jenis beras pada Arsitektur *MobileNet*.

## **1.4 Manfaat Penelitian**

Manfaat dari penelitian ini memiliki dua aspek yaitu teoritis dan praktik. Adapun manfaat tersebut sebagai berikut :

1. Secara teoritis, pemahaman dan pembelajaran dalam penggunaan metode *Convolutional Neural Network (CNN)* pada proses klasifikasi jenis beras.
2. Secara praktik, memudahkan penggunaan dalam proses klasifikasi jenis beras.

## **1.5 Batasan Masalah**

Agar pembahasan penelitian ini tidak menyimpang dari perumusan, maka diperlukan batasan dalam penelitian sebagai berikut :

1. Metode yang digunakan untuk klasifikasi jenis beras adalah metode *Convolutional Neural Network (CNN)*.
2. Penelitian ini difokuskan hanya untuk lima jenis beras, yaitu beras IR 64, beras basmathi, beras ketan, beras merah, dan beras hitam.
3. Pengambilan gambar menggunakan *Smartphone Android* dengan format gambar berekstensi JPG.

## BAB II

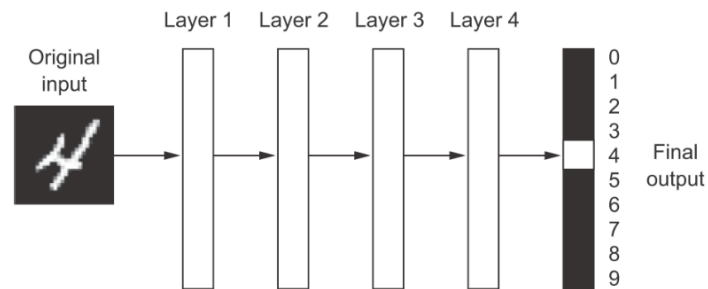
### TINJAUAN PUSTAKA

#### 2.1 *Deep Learning*

*Deep learning* merupakan salah satu dari cabang *machine learning* yang memiliki algoritma pemodelan tingkat tinggi pada data yang memiliki beberapa fungsi transformasi *non-linear* yang memiliki lapisan mendalam. Teknik dan algoritma ini dapat diterapkan untuk beberapa pembelajaran, yaitu pembelajaran terarah (*supervised learning*), tak terarah (*unsupervised learning*) dan semi terarah (*semi-supervised learning*). Ketiga pembelajaran tersebut dapat diterapkan pada beberapa aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi, dan lain-lain. *Deep learning* memiliki struktur dan jumlah jaringan saraf yang sangat banyak sampai ratusan lapisan (Dadang, 2018).

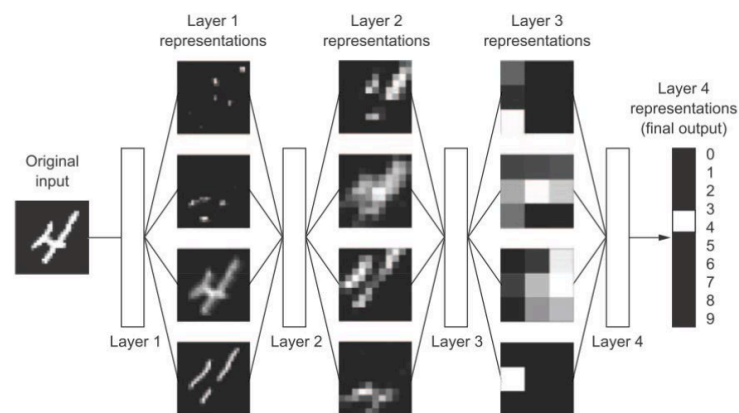
*Deep learning* merupakan jenis dari algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input kemudian diolah menggunakan sejumlah lapisan yang tersembunyi transformasi non linier dari data input untuk menghitung nilai output. *Deep Learning* memiliki fitur algoritma yang dapat digunakan untuk melakukan proses ekstraksi secara otomatis. Hal ini berarti bahwa algoritma *deep learning* dapat menangkap fitur yang relevan atau yang diperlukan untuk memecahkan masalah. Algoritma *deep learning* mampu mengurangi beban pemrograman dalam pemilihan fitur yang eksplisit. Selain itu, algoritma *deep learning* juga dapat digunakan dalam pemecahan masalah yang memerlukan pengawasan, tanpa pengawasan dan semi terawasi.

Representasi yang dipelajari pada algoritma *deep learning* dapat dilihat pada gambar 2.1.



Gambar 2.1 Jaringan saraf mendalam untuk klasifikasi digit gambar

Kemudian pada gambar 2.2, jaringan mengubah gambar digit menjadi representasi yang berbeda dan informatif dari gambar aslinya. Jaringan ini dapat disebut sebagai operasi distilasi dengan informasi yang bertingkat, yang mana informasi melewati beberapa filter kemudian menjadi output.



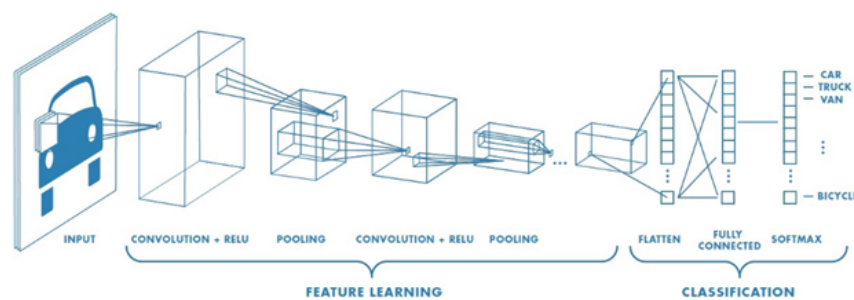
Gambar 2.2 Representasi mendalam yang dipelajari oleh model klasifikasi digit

## 2.2 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang digunakan untuk pengolahan data dua dimensi (Suartika, dkk. 2016). *Convolutional Neural Network* (CNN) merupakan jenis *neural network* yang dapat digunakan untuk pengolahan data image. *Convolutional Neural Network* terdiri dari neuron yang memiliki bias, *weight*, dan *activation*

*function*. Beberapa proses yang terdapat pada *Convolutional Neural Network*, yaitu *Convolutional Layer*, *Non-Linearity Layer (ReLU Layer)*, *Pooling Layer* dan *Fully-Connected Layer* (Paoletti, dkk. 2017).

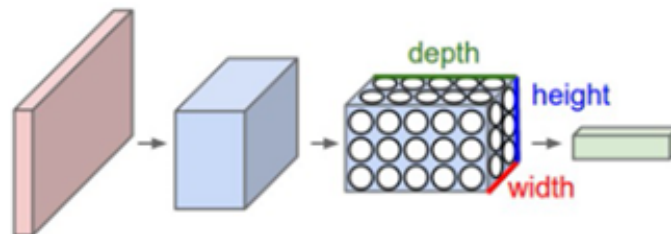
Alur proses pada *Convolutional Neural Network* (CNN) dalam pengolahan citra mulai dari proses masukan sampai klasifikasi citra menjadi hasil atau keluaran tertentu yang terlihat pada gambar 2.3.



Gambar 2.3 Alur CNN pada pengolahan citra

Proses konvolusi merupakan proses pada operasi aljabar linier yang memiliki fungsi untuk mengalikan matriks dari beberapa filter yang akan diproses. Proses tersebut merupakan salah satu jenis lapisan dari beberapa lapisan yang terdapat di dalam jaringan. Proses tersebut dinamakan lapisan konvolusi yang merupakan lapisan utama dan terpenting untuk digunakan. Selain itu terdapat jenis lapisan lain yaitu *Pooling Layer*. Lapisan tersebut digunakan untuk pengambilan nilai rata-rata atau nilai maksimal dari beberapa bagian piksel yang terdapat di dalam citra. Setiap lapisan input memiliki volume berbeda-beda yang memiliki kedalaman, lebar dan tinggi masing-masing. Besaran atau nilai yang didapatkan juga akan berbeda-beda, tergantung dari hasil proses filtrasi yang didapatkan dari lapisan yang dilakukan sebelumnya serta dari jumlah filter yang digunakan.

*Convolutional Neural Network* (CNN) juga dapat diterapkan pada klasifikasi objek berskala panjang, lebar dan tinggi seperti pada gambar 2.4.



Gambar 2.4 Contoh jaringan CNN

### 2.2.1 Arsitektur CNN

*Convolutional Neural Network* (CNN) memiliki lima lapisan atau *layer* utama sebagai berikut:

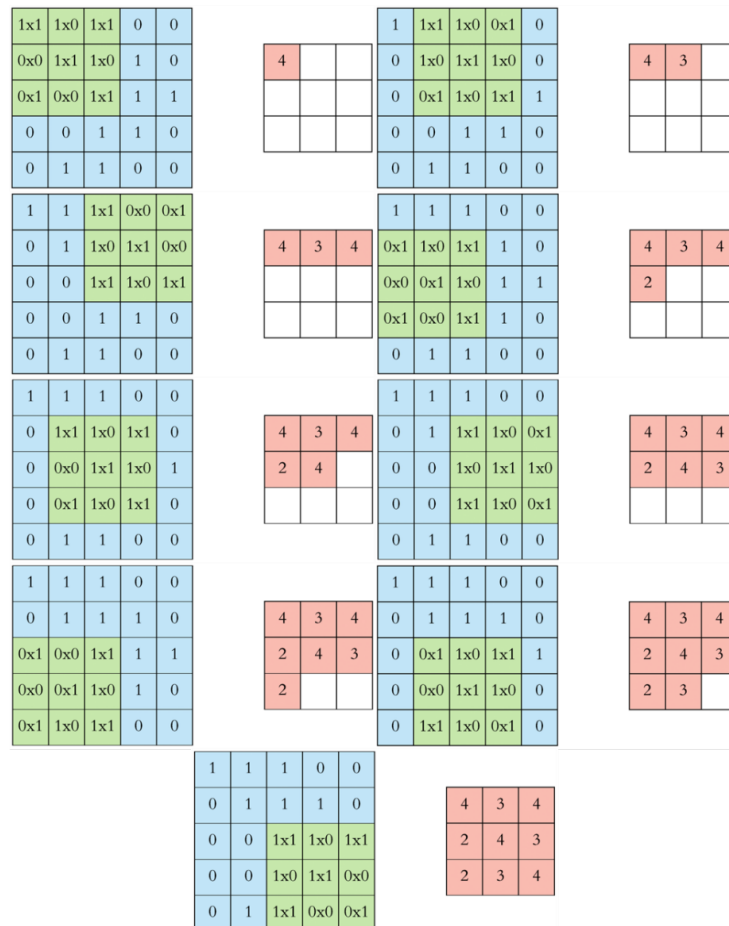
1. Lapisan Masukan (*Input Layer*)

Lapisan masukan berfungsi sebagai penampungan dari nilai piksel citra yang menjadi input atau masukan (Tandungan, 2019). Input menyesuaikan dengan ukuran dan *channel* warna dari citra. Seperti contoh jika terdapat citra yang berukuran 64x64 dan mempunyai 3 channel warna yaitu RGB (Red, Green, Blue), maka yang menjadi input adalah piksel *array* yang memiliki ukuran 64x64x3.

2. Lapisan Konvolusi (*Convolution Layer*)

Lapisan konvolusi (*convolution layer*) merupakan lapisan inti dari metode CNN (Tandungan, 2019). Lapisan ini menghasilkan sebuah citra baru dari citra input. Lapisan konvolusi memiliki susunan *neuron* untuk membentuk *filter* dua dimensi atau *filter* yang memiliki panjang dan tinggi (piksel). Lapisan ini menggunakan fungsi output sebagai *feature map* dari

input citra. Kemudian *feature map* tersebut digunakan pada lapisan aktivasi (*activation layer*). Berikut ini merupakan alur pada lapisan konvolusi.



Gambar 2.5 Alur Lapisan Konvolusi

Lapisan konvolusi memiliki tiga *hyperparameter* yang digunakan untuk mengatur ukuran volume keluaran atau output neuron yaitu kedalaman (*depth*), langkah (*stride*) dan *zero-padding*.

a. Kedalaman (*depth*)

Kedalaman (*depth*) merupakan *hyperparameter volume output* atau keluaran yang sesuai dengan jumlah filter yang digunakan, kemudian masing-masing akan belajar dan mencari sesuatu yang berbeda dari *input* atau masukan (CS231n). Kedalaman atau *depth* juga merupakan salah satu

dari ukuran *input*. Misalnya terdapat citra yang memiliki ukuran 64x64x3. Dari ukuran tersebut dapat diuraikan dengan lebar, tinggi, kedalaman.

b. Langkah (*stride*)

Langkah (*stride*) merupakan parameter penentu jumlah pergeseran filter (Chollet, 2018). Jika *stride* memiliki nilai 1, maka filter konvolusi akan bergeser 1 piksel secara vertikal dan secara otomatis akan bergeser secara horizontal. *Stride* yang semakin kecil akan menghasilkan informasi yang lebih detail, namun akan menyebabkan komputasi yang lebih detail dibandingkan dengan *stride* yang besar.

c. *Padding*

*Padding* juga bisa disebut dengan *zero padding* merupakan sebuah parameter untuk menentukan piksel bernilai 0 yang ditambahkan pada setiap sisi masukan atau inputan (Chollet, 2018). Hal ini berfungsi untuk manipulasi keluaran atau output dari lapisan konvolusi (*convolution layer*) pada *feature map*.

Penggunaan *padding* memiliki tujuan sebagai berikut.

- 1 Dimensi *output* atau keluaran dari lapisan konvolusi (*convolution layer*) selalu lebih kecil dibandingkan *input* atau masukan (kecuali penggunaan 1x1 filter dengan *stride* 1). *Output* atau keluaran ini digunakan lagi sebagai masukan dari lapisan konvolusi selanjutnya, sehingga informasi yang terbuang akan semakin banyak. Penggunaan *padding* ini juga dapat digunakan sebagai pengaturan dimensi *output* atau keluaran agar tetap sama dengan

dimensi *input* atau masukannya atau pengurangannya tidak secara drastis sehingga lapisan konvolusi dapat digunakan secara lebih dan juga dapat memperoleh *features* yang lebih banyak untuk diekstrak.

- 2 Meningkatkan performa model melalui filter konvolusi yang terfokus pada informasi sebenarnya yang berada di antara *zero padding*.

Persamaan yang digunakan untuk perhitungan dimensi *feature map* adalah sebagai berikut.

$$Output = \frac{W-F+2P}{s} + 1 \quad (2.1)$$

dengan:

W = Panjang/Tinggi *Input*

F = Panjang/Tinggi *Filter*

P = *Zero Padding*

S = *Stride*

Lapisan konvolusi memiliki skema pembagian parameter (*parameter sharing*) yang berfungsi untuk mengontrol jumlah parameter pada saat proses konvolusi melewati *input* (CS231n).

### 3 Lapisan Aktivasi (*Activation Layer*)

Lapisan aktivasi (*activation layer*) merupakan lapisan yang berfungsi untuk memasukkan *feature map* ke fungsi aktivasi (Tandungan, 2019). Fungsi aktivasi berfungsi untuk mengubah nilai ] *feature map* yang berada pada jangkauan tertentu dan sesuai dengan fungsi aktivasi yang digunakan. Hal ini bertujuan untuk meneruskan nilai yang menampilkan

fitur dominan dari citra yang kemudian akan masuk ke lapisan berikutnya. Penelitian ini menggunakan fungsi aktivasi *ReLU* dan *Softmax*.

#### A. Fungsi Aktivasi *ReLU*

Fungsi aktivasi *ReLU* adalah fungsi aktivasi yang bernilai nol jika  $x < 0$ , kemudian linier dengan kemiringan yang bernilai 1 jika  $x > 0$ . *ReLU* juga merupakan operasi yang digunakan untuk mengenalkan *nonlinearitas* dan berfungsi sebagai peningkatan representasi model. Fungsi aktivasi *ReLU* adalah  $f(x) = \max(0, x)$  (Heaton, 2015). Nilai keluaran atau output *neuron* dapat dinyatakan bernilai 0 jika inputnya memiliki nilai negatif. Jika inputnya positif, maka outputnya adalah nilai dari input itu sendiri (Kim, dkk. 2016).

Fungsi aktivasi *ReLU* memiliki kelebihan dapat mempercepat gradien stokastik daripada fungsi yang lain seperti fungsi sigmoid/tan h, hal ini dikarenakan fungsi aktivasi *ReLU* memiliki bentuk linier dan tidak menggunakan operasi eksponensial seperti yang terdapat pada fungsi sigmoid/tan h, sehingga fungsi aktivasi *ReLU* dapat melakukan proses pembuatan matriks aktivasi ketika ambang batas bernilai 0.

Selain itu, fungsi aktivasi *ReLU* memiliki kelemahan mudah rapuh atau mati pada saat proses training. Hal tersebut disebabkan karena terdapat gradien besar yang mengalir melalui fungsi aktivasi *ReLU* sehingga terjadi pembaruan bobot, kemudian neuron tidak aktif pada *datapoint* lagi. Jika hal tersebut terjadi, maka aliran gradien yang melalui unit selamanya akan bernilai nol dari titik tersebut. Kemudian unit *ReLU* dapat mati secara

*irreversible* selama proses pelatihan karena akan menyebabkan lumpuhnya *data manifold*.

#### B. Fungsi Aktivasi *Softmax*

Fungsi ini digunakan untuk menghitung probabilitas setiap kelas target atas semua kelas target yang memungkinkan dan berfungsi untuk membantu menentukan kelas target untuk masukkan. Fungsi aktivasi *softmax* juga digunakan untuk mendapatkan hasil klasifikasi (Arrofiqoh dan Harintaka, 2018). Fungsi ini biasanya diterapkan pada lapisan terakhir dari *Convolutional Neural Network*. Berikut ini merupakan persamaan dari fungsi aktivasi *Softmax*.

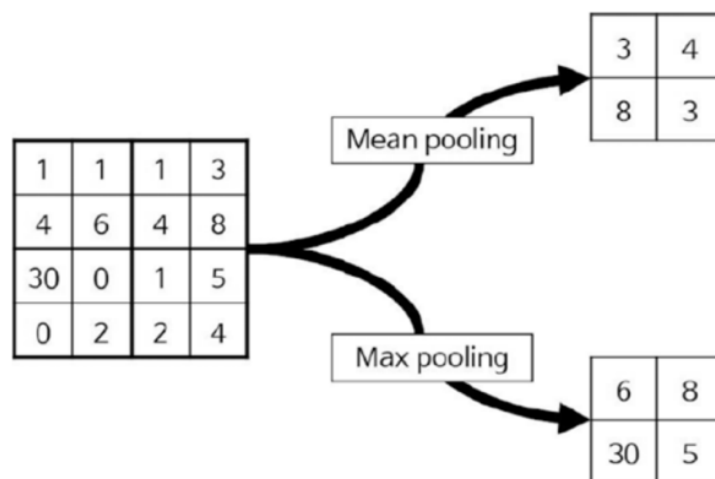
$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \text{ untuk } i = 1, 2, 3, \dots, k \quad (2.2)$$

$f_i$  menunjukkan hasil fungsi dari setiap elemen ke- $i$  pada vektor keluaran atau output kelas. Kemudian  $x$  merupakan hipotesis dari model pelatihan agar dapat diklasifikasi oleh fungsi *softmax*. Fungsi *softmax* memiliki keuntungan mengenai rentang probabilitas keluaran bernilai 0 sampai 1, dan jumlah semua probabilitas sama dengan 1 (William, dkk. 2017).

#### 4 *Pooling Layer*

*Layer* atau lapisan ini berfungsi menerima masukkan dari lapisan aktivasi yang berarti representasi ukuran spasial dan mengurangi jumlah parameternya untuk mengontrol *overfitting* (Arrofiqoh dan Harintaka, 2018). *Pooling layer* juga dapat digunakan untuk menyimpulkan data dengan reduksi menggunakan *Max pooling* (Lorentius, dkk. 2019).

*Pooling* juga bisa disebut *subsampling* atau *downsampling* yang berfungsi untuk mengurangi dimensi *feature map* dengan tanpa menghilangkan informasi penting yang terdapat di dalamnya. Proses pertama pada *pooling layer* yaitu menentukan ukuran *downsampling* yang akan digunakan pada *feature map*. Kemudian dilakukan proses *pooling* pada *feature map*. Tujuan *pooling layer* adalah mengurangi dimensi yang terdapat pada *feature map*. Hal tersebut akan menyebabkan komputasi semakin cepat karena semakin sedikit parameter yang diperbarui. Proses *pooling* pada *feature map* dapat dilihat seperti gambar berikut.

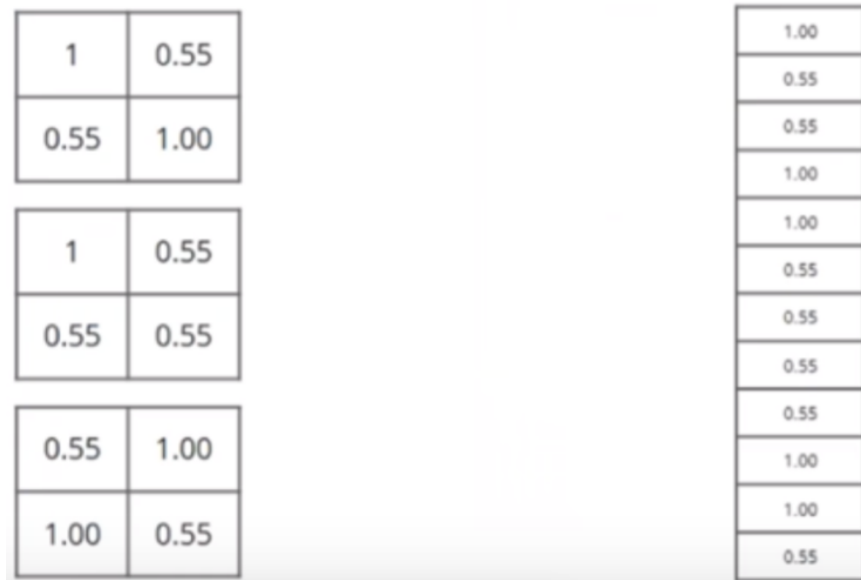


Gambar 2.6 *Pooling Layer*

## 5 Fully Connected Layer

*Layer* ini merupakan *layer* terakhir yang akan menerima hasil dari *pooling layer* untuk digunakan menjadi masukan dan data yang berupa matriks x-dimensi diubah menjadi matriks linear atau matriks 1 dimensi sehingga klasifikasi dapat dilakukan lebih mudah (Lorentius, dkk. 2019). Sebelum *layer* ini terdapat lapisan lain yang bernama lapisan *flatten*. Lapisan tersebut memiliki fungsi membentuk ulang *feature map* dari lapisan

yang telah diekstraksi menjadi vektor agar dapat digunakan sebagai masukan. *Fully connected layer* dapat dilihat pada gambar berikut.



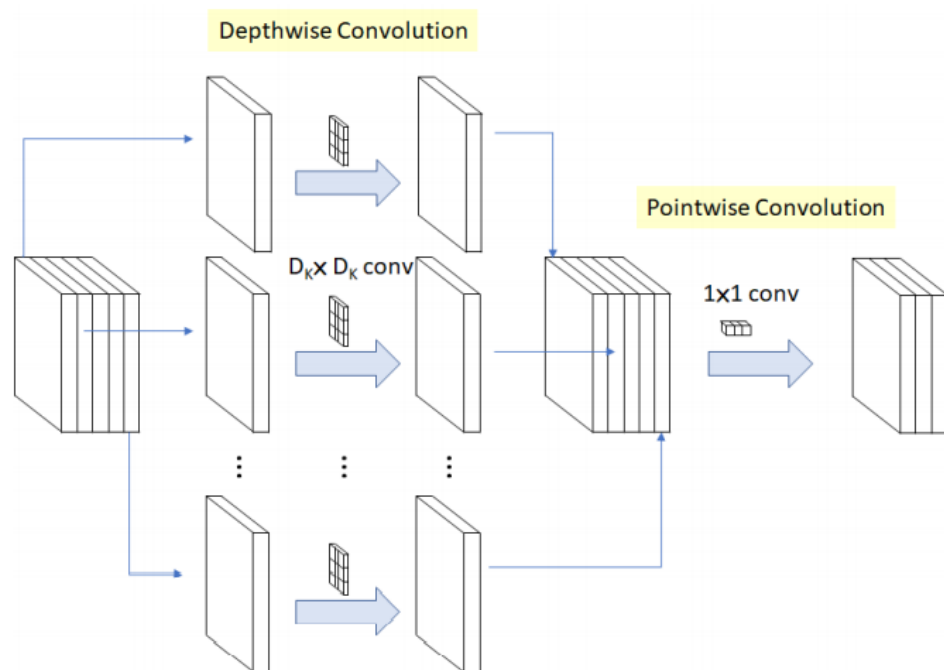
Gambar 2.7 *Fully Connected Layer*

### 2.2.2 Arsitektur *MobileNet*

Arsitektur ini merupakan salah satu arsitektur yang dapat digunakan untuk mengatasi kebutuhan pada saat *computing resource* yang berlebihan (Ekoputris). Arsitektur ini merupakan arsitektur CNN yang dibuat untuk bisa digunakan di ponsel. Penggunaan lapisan konvolusi pada arsitektur *MobileNet* menggunakan ketebalan *filter* yang sesuai dengan ketebalan dari gambar input atau masukan. Terdapat dua konvolusi pada arsitektur *MobileNet*, yaitu *depthwise convolution* dan *pointwise convolution*.

Model pada arsitektur *MobileNet* yang didasarkan pada konvolusi yang mendalam dapat dipisah-pisah (*depthwise separable convolution*) menjadi bentuk konvolusi yang menguraikan konvolusi standar (*standard convolution*) menjadi konvolusi mendalam (*depthwise convolution*) dan konvolusi 1x1 atau konvolusi

searah (*pointwise convolution*) (Google, 2017). *Depthwise convolution* memiliki penerapan satu *filter* pada setiap saluran masukan, kemudian *pointwise convolution* menggunakan konvolusi  $1 \times 1$  untuk menggabungkan hasil keluaran dari *depthwise convolution*. Sedangkan *standard convolution* mem-*filter* dan melakukan penggabungan masukan pada sebuah *set* keluaran yang baru. *Depthwise separable convolution* membagi lapisan menjadi dua bagian berdasarkan fungsinya, yaitu lapisan untuk *filter* dan lapisan untuk penggabung. Pembagian lapisan ini dapat mengurangi komputasi dan ukuran model.



Gambar 2.8 *Depthwise Separable Convolution*

Semua lapisan pada arsitektur *MobileNet* diikuti oleh *batchnorm* dan ReLU, namun terdapat pengecualian pada lapisan akhir yang terhubung dan tidak memiliki non linier dan masuk pada lapisan *softmax* untuk dilakukan pada proses klasifikasi. Berikut ini merupakan tabel arsitektur *MobileNet*.

Tabel 2.1 Arsitektur *MobileNet*

<i>Type/Stride</i>	<i>Filter Shape</i>	<i>Input Size</i>
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5x Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool $7 \times 7$	$7 \times 7 \times 1024$
FC / s1	$1024 \times 1000$	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

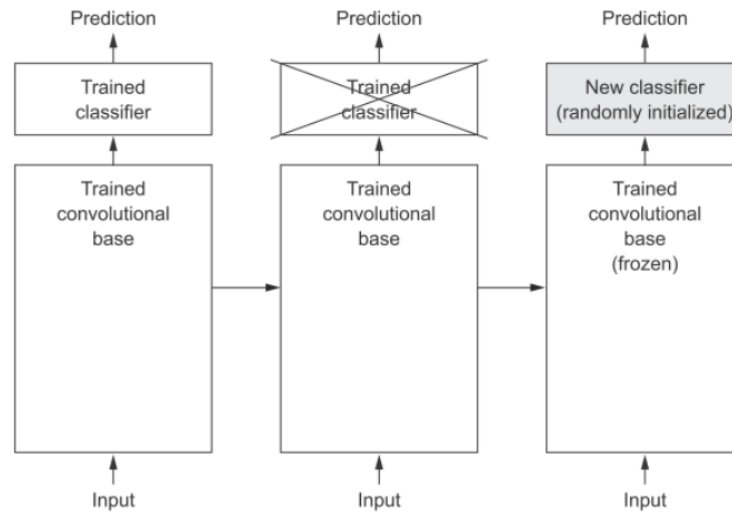
### 2.2.3 Feature Extraction (Ekstraksi Fitur)

*Deep Learning* memiliki pendekatan pada dataset yang berupa gambar dengan menggunakan jaringan yang sudah dilatih sebelumnya atau dinamakan *pretrained network*. *Network* atau jaringan ini merupakan jaringan yang telah disimpan dan dilatih pada *dataset* yang besar, biasanya terdapat pada proses klasifikasi gambar dalam skala yang besar (Chollet, 2018). Jika pada dataset asli cukup besar dan umum, maka hierarki spasial dari fitur yang telah dipelajari dan belum dilatih secara efektif dapat bertindak sebagai model generik dan fiturnya dapat digunakan untuk beberapa permasalahan komputer visual, meskipun masalah baru tersebut dapat melibatkan kelas yang berbeda dari tugas aslinya, misalnya

digunakan pada pelatihan jaringan yang terdapat di *ImageNet*, kemudian jaringan ini dapat digunakan kembali seperti identifikasi item furnitur pada gambar. *Deep learning* memiliki keuntungan yaitu probabilitas beberapa fitur yang dapat dipelajari pada beberapa masalah yang telah terlatih kemudian hal tersebut dibandingkan dengan beberapa pendekatan yang lama dan pembelajaran yang dangkal. Hal tersebut yang membuat *deep learning* lebih efektif untuk mengatasi masalah data yang kecil.

*Feature extraction* (ekstraksi fitur) menggunakan representasi yang telah dipelajari oleh jaringan sebelumnya agar dapat melakukan ekstraksi fitur dari sampel yang baru (Chollet, 2018). Fitur tersebut kemudian akan dijalankan melalui *classifier* baru yang telah dilatih dari awal.

*Convolutional Neural Network* (CNN) untuk klasifikasi data berupa gambar memiliki dua bagian yaitu dimulai dengan *pooling layer* dan *convolutional layer*, dan yang terakhir adalah klasifikasi yang terhubung erat (*densely connected layer*). Bagian yang pertama adalah *convolutional base* yang terdapat dalam model. *Convolutional Neural Network* (CNN) memiliki beberapa proses yang terdiri atas ekstraksi fitur untuk mengambil basis konvolusional dari jaringan yang telah dilatih sebelumnya, kemudian menjalankan data yang baru, dan melatih klasifikasi baru pada *output*. Hal ini dapat dilihat pada gambar berikut.

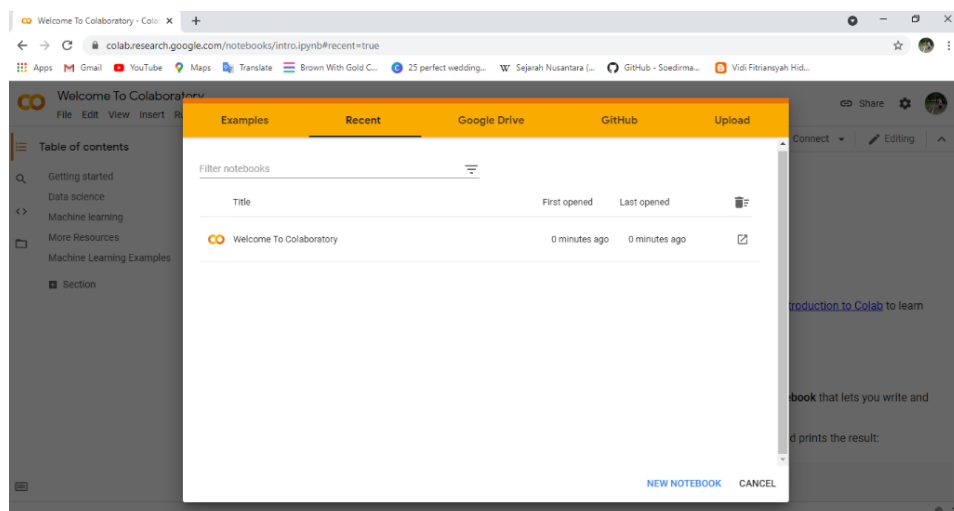


Gambar 2.9 Pertukaran Klasifikasi Baru dengan Mempertahankan Basis Konvolusi yang Sama

Gambar di atas menunjukkan bahwa hanya terdapat penggunaan *convolutional base* dan tidak terdapat penggunaan *densely connected classifier*. Hal ini dikarenakan representasi yang telah dipelajari *convolutional base* lebih generik sehingga dapat digunakan lagi.

### 2.3 Google Colaboratory

*Google Colaboratory* atau *Colaboratory* merupakan layanan yang berbasis *cloud* dari *Google* dan merupakan replikasi dari *Jupyter Notebook* di *cloud* (Mueller dan Massaron). Penggunaan *Google Colaboratory* tidak memerlukan proses penginstalan apapun di dalam sistem. Penggunaan *Google Colaboratory* tidak berbeda jauh seperti pada saat proses instalasi *desktop Jupyter Notebook*. Berikut ini merupakan tampilan dari *Google Colaboratory*.



Gambar 2.10 Tampilan *Google Colaboratory*

Penggunaan *Google Colaboratory* memerlukan akun *Google* untuk dapat diakses dan digunakan semua fitur yang terdapat di dalamnya. Penggunaan *Google Colaboratory* sama seperti *Jupyter Notebook*, yaitu melakukan tugas-tugas atau perintah-perintah yang berorientasi sel. *Google Colaboratory* dapat digunakan untuk membuat macam-macam sel dan digunakan untuk buku catatan. (Mueller & Massaron, 2019).

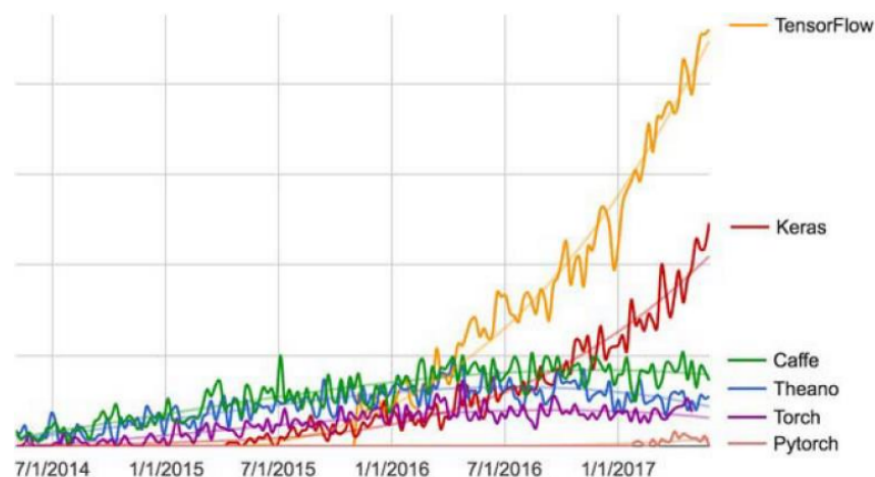
## 2.4 *Keras* dan *TensorFlow*

*Keras* merupakan salah satu *Framework* dari *deep learning* untuk *Python*. *Keras* memiliki kemampuan memudahkan untuk pendefinisian dan pelatihan pada mayoritas model *deep learning* (Webstaurant Store). Pada awalnya, *Keras* berfungsi untuk mempercepat eksperimen. *Keras* memiliki beberapa fitur, diantaranya:

- a. Memungkinkan kode dapat berjalan dengan sama-sama optimal pada CPU atau GPU.
- b. Memiliki API (*Application Programming Interface*) yang *user-friendly* sehingga mudah untuk prototipe pada model *deep learning*.

- c. Memiliki dukungan bawaan yang dapat digunakan untuk *convolutional networks*, *recurrent network*, dan kombinasi antara *convolutional networks* dan *recurrent network*.
- d. Mendukung arsitektur jaringan yang diinginkan seperti: model *multi-input* atau *multi-output*, berbagi lapisan, dan lain-lain. Hal ini berarti *Keras* dapat digunakan untuk membangun model *deep learning* apapun.

*Keras* merupakan *framework* yang berada di bawah lisensi MIT permissitif, yang mana dapat digunakan secara komersial dengan bebas atau *commercial use*. *Framework Keras* kompatibel dengan bahasa *Python* mulai dari versi 2,7 sampai 3,6.

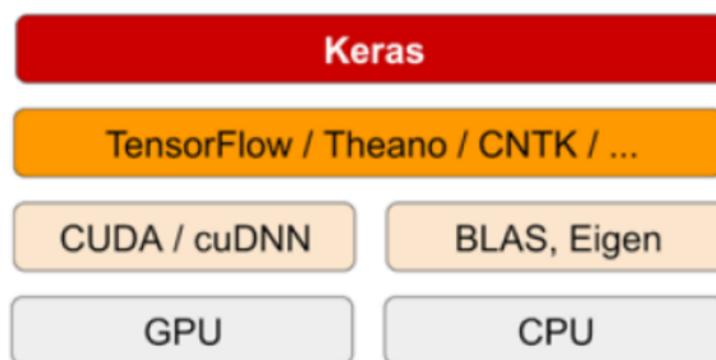


Gambar 2.11 Minat Pencarian pada *Search Engine Google* tentang Beberapa *Framework Deep Learning* dari Waktu ke Waktu

*Keras* merupakan *library* yang memiliki tingkatan-tingkatan model yang menyediakan blok-blok untuk mengembangkan model yang berada pada *deep learning* (Webstaurant Store). *Keras* memiliki kekurangan tidak dapat dilakukan untuk penanganan operasi tingkat rendah, misalnya manipulasi tensor dan

diferensiasi. Namun, terdapat *library sensor* khusus yang dioptimalkan dengan baik dan berfungsi sebagai mesin *backend* pada *Keras*.

Pada saat ini, *Keras* memiliki beberapa implementasi *backend* yaitu *backend TensorFlow*, *backend Theano*, dan *backend Microsoft Cognitive Toolkit (CNTK)*. Di masa yang akan datang *Keras* memiliki kemungkinan akan lebih berkembang dan lebih luas sehingga dapat bekerja lebih dengan mesin eksekusi yang terdapat pada *deep learning*.



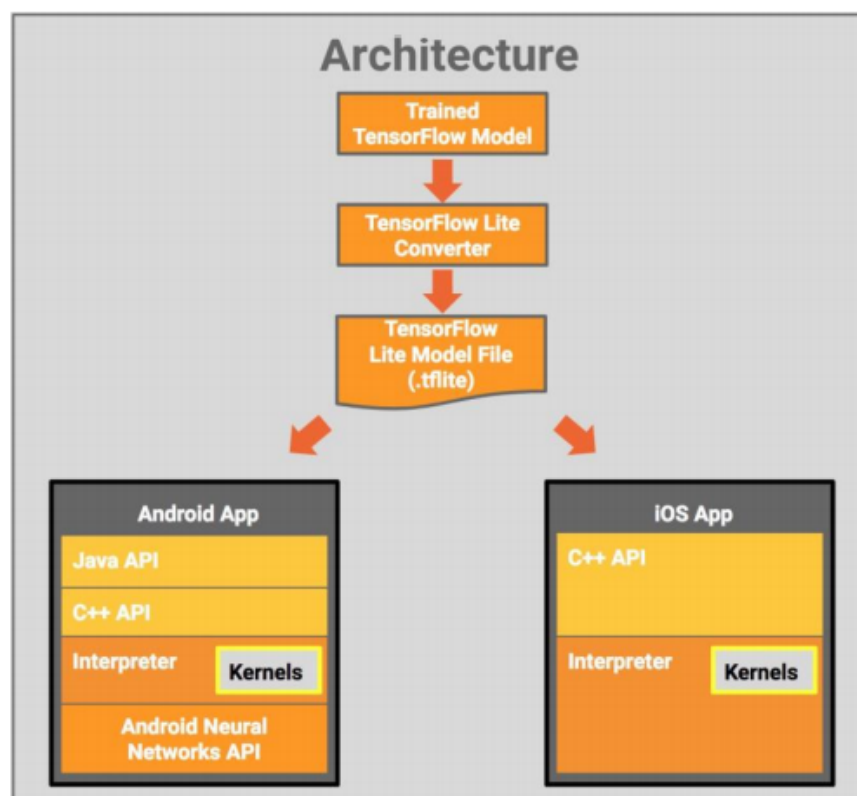
Gambar 2.12 Susunan Perangkat Lunak dan Perangkat Keras pada *Deep Learning*

*TensorFlow* dapat membantu *Keras* dapat berjalan lebih mulus pada CPU dan GPU. *TensorFlow* menggunakan *library* tingkat rendah untuk operasi pada *tensor* yang biasa disebut dengan *Eigen* pada saat berjalan di CPU. Sedangkan pada saat berjalan di GPU, *TensorFlow* menggunakan *library* pada operasi *deep learning* yaitu *library NVIDIA CUDA Deep Neural Network (cuDNN)* (Type Of Rice, 2020).

## 2.5 TensorFlow Lite

*TensorFlow Lite* merupakan jenis *TensorFlow* yang lebih ringan untuk dijalankan pada perangkat seluler (*TensorFlow*). *TensorFlow* merupakan jenis platform *end-to-end open source* yang dapat diterapkan pada *machine learning*. *TensorFlow* dapat menjalankan model-model yang terdapat di dalam mesin perangkat seluler dengan rendah. Hal tersebut juga bermanfaat pada proses

klasifikasi, regresi, dan beberapa proses lain tanpa adanya proses pengiriman dan penerimaan dengan server. *TensorFlow Lite* memiliki arsitektur seperti pada gambar berikut ini.



Gambar 2.13 Arsitektur Model *TensorFlow Lite*

*TensorFlow Lite* dapat dijalankan pada perangkat *Android* maupun *iOS* dengan bantuan *C++ API* dan memiliki *Java Wrapper* untuk pengembangan berbasis *Android*. Pada perangkat *Android* terdapat *interpreter* atau penerjemah untuk proses akselerasi dengan adanya *Android Neural Networks*. Selain itu penggunaan CPU standar juga dapat digunakan untuk proses eksekusi.

*TensorFlow Lite* digunakan untuk melatih model yang kemudian akan diproses pada mesin dan dikonversi menjadi format “.tflite”, kemudian model dengan format tersebut akan dimuat pada *interpreter* atau penerjemah seluler. (Using TensorFlow Lite on Android, 2020).

## 2.6 *Android Studio*

*Android Studio* merupakan aplikasi IDE atau perangkat lunak resmi yang dapat digunakan untuk pengembang aplikasi pada perangkat *Android* (Google). *Android Studio* memiliki beberapa fitur yang dapat dimanfaatkan untuk pengembangan aplikasi berbasis *Android* seperti:

1. Memiliki sistem *Gradle*
2. Memiliki *emulator* yang cepat
3. Memiliki *code template* dan terintegrasi dengan GitHub
4. Memiliki *framework* yang luas
5. Memiliki dukungan NDK dan C++
6. Memiliki dukungan *Google Cloud Platform*
7. Memiliki koreksi pada penulisan kode program

*Android Studio* dapat menggunakan bahasa pemrograman *Java*. *Android Studio* digunakan untuk pengembangan aplikasi android berupa penulisan, edit, penyimpanan proyek dan beberapa file lain yang berhubungan dengan proyek yang dikembangkan. *Android Studio* memiliki *Android SDK* sebagai ekstensi kode dari java yang digunakan untuk menjalankan aplikasi pada perangkat *Android*. Jadi, *Android Studio* berperan sebagai penghubung antara bahasa pemrograman *Java* dan *Android SDK* (Android Studio User Guide, 2020).

### 2.6.1 Struktur Proyek

Struktur proyek pada *Android Studio* memiliki modul yang berisi kode program (*source code*) dan file sumber (*resource file*) daya . Beberapa modul yang terdapat pada *Android Studio* yaitu:

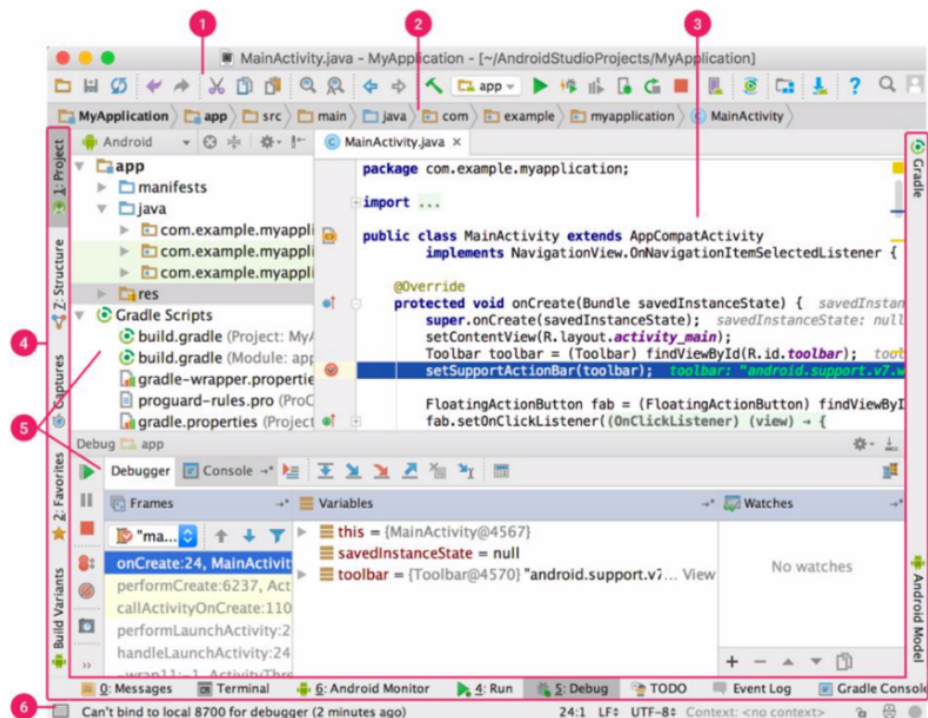
1. Modul aplikasi *Android*
2. Modul *Library*
3. Modul *Google App Engine*

Pada umumnya, *Android Studio* menunjukkan tampilan file proyek yang berisi proyek *Android*. (Android Studio User Guide, 2020) Proyek tersebut memiliki beberapa modul atau folder diantaranya:

1. *Manifests*
2. *Java*
3. *Res*

### 2.6.2 Tampilan Antarmuka *Android Studio*

Tampilan antarmuka pada *Android Studio* dapat dilihat pada gambar di bawah.



Gambar 2.14 Tampilan Antarmuka *Android Studio*

Dari gambar di atas, dapat dilihat bahwa pada tampilan *Android Studio* memiliki beberapa area yaitu:

1. Toolbar, yang berfungsi untuk melakukan beberapa tindakan seperti menjalankan aplikasi atau menjalankan *tools* yang lainnya.
2. Navigation bar, berfungsi sebagai navigasi pada proyek *Android Studio* dan juga untuk membuka atau mengubah file.
3. Editor window, berfungsi untuk pembuatan atau modifikasi kode program yang terdapat pada proyek *Android Studio*.
4. Tool window bar, yang terdapat dan berjalan di luar jendela IDE dan memiliki tombol yang dapat diperluas maupun dicitkan.
5. Tool windows, berfungsi untuk memberikan akses pada tugas-tugas tertentu seperti pencarian dan control-kontrol yang lain.
6. Status bar, yang berfungsi untuk menampilkan informasi proyek yang dikerjakan pada *Android Studio* dan memberikan informasi pesan yang lain.

### **2.6.3 Gradle System**

*Android Studio* memiliki fitur *Gradle* yang digunakan sebagai dasar pengembangan sistem dan memiliki kemampuan berdasarkan *plugin* yang telah disediakan untuk *Gradle*. Fitur *Gradle System* terdapat pada menu *Android Studio*.

Fitur ini memiliki beberapa kemampuan diantaranya:

1. Konfigurasi dan kustomisasi pada proyek.
2. Pembuatan beberapa apk yang memiliki beberapa fitur meskipun dengan menggunakan proyek dan modul yang sama.

3. Dapat menggunakan kode pada seluruh sumber proyek.

*Gradle* memiliki fleksibilitas untuk dapat melakukan proses-proses tanpa harus mengubah file inti dari aplikasi. File hasil build dari program *Android Studio* dinamakan “*build.gradle*”. File tersebut digunakan untuk mengatur *build* dengan elemen *plugin* yang telah tersedia untuk proses *gradle*. Setiap proyek pada *Android Studio* memiliki file *build* utama untuk keseluruhan isi proyek dan terdapat beberapa file build untuk setiap modul.

## 2.7 Beras

Beras merupakan salah satu makanan pokok masyarakat Indonesia, bahkan dunia (Trisnawan, dkk. 2019:16). Di Indonesia, beras merupakan salah satu komoditas yang berpengaruh untuk menyokong kebutuhan masyarakat. Beras menjadi salah satu kebutuhan pangan yang penting bagi masyarakat pedesaan maupun perkotaan. Pertambahan penduduk memiliki pengaruh terhadap kebutuhan beras. Semakin tinggi angka pertambahan penduduk di pedesaan maupun perkotaan, kebutuhan beras juga semakin tinggi (Fardhani, dkk. 2018:25). Hal tersebut mengakibatkan beras menjadi salah satu obyek yang penting untuk diperhatikan dan dipelajari.

Dalam kehidupan sehari-hari, beras menjadi makanan pokok bagi masyarakat Indonesia (Aji, dkk. 2020:40). Beras berperan sebagai makanan yang paling penting untuk dihidangkan. Mayoritas masyarakat Indonesia, khususnya Jawa berpendapat bahwa “tidak makan kalau tidak nasi”. Hal tersebut berarti bahwa seseorang dikatakan belum makan, jika belum makan nasi. Dalam hal ini yang dimaksud adalah beras, karena beras merupakan bahan untuk membuat nasi. Oleh

karena itu, masyarakat perlu memiliki ketersediaan beras yang cukup untuk memenuhi kebutuhan pangan sehari-hari.

Terdapat beberapa jenis beras yang beredar di pasaran, seperti beras IR 64, IR 42, basmathi, pandanwangi, ketan, rojo lele dan lain-lain (Trisnawan, dkk. 2019:16). Jenis-jenis beras dapat diklasifikasikan berdasarkan ukuran, bentuk, warna, tekstur, dan lain-lain.

### **2.7.1 Beras IR 64**

Beras IR 64 merupakan salah satu jenis beras yang berbentuk tegak dan memiliki tinggi sekitar 115-126 cm. Gabah pada beras IR 64 memiliki bentuk yang panjang dan tipis atau ramping, dan memiliki warna kuning bersih. Beras jenis IR 64 lebih cocok untuk ditanam pada daerah dengan dataran rendah sampai daerah dataran sedang (Suprihatno, dkk. 2010). Beras IR 64 jika sudah diolaha menjadi nasi memiliki kadar amilosa 23%. Beras IR 64 memiliki bobot per seribu butir sebanyak 24,1 gram.



Gambar 2.15 Beras IR 64

### **2.7.2 Beras Ketan**

Beras Ketan yang berwarna putih merupakan beras yang masuk dalam famili Graminae (Priyanto, 2012). Butir beras dari beras ketan memiliki zat pati yang bernilai sekitar 80-85% yang terdapat dalam endosperma yang memiliki

susunan granula pati yang memiliki ukuran 3-10 milimikron. Beras ketan juga memiliki kandungan vitamin, mineral dan air. Karbohidrat yang dimiliki oleh beras ketan berasal dari pati.



Gambar 2.16 Beras Ketan

### 2.7.3 Beras Basmathi

Beras Basmathi merupakan beras yang berasal dari India/Pakistan. Dalam Bahasa Sanskerta, basmathi 'basmathi' berarti harum atau wangi dan dalam Bahasa India ia juga mempunyai maksud "*soft rice*" yaitu lembut.

Beras Basmathi memiliki dua jenis beras, yaitu beras putih dan beras coklat. Keunikan Beras Basmathi akan terlihat setelah dimasak menjadi nasi. Beras ini akan terlihat lebih melar memanjang butiran, sedikit pera, mudah terurai dan aromanya sangat harum, beras ini memang memiliki aroma yang sangat harum. Selain itu, butiran atau buliran dari Beras Basmathi ini berbentuk panjang dan kecil melebihi ukuran beras yang biasanya agak pendek dan bulat.

Berdasarkan uji coba dari laboratorium, beras basmathi memiliki kandungan 0.09 bagian 2-acetyl-1-pyrroline per juta, yaitu 12 kali lebih dari konsentrasi pada beras biasa, karena secara alami wangi maka beras ini tak bisa ditandingi aromanya dengan beras lain.



Gambar 2.17 Beras Basmathi

#### 2.7.4 Beras Merah

Beras Merah merupakan beras yang diproses tanpa penyosohan. Beras merah digiling menjadi beras pecah kulit. Kulit ari dari beras merah masih melekat pada endosperm (Santika dan Rozakurniati, 2010).

Beras Merah memiliki beberapa kandungan yang unggul di dalamnya. Beras Merah memiliki kandungan gizi seperti serat dari asam-asam lemak esensial dan beberapa vitamin lainnya. Kandungan gizi beras merah per 100 g, terdiri atas protein 7,5 g, lemak 0,9 g, karbohidrat 77,5 g, kalsium 16 mg, fosfor 163 mg, zat besi 0,3 g, vitamin B1 0,21 mg dan antosianin (Indriani et al., 2013).

Beras Merah memiliki pigmen berwarna merah yang didalamnya terdapat senyawa antioksidan yang baik bagi kesehatan tubuh. Antioksidan merupakan molekul yang dapat menghambat oksidasi dari molekul lainnya. Reaksi oksidasi ini menghasilkan radikal bebas berantai yang dapat memberikan efek pada kerusakan atau kematian sel. Antioksidan akan menghentikan reaksi berantai tersebut dengan menghapus intermediet radikal bebas, dan menghambat reaksi oksidasi lainnya (Suprihatno, dkk., 2010).

Penyakit *Atherosclerosis* dapat dicegah dengan mengonsumsi Beras Merah, karena Beras Merah memiliki kandungan senyawa yang memberikan peningkatan antioksidan seperti asam amino, asam nikotinat, riboflavin dan berbagai mineral yang lain (Suardi, 2005).

Beras Merah memiliki nilai ekonomis yang lebih tinggi dibandingkan dengan beras biasa (putih) karena memiliki kandungan yang baik. Meskipun demikian, Beras Merah masih kalah populer jika dibandingkan dengan beras putih karena masa simpan Beras Merah lebih pendek dari beras putih.



Gambar 2.18 Beras Merah

### 2.7.5 Beras Hitam

Beras Hitam merupakan varietas atau beras jenis lokal yang memiliki pigmen (terutama antosianin) paling baik. Beras Hitam memiliki rasa dan aroma yang baik dan memiliki penampilan yang spesifik dan unik. Beras Hitam akan berwarna lebih pekat dengan rasa dan aroma yang menggugah selera makan setelah dimasak menjadi nasi (Suardi dan Ridwan, 2009).

Kristantini et al., (2012) mengemukakan bahwa beras hitam (*Oryza sativa* L.indica) merupakan salah satu jenis beras yang terdapat di dunia, di samping jenis beras yang lain seperti beras putih, beras coklat, dan beras merah. Beras Hitam

mulai populer dan dikonsumsi oleh sebagian masyarakat sebagai bahan pangan fungsional karena secara alami atau melalui proses tertentu mengandung satu atau lebih senyawa yang dianggap memiliki beberapa fungsi fisiologis yang bermanfaat terhadap kesehatan konsumennya. Warna ungu gelap dari Beras hitam disebabkan karena adanya kandungan antosianin tinggi yang terletak pada lapisan perikarp. Oki et al., (2002) dalam Narwidina (2009) mengatakan bahwa beras hitam (*Oryza sativa L.indica*) memiliki perikarp, aleuron dan endosperm berwarna merah-biru-ungu pekat yang menunjukkan adanya kandungan antosianin di dalamnya. Beras hitam mempunyai kandungan serat pangan (dietary fiber) dan hemiselulosa masing — masing bernilai 7,5% dan 5,8%, sedangkan beras putih hanya 5,4% dan 2,2%. Beras hitam berasal dari tanaman padi hitam *Oryza sativa L.* adalah nama ilmiah padi (Tjitrosoepomo, 2005).



Gambar 2.19 Beras Hitam

#### 2.7.6 Penelitian Terkait

Rifayani Fadhilah (2019) melakukan penelitian dengan memanfaatkan pengolahan citra digital untuk melakukan perancangan program klasifikasi kopi beras. Penelitian tersebut dilakukan dengan metode rekayasa. Obyek rekayasa yang digunakan adalah citra yang dapat ditangkap oleh sensor penangkap citra. Metode

rekayasa dilakukan dengan menggunakan program pengolah citra yang dapat mengklasifikasikan kopi beras sesuai dengan karakteristik citranya.

Mas'ud Effendi, dkk (2017) membuat penelitian dengan memanfaatkan metode jaringan syaraf tiruan pada pengolahan citra digital untuk mengidentifikasi jenis dan mutu kopi. Penelitian tersebut menggunakan data citra sebanyak 570 gambar dengan 30 gambar pada masing-masing mutu. Perbandingan data yang digunakan adalah 70% untuk training dan 30% untuk testing. Beberapa metode pengolahan citra digital yang dapat digunakan untuk klasifikasi jenis beras adalah Jaringan Syaraf Tiruan, *K-Means Clustering*, *Convolutional Neural Network*, dan lain-lain. Beberapa metode tersebut memiliki karakter, kelebihan dan kekurangan masing-masing. Metode yang digunakan pada penelitian ini adalah metode *Convolutional Neural Network*.

Arrofiqoh & Harintaka (2018) melakukan penelitian dengan mengimplementasikan metode *convolutional neural network* untuk proses klasifikasi tanaman pada citra resolusi tinggi. Penelitian tersebut bermula karena adanya kendala pada proses interpretasi secara manual yaitu ketika melakukan pengenalan objek secara visual. Pada penelitian tersebut, *Convolutional Neural Network* (CNN) digunakan sebagai metode untuk melihat perbedaan jenis tanaman dengan pelabelan semantik dari jenis tanaman. Terdapat 5 jenis tanaman yang digunakan pada penelitian. Hasil dari penelitian tersebut menyatakan bahwa metode *Convolutional Neural Network* (CNN) memiliki potensi dalam pendekatan pengenalan suatu objek secara otomatis dalam melihat perbedaan jenis tanaman sebagai bahan pertimbangan bagi interpreter dalam menentukan objek citra.

Dari penelitian terkait yang tercantum diatas, pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk klasifikasi jenis beras, karena metode tersebut dapat melakukan beberapa proses, seperti pengenalan objek, ekstraksi objek, dan klasifikasi jadi sangat cocok pada penelitian ini. Perbedaan dari penelitian diatas yaitu penelitian ini difokuskan hanya untuk lima jenis beras, yaitu beras IR 64, beras basmathi, beras ketan, beras merah, dan beras hitam. Program klasifikasi jenis beras dibuat menggunakan bahasa pemrograman python dengan antarmuka dan infrastruktur dari *Google Colaboratory*.

## BAB III

### METODOLOGI PENELITIAN

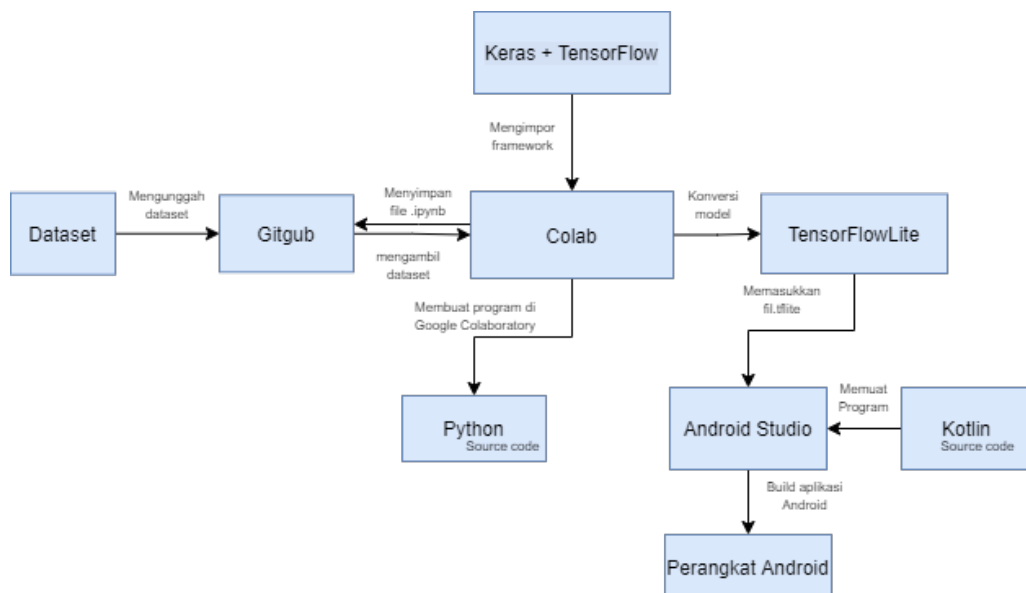
#### 3.1 Desain Sistem

Bab ini akan memberikan penjelasan tentang Langkah atau tahapan dan perancangan aplikasi penelitian agar prosesnya dapat terlihat lebih terstruktur sehingga proses penelitian dapat dipahami dengan baik. Penelitian ini dirancang untuk melakukan proses klasifikasi jenis beras dengan mengimplementasikan metode *Convolutional Neural Network (CNN)*. Beberapa pertanyaan akan timbul ketika membangun sebuah sistem, seperti contoh ‘sistem apa yang akan dibangun?’, ‘Bagaimana alur proses dari sistem tersebut?’ Bagaimana cara kerja dari sistem tersebut? Oleh karena itu, diperlukan perancangan sistem yang dapat pertanyaan-pertanyaan seperti contoh sehingga sistem akan tergambar lebih jelas dan mudah dipahami. Sistem yang akan dibangun dalam penelitian ini meliputi 4 tahap yaitu :

1. Tahapan *data preparation* digunakan untuk memanipulasi data.
2. Tahapan selanjutnya membuat pemodelan sistem dengan menentukan jumlah dari *convolutional layer*, *pooling layer*, *filter* dan *model fully connected layer* yang tersusun dari beberapa *layer*, setiap *layer* memiliki beberapa neuron yang saling berhubungan dengan menggunakan model arsitektur jaringan *Convolutional Neural Network (CNN) MobileNetV1*.  
Output dari tahapan ini adalah berupa klasifikasi jenis beras.

3. *Training* merupakan tahapan yang penting dari keberhasilan sebuah sistem yang dibangun. Jika hasil dari tahapan ini bagus, maka kemungkinan besar sistem bekerja dengan baik. Output yang dihasilkan dari tahapan proses ini adalah sebuah model *fitting* yang merupakan ciri-ciri dari klasifikasi jenis beras. Model *fitting* ini akan digunakan sebagai validasi dan perbandingan bobot dalam proses *testing*.
4. *Testing* merupakan tahapan terakhir pada penelitian ini. Tahapan ini bertujuan untuk mengetahui seberapa baik kinerja sistem dalam mengklasifikasikan jenis beras. Pada tahap *testing*, dilakukan proses validasi dengan melakukan perbandingan model *fitting* yang didapat dari tahapan *training* sebelumnya.

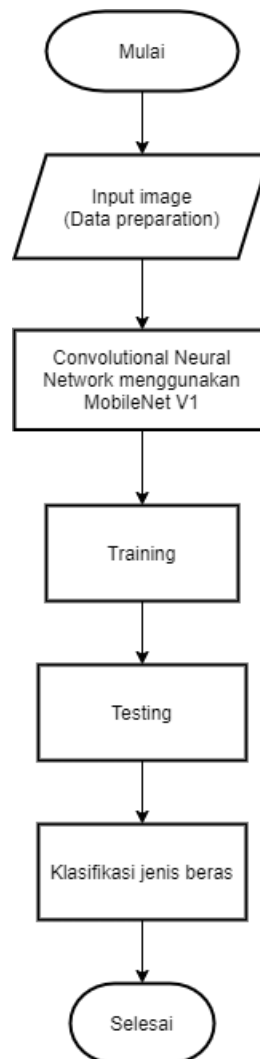
Desain arsitektur sistem yang akan dibangun pada penelitian ini digambarkan seperti gambar 3.1. berikut :



Gambar 3.1 Desain Arsitektur Sistem

Proses pada desain arsitektur sistem pada penelitian ini dimulai dari membuat *source code* untuk mengimplementasikan *Convolutional Neural Network (CNN)* dan mengimpor *Framework keras* dan *tensorflow* yang dibuat menggunakan infrastruktur *Google Colaboratory* dengan bahasa pemrograman *Python* dan disimpan dalam bentuk file *Jupyter Notebooks (.ipynb)* dan kemudian disimpan ke *Github*. Selanjutnya *dataset* pelatihan tersebut diambil dari *Github* dan disimpan ke dalam tempat penyimpanan sementara pada *Google Colaboratory*. Sistem yang dirancang ini menggunakan arsitektur jaringan *Convolutional Neural Network (CNN) MobileNetV1*, sedangkan untuk metode yang digunakan adalah metode *transfer learning* dengan mengambil *file Keras (\*.h5)* yang sudah dilatih sebelumnya pada kedua arsitektur tersebut dan melakukan *feature extraction* terhadap file tersebut. Selain itu juga ditambah satu *filter feature map* tambahan sebagai keluaran untuk klasifikasi tersebut.

Sedangkan untuk perancangan sistem yang akan dibangun pada penelitian ini digambarkan secara sederhana, seperti gambar berikut :



Gambar 3.2 Perancangan Sistem

### 3.1.1 Input image

Langkah pertama dalam melakukan penelitian ini yaitu pembuatan data *input* dengan melakukan pembagian data yang berisi 3 macam gambar, yaitu:

1. Data *training*, yang digunakan untuk proses *learning*.
2. Data *testing*, yang digunakan untuk proses *testing*.
3. Data *validation*, yang digunakan untuk validasi ketika proses *learning* berlangsung.

*Dataset* yang digunakan pada klasifikasi varietas beras yaitu *dataset training* dan *validation* merupakan *dataset* yang akan digunakan untuk melakukan pelatihan untuk memperoleh model dari arsitektur *MobileNetV1*. Sedangkan *dataset testing* digunakan untuk menguji tingkat probabilitas hasil yang benar pada arsitektur *MobileNetV1* pada *Google Colaboratory*. Seluruh *dataset* tersebut menggunakan gambar berwarna RGB (tiga saluran warna) dan diubah ukurannya menjadi 224x224 piksel sesuai dengan masukan pada arsitektur *MobileNetV1*.

Dari perancangan sistem pada gambar 3.2 dapat dijelaskan dari tahap *preparation* dan *pre-proses dataset* dilakukan dengan pengambilan gambar dari 5 macam obyek varietas beras yaitu beras ir64, beras basmathi, beras ketan, beras merah, dan beras hitam, menggunakan kamera smartphone dengan format gambar (.jpg) yang kemudian di-*crop* dan dimasukkan sebagai *dataset training* dan *testing* pada *github* yang selanjutnya akan dimasukkan ke *Google Colaboratory*.

*Dataset training* pada penelitian ini yang digunakan sebanyak 75 gambar berformat (.jpg) di setiap varietas beras sehingga totalnya 150 gambar untuk 2 varietas beras. *Dataset* dari keduanya disimpan di dalam folder yang terpisah. *Dataset training* ini diambil secara acak dari folder train yang berjumlah 75 gambar di setiap varietas beras.

*Dataset validation* yang digunakan dari folder train dengan jumlah 75 gambar. Sehingga jumlahnya sebanyak 15 gambar di setiap varietas beras dan totalnya sebanyak 30 gambar. *Dataset validation* digunakan untuk menguji dan membandingkan hasil pelatihan dengan *dataset* pelatihan di setiap epoch-nya.

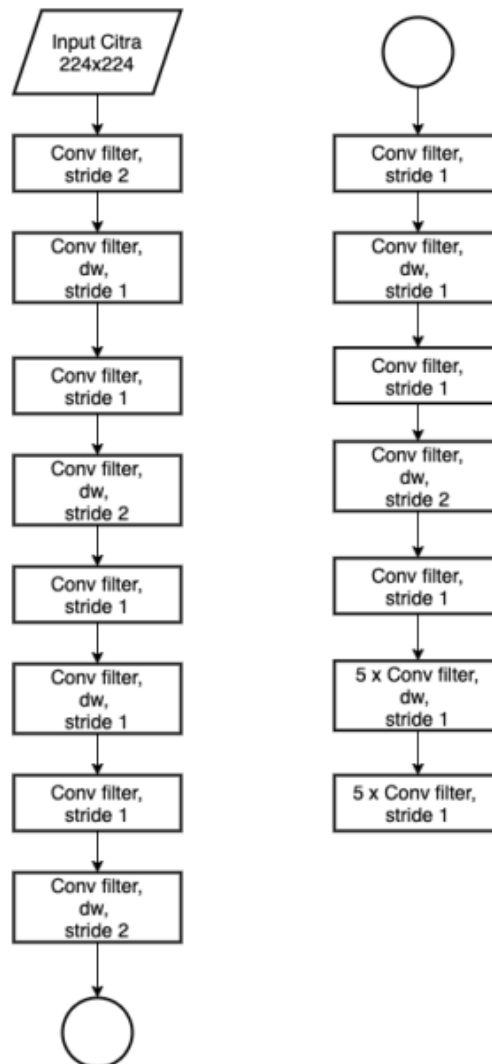
Dataset *validation* dari keduanya juga disimpan di dalam folder yang terpisah sesuai nama varietas berasnya.

Dataset *testing* yang digunakan sama seperti dataset *validation* yaitu sebanyak 15 gambar pada masing-masing varietas beras. Untuk dataset *testing* disimpan dalam folder test, terpisah dari dataset *training* dan *validation*. Dataset pengujian ini digunakan untuk menguji hasil *training* pada jaringan CNN yang digunakan.

### **3.1.2 Image processing menggunakan *Convolutional Neural Network* (CNN)**

#### ***MobileNetV1***

Arsitektur *MobileNet* menjadi jaringan dasar yang digunakan pada penelitian ini. *MobileNet* memiliki dua parameter yang digunakan untuk proses penyesuaian *trade-off* sumber daya/akurasi, termasuk *width multiplier* dan *resolution multiplier*. *Width multiplier* memungkinkan untuk membuat jaringan menjadi lebih tipis, sedangkan *resolution multiplier* mengubah dimensi input gambar untuk dapat mengurangi representasi internal di setiap lapisan. Dalam penelitian ini *MobileNet* digunakan untuk melakukan proses pembangunan lapisan konvolusi dasar di CNN. Arsitektur jaringan dasar didefinisikan, seperti gambar 3.3.



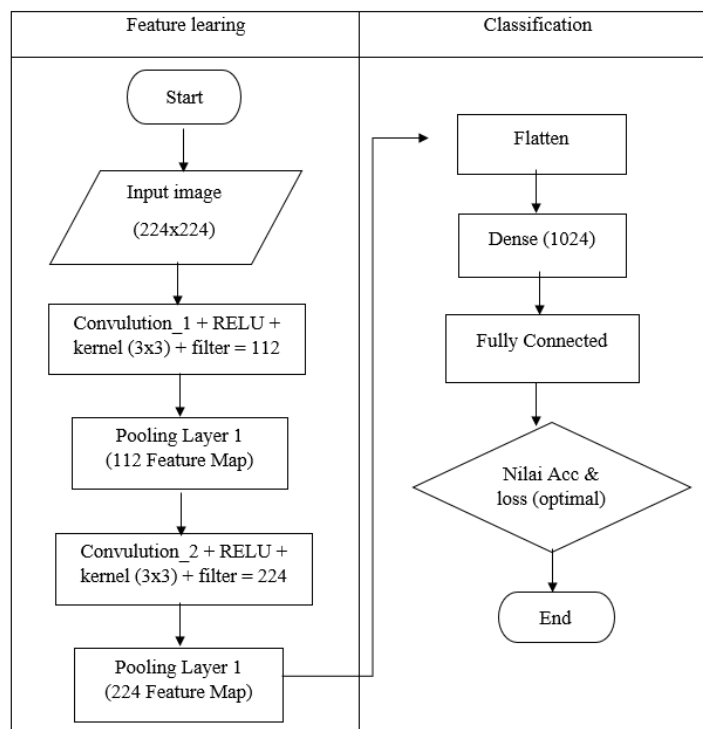
Gambar 3.3 Arsitektur *MobileNet*

Berdasarkan gambar diatas, “*Conv*” merupakan standar dari *convolution*, “*Conv dw*” merupakan *depthwise separable convolution*. *Stride* disimbolkan sebagai “s”, “s1” yang artinya *stride* memiliki ukuran 1x1, sedangkan “s2” merupakan *stride* yang memiliki ukuran 2x2.

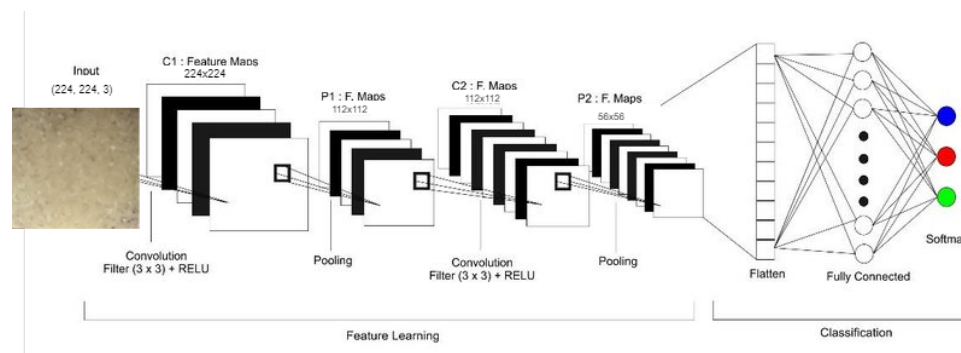
Konvolusi *separable depthwise* memiliki dua lapisan, yaitu konvolusi *depthwise* dan konvolusi *pointwise*. Konvolusi *Depthwise* digunakan untuk menerapkan *filter* tunggal pada masing-masing saluran input, sedangkan konvolusi

*pointwise*, konvolusi  $1 \times 1$  sederhana, digunakan untuk membuat kombinasi linier dari output lapisan *depthwise*. Arsitektur *MobileNet* menggunakan norma batch dan nonlinier *ReLU* untuk kedua lapisan. Pengurangan komputasi sebanding dengan jumlah saluran *feature map* keluaran dan kuadrat ukuran kernel.

Setelah dilakukan pembuatan dataset, proses selanjutnya adalah melakukan pelatihan model CNN. Berdasarkan kinerja CNN yang sangat baik dalam komputer vision, maka arsitektur yang akan dibangun pada penelitian ini terdiri dari dua tahapan yaitu *feature learning* dan *classification*. Input gambar pada model CNN menggunakan citra yang memiliki ukuran  $224 \times 224 \times 3$ . Angka 3 tersebut merupakan citra yang memiliki 3 channel yaitu *Red*, *Green*, dan *Blue* (RGB). Citra masukan atau *input* kemudian diproses melalui proses konvolusi dan *pooling* yang dilakukan pada tahapan *feature learning*. Jumlah proses konvolusi pada rancangan ini memiliki dua lapisan konvolusi. Setiap konvolusi memiliki jumlah filter dan ukuran kernel yang berbeda. Kemudian dilakukan proses *flatten* yang berfungsi untuk mengubah *feature map* dari hasil *pooling layer* menjadi bentuk *vector*. Proses ini disebut dengan proses *fully Connected layer*. Berikut adalah rancangan dari arsitektur CNN pada penelitian ini :

Tabel 3.1 *FlowChart Model*

Berdasarkan tabel diatas dijelaskan terdapat dua tahap dalam arsitektur CNN, yaitu *Feature Learning* dan *classification*. *Feature learning* merupakan teknik untuk memungkinkan sebuah system dapat berjalan secara otomatis untuk melakukan proses penentuan representasi dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut. Tahap *Classification* adalah sebuah tahap dimana hasil dari *feature learning* akan digunakan untuk proses klasifikasi berdasarkan *subclass* yang sudah ditentukan. Jika flow chart diatas diubah kedalam bentuk gambar, maka dapat dilihat seperti gambar 3.4 berikut :



Gambar 3.4 Arsitektur CNN

Konvolusi pertama menggunakan filter sebanyak 112 dan kernel dengan matriks 3x3. Kemudian dilakukan *pooling* menggunakan ukuran 2x2 dengan pergeseran *mask* dua langkah. Kemudian konvolusi kedua menggunakan filter sebanyak 224 dan kernel dengan matriks 2x2. Kemudian *flatten* melakukan perubahan *output* dari proses konvolusi dengan cara menggunakan struktur konvolusi yang dapat dipisahkan secara mendalam pada arsitektur *MobileNet* untuk membangun pengklasifikasi. Selanjutnya proses klasifikasi dengan menggunakan MLP (*Multi Layer Perceptron*) dengan jumlah neuron pada lapisan tersembunyi yang telah ditentukan. Kelas dari citra kemudian diklasifikasikan berdasarkan nilai neuron pada lapisan tersembunyi dengan menggunakan fungsi aktivasi *softmax*.

### 3.1.3 Training

Proses *training* ini merupakan bagian terpenting dari keberhasilan proses CNN, yang mana proses CNN dapat dikatakan berhasil jika proses *training* memiliki hasil bagus. Sistem akan mempelajari *dataset training* untuk klasifikasi jenis beras yang berjumlah lima jenis. Model *fitting* berisi model neuron yang harus diaktifkan ketika menunjukkan varietas berasnya. Setelah proses *training* selesai, model *fitting* kemudian disimpan untuk dipanggil kembali pada proses *testing*.

### 3.1.4 Testing

*Testing* dilakukan untuk melakukan pengujian tentang seberapa baik kinerja model pembelajaran sistem pada proses identifikasi dan klasifikasi jenis beras pada *image*. Sistem akan diberi *input* atau masukan dari data *testing* yang belum pernah digunakan sebelumnya. Dalam proses ini model *fitting* yang telah didapatkan dari proses *training* akan digunakan sebagai pembanding untuk proses klasifikasi berbagai jenis beras yang ada pada *image* inputan. Dengan demikian akan diketahui jenis beras sesuai dengan bentuk, ukuran dan nama beras.

### 3.2 Desain Pengujian

Proses pengujian dilakukan untuk melihat tingkat kesesuaian metode yang telah diterapkan dapat melakukan klasifikasi berdasarkan data yang ada. Proses klasifikasi dengan metode CNN pada penelitian ini dilakukan dengan menyediakan model dari *Tensor Flow Lite* yang sudah terlatih, maksudnya model tersebut sudah mempelajari keseluruhan data training yang berjumlah 375 data dari 5 jenis beras yang diklasifikasi. Kemudian model tersebut akan digunakan pada aplikasi android untuk proses klasifikasi data.

Proses pengujian ini dilakukan untuk melihat bagaimana sistem dapat melakukan proses klasifikasi terhadap beberapa jenis beras yaitu IR64, ketan, basmathi, merah, dan hitam. Kemudian dilakukan proses analisis pada metode yang digunakan untuk mengetahui penggunaan dan tingkat akurasi pada komputasi. Pada proses pengujian sistem, pengambilan data gambar dilakukan pada kotak kardus dengan pencahayaan lampu putih dan lampu kuning. Kotak ini digunakan untuk alas atau media untuk melakukan pengujian pada perangkat android, kemudian

intensitas cahaya yang digunakan pada setiap data gambar beras yang akan di uji coba, terdapat kondisi cahaya yang berbeda-beda untuk setiap pengujian antara lain cahaya dari lampu putih, cahaya dari lampu kuning, cahaya lampu putih dan kuning. Pemilihan lampu ini berdasarkan lampu yang umum digunakan untuk pencahayaan, baik di dalam maupun di luar ruangan. Lampu kuning memiliki kesan cahaya yang redup, romantis, hangat dan akrab serta identik dengan kondisi cahaya matahari (Grant-Hays dan Kimberley A, 2003). Selain itu lampu kuning juga memiliki intensitas cahaya yang tinggi dan berbanding lurus dengan kuat penerangan lampu (Janah, dkk. 2020:10). Sedangkan lampu putih merupakan lampu yang umum digunakan pada ruangan atau bisa dikatakan murni dan memiliki kesan terang dan dingin (Grant-Hays dan Kimberley A, 2003). Pemilihan lampu putih dan kuning ini juga berdasarkan tingkat atau daya serap energi yang lebih kecil karena memiliki nilai *dark current* dan voltase yang kecil dibandingkan dengan lampu yang berwarna selain putih dan kuning (Mutmainnah, dkk. 2020:205), Pengujian ini dilakukan dengan mengambil 75 data testing dari 5 jenis beras yang berbeda, untuk setiap jenis beras disediakan 15 data testing.

Pengujian dilakukan dengan menggunakan metode *confusion matrix* untuk mendapatkan nilai akurasi dari proses klasifikasi yang dilakukan pada *dataset*. *Confusion matrix* merupakan *tool* yang digunakan sebagai evaluasi model klasifikasi untuk memperkirakan objek benar atau salah. Sebuah *matrix* dari prediksi yang akan dibandingkan dengan kelas sebenarnya atau dengan kata lain berisi informasi nilai sebenarnya dan prediksi pada klasifikasi.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Perancangan Sistem dan Dataset Penelitian**

Penelitian ini menggunakan dua kondisi dataset dari kelima varietas beras. Dataset tersebut digunakan untuk melakukan perancangan sistem pada *Google Colaboratory*. Model arsitektur CNN yang diperoleh dari pelatihan pada *Google Colaboratory* disimpan dalam bentuk *TensorFlow Lite* yang nantinya akan digunakan untuk merancang sistem pada perangkat *Android*.

##### **4.1.1 Dataset**

Dataset yang digunakan pada klasifikasi varietas beras ini ada tiga macam, yaitu dataset pelatihan, dataset validasi, dan dataset pengujian. Dataset pelatihan dan validasi merupakan dataset yang akan digunakan untuk melakukan pelatihan untuk memperoleh model dari arsitektur yang digunakan. Sedangkan dataset pengujian digunakan untuk menguji tingkat probabilitas hasil yang benar pada arsitektur tersebut pada *Google Colaboratory*. Seluruh dataset tersebut menggunakan gambar berwarna RGB (tiga saluran warna) dan diubah ukurannya menjadi 224x224 piksel sesuai dengan masukkan pada arsitektur *MobileNetV1*.

##### **1. Jumlah Dataset Pelatihan**

Pada penelitian ini jumlah dataset pelatihan yang digunakan sebanyak 75 gambar berformat “.jpg” di setiap varietas beras sehingga totalnya 375 gambar untuk 5 varietas beras. Dataset dari kelimanya disimpan di dalam folder yang terpisah. Dataset pelatihan ini diambil

keseluruhan dari folder train yang berjumlah 75 gambar pada setiap varietas beras.

## 2. Jumlah Dataset Validasi

Dataset validasi yang digunakan sebanyak 20% dari folder train dengan jumlah 75 gambar. Sehingga jumlahnya sebanyak 15 gambar di setiap varietas beras dan totalnya sebanyak 75 gambar total dari 5 varietas beras. Dataset validasi digunakan untuk menguji dan membandingkan hasil pelatihan dengan dataset pelatihan di setiap *epoch*-nya. Dataset validasi dari ketiganya juga disimpan di dalam folder yang terpisah sesuai nama varietas berasnya.

## 3. Jumlah Dataset Pengujian

Jumlah dataset pengujian yang digunakan sama seperti dataset validasi yaitu sebanyak 15 gambar pada masing-masing varietas beras. Untuk dataset pengujian disimpan dalam folder test, terpisah dari dataset pelatihan dan validasi. Dataset pengujian ini digunakan untuk menguji hasil pelatihan pada jaringan CNN yang digunakan.

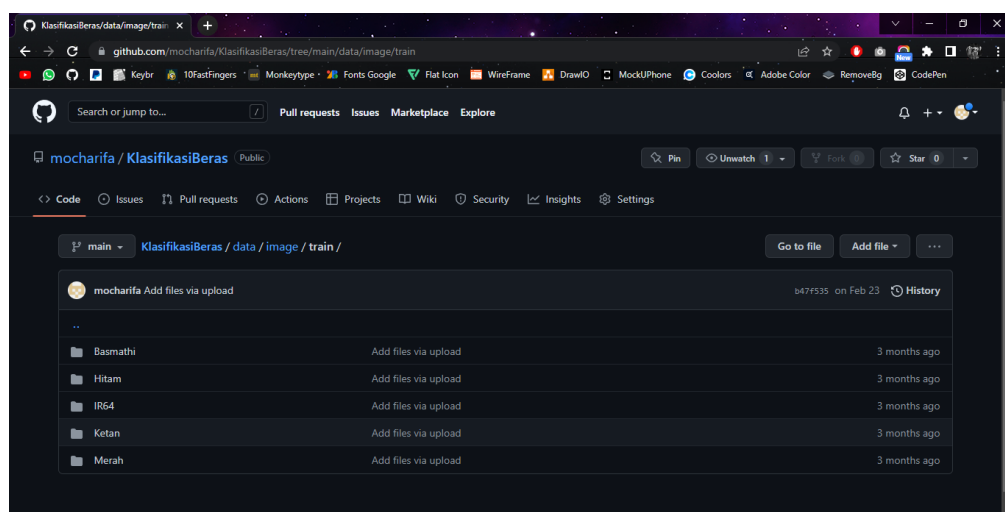
### 4.1.2 Pengambilan Dataset

Seluruh dataset dari varietas beras yang digunakan pada klasifikasi varietas beras diambil menggunakan sebuah kamera *Smartphone Android*. Pengambilan dataset dilakukan dengan mengambil gambar dari varietas beras yang akan dilatih dan diuji pada alas berwarna hitam dan putih dengan jarak antara obyek (beras) dengan kamera antara 10 cm sampai dengan 15 cm. Hal tersebut dilakukan karena

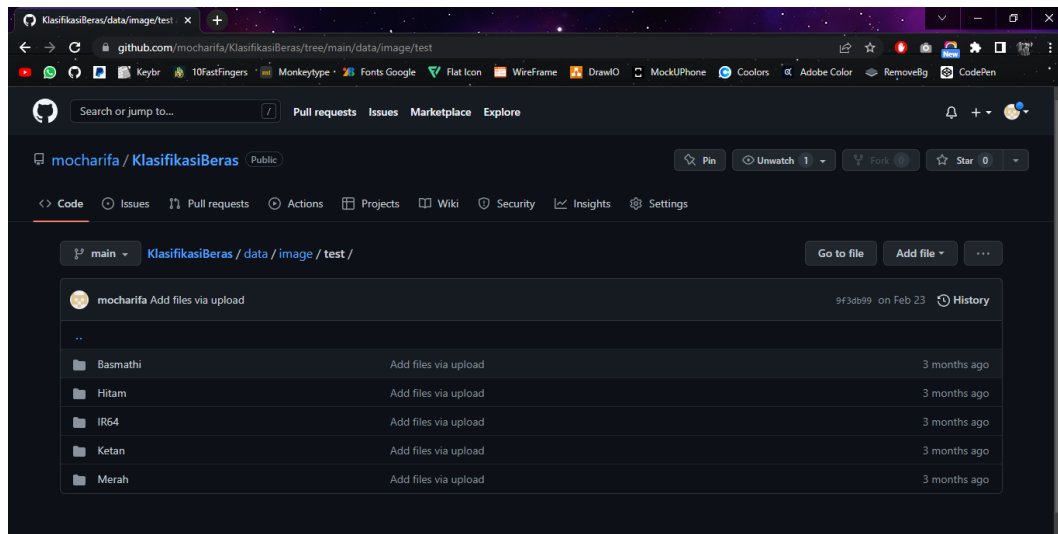
jarak tersebut sudah cukup dekat dan jelas hasilnya dan jika terlalu dekat maka kamera smartphone akan sulit untuk fokus terhadap obyek yang akan diambil. Alas berwarna hitam digunakan untuk mempermudah CNN melakukan klasifikasi obyek beras yang terlihat berbeda warnanya. Semua dataset yang diambil menggunakan kamera *smartphone* disimpan dalam format gambar “.jpg”.

### 4.1.3 Penyimpanan Dataset

Seluruh dataset pelatihan, dataset validasi, dan dataset pengujian diunggah dan disimpan pada layanan Repositori *Web Development* pada Platform *Github*. Dataset tersebut disimpan di dalam sebuah repositori dengan nama “KlasifikasiBeras”. Dataset pelatihan dan validasi disimpan di dalam folder “train” dan dataset pengujian disimpan di dalam folder “test”. Kedua folder tersebut disimpan di direktori folder “data” dengan subfolder “image”. Di setiap folder “train” dan “test” tersebut masing-masing terdapat 5 folder untuk menyimpan masing-masing dataset varietas beras, yaitu folder “Basmathi”, “Hitam”, “IR64”, “Ketan” dan “Merah”. Tampilan dataset train dan test dapat dilihat masing-masing pada gambar 4.1 dan gambar 4.2.



Gambar 4.1 Tampilan Direktori *Dataset Train* pada *Github*

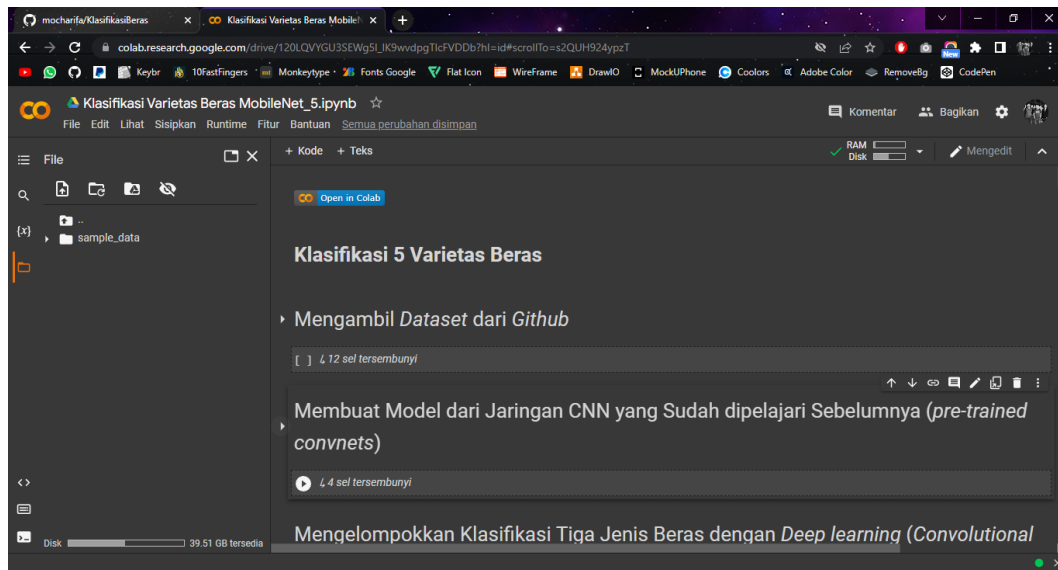


Gambar 4.2 Tampilan Direktori *Data Test* pada *Github*

Selain untuk menyimpan dataset, pada *repository* ini juga digunakan untuk menyimpan program klasifikasi varietas beras yang dibuat dengan *Google Colaboratory* serta program Aplikasi *Android* untuk klasifikasi varietas beras yang dibuat dengan *Android Studio*.

#### 4.1.4 Perancangan Program pada *Google Colaboratory*

Perancangan program pada infrastruktur *Google Colaboratory* dilakukan dengan membuat beberapa sel/baris text untuk memberi judul, keterangan dan penjelasan program di setiap barisnya serta membuat sel/baris code untuk mengimpor fungsi library dan membuat program di setiap baris yang dibuat. Program tersebut dibuat dengan *Framework Keras* dengan *backend TensorFlow*. Berikut tampilan program klasifikasi varietas beras pada *Google Colaboratory* dapat dilihat pada gambar 4.3.



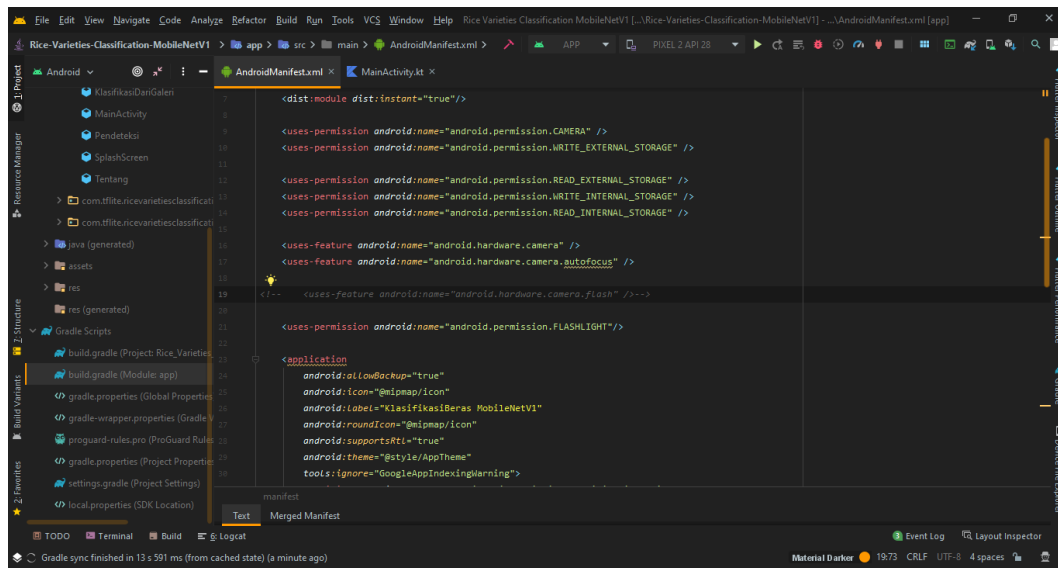
Gambar 4.3 Tampilan Program Klasifikasi Varietas Beras pada *Google Colaboratory*

Pembuatan program yang pertama adalah membuat program untuk mengambil dataset dari *Github*. Kemudian memasukan/mengimpor fungsi library yang nantinya akan digunakan untuk menjalankan program yang membutuhkan library tambahan tersebut. Selanjutnya terdapat program yang digunakan untuk mengubah ukuran 224x224 piksel sesuai pada masukan dari arsitektur *MobileNetV1*, memecah folder train menjadi dataset pelatihan dan validasi serta memuat seluruh dataset pengujian secara urut. Setelah itu dibuat program untuk menentukan nama label varietas berasnya urut secara abjad dan menyimpannya dalam bentuk text “.txt”. Selain itu membuat program untuk mengambil base model untuk melakukan feature extraction dari file model “.h5” yang sudah dilatih pada arsitektur *MobileNetV1*. Setelah itu membuat program untuk menambahkan lapisan klasifikasi untuk 5 kelas varietas beras dengan menggunakan teknik *Global Average Pooling*. Kemudian membuat program untuk melakukan pelatihan dari arsitektur tersebut dengan epochs sebanyak 100 kali. Selain itu dibuat program untuk menampilkan hasil pelatihan dalam bentuk grafik tingkat akurasi dan tingkat

kesalahannya. Sesudah itu terdapat program yang dibuat untuk menguji model hasil pelatihannya dengan dataset pengujian. Selanjutnya dibuat juga program untuk menampilkan hasil pengujian dalam bentuk confusion matrix yang berfungsi untuk mengetahui tingkat keberhasilan hasil pengujian dari setiap varietas beras. Pada tahap yang terakhir terdapat program yang dibuat untuk menyimpan dan mengkonversi model hasil pelatihan ke dalam bentuk file *TensorFlow Lite* “.tflite”. Kode sumber (source code) program Jupyter Notebooks “.ipynb” untuk klasifikasi varietas beras dengan CNN.

#### **4.1.5 Perancangan Program pada *Android Studio***

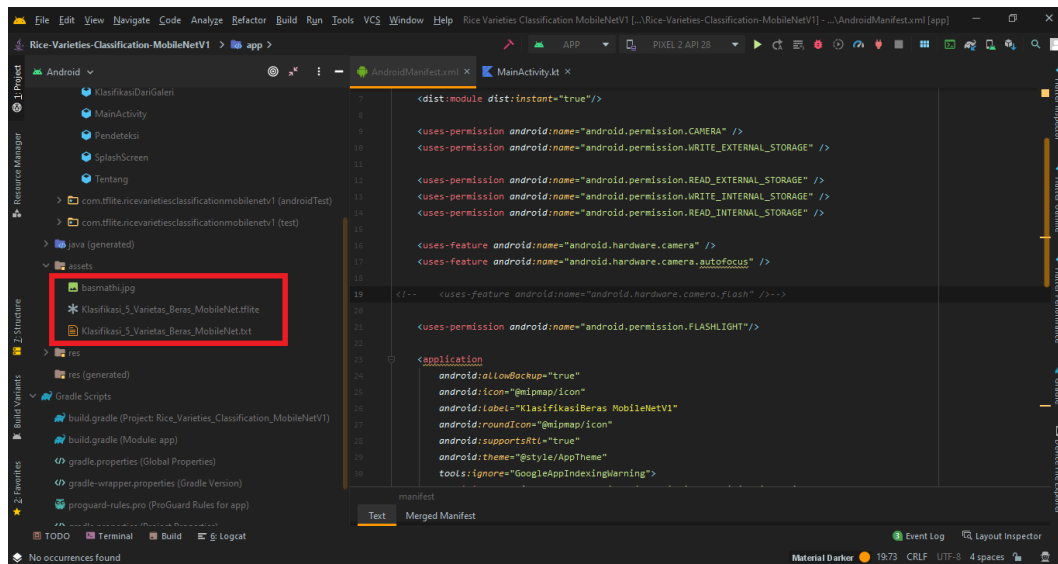
Perancangan program untuk perangkat *Android* dilakukan setelah membuat program pada *Google Colaboratory* dan mendapatkan hasil model pelatihannya yang disimpan dalam bentuk “.tflite”. Program dalam Bahasa *Kotlin* yang dibuat meliputi penggunaan fitur kamera pada *Smartphone Android*, mengimpor *library TensorFlow Lite*, membuat file *class* untuk klasifikasi obyek beras yang diambil dengan kamera maupun mengimpor gambar dari galeri foto, memasukkan file “.tflite” beserta labelnya dengan *format* “.txt” ke dalam folder assets serta membuat *class* lainnya untuk mendukung dalam membuat antarmuka aplikasi *Android*. Adapun pembuatan *layout* dalam *format* “.xml” yang berfungsi untuk membuat tampilan antarmuka pada aplikasi *Android* dari setiap *class* yang akan ditampilkan pada aplikasi *Android* untuk klasifikasi varietas beras. Gambar 4.4 berikut merupakan tampilan dari file *project* klasifikasi varietas beras pada *Android Studio*.



Gambar 4.4 Tampilan *Project* Aplikasi Klasifikasi Varietas Beras di *Android Studio*

Sebelum membuat class untuk klasifikasi varietas beras dengan kamera, maka perlu dimasukan fitur penggunaan kamera dan perijinan penggunaan kamera pada file “AndroidManifest.xml”. Selanjutnya untuk menggunakan *mobile interperter* sebagai penerjemah file “.tflite” maka diharuskan untuk melakukan *import library TensorFlow Lite* dengan cara memasukan implementasi “org.tensorflow.tensorflow-lite:1.13.1” pada dependencies yang terletak di build.gradle. Pada pembuatan program untuk klasifikasi varietas beras dengan kamera dibuat dengan 3 file class. Kelas yang pertama digunakan untuk membuat antarmuka kamera dan mengambil gambar yang selanjutnya akan diekstrak dalam bentuk array agar dapat diolah oleh model CNN. Kelas selanjutnya dibuat untuk memuat file model (.tflite) dan melakukan klasifikasi varietas beras serta memotong ukuran gambar RGB menjadi 224x224 piksel, sedangkan kelas yang ketiga untuk mengubah nilai data hasil klasifikasi dalam bentuk persentase. Pada pembuatan program klasifikasi varietas beras dilakukan dengan *import* gambar beras yang sudah tersimpan di dalam galeri foto *Android* dibuat dengan 2 file class. Kelas

pertama dibuat untuk mengimpor gambar dari galeri dan mengubah ukurannya menjadi 224x224 piksel serta menampilkan hasil prediksi dan probabilitasnya dalam persentase. Kelas berikutnya dibuat untuk melakukan klasifikasi, yaitu dengan membuat program untuk memuat file model (.tflite), mengekstrak gambar menjadi bentuk array, dan menghitung probabilitas hasil prediksi. Setelah seluruh program dibuat maka file model CNN yang sudah dilatih dan disimpan dalam bentuk “.tflite” beserta labelnya dalam bentuk “.txt” dimasukkan ke dalam folder “assets” di dalam project *Android Studio*.



Gambar 4.5 File Model (.tflite) beserta labelnya (.txt) pada folder assets

## 4.2 Pelatihan pada Google Colaboratory

Pelatihan dan pengujian pada infrastruktur *Google Colaboratory* berfungsi untuk melatih dataset dan memperoleh modelnya serta melakukan pengujian dari hasil pelatihan tersebut. Pelatihan dan pengujian tersebut dilakukan menggunakan metode *feature extraction* dengan cara mentransfer hasil pembelajaran menggunakan arsitektur dan *MobileNetV1*. Sedangkan untuk melakukan klasifikasinya menggunakan teknik *Global Average Pooling*. Pelatihan dan

pengujian pada *Google Colaboratory* ini dilakukan dengan menjalankan setiap baris program yang sudah dibuat. Pada saat pelatihan dan pengujian harus dipastikan bahwa laptop yang kita gunakan terhubung ke internet dan juga sudah terkoneksi dengan server *Google Colaboratory*. Sesudah terhubung ke server maka perlu mengubah pengaturan jenis *runtime*-nya menjadi GPU agar proses pelatihan dapat dilakukan lebih cepat. Setelah itu semua baris program dijalankan maka akan ditampilkan hasil pelatihan dan pengujiannya dalam bentuk kurva tingkat akurasi dan tingkat kesalahan dari pelatihan, validasi, dan pengujiannya. Selain itu juga model pada hasil pelatihan tersebut disimpan ke dalam direktori folder sementara pada *Google Colaboratory* yang nantinya akan diunduh dan digunakan pada aplikasi Android. Pengujian tersebut dilakukan terhadap lima varietas beras, variasi kualitas gambar dataset, dan arsitektur yang digunakan.

#### **4.2.1 Varietas Beras**

Varietas beras yang akan dilatih dan diuji adalah varietas Beras Basmathi, Beras Hitam, Beras IR-64, Beras Ketan dan Beras Merah. Kelima varietas tersebut masing-masing disimpan di dalam folder yang berbeda pada Github agar tidak tercampur pada saat pelatihan.

#### **4.2.2 Variasi Kualitas Gambar Dataset**

Dataset yang diambil untuk klasifikasi varietas beras terdapat dua kondisi, yaitu menggunakan kualitas gambar jernih dan kualitas gambar dengan *noise*. Kedua kondisi dataset tersebut diambil di dalam ruangan yang sama. Pada kondisi gambar jernih, intensitas cahaya yang mengenai obyek beras pada jarak 10 – 15 cm sekitar 1000 lux. Pada kondisi gambar dengan *noise*, intensitas cahaya yang

mengenai obyek sekitar 20 lux. Pengukuran intensitas cahaya dilakukan menggunakan aplikasi Light Meter pada perangkat Android.

#### **4.2.3 Arsitektur CNN**

Arsitektur CNN yang digunakan untuk pelatihan pada *Google Colaboratory* adalah arsitektur *MobileNetV1*. Metode CNN yang digunakan pada arsitektur tersebut adalah Feature Extraction.

#### **4.3 Pengujian pada Perangkat *Android***

Pengujian pada perangkat *Android* dilakukan setelah melakukan pengujian pada *Google Colaboratory*. Pengujian ini dilakukan menggunakan model hasil pelatihan dari *Google Colaboratory* pada masing-masing arsitektur dan variasi dataset yang digunakan. Model hasil pelatihan tersebut kemudian diimpor ke dalam *folder assets* pada file proyek *Android Studio* yang akan dibuat. Setelah diimpor dilanjutkan membuat program untuk aplikasi *Android*-nya pada *Android Studio*. Pengujian tersebut dilakukan terhadap lima varietas beras dengan menggunakan cahaya lampu putih dan lampu kuning, serta kondisi beras yang disebar dengan alas berwarna hitam. Lampu putih dan kuning dipilih berdasarkan lampu yang umum digunakan untuk pencahayaan atau penerangan, baik di dalam maupun di luar ruangan. Lampu kuning memiliki kesan cahaya yang redup dan hangat (Grant-Hays dan Kimberley A, 2003). Selain itu cahaya kuning juga identik dengan kondisi cahaya matahari. Sedangkan lampu putih merupakan lampu yang umum digunakan pada ruangan dan memiliki kesan terang dan dingin. Beberapa foto hasil pengujian tersebut dapat dilihat pada lampiran yang tersedia di penelitian ini.

### 4.3.1 Hasil Pengujian pada Arsitektur *MobileNetV1* dengan Cahaya Putih

Pada pengujian pertama dilakukan menggunakan model TensorFlow Lite arsitektur *MobileNetV1* dengan dataset kualitas yang baik menggunakan cahaya putih dari lampu engsel yang beredar di pasaran. Hasil pengujiannya dapat dilihat pada tabel 4.2 berikut.

Tabel 4.1 Hasil pengujian arsitektur *MobileNetV1* dengan Cahaya Putih

Pengujian ke-	Intensitas Cahaya Lampu	Varietas Beras	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Basmathi	Basmathi	96.52	Benar
2	114279 lux	Basmathi	Basmathi	97.15	Benar
3	114279 lux	Basmathi	Basmathi	97.75	Benar
4	114279 lux	Basmathi	Basmathi	99.16	Benar
5	114279 lux	Basmathi	Basmathi	90.13	Benar
6	114279 lux	Hitam	Hitam	100	Benar
7	114279 lux	Hitam	Hitam	100	Benar
8	114279 lux	Hitam	Hitam	100	Benar
9	114279 lux	Hitam	Hitam	100	Benar
10	114279 lux	Hitam	Hitam	100	Benar
11	114279 lux	IR 64	IR 64	77.82	Benar
12	114279 lux	IR 64	IR 64	90.34	Benar
13	114279 lux	IR 64	Ketan	57.75	Salah
14	114279 lux	IR 64	IR 64	81.12	Benar
15	114279 lux	IR 64	IR 64	89.64	Benar
16	114279 lux	Ketan	Ketan	96.35	Benar
17	114279 lux	Ketan	IR 64	64.58	Salah
18	114279 lux	Ketan	Ketan	93.92	Benar

<b>19</b>	114279 lux	Ketan	Ketan	98	Benar
<b>20</b>	114279 lux	Ketan	Ketan	98.4	Benar
<b>21</b>	114279 lux	Merah	Merah	99.99	Benar
<b>22</b>	114279 lux	Merah	Merah	99.99	Benar
<b>23</b>	114279 lux	Merah	Merah	99.99	Benar
<b>24</b>	114279 lux	Merah	Merah	99.98	Benar
<b>25</b>	114279 lux	Merah	Merah	99.9	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras IR64 dan Ketan hanya terdapat 1 kesalahan, untuk Beras Basmathi, Merah dan Hitam benar semua. Tingkat probabilitas dari kelimanya tidak stabil yang bervariasi antara 57.75% sampai 99.99%. Hal tersebut disebabkan oleh tidak meratanya cahaya yang mengenai beras sehingga terdapat bayangan yang menjadi faktor penurunan tingkat akurasi atau disebabkan dengan kamera dari smartphone yang kurang fokus. Penurunan tingkat akurasi pada Beras IR64 dan Beras Ketan disebabkan karena bayangan atau tingkat focus yang dihasilkan membuat jaringan CNN memprediksi klasifikasinya tidak sesuai yang memang tekstur bentuk lebih mirip antara kedua jenis beras tersebut. Respon deteksi pada pengujian tersebut tergolong cepat antara 1 sampai 2 detik dan bergantung pada proses yang sedang dilakukan oleh prosesor. Setelah diperoleh tabel hasil pengujiannya maka dapat dihitung persentase akurasi dan persentase kesalahan pengujian untuk seluruh varietas berasnya sebagai berikut.

- $akurasi = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah data pengujian}} \times 100\%$

$$akurasi = \frac{23}{25} \times 100\% = 92,00\%$$

- $kesalahan = \frac{\text{Jumlah prediksi salah}}{\text{Jumlah data pengujian}} \times 100\%$

$$kesalahan = \frac{2}{25} \times 100\% = 8,00\%$$

#### 4.3.2 Hasil Pengujian pada Arsitektur *MobileNetV1* dengan Cahaya Kuning

Pada pengujian kedua dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset kualitas yang baik menggunakan cahaya kuning dari lampu engsel yang beredar di pasaran. Hasil pengujiannya dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Hasil pengujian arsitektur *MobileNetV1* dengan Cahaya Kuning

Pengujian ke-	Intensitas Cahaya Lampu	Varietas Beras	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Basmathi	Basmathi	99.99	Benar
2	114279 lux	Basmathi	Basmathi	99.99	Benar
3	114279 lux	Basmathi	Basmathi	99.99	Benar
4	114279 lux	Basmathi	Basmathi	99.99	Benar
5	114279 lux	Basmathi	Basmathi	99.99	Benar
6	114279 lux	Hitam	Hitam	93.6	Benar
7	114279 lux	Hitam	Hitam	81.63	Benar
8	114279 lux	Hitam	Hitam	90.38	Benar
9	114279 lux	Hitam	Hitam	82.48	Benar
10	114279 lux	Hitam	Hitam	81.61	Benar
11	114279 lux	IR 64	IR 64	60.64	Benar
12	114279 lux	IR 64	Ketan	50.39	Salah
13	114279 lux	IR 64	Ketan	53.08	Salah
14	114279 lux	IR 64	IR 64	55.73	Benar
15	114279 lux	IR 64	IR 64	66.38	Benar

<b>16</b>	114279 lux	Ketan	Ketan	87.18	Benar
<b>17</b>	114279 lux	Ketan	Ketan	88.64	Benar
<b>18</b>	114279 lux	Ketan	Ketan	88.19	Benar
<b>19</b>	114279 lux	Ketan	Ketan	85.29	Benar
<b>20</b>	114279 lux	Ketan	Ketan	84.52	Benar
<b>21</b>	114279 lux	Merah	Merah	87.37	Benar
<b>22</b>	114279 lux	Merah	Merah	92.49	Benar
<b>23</b>	114279 lux	Merah	Merah	87.77	Benar
<b>24</b>	114279 lux	Merah	Merah	89.93	Benar
<b>25</b>	114279 lux	Merah	Merah	88.99	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras IR64 hanya terdapat 2 kesalahan, Basmathi, Hitam, Ketan dan Beras Merah benar semua. Tingkat probabilitas dari kelimanya tidak stabil yang bervariasi antara 50.39% sampai 99,99%. Hal tersebut disebabkan oleh tidak meratanya cahaya yang mengenai beras sehingga terdapat bayangan yang menjadi faktor penurunan tingkat akurasi atau disebabkan dengan kamera dari smartphone yang kurang fokus. Penurunan tingkat akurasi pada Beras IR64 yang cukup signifikan disebabkan karena bayangan yang dihasilkan membuat jaringan CNN memprediksi klasifikasinya ke Beras Ketan yang memang tekstur bentuknya hampir sama dengan Beras IR64. Respon deteksi pada pengujian tersebut tergolong cepat antara 1 sampai 2 detik dan bergantung pada proses yang sedang dilakukan oleh prosesor. Setelah diperoleh tabel hasil pengujiannya maka dapat dihitung persentase akurasi dan persentase kesalahan pengujian untuk seluruh varietas berasnya sebagai berikut.

- $akurasi = \frac{Jumlah\ prediksi\ benar}{Jumlah\ data\ pengujian} \times 100\%$

$$akurasi = \frac{23}{25} \times 100\% = 92,00\%$$

- $kesalahan = \frac{Jumlah\ prediksi\ salah}{Jumlah\ data\ pengujian} \times 100\%$

$$kesalahan = \frac{2}{25} \times 100\% = 8,00\%$$

### 4.3.3 Hasil Pengujian pada Arsitektur *MobileNetV1* dengan Cahaya Putih dan Kuning

Pada pengujian ketiga dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset kualitas yang baik menggunakan cahaya campuran putih dan kuning dari lampu engsel yang beredar di pasaran. Hasil pengujiannya dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Hasil pengujian arsitektur *MobileNetV1* dengan Cahaya Putih dan Kuning

Pengujian ke-	Intensitas Cahaya Lampu	Varietas Beras	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Basmathi	Basmathi	99.8	Benar
2	114279 lux	Basmathi	Basmathi	99.05	Benar
3	114279 lux	Basmathi	Basmathi	99.9	Benar
4	114279 lux	Basmathi	Basmathi	99.92	Benar
5	114279 lux	Basmathi	Basmathi	99.16	Benar
6	114279 lux	Hitam	Hitam	99.99	Benar
7	114279 lux	Hitam	Hitam	99.99	Benar
8	114279 lux	Hitam	Hitam	99.99	Benar
9	114279 lux	Hitam	Hitam	99.99	Benar
10	114279 lux	Hitam	Hitam	99.99	Benar
11	114279 lux	IR 64	IR 64	89.43	Benar

<b>12</b>	114279 lux	IR 64	IR 64	71.48	Benar
<b>13</b>	114279 lux	IR 64	IR 64	86.15	Benar
<b>14</b>	114279 lux	IR 64	IR 64	78.73	Benar
<b>15</b>	114279 lux	IR 64	IR 64	80.02	Benar
<b>16</b>	114279 lux	Ketan	IR 64	92.2	Salah
<b>17</b>	114279 lux	Ketan	Ketan	98.56	Benar
<b>18</b>	114279 lux	Ketan	Ketan	99.02	Benar
<b>19</b>	114279 lux	Ketan	Ketan	98.68	Benar
<b>20</b>	114279 lux	Ketan	Ketan	97.94	Benar
<b>21</b>	114279 lux	Merah	Merah	96.48	Benar
<b>22</b>	114279 lux	Merah	Merah	98.47	Benar
<b>23</b>	114279 lux	Merah	Merah	99.35	Benar
<b>24</b>	114279 lux	Merah	Merah	97.9	Benar
<b>25</b>	114279 lux	Merah	Merah	98.22	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras Ketan hanya terdapat 1 kesalahan, Basmathi, Hitam, IR-64 dan Beras Merah benar semua. Tingkat probabilitas dari kelimanya tidak stabil yang bervariasi antara 71,48% sampai 99,9%. Hal tersebut disebabkan oleh tidak meratanya cahaya yang mengenai beras sehingga terdapat bayangan yang menjadi faktor penurunan tingkat akurasi dibanding dengan cahaya flashlight. Penurunan tingkat akurasi pada Beras Ketan yang cukup signifikan disebabkan karena bayangan yang dihasilkan membuat jaringan CNN memprediksi klasifikasinya ke Beras IR- 64 yang memang tekstur warnanya lebih gelap daripada Beras Ketan. Respon deteksi pada pengujian tersebut tergolong cepat antara 1 sampai 2 detik dan bergantung pada proses yang

sedang dilakukan oleh prosesor. Setelah diperoleh tabel hasil pengujiannya maka dapat dihitung persentase akurasi dan persentase kesalahan pengujian untuk seluruh varietas berasnya sebagai berikut.

- $akurasi = \frac{Jumlah\ prediksi\ benar}{Jumlah\ data\ pengujian} \times 100\%$

$$akurasi = \frac{24}{25} \times 100\% = 96,00\%$$

- $kesalahan = \frac{Jumlah\ prediksi\ salah}{Jumlah\ data\ pengujian} \times 100\%$

$$kesalahan = \frac{1}{25} \times 100\% = 4,00\%$$

#### 4.3.4 Hasil Pengujian pada Arsitektur *MobileNetV1* per Varietas Beras

Pada sub bab ini akan menjelaskan hasil dari masing-masing beras pada tiga kondisi cahaya, yaitu cahaya putih, cahaya kuning, dan campuran cahaya putih dan kuning yang akan disajikan dalam bentuk tabel.

##### 1. Beras Basmathi

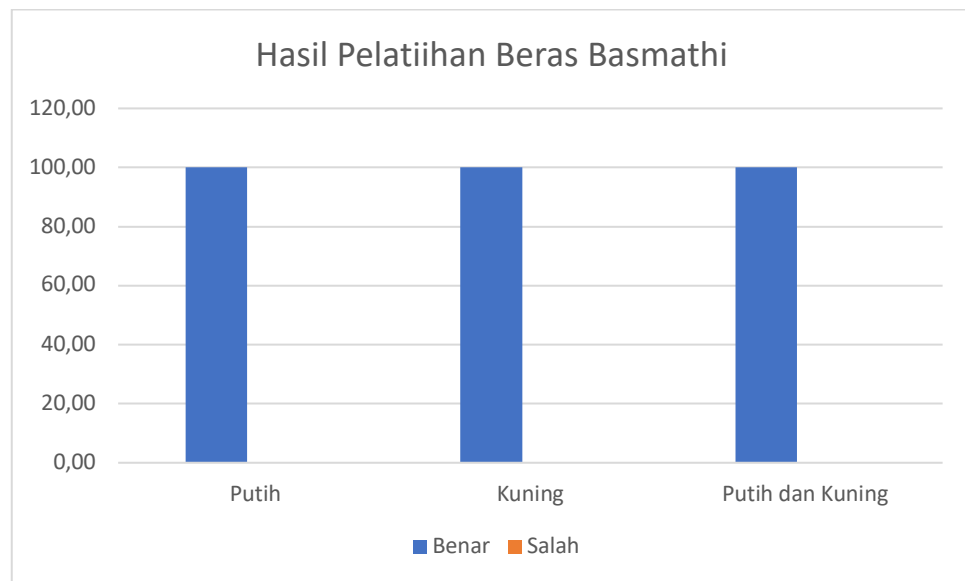
Pada pengujian pertama dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset untuk varietas Beras Basmathi. Hasil pengujiannya dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Hasil pengujian arsitektur *MobileNetV1* dengan Beras Basmathi

Pengujian ke-	Intensitas Cahaya Lampu	Cahaya Lampu	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Putih	Basmathi	96.52	Benar
2	114279 lux	Putih	Basmathi	97.15	Benar
3	114279 lux	Putih	Basmathi	97.75	Benar
4	114279 lux	Putih	Basmathi	99.16	Benar
5	114279 lux	Putih	Basmathi	90.13	Benar

<b>6</b>	114279 lux	Kuning	Basmathi	99.99	Benar
<b>7</b>	114279 lux	Kuning	Basmathi	99.99	Benar
<b>8</b>	114279 lux	Kuning	Basmathi	99.99	Benar
<b>9</b>	114279 lux	Kuning	Basmathi	99.99	Benar
<b>10</b>	114279 lux	Kuning	Basmathi	99.99	Benar
<b>11</b>	114279 lux	Putih & Kuning	Basmathi	99.8	Benar
<b>12</b>	114279 lux	Putih & Kuning	Basmathi	99.05	Benar
<b>13</b>	114279 lux	Putih & Kuning	Basmathi	99.9	Benar
<b>14</b>	114279 lux	Putih & Kuning	Basmathi	99.92	Benar
<b>15</b>	114279 lux	Putih & Kuning	Basmathi	99.16	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras Basmathi benar semua. Tingkat probabilitas dari semua kondisi cahaya hampir stabil antara 90% sampai 99,9%. Hal ini dapat disimpulkan bahwa tingkat akurasi kemiripan untuk beras basmathi mencapai 100% karena tidak ada kesalahan prediksi pada ketiga jenis cahaya lampu, meskipun tingkat probabilitasnya kurang dari 100%. Kurva tingkat akurasi hasil pelatihan dari Beras Basmathi dapat dilihat pada gambar 4.6 sebagai berikut.



Gambar 4.6 Hasil Pelatihan Beras Basmathi

## 2. Beras Hitam

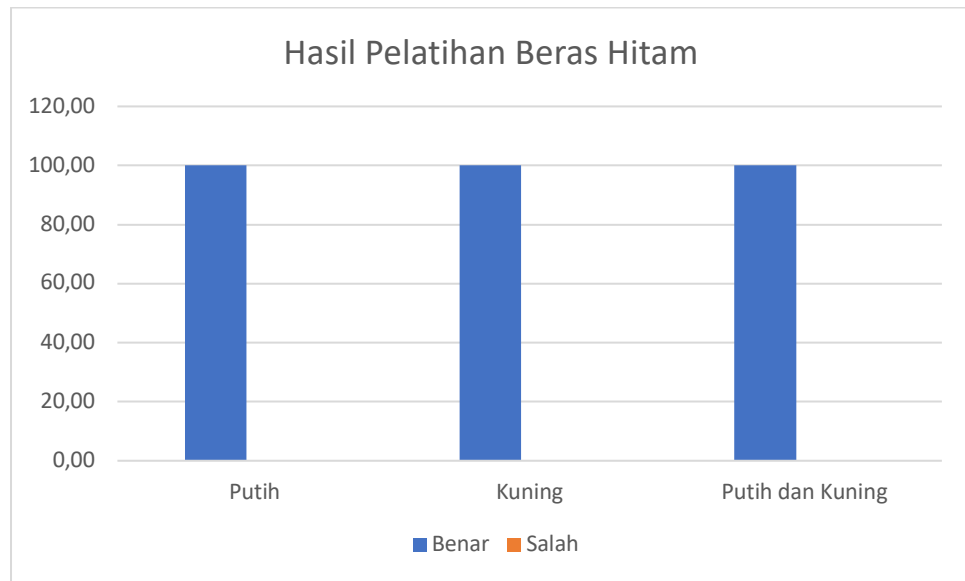
Pengujian dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset untuk varietas Beras Hitam. Hasil pengujiannya dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Hasil pengujian arsitektur *MobileNetV1* dengan Beras Hitam

Pengujian ke-	Intensitas Cahaya Lampu	Cahaya Lampu	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Putih	Hitam	100	Benar
2	114279 lux	Putih	Hitam	100	Benar
3	114279 lux	Putih	Hitam	100	Benar
4	114279 lux	Putih	Hitam	100	Benar
5	114279 lux	Putih	Hitam	100	Benar
6	114279 lux	Kuning	Hitam	93.6	Benar
7	114279 lux	Kuning	Hitam	81.63	Benar
8	114279 lux	Kuning	Hitam	90.38	Benar
9	114279 lux	Kuning	Hitam	82.48	Benar

<b>10</b>	114279 lux	Kuning	Hitam	81.61	Benar
<b>11</b>	114279 lux	Putih & Kuning	Hitam	99.99	Benar
<b>12</b>	114279 lux	Putih & Kuning	Hitam	99.99	Benar
<b>13</b>	114279 lux	Putih & Kuning	Hitam	99.99	Benar
<b>14</b>	114279 lux	Putih & Kuning	Hitam	99.99	Benar
<b>15</b>	114279 lux	Putih & Kuning	Hitam	99.99	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras Hitam benar semua. Tingkat probabilitas dari masing-masing kondisi cahaya, cahaya putih dan campuran cahaya putih dan kuning sangat stabil yaitu 100% dan 90%, cahaya kuning antara 80-94. Hal ini dapat disimpulkan bahwa tingkat akurasi kemiripan untuk beras hitam pada kondisi cahaya putih dan campuran cahaya putih dan kuning sangat stabil, sedangkan pada cahaya kuning tingkat probabilitasnya menurun. Hal ini kemungkinan disebabkan karena kondisi cahaya kuning yang cenderung lebih redup dari dua kondisi cahaya yang lain, selain itu alas yang berwarna hitam yang memiliki warna sama dengan beras, sehingga membuat beras hitam menjadi lebih sulit untuk diidentifikasi oleh sistem. Kurva tingkat akurasi hasil pelatihan dari Beras Hitam dapat dilihat pada gambar 4.7 sebagai berikut.



Gambar 4.7 Hasil Pelatihan Beras Hitam

### 3. Beras IR 64

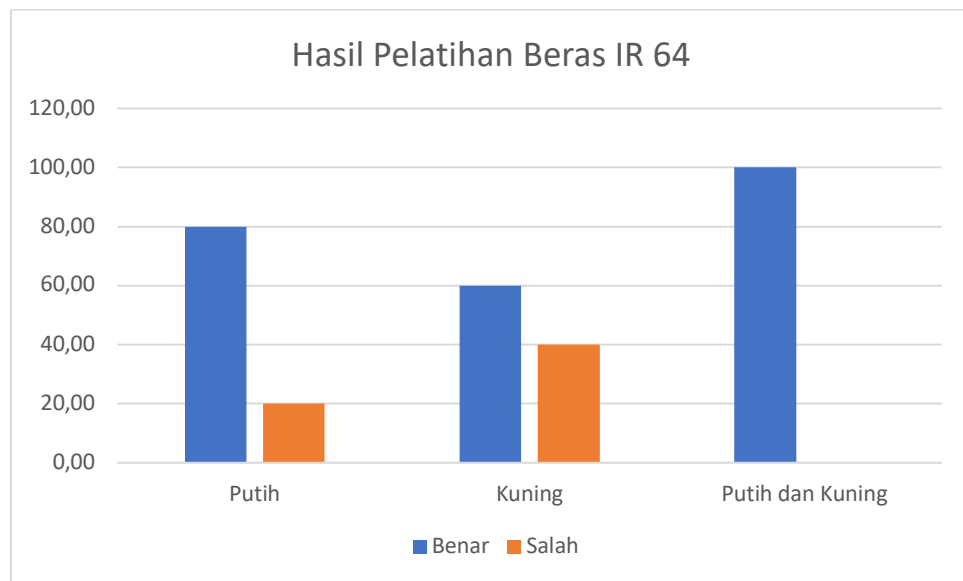
Pengujian dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset untuk varietas Beras IR 64. Hasil pengujiannya dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6 Hasil pengujian arsitektur *MobileNetV1* dengan Beras IR 64

Pengujian ke-	Intensitas Cahaya Lampu	Cahaya Lampu	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Putih	IR 64	77.82	Benar
2	114279 lux	Putih	IR 64	90.34	Benar
3	114279 lux	Putih	Ketan	57.75	Salah
4	114279 lux	Putih	IR 64	81.12	Benar
5	114279 lux	Putih	IR 64	89.64	Benar
6	114279 lux	Kuning	IR 64	60.64	Benar
7	114279 lux	Kuning	Ketan	50.39	Salah
8	114279 lux	Kuning	Ketan	53.08	Salah
9	114279 lux	Kuning	IR 64	55.73	Benar

<b>10</b>	114279 lux	Kuning	IR 64	66.38	Benar
<b>11</b>	114279 lux	Putih & Kuning	IR 64	89.43	Benar
<b>12</b>	114279 lux	Putih & Kuning	IR 64	71.48	Benar
<b>13</b>	114279 lux	Putih & Kuning	IR 64	86.15	Benar
<b>14</b>	114279 lux	Putih & Kuning	IR 64	78.73	Benar
<b>15</b>	114279 lux	Putih & Kuning	IR 64	80.02	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras IR 64 memiliki 3 kali pengujian yang salah prediksi, yaitu 1 pengujian pada lampu putih dan 2 pengujian pada lampu kuning. Tingkat probabilitas dari masing-masing kondisi cahaya, cahaya putih antara 57-90%, cahaya kuning 50-66%, serta campuran cahaya putih dan kuning antara 71-89%. Hal ini dapat disimpulkan bahwa tingkat akurasi kemiripan untuk Beras IR 64 pada kondisi cahaya putih dan cahaya kuning kurang stabil, sedangkan pada campuran cahaya putih dan kuning tingkat probabilitasnya naik. Beberapa kesalahan pada pengujian kemungkinan disebabkan karena secara tampilan Beras IR 64 hampir memiliki kesamaan dari segi tekstur dan warna dengan Beras Ketan, sehingga ada beberapa pengujian yang hasil prediksinya adalah Beras Ketan. Kurva tingkat akurasi hasil pelatihan dari Beras IR 64 dapat dilihat pada gambar 4.8 sebagai berikut.



Gambar 4.8 Hasil Pelatihan Beras IR 64

#### 4. Beras Ketan

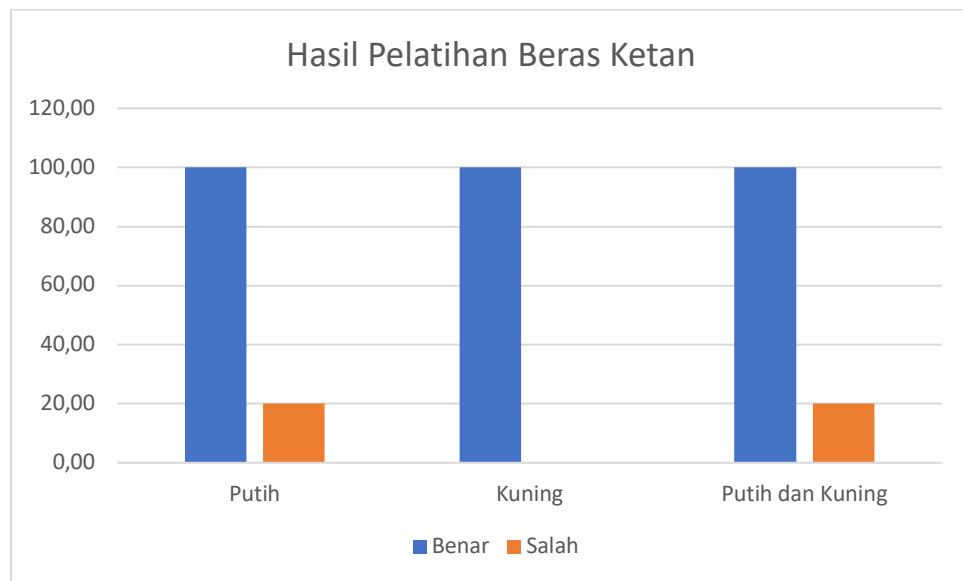
Pengujian dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset untuk varietas Beras Ketan. Hasil pengujiannya dapat dilihat pada tabel 4.7 berikut.

Tabel 4.7 Hasil pengujian arsitektur *MobileNetV1* dengan Beras Ketan

Pengujian ke-	Intensitas Cahaya Lampu	Cahaya Lampu	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Putih	Ketan	96.35	Benar
2	114279 lux	Putih	IR 64	64.58	Salah
3	114279 lux	Putih	Ketan	93.92	Benar
4	114279 lux	Putih	Ketan	98	Benar
5	114279 lux	Putih	Ketan	98.4	Benar
6	114279 lux	Kuning	Ketan	87.18	Benar
7	114279 lux	Kuning	Ketan	88.64	Benar
8	114279 lux	Kuning	Ketan	88.19	Benar
9	114279 lux	Kuning	Ketan	85.29	Benar

<b>10</b>	114279 lux	Kuning	Ketan	84.52	Benar
<b>11</b>	114279 lux	Putih & Kuning	IR 64	92.2	Salah
<b>12</b>	114279 lux	Putih & Kuning	Ketan	98.56	Benar
<b>13</b>	114279 lux	Putih & Kuning	Ketan	99.02	Benar
<b>14</b>	114279 lux	Putih & Kuning	Ketan	98.68	Benar
<b>15</b>	114279 lux	Putih & Kuning	Ketan	97.94	Benar

Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras Ketan memiliki 2 kali pengujian yang salah prediksi, yaitu 1 pengujian pada lampu putih dan 1 pengujian pada campuran lampu putih dan kuning. Tingkat probabilitas dari masing-masing kondisi cahaya, cahaya putih antara 64-98%, cahaya kuning 84-88%, serta campuran cahaya putih dan kuning antara 92-99%. Hal ini dapat disimpulkan bahwa tingkat akurasi kemiripan untuk beras ketan pada kondisi cahaya putih dan campuran cahaya putih dan kuning kurang stabil, sedangkan pada cahaya kuning tingkat probabilitasnya cukup stabil di angka 80 an persen. Beberapa kesalahan pada pengujian kemungkinan disebabkan karena Beras Ketan memiliki tekstur dan warna yang hampir sama dengan Beras IR 64, sama halnya pada pengujian Beras IR 64 yang beberapa hasil prediksinya menghasilkan hasil prediksi Beras Ketan. Kurva tingkat akurasi hasil pelatihan dari Beras Ketan dapat dilihat pada gambar 4.9 sebagai berikut.



Gambar 4.9 Hasil Pelatihan Beras Ketan

## 5. Beras Merah

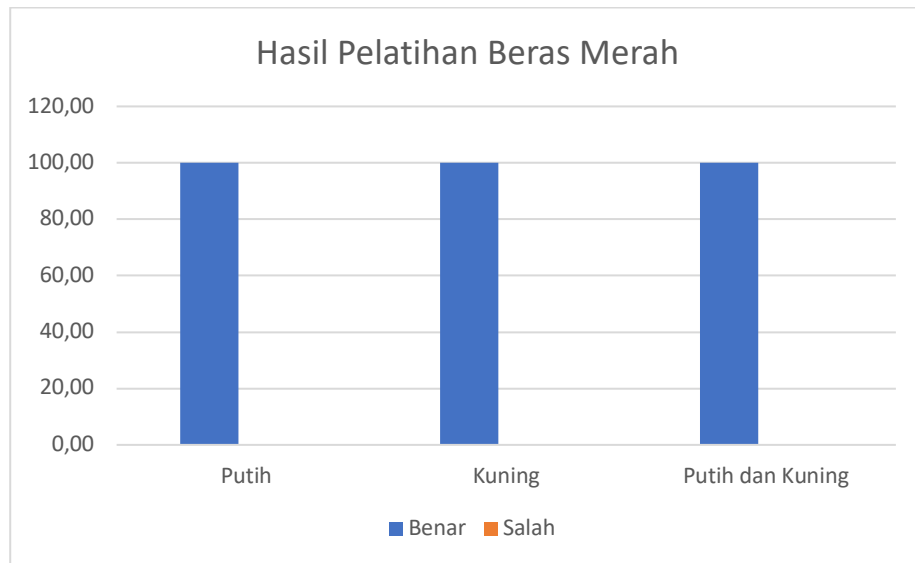
Pengujian dilakukan menggunakan model *TensorFlow Lite* arsitektur *MobileNetV1* dengan dataset untuk varietas Beras Merah. Hasil pengujiannya dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Hasil pengujian arsitektur *MobileNetV1* dengan Beras Merah

Pengujian ke-	Intensitas Cahaya Lampu	Cahaya Lampu	Hasil Prediksi	Tingkat Probabilitas (%)	Benar/Salah
1	114279 lux	Putih	Merah	99.99	Benar
2	114279 lux	Putih	Merah	99.99	Benar
3	114279 lux	Putih	Merah	99.99	Benar
4	114279 lux	Putih	Merah	99.98	Benar
5	114279 lux	Putih	Merah	99.9	Benar
6	114279 lux	Kuning	Merah	87.37	Benar
7	114279 lux	Kuning	Merah	92.49	Benar
8	114279 lux	Kuning	Merah	87.77	Benar
9	114279 lux	Kuning	Merah	89.93	Benar

<b>10</b>	114279 lux	Kuning	Merah	88.99	Benar
<b>11</b>	114279 lux	Putih & Kuning	Merah	96.48	Benar
<b>12</b>	114279 lux	Putih & Kuning	Merah	98.47	Benar
<b>13</b>	114279 lux	Putih & Kuning	Merah	99.35	Benar
<b>14</b>	114279 lux	Putih & Kuning	Merah	97.9	Benar
<b>15</b>	114279 lux	Putih & Kuning	Merah	98.22	Benar

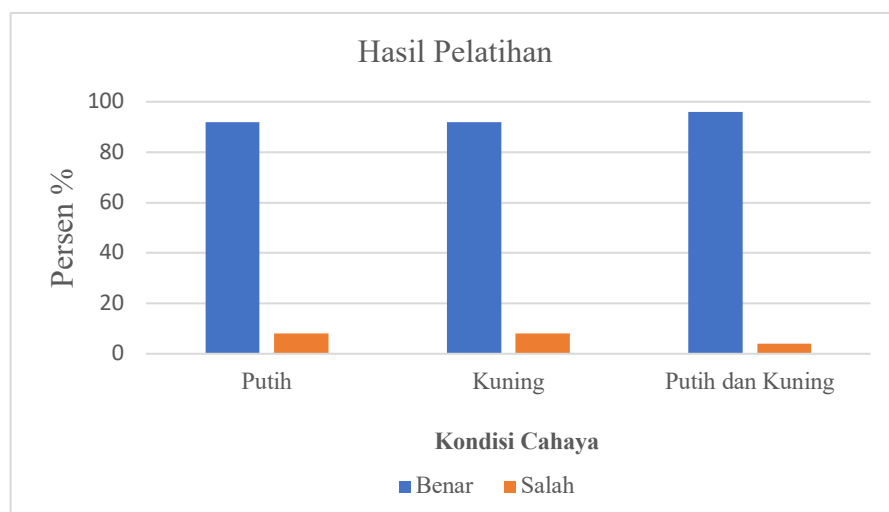
Dari tabel pengujian tersebut dapat dilihat bahwa hasil prediksi untuk Beras Merah benar semua. Tingkat probabilitas dari semua kondisi cahaya cukup stabil yaitu 87-99%. Hal ini dapat disimpulkan bahwa tingkat akurasi kemiripan untuk Beras Merah pada semua kondisi cahaya cukup stabil. Hal ini kemungkinan disebabkan karena kondisi cahaya dan tingkat fokus pada objek yang stabil sehingga hasil prediksi benar semua dan tingkat probabilitasnya cukup stabil. Kurva tingkat akurasi hasil pelatihan dari Beras Merah dapat dilihat pada gambar 4.10 sebagai berikut.



Gambar 4.10 Hasil Pelatihan Beras Merah

#### 4.3.5 Perbandingan Hasil Pelatihan Semua Varietas Beras pada Semua Kondisi Cahaya

Untuk mengetahui perbedaan tingkat akurasi pelatihan serta tingkat kesalahan pelatihan dari hasil pelatihan yang dilakukan maka dibuatlah grafik perbandingan tingkat akurasi dan tingkat kesalahan dari setiap hasil pelatihannya. Kurva tingkat akurasi hasil pelatihannya dapat dilihat pada gambar 4.11 sebagai berikut.



Gambar 4.11 Kurva Hasil Pelatihan

Berdasarkan kurva tersebut dapat dilihat bahwa seluruh tingkat akurasi pelatihan hampir mencapai nilai maksimumnya sebesar 100. Tingkat akurasi pelatihan yang cukup tinggi dan baik dihasilkan oleh arsitektur *MobileNetV1*, namun sangat rendah pada tingkat kesalahan prediksi. Semakin tinggi tingkat akurasi dan semakin sedikit tingkat kesalahan, maka arsitektur CNN dapat melakukan klasifikasi dengan baik dan dapat disimpulkan hasil pelatihan tersebut dalam kondisi goodfit.

Selain kurva, perbandingan rata-rata akurasi benar dan salah dapat dilihat dalam tabel 4.9 dari masing-masing varietas pada masing-masing cahaya.

Tabel 4.9 Rata-rata Akurasi pada Masing-masing Varietas Beras

No.	Intensitas Cahaya Lampu	Cahaya Lampu	Varietas Beras	Akurasi Benar (%)	Akurasi Salah (%)
1	114279 lux	Putih	Basmathi	96.14	0
2	114279 lux	Kuning	Basmathi	99.99	0
3	114279 lux	Putih & Kuning	Basmathi	99.57	0
4	114279 lux	Putih	Hitam	100	0
5	114279 lux	Kuning	Hitam	85.94	0
6	114279 lux	Putih & Kuning	Hitam	99.99	0
7	114279 lux	Putih	IR 64	84.73	57.75
8	114279 lux	Kuning	IR 64	60.92	51.74
9	114279 lux	Putih & Kuning	IR 64	81.16	0
10	114279 lux	Putih	Ketan	96.67	64.58
11	114279 lux	Kuning	Ketan	86.76	0
12	114279 lux	Putih & Kuning	Ketan	98.55	92.2
13	114279 lux	Putih	Merah	99.97	0

<b>14</b>	114279 lux	Kuning	Merah	89.31	0
<b>15</b>	114279 lux	Putih & Kuning	Merah	98.08	0

Sedangkan tabel 4.10 menampilkan persentase akurasi benar dan salah yang dilakukan pada lima kali pengujian pada masing-masing beras dan masing-masing cahaya.

Tabel 4.10 Persentase Akurasi Benar dan Salah pada Masing-masing Varietas Beras

No.	Intensitas Cahaya Lampu	Cahaya Lampu	Varietas Beras	Akurasi Benar (%)	Akurasi Salah (%)
<b>1</b>	114279 lux	Putih	Basmathi	100	0
<b>2</b>	114279 lux	Kuning	Basmathi	100	0
<b>3</b>	114279 lux	Putih & Kuning	Basmathi	100	0
<b>4</b>	114279 lux	Putih	Hitam	100	0
<b>5</b>	114279 lux	Kuning	Hitam	100	0
<b>6</b>	114279 lux	Putih & Kuning	Hitam	100	0
<b>7</b>	114279 lux	Putih	IR 64	80	20
<b>8</b>	114279 lux	Kuning	IR 64	60	40
<b>9</b>	114279 lux	Putih & Kuning	IR 64	100	0
<b>10</b>	114279 lux	Putih	Ketan	80	20
<b>11</b>	114279 lux	Kuning	Ketan	100	0
<b>12</b>	114279 lux	Putih & Kuning	Ketan	80	20
<b>13</b>	114279 lux	Putih	Merah	100	0
<b>14</b>	114279 lux	Kuning	Merah	100	0
<b>15</b>	114279 lux	Putih & Kuning	Merah	100	0

#### 4.4 Pembahasan dan Integrasi Islam

Klasifikasi merupakan suatu kegiatan mengelompokkan data secara sistematis berdasarkan indikasi tertentu dan sesuai dengan kaidah yang ditetapkan atau bisa dikatakan bahwa klasifikasi merupakan pembagian sesuatu menurut kelas-kelas berdasarkan ciri persamaan atau perbedaan. Objek yang digunakan untuk klasifikasi pada penelitian ini adalah beras.

Dalam Al-Quran, Allah menyebutkan beberapa ayat tentang pertanian yang menjelaskan tentang tumbuhan, biji-bijian, dan buah-buahan di bumi. Sebagaimana ayat berikut ini.

إِنَّ اللَّهَ فَالِكُ الْحَبِّ وَالنَّوَى ۖ يُخْرِجُ الْحَيَّ مِنَ الْمَيِّتِ وَمُخْرِجُ الْمَيِّتِ مِنَ الْحَيِّ ۚ ذَٰلِكُمْ اللَّهُ فَالِي تُوْفِكُونَ (الانعام

: ٩٥)

*“Sesungguhnya Allah menumbuhkan butir tumbuh-tumbuhan dan biji buah-buahan. Dia mengeluarkan yang hidup dari yang mati dan mengeluarkan yang mati dari yang hidup. (Yang memiliki sifat-sifat) demikian ialah Allah, maka mengapa kamu masih berpaling?.”* (Q.S. Al-An’am (6) : 95).

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً ۖ فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا نُخْرِجُ مِنْهُ حَبًّا مُتَرَاكِبًا ۖ وَمِنَ النَّخْلِ مِنْ طَلْعِهَا قِنْوَانٌ دَانِيَةٌ وَجَنَّاتٍ مِنْ أَعْنَابٍ وَالزَّيْتُونَ وَالرُّمَّانَ مُشْتَبِهًا وَغَيْرَ مُتَشَابِهٍ ۗ انظُرُوا إِلَى ثَمَرِهِ إِذَا أَثْمَرَ وَيَنْعِهِ ۗ إِنَّ فِي ذَٰلِكُمْ لَآيَاتٍ لِّقَوْمٍ يُؤْمِنُونَ (الانعام : ٩٩)

*“Dan Dialah yang menurunkan air dari langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan, maka kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau itu butir yang banyak; dan dari mayang kurma, mengurai tangkai-tangkai yang menjulai, dan kebun-kebun anggur, dan (Kami keluarkan pula) zaitun dan delima yang serupa dan yang tidak serupa, perhatikanlah buahnya pada waktu berbuah, dan menjadi masak. Sungguh, pada yang demikian itu ada tanda-tanda (kekuasaan Allah) bagi orang-orang yang beriman.”* (Q.S. Al-An’am (6) : 99).

وَفِي الْأَرْضِ قِطْعٌ مُتَجَوِّرَاتٌ وَجَنَّتْ مِنْ أَعْنَابٍ وَزَّرَعٌ وَنَخِيلٌ صِنَوَانٌ وَعَيْرُ صِنَوَانٍ يُسْقَى بِمَاءٍ وَاحِدٍ وَتَفْضُلٌ  
بَعْضَهَا عَلَى بَعْضٍ فِي الْأَكْلِ، إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّقَوْمٍ يَعْقِلُونَ (الرعد : ٤)

*“Dan di bumi terdapat bagian-bagian yang berdampingan, kebun-kebun anggur, tanaman-tanaman, pohon kurma yang bercabang, dan yang tidak bercabang; disirami dengan air yang sama, tetapi Kami lebihkan tanaman yang satu dari yang lainnya dalam hal rasa. Sungguh, pada yang demikian itu terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang mengerti.” (Q.S. Ar-Ra’d (13) : 4).*

Tiga ayat tersebut merupakan ayat yang memiliki keterkaitan dengan bahan pangan atau hasil bumi. Ayat tersebut dirasa memiliki isi tentang makanan di bumi yang bermacam-macam jenisnya, yang salah satunya adalah padi atau beras. Selain itu ayat tersebut juga berisi tentang Allah telah menumbuhkan macam-macam butir tumbuhan dan itu merupakan salah satu dari tanda-tanda kekuasaan Allah. Salah satu jenis butir tumbuhan yang bermacam-macam itu adalah beras.

” يُسْقَى بِمَاءٍ وَاحِدٍ وَتَفْضُلٌ بَعْضَهَا عَلَى بَعْضٍ فِي الْأَكْلِ ”

Ibnu Katsir, salah satu mufassir menafsirkan kalimat di atas “disirami dengan air yang sama. Kami lebihkan sebagian tanaman-tanaman itu atas sebagian yang lain tentang rasanya.” Maksudnya, perbedaan dalam jenis buahbuahan dan tanaman itu dari segi bentuk, warna, rasa, daun, bunganya, ada yang manis, asam, pahit, dan macam lainnya, atas izin Allah. Dan inilah tanda-tanda kebesaran Allah. (Katsir, 2013)

Sedangkan dalam dalam Kitab Al-Lubab, M. Quraish Shihab sedikit memberi penjelasan mengenai Surah Ar-Ra’du [13] ayat 4, bahwa Allah menginformasikan kepada manusia tentang bumi yang di dalamnya terdapat kepingan-kepingan tanah yang saling berdampingan, dengan kualitas yang berbeda. Ada yang tandus, subur, serta ditumbuhi oleh tumbuhan yang berbeda. Ada yang

ditanami kebun-kebun anggur, tanaman persawahan, ada juga yang perkebunan pohon-pohon kurma, dan lain sebagainya. Dan Allah melebihkan itu atas sebagian yang lain, baik dalam segi besar dan kecilnya, warna, dan bentuknya, serta perbedaan lainnya. (Shihab, 2012)

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka diperoleh kesimpulannya sebagai berikut.

1. Perancangan arsitektur CNN *MobileNetV1* untuk klasifikasi varietas beras dilakukan dengan metode *Feature Extraction* karena metode tersebut cocok digunakan pada jumlah *dataset* yang sedikit dengan hasil yang cukup baik.
2. Pengujian klasifikasi varietas beras pada perangkat *Android* dilakukan dengan cara mengambil obyek beras yang akan dideteksi menggunakan fitur kamera dengan keadaan beras yang disebar di alas berwarna hitam dengan kondisi cahaya putih, kuning, dan campuran cahaya putih dan kuning.
3. Faktor yang mempengaruhi tingkat akurasi dan tingkat kesalahan dari hasil pengujian pada CNN adalah jarak obyek yang akan diklasifikasi, kualitas gambar obyek, serta kondisi pencahayaan.
4. CNN dapat melakukan klasifikasi obyek gambar dengan baik jika obyek yang diklasifikasi sangat mirip dengan dataset yang sudah dilatih.
5. Hasil pengujian arsitektur *MobileNetV1* pada perangkat *Android* untuk semua kondisi menghasilkan nilai akurasi rata-rata sebesar 93,33%. sedangkan hasil pengujian pada perangkat *Android* untuk masing-masing

beras dalam 5 kali pengujian pada 3 kondisi cahaya yang berbeda adalah Beras Basmathi, Hitam dan Merah mencapai akurasi 100% pada semua kondisi cahaya. Beras IR 64 mencapai akurasi 80% untuk cahaya putih, 60% untuk cahaya kuning, dan 100% untuk campuran cahaya putih dan kuning. Beras Ketan mencapai akurasi 80% untuk cahaya putih, 100% untuk cahaya kuning, dan 80% untuk campuran cahaya putih dan kuning. Model hasil pelatihan arsitektur *MobileNetV1* dapat dikonversi ke *TensorFlow Lite* dengan baik sehingga hasil klasifikasi pada aplikasi *Android* memiliki tingkat akurasi yang tinggi.

6. Waktu respon deteksi arsitektur *MobileNetV1* termasuk cepat yaitu sekitar 1000 ms di perangkat *Android* yang digunakan.

## 5.2 Saran

Saran yang perlu dikembangkan untuk tugas akhir ini agar lebih baik sebagai berikut.

1. Dataset pelatihan ditambahkan lagi jumlah dan variasi kondisi berasnya agar hasil prediksinya semakin meningkat.
2. Dataset pengujian perlu ditambahkan lagi agar tingkat akurasi hasil pengujian tiap varietas berasnya tidak terlalu berbeda jauh.
3. Perlu adanya penambahan varietas beras yang lebih beragam untuk dilakukan klasifikasi menggunakan CNN.
4. Proses pendeteksian varietas beras pada aplikasi *Android* dikembangkan lagi agar dapat mendeteksi secara realtime sehingga pengguna tidak perlu menekan tombol “Deteksi” pada aplikasi tersebut.

## DAFTAR PUSTAKA

- Aji, Deddy Kusbianto Purwoko. Agung Nugroho Pramudhita., dan Nurhalimah. (2020). Perbandingan Metode FTS dan MA pada Peramalan Persediaan Beras di Kabupaten Malang. Politeknik Negeri Malang.
- Anonim, "CS231n *Convolutional Neural Networks for Visual Recognition*". [Daring]. <https://cs231n.github.io/convolutional-networks/>. [Diakses: 14-Dec-20].
- Arrofiqoh, Erlyna Nour dan Harintaka. (2018). Implementasi Metode *Convolutional Neural Network* untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi. *Jurnal Geomatika*. Vol : 24, No : 2. 61-68. <http://dx.doi.org/10.24895/JIG.2018.24-2.810>
- As Shidiq Cater Indonesia. Kandungan-kandungan Beras Basmati. [Daring]. <https://www.asshidiqaqiqah.com/kandungan-kandungan-beras-basmati/>. [Diakses: 25-Mei-2021].
- Badan Standardisasi Nasional. (2008). Jakarta.
- Chollet, Francois. (2018). *Deep Learning with Python*. Shelter Island: Manning Publication Co.
- Dadang, W. (2018). Memahami Kecerdasan Buatan Berupa *Deep Learning* dan *Machine Learning*. Retrieved January 4, 2019, from <https://warstek.com/2018/02/06/deepmachinelearning/>
- Deng, L & Yu, D. (2013). *Deep Learning: Methods and Applications*. *Foundations and trends® in Signal Processing*, 7(3-4), 197-387. <https://doi.org/10.1136/bmj.319.7209.0a>
- E. Z., Parveen. (2017). *Assessment of Quality of Rice Grain using Optical and Image Processing Technique*. 265-270.
- Effendi, Mas'ud. Ullivia Fatasya., dan Usman Effendi. (2017). Identifikasi Jenis dan Mutu Kopi menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan. *Jurnal Ilmiah Teknologi Pertanian AGROTECHNO*. Vol. 2, No. 1. 140-146.
- Ekoputris, Rizqi Okta. "MobileNet: Deteksi Objek pada Platform Mobile". [Daring]. Tersedia pada: <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>. [Diakses: 28-Nov-2019].
- Fadhilah, Rifayani. (2019). Perancangan Program Klasifikasi Kopi Beras Menggunakan Pengolahan Citra Digital. Jatinangor: Universitas Padjajaran.

- Fardhani, Ayu Artika. Desi Insani Natalia Simanjuntak., dan Anjar Wanto. (2018). Prediksi Harga Beras di Pasar Tradisional di 33 Kota di Indonesia Menggunakan Algoritma *Backpropagation*. *Jurnal Infomedia* Vol.3 No. 1. 25-30.
- Google. “*Android Studio User Guide*”. [Daring]. Tersedia pada: <https://developer.android.com/studio/intro>. [Diakses: 3-Mar-2021].
- Google Inc. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv:1704.04861v1 [cs.CV] 17 Apr 2017.
- Grant-Hays, Brenda dan Kimberley A. Mikula. (2003). *Color in Small Spaces*. New York. McGraw Hill.
- Heaton, J. (2015). *Artificial Intelligence for Humans: Deep learning and neural networks of Artificial Intelligence for Humans Series*. Createspace Independet Publishing Platform.
- Janah, Anisa Furtakhul, Upik Nurbaiti., dan Fianti. (2020). Pengaruh Warna dan Bentuk Lampu terhadap Tingkat Kenyamanan Manusia di Ruang Tidur. *EnviroScienceae*. Vol : 16, No : 1. 7-11.
- Katsir, Ibnu. (2013). *Tafsir Ibnu Katsir*. Pustaka Imam Asy-Syafi’i, Jakarta. 6-7.
- Kim, J., Sangjun, O., Kim, Y., & Lee, M. (2016). *Convolutional Neural Network with Biologically Inspired Retinal Structure*. *Procedia Computer Science*, 88, 145-154. <https://doi.org/10.1016/j.procs.2016.07.418>
- Lorentius, C.A dkk. (2019). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network. *Jurnal Infra* Vol.7, No.1.
- Mueller, John Paul dan Luca Massaron. “*What is Google Colaboratory*”. [Daring]. Tersedia pada: <https://www.dummies.com/programming/python/what-is-google-colaboratory/>. [Diakses: 29-Jan-2021].
- Mutmainnah, dkk. (2020). Karakteristik Listrik dan Optik pada *LED* dan *Laser*. *Jurnal Teori dan Aplikasi Fisika*. Vol : 08, No : 02. 203-208.
- Neurohive. “*VGG16 – Convolutional Network for Classification and Detection*.” [Daring]. Tersedia pada: <https://neurohive.io/en/popular-networks/vgg16/>. [Diakses: 28-Nov-2019].
- Paoletti, M.E., Haut, J.M., Plaza J., & Plaza A. (2017). “*A new deep convolutional neural network for fast hyperspectral image classification*”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 16.
- Priyanto, T. (2012). *Beras Ketan & Sifat Fisika-Kimianya*. [Daring]. <http://www.alatcetakrengginang.com/2012/02/beras-ketan-sifat-fisika-kimianya.html>. [Diakses: 25-Mei-2021].

- Sada, H.J. (2017). *Kebutuhan Dasar Manusia Dalam Perspektif Pendidikan Islam*. Al-Tadzkiyyah : Jurnal Pendidikan Islam. Vol : 8, No : II. 213-226.
- Shihab, M. Quraish. (2012). *Al-Lubab; Makna, Tujuan, dan Pelajaran dari Surah-Surah Al-Qur'an*. Lentera Hati, Tangerang. 59-60.
- Suartika, I Wayan E. P dkk. (2016). *Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101*. Jurnal Teknik ITS Vol.5, No.1 : 2016.
- Suprihatno, B, Daradjat dkk. (2010). *Deskripsi Varietas Padi*. Balai Besar Penelitian Tanaman Padi.
- Tandung, Sofyan. (2019). "*Pengenalan Convolutional Neural Network – Part 1*". [Daring]. Tersedia pada: <http://sofyantandung.com/pengenalan-convolutional-neural-network-part-1/>. [Diakses: 25-Jan-21].
- TensorFlow. "*Using TensorFlow Lite on Android*". [Daring]. Tersedia pada: <https://medium.com/tensorflow/using-tensorflow-lite-on-android-9bbc9cb7d69d>. [Diakses: 2-Feb-2021].
- Trisnawan, Atriyana. Wahyudi Harianto., dan Syahminan. (2019). *Klasifikasi Beras Menggunakan Metode K-Means Clustering Berbasis Pengolahan Citra Digital*. Jurnal Terapan Sains & Teknologi (RAINSTEK). Vol : 1, No : 1. 16-24.
- Webstaurant Store. "*Types of Rice*". [Daring]. Tersedia pada: <https://www.webstaurantstore.com/guide/658/types-of-rice.html>. [Diakses: 1-Feb-2021].
- Sugiarto, William dkk. (2017). *Estimasi Arah Tatapan Mata dengan Menggunakan Average Pooling Convolutional Neural Network*. Dinamika Teknologi Vol. 9, No.2 : 2017.