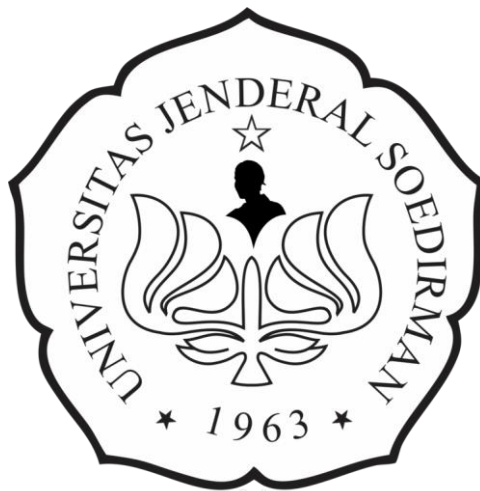


**PROPOSAL PENELITIAN**

**KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL**  
**MENGUNAKAN METODE**  
***CONVOLUTIONAL NEURAL NETWORK (CNN)* BERBASISI ANDROID**



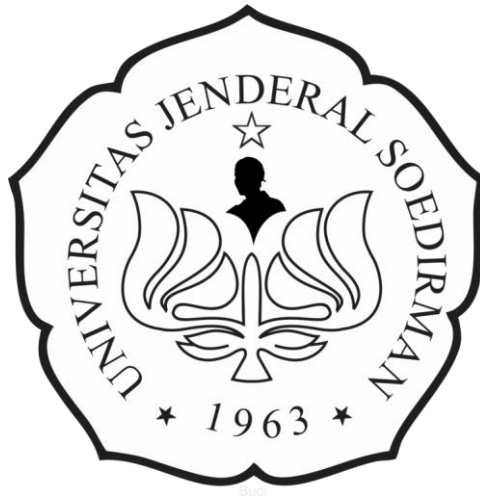
Diajukan Sebagai Salah Satu Persyaratan untuk Memenuhi Mata Kuliah  
Tugas Akhir pada Pendidikan Strata Satu Fakultas Teknik  
Universitas Jenderal Soedirman

**Disusun Oleh:**  
**Noer Aditama**  
**H1C013039**

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI**  
**UNIVERSITAS JENDERAL SOEDIRMAN**  
**FAKULTAS TEKNIK**  
**JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO**  
**PURWOKERTO**  
**2020**

**HALAMAN JUDUL**

**LAPORAN TUGAS AKHIR**  
**KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL**  
**MENGUNAKAN METODE**  
***CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASISI ANDROID**



Diajukan Sebagai Salah Satu Persyaratan untuk Memenuhi Mata Kuliah  
Tugas Akhir pada Pendidikan Strata Satu Fakultas Teknik  
Universitas Jenderal Soedirman

**Disusun Oleh:**  
**Noer Aditama**  
**H1C013039**

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI**  
**UNIVERSITAS JENDERAL SOEDIRMAN**  
**FAKULTAS TEKNIK**  
**JURUSAN/PROGRAM STUDI TEKNIK ELEKTRO**  
**PURWOKERTO**  
**2020**

## HALAMAN PENGESAHAN

Proposal Tugas akhir dengan Judul :

**KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL  
MENGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASISI ANDROID**

Disusun Oleh:

Noer Aditama

H1C013039

Disetujui dan disahkan

Pada tanggal, .....

Mengetahui,

Pembimbing I

Pembimbing II

Imron Rosyadi, S.T., M.Sc.

NIP. 197909242003121003

Mengetahui:

Dekan Fakultas Teknik

Dr. Eng. Suroso, S.T., M.Eng.

NIP. 197812242001121002

## HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Laporan Tugas Akhir dengan judul **“KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID”** ini tidak terdapat karya yang pernah diajukan, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Purbalingga, April 2020

[materai]  
Ttd

Noer Aditama  
H1C013039

## **HALAMAN MOTTO DAN PERSEMBAHAN**

### **MOTTO**

*“Sukses itu penting, tapi bersyukur jauh lebih sukses”*

### **PERSEMBAHAN**

1. Orang tua saya dan keluarga
2. Seluruh mahasiswa Teknik Elektro Unsoed.
3. Seluruh mahasiswa Unsoed.
4. Kepada siapa saja yang membutuhkan hasil penelitian saya.

**RINGKASAN**  
**KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL**  
**MENGUNAKAN METODE**  
***CONVOLUTIONAL NEURAL NETWORK (CNN)* BERBASISI ANDROID**

Noer Aditama

Beras merupakan makanan pokok bagi masyarakat di Indonesia. Oleh karena itu beras yang beredar di pasaran harus memiliki standar dan kualitas yang baik. Namun harga beras semakin melonjak menyebabkan banyak beras yang kualitasnya kurang baik beredar di pasaran, sehingga dibutuhkan standar kualitas beras yang dapat digunakan langsung oleh konsumen beras. Standar pengujian kualitas beras dapat dilakukan secara laboratorium dan visual. Pengujian secara visual adalah salah satu cara yang digunakan oleh pihak Bulog untuk menguji kualitas beras dan dapat diterapkan secara langsung menggunakan citra beras melalui kamera *smartphone* konsumen beras.

Dalam hal ini penulis membuat penelitian tentang klasifikasi kualitas beras menggunakan *deep learning* berbasis android agar mudah digunakan menggunakan *Smartphone*. Dalam *deep learning* terdapat *Convolutional Neural Network (CNN)* yang memiliki berbagai macam arsitektur yang dapat digunakan untuk melakukan klasifikasi kualitas beras berdasarkan citranya. Jenis arsitektur CNN yang digunakan pada klasifikasi kualitas beras adalah VGG16Net dan MobileNet dengan menggunakan metode ekstraksi fitur. Kedua arsitektur tersebut digunakan untuk melakukan pelatihan dan pengujian pada infrastruktur *Google Colaboratory*. Dari hasil pelatihan tersebut kemudian disimpan dalam file *TensorFlow Lite* dan diimpor ke dalam *project* pada *Android Studio* agar dapat diimplementasikan pada aplikasi android. Cara penggunaannya hanya dengan memasukkan obyek yang akan diidentifikasi ke dalam aplikasi melalui media pengambilan gambar secara langsung menggunakan kamera atau *import gallery*.

Kata kunci : Klasifikasi kualitas beras, *deep learning*, CNN, *Google Colaboratory*, *Android*

**SUMARRY**  
**RICE QUALITY CLASSIFICATION WITH DIGITAL IMAGES USING**  
**CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD ON ANDROID**

Noer Aditama

*Rice is the staple food for most people in Indonesia. Therefore, rice on the market must have a standard and good quality. However, rice prices increase causing a lot of rice with poor quality on the market, so it needs rice quality standards that can be used directly by consumers of rice. Standard testing the quality of rice can be carried out laboratory and visually. Standard testing visual quality of rice is one of the ways used by Bulog to test the quality of rice, and can be applied directly using the image of rice through the smartphone's camera consumers of rice.*

*In this case the authors make research on the identification of plant diseases using Android-based deep learning so that it is easy to use using smartphones. In the deep learning there is the Convolutional Neural Network has a variety of architectures that can be used to classification rice quakit based on their image. CNN architectures types used in the classification of rice quality are VGG16Net and MobileNet using the feature extraction method. Both architectures are used for training and testing on the Google Colaboratory infrastructures. Then the training data is stored and converted into a TensorFlow Lite file and imported a project in Android studio so that it can be implemented in an Android application. How to use it only by entering the object that will be identified into the application Plant Diseases through media shooting directly using a camera or import gallery.*

*Keywords : rice quality classification, deep learning, CNN, Google Colaboratory, Android*

## KATA PENGANTAR

Segala puji bagi Allah SWT, pencipta seluruh alam semesta, karena atas segala limpahan rahmat dan karunia-Nya, penulis mampu menyusun laporan tugas akhir dengan judul **“KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL MENUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID”** ini dan dapat menyelesaikan dengan baik.

Tugas akhir merupakan salah satu mata kuliah wajib bagi mahasiswa Program Studi Sarjana Teknik Elektro Fakultas Teknik Universitas Jenderal Soedirman, Purwokerto. Melalui tugas akhir, mahasiswa diharapkan dapat menambah wawasan tentang bidang ilmu elektro yang sedang digeluti. Selain itu, dengan adanya tugas akhir diharapkan mahasiswa dapat bertambah pengalaman tentang dunia kerja yang sebenarnya sekaligus mampu menimba ilmu yang tidak sempat tersampaikan pada bangku perkuliahan.

Segala kendala yang dialami penulis dalam proses penyusunan laporan tugas akhir ini Alhamdulillah dapat teratasi berkat bantuan dan dukungan dari berbagai pihak. Dalam hal ini, penulis mengucapkan terimakasih kepada :

1. Kedua orangtua dan keluarga penulis yang senantiasa mendukung dan mendoakan untuk kelancaran studi.
2. Ibu Farida Asriani, S.T., M.Eng. selaku Kepala Jurusan Teknik Elektro Universitas Jenderal Soedirman
3. Bapak Imron Rosyadi, S.T., M.Sc selaku Dosen Pembimbing I Proposal Tugas Akhir, atas segala bimbingan dan masukan dalam penyusunan Proposal Tugas Akhir.
4. ..selaku Dosen Pembimbing II Proposal Tugas Akhir, atas segala bimbingan dan masukan dalam penyusunan Proposal Tugas Akhir.
5. Teman - teman saya yang telah membantu kelancaran fikiran dan membuka wawasan san serta menyemangati saya disaat gundah mengerjakan tugas akhir.



6. Seluruh pihak yang telah membantu baik secara langsung maupun tidak langsung dalam pelaksanaan tugas akhir.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penulisan serta penyusunan laporan tugas akhir ini. Penulis memohon maaf apabila terdapat kekurangan serta kesalahan. Saran dan kritik yang membangun dari para pembaca sangat penulis harapkan untuk penulisan karya – karya lain dikemudian hari yang lebih baik. Semoga apa yang terkandung di dalam laporan ini mampu memberikan manfaat bagi para pembacanya.

Purbalingga, April 2020

**Noer Aditama**

## DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN .....	iv
HALAMAN MOTTO DAN PERSEMBAHAN.....	v
RINGKASAN .....	vi
SUMMARY .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB I.....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II.....	5
2.1 Keaslian Penelitian .....	5
2.2 <i>Deep learning</i> .....	6
2.3 <i>Convolutional Neuro network (CNN)</i> .....	8
2.3.1 <i>Input layer</i> .....	8
2.3.2 <i>Convolution Layer</i> .....	9
2.3.3 <i>Activaton Layer</i> .....	10
2.3.4 <i>Pooling Layer</i> .....	11
2.3.5 <i>Fully Connected Layer</i> .....	12
2.4 Arsitektur CNN.....	13
2.4.1 Arsitektur VGG16.....	13
2.4.2 Arsitektur <i>MobileNet</i> .....	15
2.5 <i>Google Colaboratoty</i> .....	17
2.6 <i>Python</i> .....	18
2.7 <i>Framework Keras</i> .....	18
2.8 <i>Tensor Flow Lite</i> .....	19
2.9 <i>Android Studio</i> .....	20
2.10 <i>Beras</i> .....	20
2.10.1 <i>Beras IR 64</i> .....	21
BAB III .....	22
3.1 Waktu dan Tempat Penelitian .....	22
3.2 Alat dan Bahan .....	22
3.3 Alur dan Tahap Penelitian .....	22
3.3.1 Tahap Persiapan .....	22
3.3.2 Tahap Penelitian.....	23
3.3.3 Tahap Analisis Data .....	25
3.3.4 Tahap Akhir.....	25
3.4 Waktu dan Jadwal Palaksanaan Tugas Akhir .....	25

## DAFTAR TABEL

TABEL 2.1 : Perbandingan Penelitian Terdahulu .....	6
TABEL 2.2 : <i>Arsitektur MobileNet</i> .....	16
TABEL 3.7 : Jadwal Penelitian .....	25

## DAFTAR GAMBAR

GAMBAR 2.1 : Jaringan saraf mendalam untuk klasifikasi digit gambar .....	7
GAMBAR 2.2 : Representasi mendalam .....	8
GAMBAR 2.3 : <i>Proses Input Layer</i> .....	9
GAMBAR 2.4 : <i>Alur Convolution Layer</i> .....	10
GAMBAR 2.5 : <i>Activation functions</i> .....	11
GAMBAR 2.6 : <i>FeatMap</i> .....	11
GAMBAR 2.7 : <i>Proses Pooling</i> .....	12
GAMBAR 2.8 : <i>Fully Connected Layer</i> .....	12
GAMBAR 2.9 : <i>Contoh CNN</i> .....	13
GAMBAR 2.10 : <i>Arsitektur VGG16</i> .....	14
GAMBAR 2.11 : <i>Depthwise separable convolution</i> .....	15
GAMBAR 2.12 : Standard convolution Depthwise separable convolution. ....	17
GAMBAR 2.13 : <i>Arsitektur Konversi model ke TensorFlow Lite</i> .....	19
GAMBAR 2.14 : <i>Beras IR 64</i> .....	21
GAMBAR 3.1 : <i>Desain arsitektur sistem</i> .....	23
GAMBAR 3.2 : <i>Diagram Alur sistem</i> .....	24
GAMBAR 3.3 : <i>Digram Alur Penelitian</i> .....	25

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Beras merupakan makanan pokok bagi sebagian besar masyarakat di Indonesia. Oleh karena itu beras yang beredar di pasaran harus memiliki standar dan kualitas yang baik. Namun harga beras yang semakin melonjak menyebabkan banyak beras dengan kualitas yang kurang baik beredar di pasaran, sehingga dibutuhkan standar kualitas beras.

Kualitas beras dapat dikelompokkan berdasarkan bentuk dan warna. Warna dan bentuk suatu beras dapat menentukan kualitas suatu beras. Semakin putih, bersih dan utuh suatu beras memndakan kualitas beras semakin bagus. Nilai putih dan bersih beras diperoleh dengan menganalisis nilai HSV pada citra beras, sedangkan nilai utuh diperoleh dengan menganalisis luas objek keras [1].

*Deep learning* adalah salah satu metode pembelajaran mesin yang memungkinkan komputer untuk mempelajari tugas tugas yang disesuaikan dengan tugas manusia [2]. *Deep learning* dapat diimplementasikan untuk klasifikasi suatu objek berdasarkan citra dengan cara secara otomatis. Sebuah model *deep learning* dirancang untuk terus menerus menganalisis data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan.

*Convolutional Neuro Network* (CNN) merupakan salah satu jenis algoritma *deep learning* yang dapat menerima *input* berupa gambar, menentukan aspek atau objek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk belajar mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya [3]. CNN dapat dilatih untuk memahami detail sebuah gambar dengan lebih baik.

Saat ini sebagian besar konsumen beras telah menggunakan *smartphone* berbasis android kamera yang cukup mumpuni maka untuk memudahkan konsumen beras dibutuhkan aplikasi pengujian kualitas beras secara visual berbasis android.

Oleh karena itu penulis akan melakukan penelitian berjudul  
**“KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL**

## **MENGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID”**

### **1.2 Rumusan Masalah**

Berdasarkan pada latar belakang di atas, maka dapat di rumuskan beberapa rumusan masalah yaitu:

1. Bagaimana perancangan arsitektur CNN untuk klasifikasi kualitas beras?
2. Bagaimana pelatihan dan pengujian untuk klasifikasi kualitas beras dengan CNN?
3. Bagaimana pengujian kalsifikasi kualitas beras pada aplikasi android?

### **1.3 Batasan Masalah**

Agar dalam tugas akhir ini didapatkan hasil yang optimal maka permasalahan dibatasi sebagai berikut:

1. Metode *deep learning* yang digunakan untuk klasifikasi kualitas bera menggunakan *Convolutional Neuro Network* (CNN).
2. Program yang dibuat untuk membedakan tiga kualitas beras, yaitu premium, medium dan tidak layak pada beras IR64.
3. Program kalsifikasi kualitas beras menggunakan bahasa pemrograman *python* dengan antarmuka dan infrastuktur *Google Colaboratory*.
4. Pemodelan *deep learning* menggunakan *framework Keras* dan *TensorFlow*.
5. Aplikasi android pada *smartphone* menggunakan *Andorid Studio*.
6. Penentuan jumlah *dataset* pelatihan, validasi serta pengujian tidak dibahas dalam penelitian ini.

### **1.4 Tujuan Penelitian**

Pada penelitian ini bertujuan untuk:

1. Merancang arsitektur CNN untuk klasifikasi kualitas beras dan implementasinya padda aplikasi Android.

2. Melatih arsitektur CNN untuk klasifikasi kualitas beras.
3. Menguji arsitektur CNN yang sudah dilatih sebelumnya.

### **1.5 Manfaat Penelitian**

Beberapa manfaat yang diperoleh dari penelitian ini sebagai berikut:

1. Menentukan kualitas beras dapat dilakukan secara otomatis menggunakan *smartphone*.
2. Mempermudah konsumen beras dalam menentukan kualitas beras di pasaran.

### **1.6 Sistematika Penulisan**

Adapun sistematika penulisan yang digunakan dalam penelitian ini, sebagai berikut:

## **BAB I PENDAHULUAN**

Bab ini merupakan pendahuluan yang berisi tentang judul penelitian, latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi tentang teori dasar kajian pustaka yang mendasari berbagai gagasan tentang penelitian terdahulu serta teori-teori pendukung dilakukan penelitian seperti penjelasan tentang *deep learning*, *convolutional neuro network* seperti arsitektur, komponen, serta penjelasan *Google Colabratory* dan penjelasan *framework keras dan Tensorfow*.

## **BAB III METODE PENELITIAN**

Bab ini berisi tentang urutan langkah atau metode yang akan digunakan dalam penelitian, meliputi waktu dan tempat, alat dan bahan, metode penelitian, variabel penelitian, sumber data, alur penelitian dan jadwal penelitian.

#### **BAB IV PEMBAHASAN**

Bab ini berisi tentang hasil dan analisa dari penelitian mengenai klasifikasi kualitas beras pada aplikasi android. Pada bab ini juga membahas serta mengkaji tentang hasil dan analisa sesuai rumusan masalah.

#### **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan dan saran dari hasil penelitian yang di dapat.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Keaslian Penelitian

Pada penelitian tugas akhir ini peneliti mengambil judul “ **KLASIFIKASI KUALITAS BERAS DENGAN CITRA DIGITAL MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) BERBASIS ANDROID** “. Belum ada yang mengambil penelitian tentang klasifikasi kualitas beras, tetapi peneliti mengambil literatur melalui penelitian sebelumnya dan beberapa jurnal sebelumnya sebagai berikut:

1. Arif Budiman Harahap (2016) penelitian berjudul Pengembangan Aplikasi Android untuk Mengidentifikasi Kualitas Beras dengan Citra Digital. Membahas tentang kualitas beras menggunakan metode pengembangan aplikasi *waterfall* dengan dikalsifikasikan menggunakan *rules* yang didapatkan dari pohon keputusan C4.5.
2. Tiara Shafira (2018) penelitian berjudul Implementasi *Convolutionsl Neural network* untuk Klasifikasi Citra Tomat Menggunakan Keras. Membahas tentang mengklasifikasikan tomat layak dan tidak layak pada jenis tomat merah.
3. Jepri (2019) penelitiannya berjudul identifikasi Penyakit Pada Daun Tomat dan Daun Singkong Berdasarkan Citra Daun Menggunakan Metode *Convolutionsl Neural network (CNN)* Berbasis *Android*. Membahas tentang 10 penyakit tanaman pada tomat dan 5 penyakit pada daun singkong, menggunakan 3 buah arsitektur yaitu VGG16, InceptionV3 dan MobileNET.
4. Vidi fitriansyah Hidarlan (2019) penelitiannya yang berjudul Rancang bangun Klasifikasi Varietas Beras berdasarkan Citra Menggunakan Metode *Convolutionsl Neural network (CNN)* Berbasis *Android*. Membahas tentang mengkalsifikasikan 3 varietas beras yaitu Bashmati, IR64 dan Ketan, mrnggunsksn sritektur VGG16 NET dan Mobile NET.

Perbandingan penelitian penulis dengan penelitian terdahulu dijelaskan pada Tabel 2.1 sebagai berikut.

Tabel 2.1 Perbandingan Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Perincian Penelitian	Perbedaan Dengan Penelitian Penulis
1	Arif Budiman Harahap (2016)	Pengembangan Aplikasi Android untuk Mengidentifikasi Kualitas Beras dengan Citra Digital	<ul style="list-style-type: none"> <li>Bertujuan mengkalsifikasi beras menggunakan <i>waterfall</i> dan <i>rues</i> yang didapat dari pohon keputusan C4.5</li> </ul>	<ul style="list-style-type: none"> <li>Mengkalsifikasikan tidak hanya kualitas varietas beras, tetapi juga kualitas masing masing farietas mulai dari premium, medium, dan tidak layak.</li> <li>Menggunakan <i>Google collaboratory</i>.</li> </ul>
2	Tiara Shafira (2018)	Implementasi <i>Convolutionsl Neural network</i> untuk Klasifikasi Citra Tomat Menggunakan Keras	<ul style="list-style-type: none"> <li>Bertujuan untuk mengklasifikasikan tomat layak dan tidak layak pada jenis tomat merah</li> <li>Menggunakan software Rstudio versi 1.1.383</li> </ul>	<ul style="list-style-type: none"> <li>Mengkalsifikasikan tidak hanya kualitas varietas beras, tetapi juga kualitas masing masing farietas mulai dari premium, medium, dan tidak layak.</li> <li>Menggunakan <i>Google collaboratory</i></li> </ul>
3	Jepri (2019)	Identifikasi Penyakit Pada Daun Tomat dan Daun Singkong Berdasarkan Citra Daun Menggunakan Metode <i>Convolutionsl Neural network (CNN)</i> Berbasis <i>Android</i>	<ul style="list-style-type: none"> <li>Melakukan Penelitian terhadap 10 penyakit pada tomat dan 5 penyakit pada daun singkong. Menggunakan arsitektur InceptionV3, VGG16 Net dan Mobile Net</li> </ul>	<ul style="list-style-type: none"> <li>Mengkalsifikasikan tidak hanya kualitas varietas beras, tetapi juga kualitas masing masing farietas mulai dari premium, medium, dan tidak layak.</li> </ul>
4	Vidi fitriansyah Hindarlan (2019)	Rancang bangun Klasifikasi Varietas Beras berdasarkan Citra Menggunakan Metode <i>Convolutionsl Neural network (CNN)</i> Berbasis <i>Android</i>	<ul style="list-style-type: none"> <li>Melakukan Penelitian terhadap 3 varietas beras yaitu IR64, ketan, Basmathi.</li> <li>Menggunakan arsitektur VGG16 Net dan Mobile Net</li> </ul>	<ul style="list-style-type: none"> <li>Mengkalsifikasikan tidak hanya kualitas varietas beras, tetapi juga kualitas masing masing farietas mulai dari premium, medium, dan tidak layak.</li> </ul>

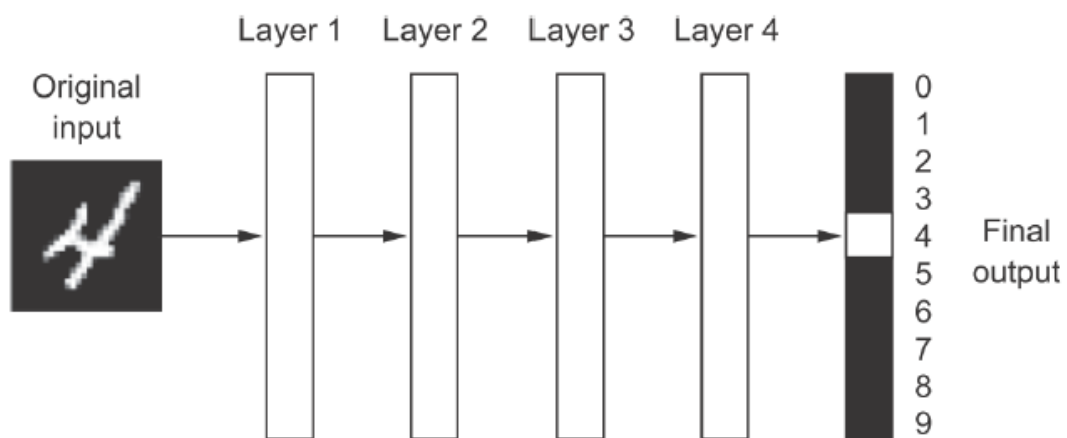
## 2.2 Deep learning

*Deep learning* adalah salah satu metode pembelajaran mesin yang memungkinkan komputer untuk mempelajari tugas tugas yang disesuaikan dengan tugas manusia [2]. *Deep learning* dapat diimplementasikan untuk klasifikasi suatu objek berdasarkan citra dengan cara secara otomatis. Sebuah model *deep learning*

diraancang untuk terus menerus menganalisis data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan. *Deep Learning* bukan merupakan referensi untuk segala jenis pemahaman yang lebih dalam yang dicapai oleh pendekatan, melainkan ia mewakili gagasan lapisan representasi yang berurutan.

Banyak lapisan (*laayer*) yang berkontribusi pada model data disebut kedalaman model. Nama-nama lain yang sesuai untuk bidang ini dapat berupa representasi pembelajaran berlapis dan pembelajaran representasi hierarkis. *Deep learning* modern sering melibatkan puluhan atau bahkan ratusan lapisan representasi berturut-turut dan mereka semua belajar secara otomatis dari data pelatihan.

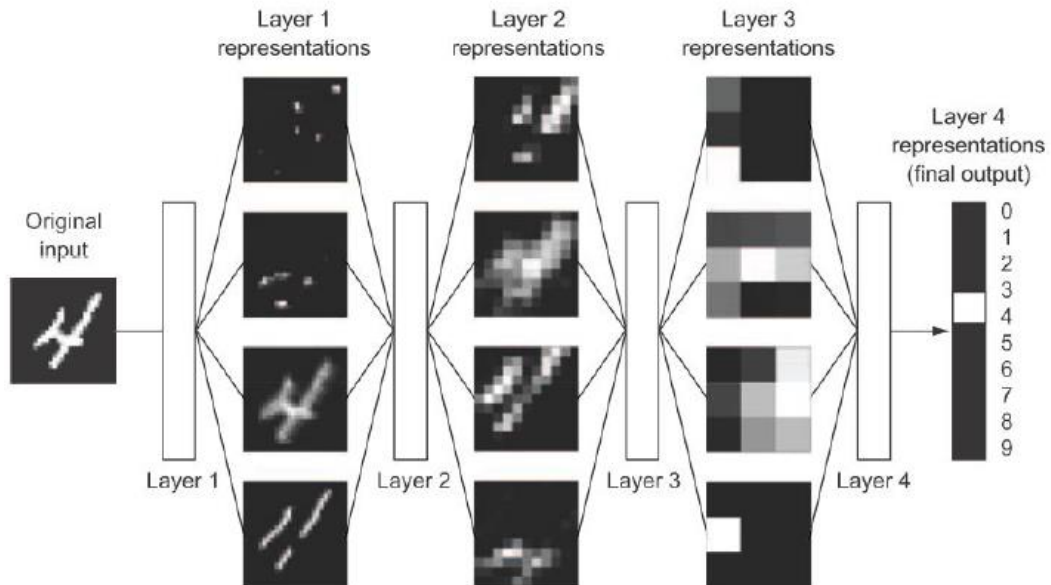
Dalam *Deep Learning*, representasi berlapis ini dipelajari melalui model yang disebut jaringan saraf. Istilah jaringan saraf adalah referensi untuk neuribiologi, tetapi meskipun beberapa konsep dalam *deep learning* dikembangkan dalam bagian dengan menggambarkan inspirasi dari pemahaman kita tentang otak, model *deep learning* bukanlah model otak. Tidak ada bukti bahwa otak mengimplementasikan sesuatu seperti pembelajaran yang digunakan yang digunakan dalam model *deep learning* modern. Representasi yang dipelajari pada algoritma *deep leearning* dapat dilihat bagaimana jaringan lapisan banyak mengubah gambar digit untuk mengenali digit apa itu seperti pada gambar 2.1.



Gambar 2.1 Jaringan saraf mendalam untuk klasifikasi digit gambar

Pada gambar 2.2 jaringan mengubah digit menjadi representasi yang semakin berbeda dari gambar asli dan semakin informatif tentang hasil akhir. Kita dapat

mengganggu jaringan yang dalam sebagai operasi distilasi informasi yang bertingkat, di mana informasi melewati filter berturut-turut dan keluar semakin dimurnikan, dimana informasi melewati filter berturut-turut dan keluar semakin dimurnikan.



Gambar 2.2 representasi mendalam yang dipelajari oleh model kalsifikasi digit

### 2.3 Convolutional Neuro network (CNN)

*Convolutional Neuro Network* (CNN) merupakan salah satu jenis algoritma *deep learning* yang dapat menerima *input* berupa gambar, menentukan aspek atau objek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk belajar mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya [3]. CNN dapat dilatih untuk memahami detail sebuah gambar dengan lebih baik. CNN termasuk dalam jenis *depp neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada citra. Pada klasifikasi citra, *multylayer perceptron* (MLP) kurang sesuai untuk digunakan karena tidak menyimpan informasi spesial dari data citra dan menganggap setiap piksel adalah filter yang independen sehingga menghasilkan hasil yang kurang baik.

#### 2.3.1 Input Layer

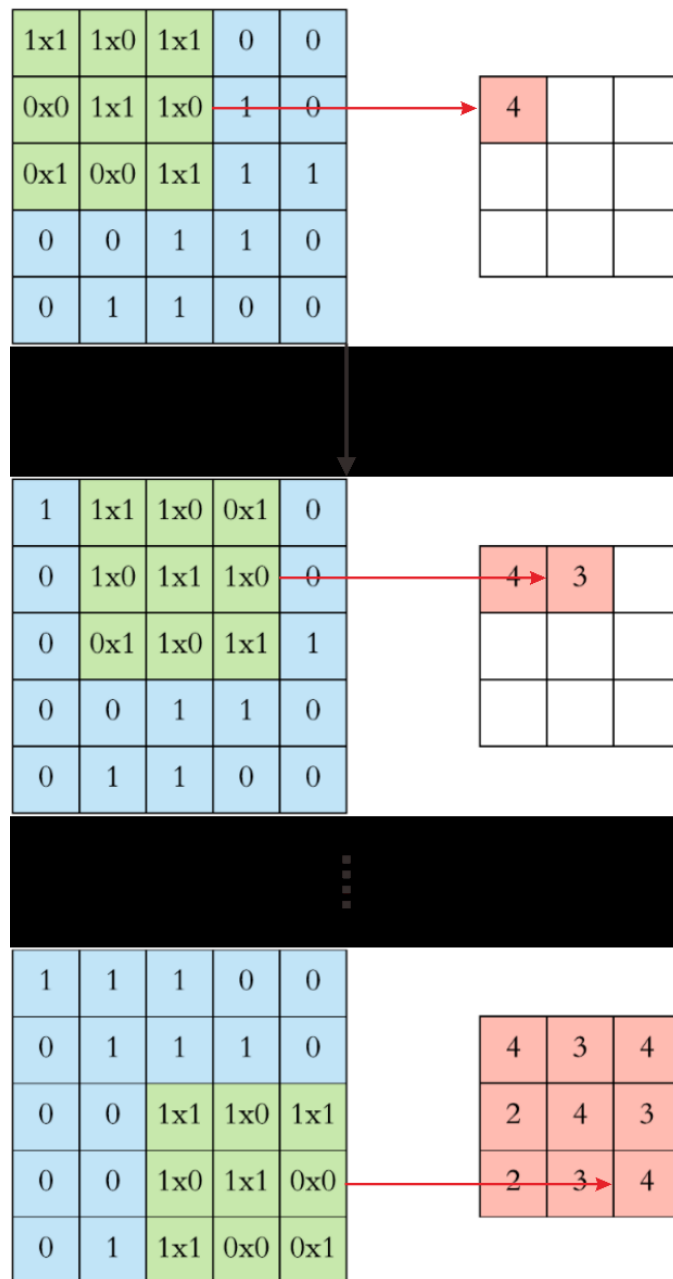
*Input* layer menampung nilai *pixel* dari citra yang menjadi masukan [9]. Untuk citra dengan ukuran  $64 \times 64$  dengan 3 *channel* warna, RGB (Red, Green, Blue) maka yang menjadi masukan akan adalah *pixel array* yang berukuran  $64 \times 64 \times 3$ .



*Gambar 2.3 Proses Input Layer*

### **2.3.2 Convolution Layer**

*Convolution Layer* adalah inti dari CNN. *Convolution Layer* menghasilkan citra baru yang menunjukkan fitur dari citra *input* [9]. Dalam proses tersebut, *Convolution Layer* menggunakan filter pada setiap citra yang menjadi masukan. Filter pada layer ini berupa *array* 2 dimensi bisa berukuran  $5 \times 5$ ,  $3 \times 3$  atau  $1 \times 1$ . Proses *convolution* dengan menggunakan filter pada layer ini akan menghasilkan *feature map* yang akan digunakan pada *activation layer*. Gambar di bawah ini menunjukkan alur pada *Convolution Layer*.



Gambar 2.4 Alur Convolution Layer

### 2.3.3 Activation Layer

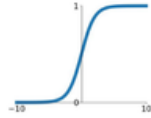
*Activation Layer* adalah layer di mana *feature* map dimasukkan ke dalam *activation Function*. *Activation function* digunakan untuk mengubah nilai-nilai pada *feature* map pada *range* tertentu sesuai dengan *activation function* yang digunakan. ini bertujuan untuk meneruskan nilai yang menampilkan fitur dominan

dari citra yang masuk ke layer berikutnya. *Activation function* yang umum digunakan bisa dilihat pada gambar 2.5 berikut

## Activation Functions

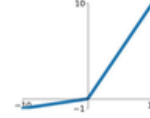
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



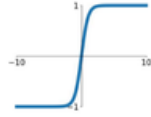
### Leaky ReLU

$$\max(0.1x, x)$$



### tanh

$$\tanh(x)$$

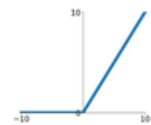


### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

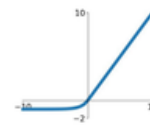
### ReLU

$$\max(0, x)$$



### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Gambar 2.5 Activation functions

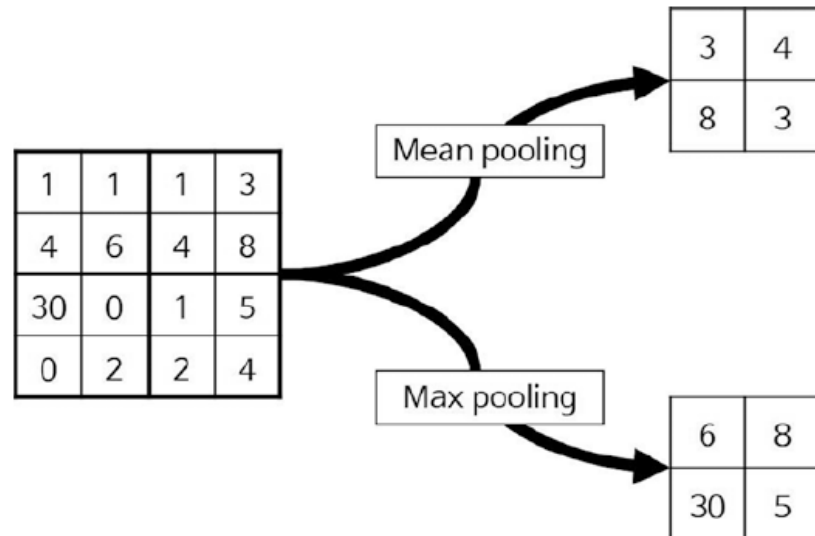
### 2.3.4 Pooling Layer

*Pooling layer* menerima *input* dari *activation layer* kemudian mengurangi jumlah parameternya. *Pooling* juga biasa disebut sub sampling atau *down sampling* yang mengurangi dimensi dari *feature map* tanpa menghilangkan informasi penting di dalamnya. Proses dalam *pooling* layer cukup sederhana. pertama-tama kita menentukan ukuran *down sampling* yang akan digunakan pada *feature map*, misalnya  $2 \times 2$ . Setelah itu kita akan melakukan proses *pooling* pada *feature map*, sebagai contoh kita akan menggunakan *feature map* berukuran  $4 \times 4$  berikut.

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

Gambar 2.6 FeatMap

Setelah itu kita akan menggunakan matrix  $2 \times 2$  untuk melakukan proses *pooling*. proses *Pooling* sendiri ada beberapa macam seperti *Max pooling*, *Mean pooling*, dan *Sum pooling*.

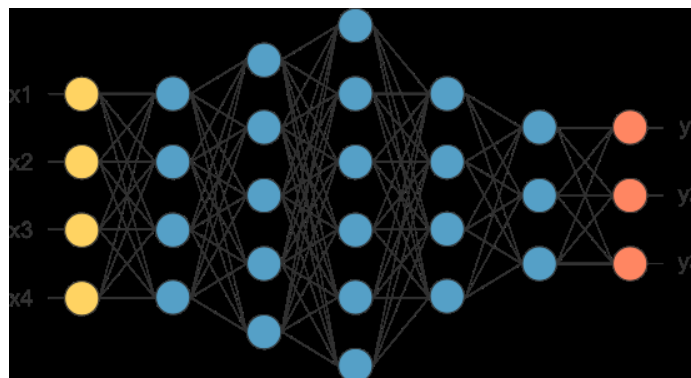


Gambar 2.7 Proses Pooling

### 2.3.5 Fully Connected Layer

Setelah melewati proses-proses di atas, hasil dari *pooling* layer digunakan menjadi masukan untuk *Fully connected layer*. Layer ini memiliki kesamaan struktur dengan *Artificial Neural Network* (ANN) pada umumnya yaitu memiliki *input layer*, *hidden layer* dan *output layer* yang masing-masing memiliki *neuron-neuron* yang saling terhubung dengan neu-

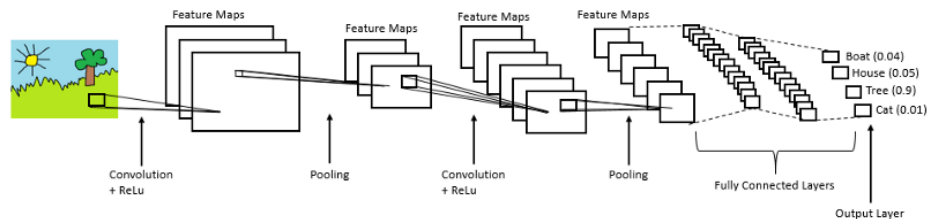
ron-neuron di layer tetangganya. gambar di bawah ini merupakan contoh *Fully Connected Layer*.



Gambar 2.8 Fully Connected Layer



Pada gambar di atas dapat dilihat sebelum hasil *pooling* digunakan sebagai *input*, hasil *pooling* terlebih dahulu diubah menjadi vektor ( $x_1, x_2, x_3, \text{dst}$ ) kemudian dari sini diproses ke dalam *Fully Connected Layer*. Pada layer terakhir di dalam *Fully Connected layer* akan digunakan *activation function sigmoid* atau *softmax* untuk menentukan klasifikasi dari citra masukan yang dari *Input Layer* CNN.

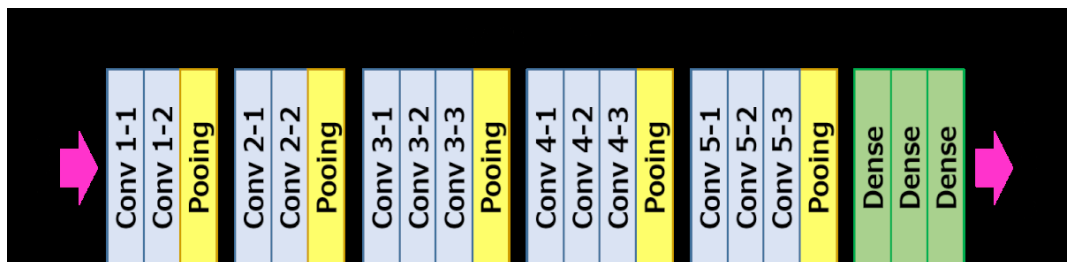


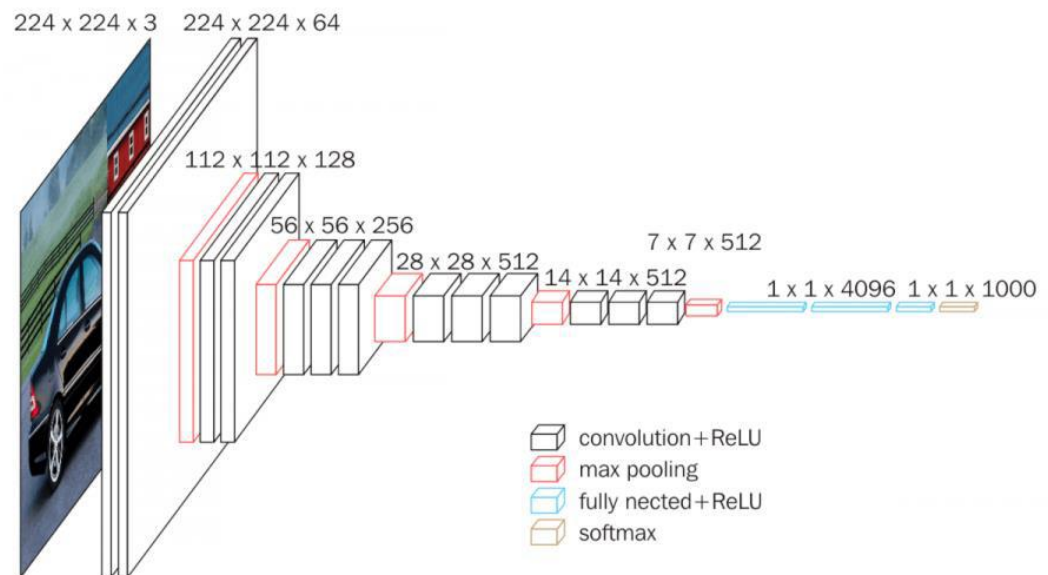
Gambar 2.9 Contoh CNN

## 2.4 Arsitektur CNN

### 2.4.1 Arsitektur VGG16

VGG16 adalah model jaringan saraf convolutional yang diusulkan oleh K. Simonyan dan A. Zisserman dari University of Oxford dalam makalah “Jaringan Konvolusional Sangat Dalam untuk Pengenalan Gambar Skala Besar” [5]. Model ini mencapai akurasi tes top-5 92,7% di ImageNet, yang merupakan dataset lebih dari 14 juta gambar dengan 1000 kelas. VGG16 adalah salah satu model terkenal yang diajukan ke ILSVRC-2014.





Gambar 2.10 Arsitektur VGG16

Masukan ke lapisan *conv1* berukuran tetap 224x224 gambar RGB. Gambar dilewatkan melalui tumpukan lapisan konvolusional (*conv.*), dimana filter digunakan dengan bidang reseptif yang sangat kecil sebesar 3×3. Dalam salah satu konfigurasi, ia juga menggunakan *filter* konvolusi 1×1, yang dapat dilihat sebagai transformasi linear dari saluran masukan (diikuti oleh non-linearitas). Langkah (*stride*) konvolusi ditetapkan pada 1 piksel, spasial *padding* dari masukan lapisan konvolusi sedemikian rupa sehingga resolusi spasial/ukuran tersebut dipertahankan setelah konvolusi, yaitu *padding* sebesar 1 piksel untuk 3×3 lapisan konvolusi. Spasial *pooling* dilakukan oleh lima lapisan *max-pooling*, yang mengikuti beberapa lapisan konvolusi (tidak semua lapisan konv diikuti oleh *max-pooling*). *Max-pooling* dilakukan dengan ukuran piksel 2×2, dan 2 *stride*. Tiga lapisan *Fully-Connected* (FC) mengikuti tumpukan lapisan konvolusional (yang memiliki kedalaman berbeda dalam arsitektur yang berbeda): dua yang pertama memiliki ukuran masing-masing 4096 kanal, yang ketiga melakukan klasifikasi ILSVRC 1000 keluaran dan dengan demikian memuat 1000 saluran (satu untuk setiap kelas). Lapisan terakhir adalah lapisan *soft-max*. Konfigurasi *fully connected layers* adalah sama di semua jaringan.

Setiap *hidden layers* dilengkapi dengan fungsi aktivasi ReLU. Juga dicatat bahwa tidak ada jaringan (kecuali satu) yang berisi Normalisasi Respon Lokal

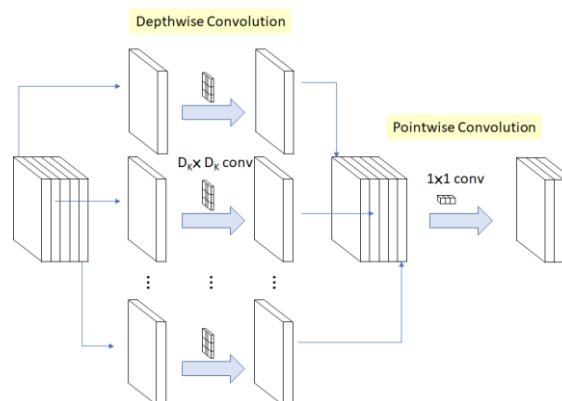
(LRN), normalisasi seperti itu tidak meningkatkan kinerja pada dataset ILSVRC, tetapi mengarah pada peningkatan konsumsi memori dan waktu komputasi.

#### 2.4.2 Arsitektur *MobileNet*

*MobileNets*, merupakan salah satu arsitektur CNN yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih [11]. Seperti namanya, *Mobile*, para peneliti dari *Google* membuat arsitektur CNN yang dapat digunakan untuk ponsel. Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN pada umumnya adalah penggunaan lapisan konvolusi dengan ketebalan *filter* yang sesuai dengan ketebalan dari gambar masukan. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.

##### 1. *Depthwise Separable Convolution*

Model *MobileNet* didasarkan pada konvolusi mendalam yang dapat dipisahkan (*depthwise separable convolution*), yaitu bentuk konvolusi yang dibentuk dengan menguraikan konvolusi standar (*standard convolution*) menjadi konvolusi mendalam (*depthwise convolution*) dan konvolusi  $1 \times 1$  yang disebut konvolusi searah (*pointwise convolution*) [5]. Untuk *MobileNets*, *depthwise convolution* menerapkan satu *filter* ke setiap saluran masukannya. *Pointwise convolution* kemudian menerapkan konvolusi  $1 \times 1$  untuk menggabungkan keluaran dari *depthwise convolution*. Konvolusi standar melakukan *filter* dan menggabungkan masukan ke dalam *set* keluaran baru dalam satu langkah. *Depthwise separable convolution* membaginya menjadi dua lapisan, lapisan terpisah untuk *filter* dan lapisan terpisah untuk menggabungkannya. Penguraian ini memiliki efek mengurangi komputasi dan ukuran model secara drastis.



Gambar 2.11 Depthwise separable convolution

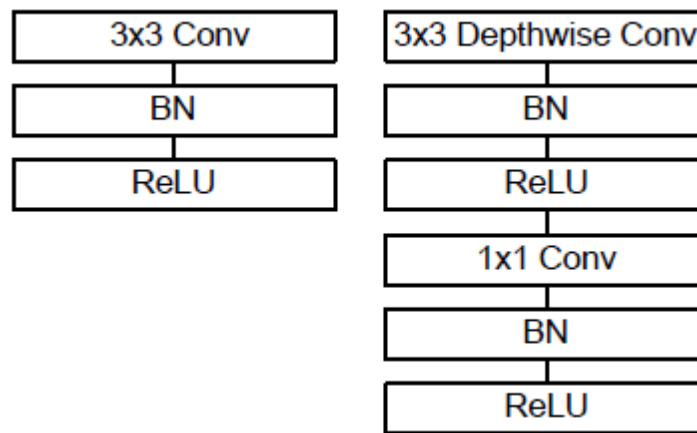
## 2. Struktur Jaringan

Arsitektur *MobileNet* didefinisikan dalam Tabel 2.1. Semua lapisan diikuti oleh *batchnorm* (*backnormalization*) dan ReLU nonlinier dengan pengecualian lapisan akhir yang terhubung penuh yang tidak memiliki nonlinier dan dimasukkan ke dalam lapisan *softmax* untuk klasifikasi.

Tabel 2.2 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	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
<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 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	$1 \times 1 \times 1000$

Pada sebuah lapisan konvolusi standar hanya menggunakan lapisan konvolusi biasa sebesar  $3 \times 3$ . Sedangkan pada sebuah lapisan *depthwise separable convolution* dipisahkan menjadi dua konvolusi yaitu  $3 \times 3$  *depthwise convolution* dan  $1 \times 1$  *pointwise convolution* serta *batchnorm* dan ReLU nonlinier di setiap lapisan konvolusinya seperti pada gambar 2.12 berikut.



Gambar 2.12 Standard convolution (kiri), Depthwise separable convolution(kanan) dengan ReLU dan BN.

## 2.5 Google Colaboratoty

*Google Colaboratoty* adalah salah satu produk Google berbasis *cloud* yang bisa kita gunakan secara gratis. Perbedaannya adalah Google Colab dibuat khusus untuk para *programmer* atau *researcher* yang mungkin kesulitan untuk mendapatkan akses komputer dengan spek tinggi. Google Colab adalah *coding environment* bahasa pemrograman *Python* dengan format “*notebook*” (mirip dengan *Jupyter notebook*), atau dengan kata lain Google seakan meminjamkan kita komputer secara gratis! untuk membuat program oleh Google [6]. Beberapa kelebihan dari Google Colab adalah sebagai berikut:

1. *Free GPU!* Google Colab memudahkan kita untuk menjalankan program pada komputer dengan spek tinggi (GPU Tesla, RAM ~12GB, *Disk* ~300GB yang masih bisa sambung dengan *Google Drive*, akses internet cepat untuk *download file* besar) dan *running* dalam waktu yang lama (Google *Colab* mengizinkan kita untuk menjalankan program hingga 12 jam).
2. Mudah berintegrasi! Google *Colab* terbilang sangat fleksibel dalam hal integrasi. Kita dapat dengan mudah menghubungkan Google *Colab* dengan *jupyter notebook* di komputer kita (*local runtime*), menghubungkan dengan *Google Drive*, atau dengan *Github*.
3. *Collaborate!* Google Colab juga memudahkan kita berkolaborasi dengan orang lain dengan cara membagi *kodingan* secara *online*.

4. Fleksibel! Salah satu yang saya favoritkan adalah kita bisa dengan mudah menjalankan *deep learning* program via HP! ya karena pada esensinya Google *Colab* hanya perlu *running* di browser, kita bisa mengawasi proses *training* (atau bahkan *coding*) via browser *smartphone* kita selama *smartphone* kita terhubung dengan Google Drive yang sama.

## 2.6 Python

*Python* adalah bahasa pemrograman interpretatif multiguna yang memakai filosofi perancangan dengan fokus kepada tingkat keterbacaan kode. Sebagai bahasa pemrograman, *Python* menggabungkan kemampuan, kapabilitas dan sintaksis kode serta fungsi pustaka yang berkualitas tinggi [7]. Banyak sekali fitur yang dimiliki *Python* sehingga menarik digunakan. Berikut fitur-fitur *Python* yang menjadi keunggulan darinya:

1. Berorientasi kepada objek.
2. Mudah dikembangkan dengan menciptakan modul-modul baru. Modul tersebut juga bisa dibangun dengan bahasa *Python*.
3. Memiliki tata bahasa yang mudah dipelajari
4. Didukung sistem pengelolaan memori secara otomatis sehingga membutuhkan kinerja saat *coding*.
5. *Python* juga memiliki banyak fasilitas pendukung sehingga ketika mengoperasikannya, terhitung mudah dan cepat.

## 2.7 Framework Keras

Saat ini kita mengenal cukup banyak *framework* untuk penggunaan *deep learning*. Beberapa yang cukup populer di antaranya adalah: *Tensorflow*, Keras, *MXNet*, dan *PyTorch*.

Pada analisis ini, saya hanya akan menggunakan *framework* Keras dengan bantuan *R interface* dari RStudio. Keras adalah high-level neural network API yang dikembangkan dengan *Python* dengan fokus tujuan untuk mempercepat proses riset atau percobaan [8]. Beberapa fitur utama dari Keras:

1. Mampu menjalankan *source code* yang sama menggunakan CPU atau GPU dengan lancar.

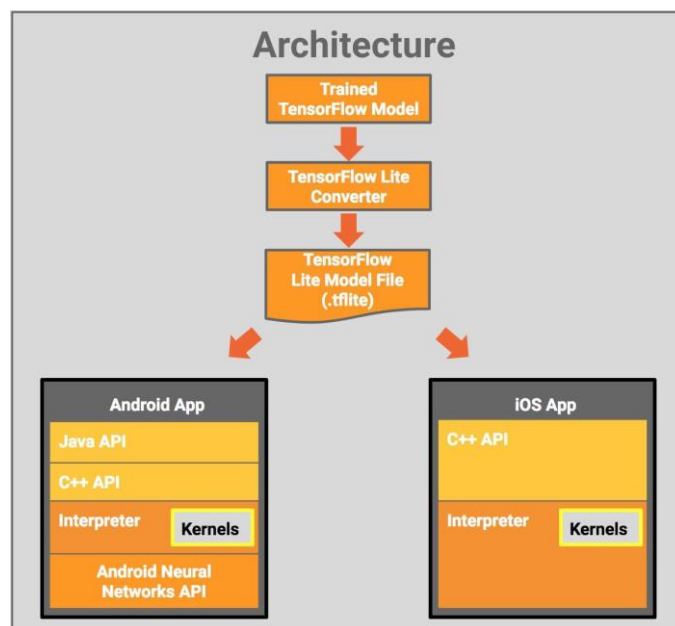
2. API yang *user-friendly* sehingga mempermudah penggunaanya dalam proses prototipe model *deep learning*.
3. Dukungan *built-in* untuk CNN atau *Convolutional Neural Networks* (*Computer Vision*), RNN atau *Recurrent Neural Networks* (untuk *sequence processing*), dan kombinasi keduanya.
4. Dapat digunakan untuk hampir semua jenis dari model *deep learning*.

## 2.8 TensorFlow Lite

*TensorFlow Lite* adalah solusi mudah untuk perangkat seluler dan *embedded devices*. Ini memungkinkan untuk menjalankan model yang dipelajari mesin pada perangkat seluler dengan latensi rendah, sehingga dapat memanfaatkannya untuk melakukan klasifikasi, regresi, atau apa pun yang mungkin Anda inginkan tanpa harus melakukan perjalanan bolak-balik ke server [9].

TensorFlow Lite terdiri dari runtime tempat Anda dapat menjalankan model yang sudah ada sebelumnya, dan seperangkat alat yang dapat Anda gunakan untuk mempersiapkan model Anda untuk digunakan pada perangkat seluler dan *embedded devices*.

TensorFlow Lite saat ini dalam pratinjau pengembang, sehingga mungkin tidak mendukung semua operasi di semua model TensorFlow. Meskipun demikian, ini berfungsi dengan model Klasifikasi Gambar umum.



Gambar 2.13 Arsitektur Konversi model ke *TensorFlow Lite*

## 2.9 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA [18]. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android, seperti:

1. Sistem build berbasis Gradle yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
5. *Template* kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. *Framework* dan fitur pengujian yang lengkap.
7. Fitur *lint* untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

## 2.10. Beras

Beras adalah salah satu komoditas penting bagi Indonesia. Hal ini mengingat hampir seluruh masyarakat Indonesia mengkonsumsi beras sebagai makanan pokoknya. Ini sebabnya Indonesia merupakan konsumen pangan dengan bahan pangan beras terbesar.

Selain itu, beras sangat berpengaruh bagi perekonomian Indonesia karena lebih dari 60% penduduk Indonesia berprofesi sebagai petani penghasil beras. Dengan demikian beras tidak hanya dibutuhkan untuk dikonsumsi tetapi juga merupakan sumber pendapatan dan penyerapan tenaga kerja [10].



### 2.10.1 Beras IR 64

Beras IR 64 merupakan salah satu jenis beras pada sawah yang memiliki bentuk tegak, tinggi sekitar 115-126 cm. Gabahnya berbentuk ramping dan panjang dan berwarna kuning bersih. Padi jenis ini cocok ditanam di daerah lahan sawah irigasi daratan rendah sampai sedang [11]. Beras IR 64 adalah jenis beras yang pulen jika dimasak menjadi nasi karena memiliki kadar amilosa sebanyak 23%. Bobot beras per seribu butir beras IR 64 adalah 24.1 gram.



Gambar 2.14 Beras IR 64

## BAB III

### METODE PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan dalam waktu 4 bulan dimulai dari bulan Maret 2020 hingga bulan Juni 2020 dan bertempat di Kampus Fakultas Teknik Universitas Jenderal Soedirman.

#### 3.2 Alat dan Bahan

Alat dan bahan penunjang penelitian sebagai berikut:

1. Satu unit notebook Acer E5-475G-38LQ dengan processor intel core i3, kapasitas RAM 4GB, dan sistem operasi windows 10 64-bit.
2. Komputer virtual *Google Collaboratory* dengan spesifikasi GPU: 1x Tesla K80, komputasi 3.7, mempunyai inti CUDA 2496, 12GB GDDR5 VRAM, CPU: 1x *single core hyper threaded Xeon Processors @2.3 Ghz*, RAM:~12.6GB, *Disk:~33GB*.
3. Satu unit *smartphone* Samsung A20 SM-A205F/DS dengan spesifikasi *processor Exynos 7884 octa core 2x1.6 Ghz & 6x1.35 Ghz* , RAM 3GB, dan storage 32 GB.
4. Sistem operasi *Windows* 10 64 bit
5. Peramban Internet *Google Chrome* versi 81.0.4044.129 64 bit.
6. *Google Colaboratory (Jupiter Notebook versi cloud)*.
7. *Android Studio*.
8. Sistem operasi Android *Pie* 9
9. Layanan repository *Weeb development* pada *Platform Github*.
10. Dataset pelatihan dan pengujian yang berupa gambar jenis dan kualitas beras IR64 “.jpg” yang diambil menggunakan kamera *smartphone*.

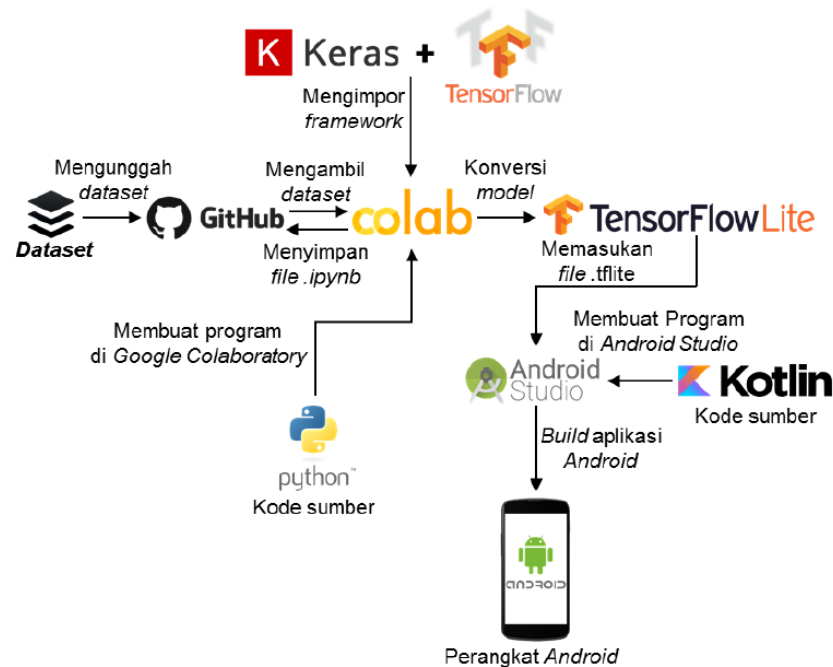
#### 3.3 Alur dan Tahap Penelitian

##### 3.3.1 Tahap Persiapan

Tahap ini merupakan tahap awal sebelum melakukan penelitian yang meliputi pembuatan pra-proposal, membuat perumusan masalah, identifikasi masalah, memahami permasalahan melalui berbagai macam sumber referensi (jurnal, buku, artikel, dan lain – lain), serta menentukan objek serta lokasi penelitian.

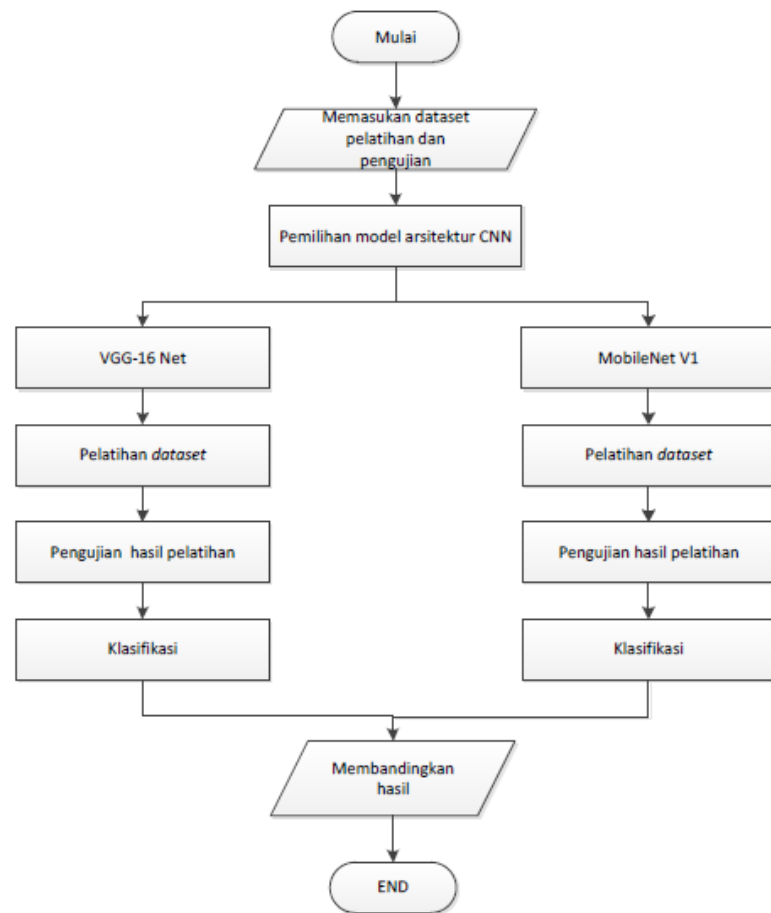
### 3.3.2 Tahap Penelitian

Tahap ini merupakan tahap *pre-prose dataset*, tahap desain arsitektur dan tahap pengujian. Desain arsitektur dari sistem yang akan dibuat disajikan pada gambar 3.1 berikut.



Gambar 3.1 Desain arsitektur sistem

Sedangkan untuk diagram alir sistem yang dirancang pada penelitian ini dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 Diagram alur sistem

Adapun diagram alir dari tahapan-tahapan penelitian yang dilakukan adalah seperti yang disajikan pada gambar 3.3. berikut.



