

# PERANCANGAN APLIKASI ANDROID PENDETEKSI MASKER MUKA DENGAN TEKNIK DEEP LEARNING

## DESIGNING AN ANDROID APPLICATION FOR FACE MASK DETECTION WITH DEEP LEARNING TECHNIQUES

Muhammad Fakhurrozi Sutisna<sup>\*1</sup>, Imron Rosyadi, S.T., M.Sc.<sup>2</sup>, Muhammad Syaiful Aliim, S.T., M.T.<sup>3</sup>

\*Email: [muhammad.sutisna@mhs.unsoed.ac.id](mailto:muhammad.sutisna@mhs.unsoed.ac.id)

<sup>1</sup>Mahasiswa Pemakalah, <sup>2</sup>Dosen Pembimbing I, <sup>3</sup>Dosen Pembimbing II

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman  
Jalan Mayor Jenderal Soengkono KM 5 Blater Purbalingga 53371

---

**Abstrak**— *Coronavirus Disease 2019* (Covid-19) merupakan penyakit menular yang disebabkan oleh virus corona jenis baru, penyakit ini dapat menular dengan cepat secara *droplet*. Sehingga semua orang harus menjaga dan melindungi diri dengan berbagai macam protokol kesehatan, salah satunya adalah memakai masker ketika keluar rumah. Dalam kebiasaan baru penggunaan masker muka merupakan suatu kewajiban. Maka dari itu diperlukan sebuah aplikasi untuk mendeteksi penggunaan masker muka agar orang-orang lebih disiplin dan peduli terhadap penggunaan masker muka. Pendeteksi masker muka dibangun dengan menggunakan teknik *deep learning*. Dengan arsitektur yang digunakan yaitu jenis *MobileNetV2* dan *VGG16Net* serta menggunakan deteksi muka dengan *SSD ResNet* dan *MTCNN*. Pelatihan dan pengujian model dilakukan pada infrastruktur *Google Colaboratory*. Hasil dari pelatihan model tersebut kemudian disimpan dan dikonversi ke dalam bentuk *file TensorFlow Lite* yang selanjutnya diimpor ke dalam *project* pada *Android Studio* agar dapat diimplementasikan pada aplikasi Android.

**Kata kunci** — deteksi masker muka, deep learning, MobileNetV2, VGG16Net, google colaboratory, android

---

**Abstract**— *Coronavirus Disease 2019* (Covid-19) is an infectious disease caused by a newly of coronavirus, this disease can be transmitted quickly in droplets. So that everyone must maintain and protect themselves with various kinds of health protocols, one of which is wearing a face mask when leaving the house. In the new normal, using face masks is also an obligation. Therefore an application is needed to detect the use of face masks so that people are more disciplined and care about using face masks. Face mask detection is built using deep learning techniques. The architecture used is the type of MobileNetV2 and VGG16Net also using face detection with SSD ResNet and MTCNN. Model training and testing on the Google Colaboratory infrastructure. The results of the model training are then stored and converted into a TensorFlow Lite file which is then imported into the project in Android Studio, so that it can be implemented in the Android application.

**Keywords** — face mask detection, deep learning, MobileNetV2, VGG16Net, google colaboratory, android

---

## I. PENDAHULUAN

### A. Latar Belakang

*Coronavirus Disease 2019* (Covid-19) merupakan penyakit menular yang disebabkan oleh virus corona jenis baru, setidaknya ada dua jenis virus corona yang diketahui menyebabkan penyakit yang dapat menimbulkan gejala berat seperti *Middle East Respiratory Syndrome* (MERS) dan *Severe Acute Respiratory Syndrome* (SARS) [1]. Sehingga diketahui bahwa virus ini dapat menular begitu cepat. Ketika seseorang terkena infeksi Covid-19 maka akan mengalami beberapa gejala umum antara lain gejala

gangguan pernapasan akut seperti demam, batuk, dan sesak napas. Kasus ini diduga berhubungan dengan Pasar Ikan di Wuhan, China pada akhir Desember 2019 [2].

Berdasarkan studi epidemiologi dan virologi, membuktikan bahwa Covid-19 utamanya ditularkan dari orang yang bergejala (simptomatik) ke orang lain yang berada jarak dekat melalui *droplet* [2]. Penularan *droplet* terjadi ketika seseorang berada pada jarak yang dekat yaitu sekitar 1 (satu) meter dengan seseorang yang memiliki gejala pernapasan seperti batuk atau bersin. Sehingga *droplet* berisiko akan mengenai mulut dan hidung. Oleh karena itu,

diperlukan pencegahan Covid-19 dengan beberapa cara seperti memakai masker, menjaga jarak, rajin mencuci tangan, dan selalu menerapkan protokol kesehatan.

Salah satu keharusan menerapkan protokol kesehatan yaitu dengan menggunakan masker ketika keluar dari rumah. Penggunaan masker merupakan cara perlindungan diri dan untuk saat ini diwajibkan untuk semua orang, baik orang sehat maupun sakit. Orang sehat dapat menggunakan masker kain saat hendak keluar rumah, sedangkan bagi orang yang memiliki gejala infeksi pernapasan seperti batuk atau bersin, dicurigai memiliki gejala Covid-19, dan petugas kesehatan menggunakan masker bedah [3]. Namun, banyak sekali orang-orang yang masih mengabaikan penggunaan masker ketika memasuki sebuah tempat yang ramai. Maka diperlukan sebuah terobosan untuk deteksi masker pada suatu tempat seperti ketika memasuki lingkungan kampus, atau pun tempat-tempat umum lainnya agar penggunaan masker dapat lebih terkontrol dengan baik.

*Machine learning* adalah sebuah cabang ilmu kecerdasan buatan yang bertujuan untuk memungkinkan sebuah mesin dapat melakukan pekerjaannya dengan terampil dengan menggunakan perangkat lunak yang cerdas [4]. *Machine learning* nantinya dapat melakukan tugas-tugas tertentu dengan cara mempelajari algoritma dan model statistika yang ada. Pada *machine learning* terdapat tiga tipe yaitu *Supervised Learning*, *Unsupervised Learning* dan *Deep Learning*.

*Deep learning* merupakan sebuah kecerdasan buatan yang memiliki banyak lapisan tersembunyi untuk memproses data yang masih mentah [5]. Dasarnya *deep learning* menggunakan representasi yang sederhana tetapi dapat membangun konsep yang kompleks. Sehingga *deep learning* dapat dilatih untuk mendeteksi suatu objek serta klasifikasi objek. Terdapat beberapa algoritma yang dapat digunakan untuk melakukan pelatihan model sebuah *deep learning* misalnya untuk klasifikasi objek dengan *MobileNet*, *VGGNet*, dan sebagainya sedangkan untuk deteksi objek dapat menggunakan *YOLO*, *SSD ResNet*, *MTCNN*, dan sebagainya.

Membentuk suatu sistem pendeteksi objek dapat menggunakan dua kombinasi algoritma yaitu dengan melakukan klasifikasi kemudian mendeteksi objek tersebut. Maka perlu dilakukan teknik pelatihan untuk klasifikasi objek dengan menggunakan *Convolutional Neural Network* (CNN), yaitu sebuah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi [6]. Setelah

dilakukan pelatihan model dengan CNN maka selanjutnya model dikombinasikan dengan model untuk mendeteksi suatu objek. Ada banyak model yang dapat digunakan misalnya *Multi-task Cascaded Convolutional Neural Network* (MTCNN) dan modul *SSD ResNet*.

Perkembangan teknologi tentunya sangat cepat, ponsel cerdas menjadi bagian besar dari kehidupan sehari-hari bagi banyak orang di seluruh dunia. Berdasarkan survei [7] jumlah pengguna ponsel cerdas yaitu mencapai 3 miliar dan angka tersebut diperkirakan akan meningkat dalam beberapa tahun mendatang. Dengan menerapkan *deep learning* pada ponsel cerdas *android*, deteksi masker muka akan menjadi lebih mudah dan fleksibel yang mana dapat dilakukan oleh sebagian besar pengguna karena hanya dengan menggunakan sebuah perangkat *android*.

Sehingga berdasarkan yang telah dipaparkan sebelumnya, penulis akan melakukan penelitian tugas akhir dengan judul “PERANCANGAN APLIKASI ANDROID PENDETEKSI MASKER MUKA DENGAN TEKNIK DEEP LEARNING”.

## B. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian adalah sebagai berikut.

1. Bagaimana perancangan arsitektur *deep learning* untuk pendeteksi masker muka pada perangkat *Android*?
2. Bagaimana pengujian arsitektur *deep learning* untuk deteksi masker muka pada perangkat *Android*?

## C. Batasan Masalah

Agar penelitian tugas akhir mendapatkan hasil yang optimal maka permasalahan dibatasi sebagai berikut.

1. Metode *deep learning* yang digunakan untuk pendeteksi masker muka terdiri dari dua komponen yaitu algoritma pendeteksi muka dengan *SSD ResNet* dan *MTCNN*, serta algoritma klasifikasi wajah dengan CNN (*MobileNetV2* dan *VGG16Net*).
2. Program dibuat untuk membedakan pengguna memakai masker atau tidak memakai masker secara *realtime* menggunakan *Android*.
3. Program pendeteksi masker muka menggunakan bahasa pemrograman *Python* dengan antarmuka dan infrastruktur *Google Colaboratory*.

4. Pembuatan model *deep learning* pada deteksi masker muka dengan citra menggunakan *Framework Keras* dan *Tensorflow*.
5. Aplikasi *Android* untuk pendeteksi masker muka dibuat menggunakan *Android Studio* dengan bahasa pemrograman *Java*.

#### D. Tujuan

Dalam penelitian tugas akhir ini terdapat tujuan yaitu sebagai berikut.

1. Merancang arsitektur *deep learning* untuk pendeteksi masker muka dan penerapannya pada aplikasi *Android*.
2. Menguji arsitektur *deep learning* untuk deteksi masker muka dengan perangkat *Android*.

#### E. Manfaat

Manfaat yang diharapkan dalam pelaksanaan penelitian tugas akhir ini adalah sebagai berikut.

1. Mampu menerapkan ilmu yang didapat pada bangku perkuliahan untuk menyelesaikan tugas akhir.
2. Dapat membantu mencegah penyebaran Covid-19.
3. Mempermudah petugas dalam melakukan pemeriksaan penggunaan masker.

## II. TINJAUAN PERUSAHAAN

### A. Penelitian Terdahulu

Dalam penyusunan penelitian tugas akhir “Deteksi Penggunaan Masker dengan Metode *Convolutional Neural Network* (CNN) Berbasis *Android* sebagai Bentuk Pencegahan Covid-19” terdapat beberapa penelitian terdahulu yang berhubungan yaitu membahas klasifikasi obyek dengan citra menggunakan *machine learning*. Penelitian tersebut dapat dilihat sebagai berikut.

1. Penelitian I Wayan Suartika E. P., Arya Yudhi Wijaya, dan Rully Soelaiman dengan judul “Klasifikasi Citra Menggunakan *Convolutional Neural Network* (CNN) pada *Caltech 101*” [6]. Penelitian ini melakukan klasifikasi 5 kategori unggas yaitu *Emu*, *Flamingo*, *Ibis*, *Pigeon* dan *Rooster* dengan 390 citra dari data *Caltech 101* dan menggunakan 3 lapisan CNN, antara lain *convolutional layer*, *subsampling layer*, dan *fully connected layer*.
2. Penelitian Musakkarul Mu’min Lambacing dan Ferdiansyah dengan judul “Rancang Bangun *New Normal Covid-19 Masker Detektor* dengan Notifikasi Telegram Berbasis *Internet of Things*”

[8]. Penelitian ini melakukan pelatihan model dengan data dibagi menjadi 2 variabel yaitu *with masker* sebanyak 1915 citra dan *without masker* sebanyak 1918 citra. Kemudian menggunakan Raspberry Pi sebagai otak utamanya serta modul kamera juga PIR sensor sebagai deteksi orang dan maskernya. Selanjutnya data yang didapat akan dikirim melalui notifikasi Telegram.

3. Penelitian Mohamed Loey, Gunnasekaran Manogaran, Mohamed Hamed N. Taha dan Nour Eldeen M. Khalifa dengan judul “*A Hybrid Deep Transfer Learning Model with Machine Learning Methods For Face Mask Detection In The Era of The Covid-19 Pandemic*” [9]. Pada penelitian ini menggunakan dua model yaitu pertama dengan *transfer learning* arsitektur *ResNet50* yang digunakan untuk fase ekstraksi fitur. Kemudian menggunakan pembelajaran mesin tradisional seperti pohon keputusan, *Support Vector Machine* (SVM) dan algoritma ansambel yang digunakan untuk fase pelatihan, validasi dan pengujian. Dataset yang digunakan menggunakan tiga jenis yaitu wajah yang bermasker nyata, bermasker simulasi dan ketika berada di alam liar.
4. Penelitian Vidi Fitriansyah Hidarlan dengan judul “Rancang Bangun Klasifikasi Varietas Beras Berdasarkan Citra Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis *Android*” [10]. Pada penelitian ini membahas cara mengklasifikasikan 3 varietas beras yaitu Bashmati, IR64, dan Ketan dengan menggunakan arsitektur *VGG-16Net* dan *MobileNet* yang selanjutnya diaplikasikan pada *smartphone android*.

### B. *Coronavirus Disease 2019* (Covid-19)

*Coronavirus Disease 2019* (Covid-19) adalah penyakit menular yang disebabkan oleh *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2) [1]. Tanda dan gejala ketika terinfeksi antara lain gejala pernapasan akut seperti demam, batuk, dan sesak napas. Dengan kasus yang berat akan menyebabkan pneumonia, sindrom pernapasan akut, gagal ginjal dan bahkan kematian. Pertama kali mendapatkan laporan Covid-19 berasal dari Kota Wuhan, Provinsi Hubei, China pada 31 Desember 2019.

#### 1) Pencegahan dan Pengendalian Covid-19

Berdasarkan hal tersebut maka Pemerintah Indonesia mengeluarkan pedoman untuk melakukan pencegahan dan pengendalian penularan. Mengingat

cara penularannya berdasarkan infeksi droplet dari individu ke individu, maka dapat terjadi dimana saja dan kapan saja masuk ke dalam tubuh melalui hidung, mulut dan mata.

## 2) Pentingnya Memakai Masker

Seperti yang diketahui bahwa penularan virus corona dapat melalui droplet atau percikan ketika seseorang berbicara, batuk, bersin, atau sebagainya. Oleh karenanya, masker dibuat untuk melindungi diri dari droplet yang dikeluarkan oleh orang lain agar tidak masuk ke hidung dan mulut kita atau sebaliknya. Terdapat tiga jenis masker yang disarankan kepada masyarakat agar dapat memutus penyebaran virus corona [11], antara lain masker kain, masker bedah, dan masker N95.

## C. Deep Learning

Deep learning merupakan salah satu cabang dari ilmu pembelajaran mesin (*machine learning*) yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam [12]. Struktur dari pemodelan jaringan deep learning ditampilkan pada Gambar 1.

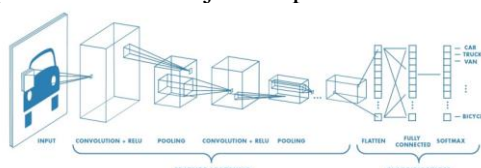


Gambar 1 Struktur Arsitektur Deep Learning

Terdapat dua istilah penting dalam pembangunan model yaitu pelatihan dan pengujian [12]. Pelatihan atau *training* adalah proses konstruksi model sedangkan pengujian atau *testing* merupakan proses menguji kinerja dari model hasil pembelajaran. Rasio (*training, testing*) yang dapat digunakan yaitu (90% : 10%), (80% : 20%), (75% : 25%), atau bahkan (50% : 50%).

## D. Convolutional Neural Network (CNN)

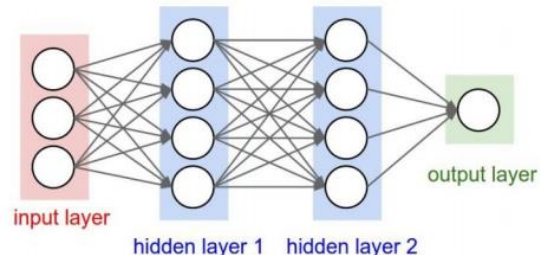
Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi [6]. CNN mampu mengenali informasi suatu objek seperti citra, teks, potongan suara dan sebagainya berupa data. Namun paling banyak pada pemrosesan citra. Alur proses CNN ditunjukkan pada Gambar 2.



Gambar 2 Alur Proses CNN

## 1) Konsep Convolutional Neural Network (CNN)

Cara kerja pada CNN memiliki kesamaan dengan MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi [6]. Gambar 3 akan menunjukkan sebuah arsitektur sederhana dari MLP.



Gambar 3 Arsitektur Sederhana MLP

## 2) Arsitektur Jaringan MobileNetV2

Arsitektur *MobileNetV2* merupakan sebuah arsitektur pengembangan dari arsitektur *MobileNetV1* yang rilis pada April 2017 [13]. *MobileNetV2* masih menggunakan *depthwise* dan *pointwise convolution*. Dengan menambahkan dua fitur baru yaitu *linear bottleneck* dan *shortcut connections* antar *bottlenecks*.

## 3) Arsitektur Jaringan VGG16Net

*VGG16* merupakan model jaringan saraf konvolusi yang diusulkan oleh K. Simonyan dan A. Zisserman dari Universitas Oxford dalam tulisan “*Very Deep Convolutional Networks for Large-Scale Image Recognition*” [14]. Pada *VGG16* memiliki 16 lapisan seperti namanya, dengan 4096-4096-4096 neuron pada *hidden layer* dan 1000 neuron pada *output layer*.

## E. Sistem Pengenalan Wajah

Pengenalan wajah adalah suatu metode yang berorientasi pada wajah. Pengenalan ini dibagi menjadi dua bagian yaitu dikenali atau tidak dikenali setelah itu kemudian dilakukan perbandingan dengan pola yang sebelumnya disimpan di dalam database [15]. Pengenalan wajah memiliki tiga tahap a) lokasi deteksi wajah, b) ekstraksi fitur, c) klasifikasi citra wajah [16].

## 1) Multi-task Cascade Convolutional Neural Network (MTCNN)

*Multi-task Cascaded Convolutional Neural Network* (MTCNN) merupakan sebuah model pengenalan wajah yang diperkenalkan oleh Kaipeng Zhang dkk pada tahun 2016 dalam tulisan “*Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks*” [17]. Susunan strukturnya

terdiri dari tiga tahap yaitu *fully convolutional network*, *on-maximum suppression* (NMS) dan selanjutnya ke CNN untuk penandaan wajah.

## 2) Modul DNN OpenCV

Dalam modul *Deep Neural Network* (DNN) OpenCV ini adalah *model caffe* yang didasarkan pada *Single Shot-Multibox Detector* (SSD) dan menggunakan arsitektur *ResNet-10* sebagai tulang punggungnya [18]. Untuk menggunakan modul DNN OpenCV dengan *model caffe* maka dibutuhkan *prototxt* yang mendefinisikan arsitektur model *ResNet10* dan *caffemodel* yang merupakan file yang berisi bobot untuk lapisan sebenarnya.

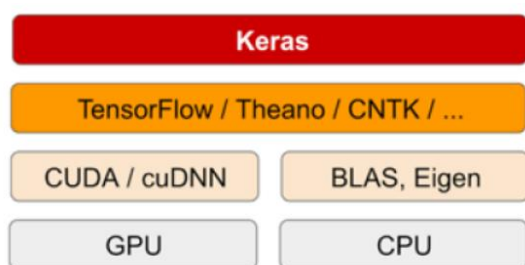
## F. Google Colaboratory

*Google Colaboratory* atau biasa disebut *Google Colab* merupakan salah satu produk *Google* berbasis *cloud* yang dapat digunakan secara gratis. Perbedaannya dengan produk *Google* lainnya yaitu pada *Google Colab* biasa diaplikasikan sebagai *coding environment* bahasa pemrograman *Python* dengan format “*notebook*” yang mirip dengan *Jupyter Notebook* [19].

## G. Framework TensorFlow dan Keras

*TensorFlow* merupakan sebuah *platform* sumber terbuka untuk *deep learning* [20]. Ini adalah sebuah *library* dengan API level tinggi. Dengan begitu *TensorFlow* dapat membantu dalam pembuatan *neural network* dalam skala yang besar. Sedangkan *Keras* adalah sebuah *framework deep learning* untuk *python* yang menyediakan cara mudah untuk mendefinisikan dan melatih hampir semua jenis model *deep learning* [21]. *Keras* awalnya dikembangkan dengan tujuan untuk mempercepat kemungkinan bereksperimen.

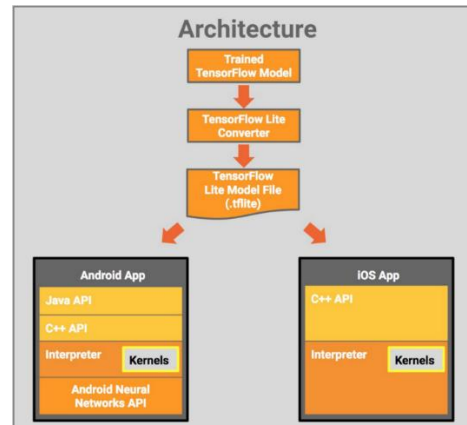
Berikut Gambar 4 susunan dari perangkat lunak dan perangkat keras *deep learning*.



**Gambar 4** Susunan Perangkat Lunak dan Perangkat Keras Deep Learning

## H. TensorFlow Lite

*TensorFlow Lite* adalah seperangkat alat untuk membantu *developer* menjalankan *TensorFlow* di perangkat seluler, tersemat, dan *Internet of Things* (IoT) [22]. Ini memungkinkan inferensi *machine learning* di perangkat dengan latensi rendah dan ukuran biner. . Arsitektur *TensorFlow Lite* itu sendiri ditunjukkan pada Gambar 5 [23].



**Gambar 5** Arsitektur TensorFlow Lite

## I. Android Studio

*Android Studio* adalah lingkungan pengembangan terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi *Android* yang didasarkan pada *IntelliJ IDEA* [24]. Selain sebagai editor kode dan fitur *developer IntelliJ* yang andal, *Android Studio* juga menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi *Android*.

# III. METODE PENELITIAN

## A. Waktu dan Tempat

Penelitian dilaksanakan dalam waktu 4 bulan mulai dari bulan Oktober 2020 sampai bulan Januari 2021 di Kampus Fakultas Teknik Universitas Jenderal Soedirman yang terletak di Km 5, Jalan Mayor Jenderal Sungkono, Desa Blater, Kabupaten Purbalingga, Jawa Tengah.

## B. Alat dan Bahan

Dalam penelitian ini, daftar alat dan bahan yang akan digunakan selama penelitian sebagai berikut.

### 1. Perangkat Keras

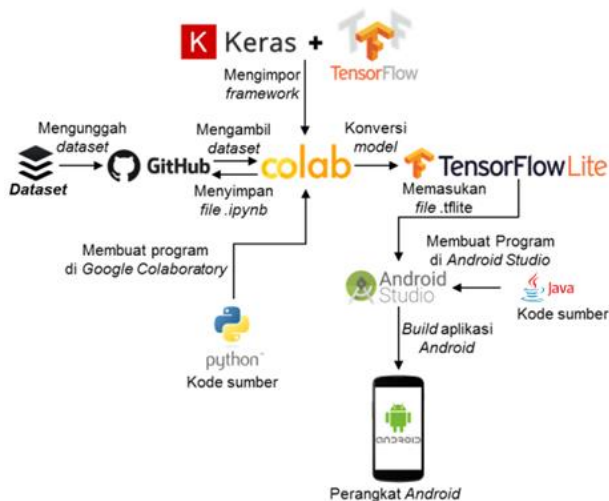
- a. Laptop Asus Vivobook A407MA dengan spesifikasi Intel Celeron N4000 dan RAM 4GB.



- b. Komputer virtual Google Colabulatory dengan spesifikasi GPU Nvidia Tesla T4, RAM 13 GB, Disk ~38GB.
  - c. Smartphone Android Mi A1.
2. Perangkat Lunak
- a. Sistem operasi Windows 10 64 bit
  - b. Peramban Internet Google Chrome versi 82.0.4240.111 64 bit
  - c. Google Colaboratory (Jupyter Notebook versi awan)
  - d. Android Studio 4.1
  - e. Sistem Operasi Android Pie
  - f. Layanan Repositori Web Development Github.
3. Dataset pelatihan dan pengujian berupa gambar orang menggunakan masker dan tidak menggunakan masker dari github/chandrikadeb7 “Face Mask Detection”.

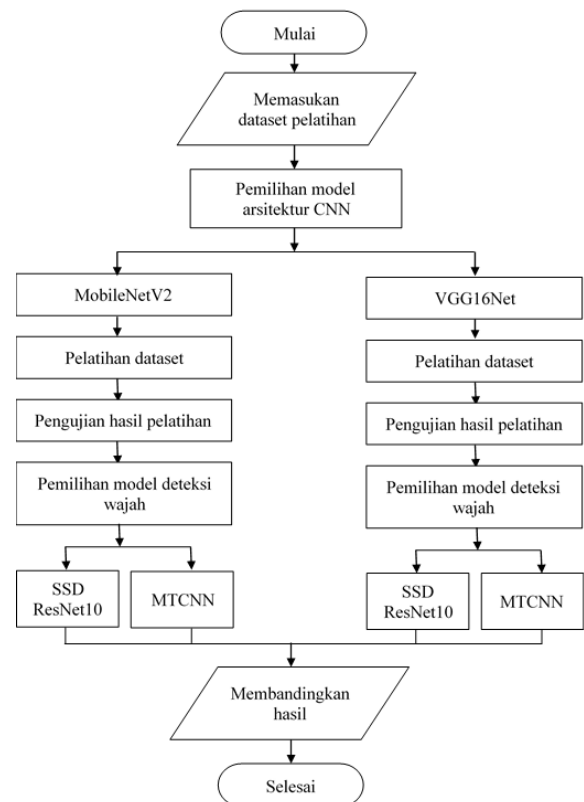
### C. Alur dan Tahap Penelitian

Penelitian dilaksanakan melalui beberapa tahapan yaitu tahap persiapan dengan studi literatur, tahap persiapan data dan *pre-proses dataset*, tahap desain arsitektur, tahap pengujian dan tahap evaluasi. Arsitektur sistem yang akan dibuat seperti yang disajikan pada Gambar 6 berikut.



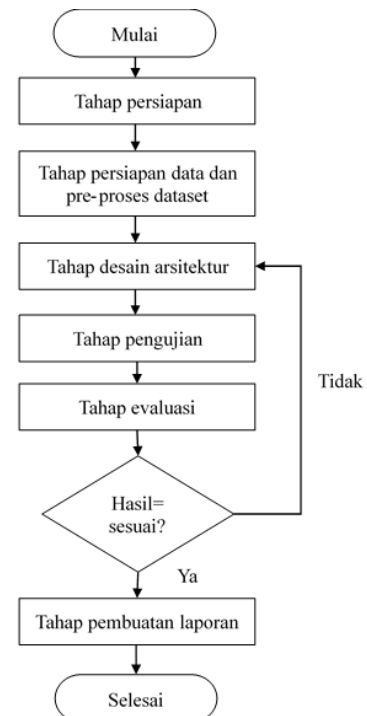
**Gambar 6** Desain Arsitektur Sistem

Sedangkan untuk diagram alir sistem penelitian yang akan dilakukan yaitu seperti yang disajikan pada Gambar 7.



**Gambar 7** Diagram Alir Sistem

Dalam penelitian ada beberapa tahapan-tahapan yang akan dilakukan yaitu seperti yang disajikan pada Gambar 8.



**Gambar 8** Alur Penelitian

### 3) Tahap Persiapan

Tahap pertama yaitu persiapan, dilakukan pembuatan pra-proposal, merumuskan masalah, dan mengidentifikasi masalah serta memahami permasalahan yang akan diangkat menjadi topik. Untuk mendukung kegiatan persiapan maka harus dilakukan sebuah studi pustaka seperti mengumpulkan referensi baik berupa jurnal, buku, artikel maupun referensi lain yang berkaitan dengan penelitian terdahulu.

### 4) Tahap Persiapan Data dan Pre-Proses Dataset

Pada tahap ini menyiapkan data yang dibutuhkan dan melakukan *pre-proses dataset* dengan cara mengumpulkan gambar orang yang menggunakan masker dan tidak menggunakan masker dan mempersiapkan model untuk deteksi wajah seperti *SSD ResNet10* dan *MTCNN*. *Dataset* pelatihan menggunakan *dataset* proyek *chandrikadeb7* dengan judul “*Face Mask Detection*” pada *Github*. Selanjutnya *dataset* tersebut dimasukkan ke *Google Colaboratory*.

### 5) Tahap Desain Arsitektur

Proses awal desain arsitektur dimulai dari membuat kode sumber untuk program *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 akan disimpan ke *Github*. Selanjutnya *dataset* pelatihan akan diambil dan disimpan ke dalam tempat penyimpanan sementara pada *Google Colaboratory* dari *Github*.

Sistem yang dirancang menggunakan arsitektur jaringan CNN yang bernama *MobileNetV2* dan *VGG16Net* dengan cara membuat model jaringan CNN yang sudah dipelajari sebelumnya (*pre-trained convnets*) pada *base model* dan *file* tersebut akan dilakukan *feature extraction*. Selanjutnya akan ditambahkan model arsitektur tambahan pada *head model*.

### 6) Tahap Pengujian

Ketika pembuatan arsitektur sudah terlaksanakan, maka dilakukan tahap pengujian. Seluruh dataset pelatihan dilatih terhadap masing-masing arsitektur jaringan CNN yang dibuat dengan panjang *epochs* atau iterasi yang ditentukan, misalnya 25 iterasi atau bahkan lebih. Setelah mencapai hasil prediksi yang memuaskan, akan dilakukan pengujian pada *Google Colaboratory* dan aplikasi *Android*. Pengujian yang

dilakukan pada *Google Colaboratory* dengan menggunakan bantuan pendeteksi wajah seperti *SSD ResNet10* dan *MTCNN* sedangkan pada *Android* dengan mengkonversi hasil model pelatihan ke dalam format *TensorFlow Lite “.tflite”* untuk selanjutnya digunakan pada *Android Studio*. File “.tflite” tersebut akan dimasukan dalam folder “*assets*” pada direktori *project Android Studio*.

Ukuran pengujian yaitu dilakukan pada citra berupa akurasi, presisi, *recall*, nilai F1 atau *harmonic mean* dari presisi dan *recall*, serta pengujian mAP (*mean Average Precision*). Sedangkan untuk video berupa pengujian *Frame Per Second* (FPS) atau berapa banyak gambar yang dihasilkan setiap detik. Selain itu terdapat beberapa variasi pengujian yang akan dilakukan seperti perbedaan jarak, tingkat pencahayaan, dan jenis masker muka yang digunakan.

### 7) Tahap Evaluasi

Tahap ini dilakukan untuk memperbaiki sistem jika terjadi kesalahan pada sistem aplikasi tersebut. Sehingga akan dihasilkan aplikasi yang baik dan layak digunakan untuk banyak orang. Jika tidak terjadi kesalahan dan aplikasi mudah digunakan oleh banyak orang, maka dapat disimpulkan aplikasi layak untuk digunakan.

## D. Jadwal Penelitian

Penelitian dilaksanakan dalam waktu 4 bulan mulai dari bulan Oktober 2020 sampai bulan Januari 2021 dengan rincian jadwal kegiatan sebagai berikut.

No.	Kegiatan	Oktober				November				Desember				Januari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi pustaka																
2.	Persiapan dan pre-processing																
3.	Desain arsitektur																
4.	Pengujian sistem																
5.	Evaluasi sistem																
6.	Pembuatan laporan																

## DAFTAR PUSTAKA

- [1] Kemenkes RI dan Direktorat Jenderal P2P, *Pedoman Pencegahan dan Pengendalian Coronavirus Disease (COVID-19) Rev Ke - 4*, vol. 4. 2020.
- [2] Kemenkes RI, *Pedoman Pencegahan dan Pengendalian Coronavirus Disease (COVID- 19) Rev Ke - 5*, vol. 5. 2020.
- [3] Kemenkes RI, “QnA : Pertanyaan dan Jawaban

- Terkait COVID-19,” 2020. [Online]. Available: <https://covid19.kemkes.go.id/qna-pertanyaan-dan-jawaban-terkait-covid-19/#.X6O9pYgzbIU>. [Accessed: 05-Nov-2020].
- [4] M. Mohammed, M. B. Khan, and E. B. M. Bashie, *Machine learning: Algorithms and Applications*, no. July. 2016.
- [5] Learners Evolve Machine, “Mengenal Machine Learning,” *medium.com*, 2019. [Online]. Available: <https://medium.com/evolve-machine-learners/mengenal-machine-learning-6c4a48db48b0>. [Accessed: 06-Nov-2020].
- [6] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [7] S. O’Dea, “Number of smartphone users from 2016 to 2021,” *statista*, 2020. [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. [Accessed: 06-Nov-2020].
- [8] M. M. Lambacing and F. Ferdiansyah, “Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet of Things,” *Dinamik*, vol. 25, no. 2, pp. 77–84, 2020, doi: 10.35315/dinamik.v25i2.8070.
- [9] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic,” *Meas. J. Int. Meas. Confed.*, vol. 167, no. May 2020, p. 108288, 2021, doi: 10.1016/j.measurement.2020.108288.
- [10] V. F. Hidarlan, “Rancang Bangun Klasifikasi Varietas Beras Berdasarkan Citra Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android,” pp. 1–17, 2020.
- [11] Wahyudi AMK, “PENTINGNYA MENGGUNAKAN MASKER DIMASA PANDEMI,” *RS Harapan Ibu*, 2020. [Online]. Available: <https://www.rsuharapanibu.co.id/pentingnya-menggunakan-masker-dimasa-pandemi/>. [Accessed: 10-Nov-2020].
- [12] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, vol. 1.4. 2020.
- [13] R. O. Ekoputris, “MobileNet: Deteksi Objek pada Platform Mobile,” *medium.com*, 2018. [Online]. Available: <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>. [Accessed: 09-Nov-2020].
- [14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [15] B. Arifin, “Aplikasi Sensor Passive Infra Red (PIR) Untuk Pendeteksian Makhluk Hidup Dalam Ruang,” *Pros. SNST ke-4*, no. 2011, pp. 39–44, 2013.
- [16] Derisma, “Faktor-Faktor yang Mempengaruhi Sistem Pengenalan Wajah Menggunakan Metode Eigenface pada Perangkat Mobile Berbasis Android,” *J. Politek. Caltex Riau*, vol. 2, no. 2, pp. 127–136, 2016.
- [17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.
- [18] V. Agarwal, “Face Detection Models: Which to Use and Why?,” *medium.com*, 2020. [Online]. Available: <https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c>. [Accessed: 11-Dec-2020].
- [19] R. Adam, “Mengenal Google Colab,” *Structimly*, 2019. [Online]. Available: <https://structilmy.com/2019/05/mengenal-google-colab/>. [Accessed: 09-Nov-2020].
- [20] A. Choudhury, “TensorFlow vs Keras: Which One Should You Choose,” 2019. [Online]. Available: <https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/>. [Accessed: 09-Nov-2020].
- [21] F. Chollet and J. J. Allaire, *Deep Learning with R*, 1st editio. United States of America: Manning Publications, 2018.
- [22] Creative Commons Attribution 4.0, “Panduan TensorFlow Lite,” *Google*, 2020. [Online]. Available: <https://www.tensorflow.org/lite/guide>. [Accessed: 10-Nov-2020].
- [23] L. Moroney, “Using TensorFlow Lite on Android,” *medium.com*, 2018. [Online]. Available: <https://medium.com/tensorflow/using-tensorflow-lite-on-android-9bbc9cb7d69d>. [Accessed: 10-Nov-2020].
- [24] Developers Android, “Mengenal Android Studio,” *Google Developer*, 2020. [Online]. Available: <https://developer.android.com/studio/intro?hl=id>. [Accessed: 10-Nov-2020].