# BAB II LANDASAN TEORI

## 2.1 *Coronavirus Disease 2019* (COVID-19)

*Coronavirus Disease 2019 (COVID-19)* adalah penyakit menular yang disebabkan oleh jenis virus *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2). Ini merupakan penemuan virus corona jenis baru yang belum pernah diidentifikasi sebelumnya pada manusia sebelum mulainya wabah di Wuhan, Tiongkok, bulan Desember 2019. Demam, batuk kering, dan rasa lelah merupakan gejala COVID-19 yang paling umum. Beberapa pasien mungkin juga mengalami gejala lain yang jarang seperti rasa nyeri dan sakit, hidung tersumbat, sakit kepala, konjungtivitis, sakit tenggorokan, diare, kehilangan indera rasa atau penciuman, ruam pada kulit, atau perubahan warna jari tangan atau kaki. Meskipun gejala-gejala ini muncul secara bertahap dan cenderung bersifat ringan, beberapa orang mungkin hanya mengalami gejala ringan meskipun telah terinfeksi.(*Pedoman Pencegahan Dan Pengendalian CORONAVIRUS DISEASE (COVID-19) Revisi Ke-5 - Protokol | Covid19.Go.Id*, n.d.)

### 2.1.1 Pencegahan dan Pengendalian COVID-19

Berdasarkan dalam buku pedoman pencegahan dan pengendalian COVID-19 (*Pedoman Pencegahan Dan Pengendalian CORONAVIRUS DISEASE (COVID-19) Revisi Ke-5 - Protokol | Covid19.Go.Id*, n.d., p. 16), Pemerintah Indonesia merilis panduan untuk mencegah dan mengendalikan penyebaran Covid-19. Virus dapat menyebar melalui droplet dari satu orang ke orang lain melalui hidung, mulut, atau mata, sehingga dapat terjadi di mana saja dan kapan saja. Oleh karena itu, beberapa tindakan dapat dilakukan untuk mencegah penyebaran virus Covid-19 sebagai berikut :

1. Menjaga kebersihan tangan dengan mencuci tangan menggunakan sabun dan air mengalir selama 40-60 detik, kemudian bilas dan keringkan dengan handuk bersih atau kertas sekali pakai. Jika tidak memungkinkan untuk mencuci tangan, alternatif lainnya adalah menggunakan pembersih tangan berbasis alkohol (*Handsanitizer*) minimal 20-30 detik.
2. Menghindari penyebaran *droplet* dari bersin atau batuk, disarankan untuk menutup mulut dan hidung menggunakan tisu atau bagian dalam lengan atas. Kemudian, tisu yang digunakan harus dibuang ke tempat sampah yang tertutup dan tangan dicuci dengan sabun dan air mengalir atau menggunakan *handsanitizer*.
3. Menjaga jarak minimal 1 meter dengan orang lain agar terhindar terkena *droplet* dari orang-orang yang batuk atau bersin.
4. Menggunakan masker kain bila berada di luar rumah, setelah 4 jam dipakai, dan cuci hingga bersih setelah dipakai.
5. Disarankan menggunakan masker jika ingin berobat ke fasyankes.

### 2.1.2 Pentingnya Penggunaan Masker

(*Pedoman Pencegahan Dan Pengendalian CORONAVIRUS DISEASE (COVID-19) Revisi Ke-5 - Protokol | Covid19.Go.Id*, n.d., p. 110) Seperti yang diketahui, virus *corona* dapat menyebar melalui *droplet* atau percikan saat seseorang berbicara, batuk, bersin, dan lain sebagainya. Oleh karena itu, masker digunakan untuk melindungi diri dari *droplet* yang dikeluarkan oleh orang lain agar tidak masuk ke dalam hidung dan mulut kita, begitu juga sebaliknya. Ada tiga jenis masker yang direkomendasikan untuk digunakan oleh masyarakat agar dapat membantu memutuskan penyebaran virus *corona*, antara lain sebagai berikut :

1. Masker kain

Sesuai anjuran dalam pedoman Kementerian Kesehatan RI, masyarakat disarankan menggunakan masker kain ketika bepergian keluar rumah, misalnya seperti sedang bekerja atau keperluan lain. Masker kain dapat meminimalisir untuk mencegah penularan virus *corona*.

1. Masker bedah

Masker ini sering di jumpai di tenaga medis saat sedang bertugas karena memiliki tiga fungsi utama agar pencegahan lebih efektif dan hanya sekali pakai.

1. Masker N95

Harga masker ini lebih tinggi dibanding masker bedah karena kinerjanya yang sangat efektif dalam mencegah penyebaran virus corona. Selain dapat menghalangi percikan air liur, masker ini juga mampu menghambat partikel kecil di udara yang bisa membawa virus. Meski demikian, masker ini tidak disarankan untuk digunakan sehari-hari dan disarankan hanya untuk petugas medis yang terlibat secara langsung dengan pasien Covid-19.

## 2.2 Citra Digital

Dalam sebuah citra digital terdiri dari kumpulan piksel-piksel, setiap piksel dalam citra memiliki nilai intensitas yang mempresentasikan tingkat kecerahan atau warna pada lokasi piksel tersebut dan memiliki 2 dimensi f(x,y) yang dimana x dan y merupakan tingkat kecerahan suatu citra pada suatu titik (Gonzalez, 2008)

Dalam setiap titik citra dapat di tuliskan sebagai berikut:

0 < *f(x,y)* < ∞

Citra digital juga bisa di representasikan dalam sebuah *matrix* M x N, yang dimana M berupa jumlah baris dan N merupakan jumlah kolom citra. Nilai yang dapat dalam matriks disebut sebagai piksel yang mempresentasikan cahaya atau warna citra tersebut.

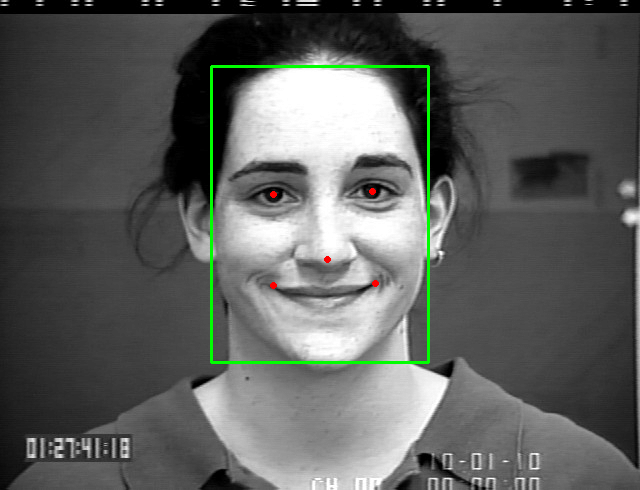
## 2.3 *Multi-Task Cascaded Convolutional Networks*

MTCNN (*Multi-Task Cascaded Convolutional Networks*) adalah sebuah model *deep learning* untuk pengenalan wajah dan deteksi *landmark* pada wajah yang dikembangkan oleh Zhang et al. dalam jurnal "*Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*".

Model MTCNN terdiri dari tiga tahap yang saling terkait: tahap deteksi wajah (*face detection*), tahap pencarian landmark (*landmark localization*), dan tahap penapisan wajah palsu (*face false positive filtering*). Setiap tahap menggunakan jaringan konvolusi yang terpisah untuk menghasilkan output yang semakin tepat dalam mengenali wajah.

Tahap pertama (deteksi wajah) menggunakan jaringan konvolusi untuk mencari kotak pembatas (*bounding box*) pada wajah dalam gambar. Tahap kedua (pencarian *landmark*) menggunakan jaringan konvolusi untuk menentukan posisi *landmark* pada wajah, seperti mata, hidung, dan mulut. Tahap ketiga (penapisan wajah palsu) kemudian menggunakan sebuah jaringan konvolusi lagi untuk memfilter wajah yang tidak relevan atau palsu.

Dengan menggabungkan tiga tahap ini, MTCNN dapat mengenali wajah dengan akurasi yang tinggi dan dapat mengatasi beberapa masalah seperti variasi posisi dan ukuran wajah dalam gambar. Model ini dapat digunakan dalam berbagai aplikasi seperti pengenalan wajah, deteksi emosi, pengenalan gender, dan masih banyak lagi.



Gambar 2.1 Deteksi dengan MTCNN

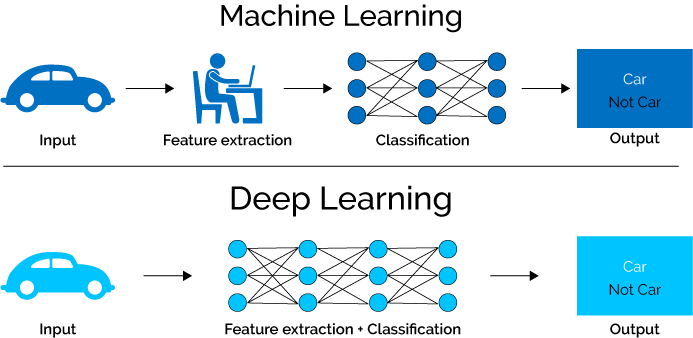
## 2.4 Sistem Pengenalan Wajah

Pengenalan wajah adalah teknologi komputer untuk mengenali dan menentukan lokasi wajah, ukuran wajah, deteksi fitur wajah, dan pengabaian citra latar, selanjutnya dilakukan mengidentifikasi citra wajah. Pengenalan wajah ini dibagi menjadi dua yaitu bagian dikenali dan tidak dikenali, dalam proses pengenalan wajah, citra dapat diambil dari jarak jauh tanpa menyentuh orang yang sedang diidentifikasi. (Bridga. 2016)

Ada banyak model yang dapat digunakan untuk pengenalan wajah terutama dalam penelitian ini akan menggunakan *Multi-Task Cascaded Convolutional Networks* (MTCNN).

## 2.5 *Deep Learning*

Mengatakan *Deep Learning* adalah salah satu metode pembelajaran mesin yang menggunakan jaringan saraf dengan banyak layer untuk memproses data dan menghasilkan output yang akurat. Jaringan saraf ini terdiri dari banyak neuron yang saling terhubung dan bekerja secara bersama-sama untuk mengekstraksi fitur dari data yang kompleks. Teknik ini telah digunakan dalam berbagai aplikasi, seperti pengenalan wajah, pengenalan suara, pemrosesan bahasa alami, dan lain-lain (Hao et al., 2016) .



Gambar 2.2 Struktur arsitektur *deep learning*

Menurut (Neapolitan & Jiang, 2018) Dalam konteks pembangunan model, ada dua istilah yang krusial, yaitu pelatihan dan pengujian. Pelatihan atau yang dikenal dengan istilah *training*, merupakan proses pembentukan model, sedangkan pengujian atau *testing*, adalah proses untuk mengevaluasi kinerja model yang telah dibangun. Data set yang digunakan dalam pelatihan biasanya terdiri dari kumpulan data, yang dapat berupa sampel data dalam bidang statistika atau citra. Secara umum, data set dapat dikelompokkan ke dalam tiga jenis yang saling tidak beririsan antara lain sebagai berikut:

1. *Training Set*

*Training set* merupakan perkumpulan banyaknya data yang akan digunakan pelatihan dalam membangun sebuah model.

1. *Development set*

*Development set* atau biasa disebut dengan *validation set* merupakan perkumpulan yang digunakan untuk melakukan *tuning* pada mesin yang sudah dilatih menggunakan *training set* , maka akan dilakukan validasi menggunakan *Validation set.* Sehingga akan memudahkan dalam proses generalisasi dan model mampu mengenali pola secara generik.

1. *Testing set*

*Testing set* merupakan kumpulan data yang digunakan untuk menguji model setelah proses pelatihan selesai seberapa akurat pada data yang pernah dilihat sebelumnya.

Pembagian *dataset* dapat bervariasi tergantung pada situasi yang ada. Biasanya, pembagian dataset (*training: validation: testing*) adalah (80%: 10%: 10%) atau (90%: 5%: 5%). Namun, jika dataset yang digunakan kecil, maka validation set mungkin tidak diperlukan sehingga dataset hanya dibagi menjadi training set dan testing set saja. Jenis pembagian ini disebut sebagai *closed testing*. *Rasio* yang dapat digunakan untuk pembagian dataset (*training, testing*) meliputi (90%: 10%), (80%: 20%), (75%: 25%), atau bahkan (50%: 50%).

## 2.6 *Convolutional Neural Network* (CNN)

(Putra, 2016) *Convolutional Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengelola data dua dimensi, seperti citra teks, potongan suara dan sebagainya yang berupa data. CNN dapat mengenali dan mengambil gambat input, menetapkan bobot dan bias yang dapat dipelajari untuk berbagai beberapa objek.

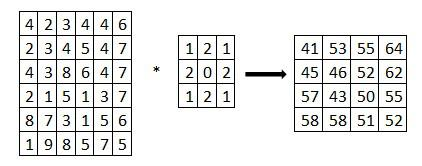


Gambar 2.3 Ilustrasi Arsitektur *Convolutional Neural Network*

### 2.6.1 Convolutional Layer

(Zufar et al., 2016) mengatakan bahwa *Convolutional Layer* merupakan lapisan yang pertama kali menerima masukan citra yang langsung pada arsitektur. Operasi pada lapisan ini sama dengan operasi konvolusi yaitu melakukan operasi kombinasi *linier filter* terhadap daerah lokal. Setiap filter yang akan mengalami operasi “dot” dari data awal dan nilai filter dalam tahapan awal filter setiap *citra* yang di *input* dengan matrix 6x6 lalu dimasukkan akan dikali dengan setiap kernel 3x3 untuk proses konvolusi. Citra input akan dibagi menjadi matriks 3x3, dan setiap matriks tersebut akan dikalikan dengan kernel yang tersedia. Hasil perkalian dari setiap matriks 3x3 dan kernel akan dijumlahkan, dan output dari penjumlahan tersebut akan ditempatkan di *cell* pada citra hasil baru.

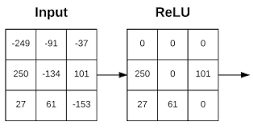
Proses ini ditunjukkan dalam Gambar 1.3, yang merupakan contoh dari layer konvolusi.



Gambar 2.4 Proses *convolutional layer*

### 2.6.2 *Rectifield Linear Unit*

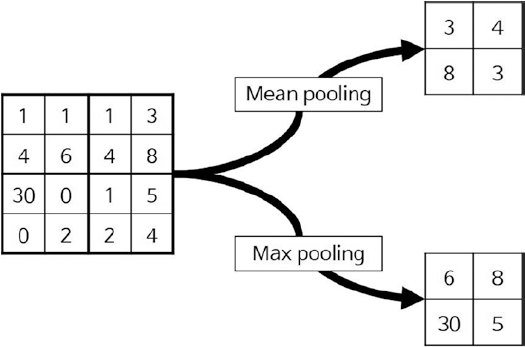
Aktivasi ReLu (*Rectifield Linear Unit)* adalah fungsi aktivasi yang paling umum digunakan pada layer CNN dan jaringan saraf dalam pemrosesan gambar. Fungsi f(x)=max(0,x) ini menghasilkan nilai nol untuk semua nilai input yang negatif, dan menghasilkan nilai input itu sendiri untuk nilai input yang positif. Dengan kata lain, jika nilai input kurang dari atau sama dengan nol, maka nilai output akan nol, sedangkan jika nilai input lebih besar dari nol, maka nilai output akan sama dengan nilai input (Agarap, 2018).



Gambar 2.5 Contoh Penerapan Aktivasi ReLu

### 2.6.3 *Pooling Layer*

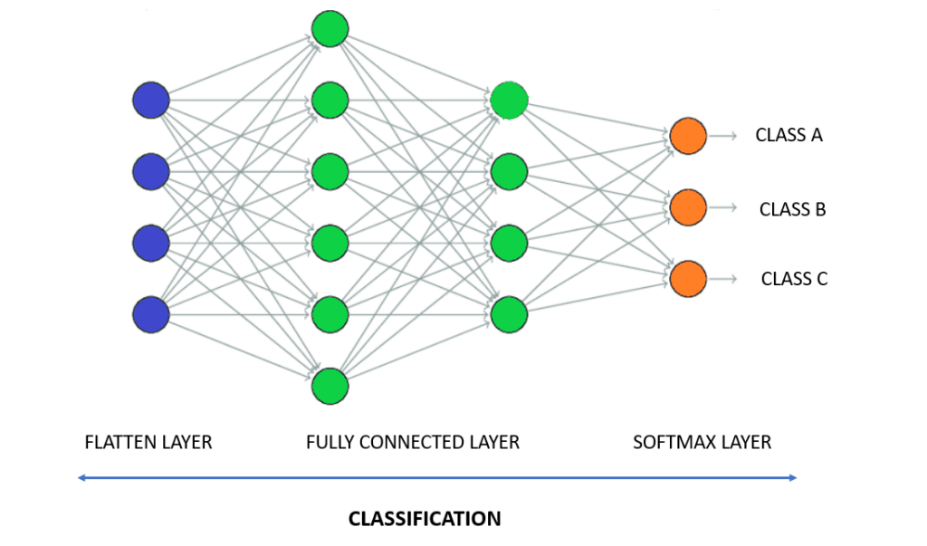
*Pooling Layer* merupakan salah satu tahapan proses dari *convolutional neural network* setelah dari tahap aktivasi biasa disebut *subsampling* atau *downsampling* yang bertujuan untuk mengurangi dimensi dari *feature map* tanpa mengurangi informasi penting di dalamnya. Langkah awal pada layer *pooling* adalah menentukan ukuran *downsampling* yang akan diterapkan pada feature map. Setelah itu, dilakukan proses *pooling* pada *feature map* tersebut. Tujuan dari *layer pooling* adalah mengurangi dimensi pada *feature map* sehingga dapat mempercepat proses komputasi karena jumlah parameter yang perlu diperbarui menjadi lebih sedikit (Gholamalinezhad & Khosravi, 2020). Proses *pooling* pada *feature map* dapat dijelaskan melalui gambar berikut.



Gambar 2.6 Matriks *feature map* 4x4 dengan proses *pooling* 2x2

### 2.6.4 *Fully Connected Layer*

*Layer* ini adalah *layer* akhir yang menerima hasil dari *layer pooling* sebagai input dan mengubah data dalam bentuk matriks x-dimensi menjadi *matriks linear* atau matriks 1 dimensi agar klasifikasi menjadi lebih mudah untuk dilakukan (Basha et al., 2019).

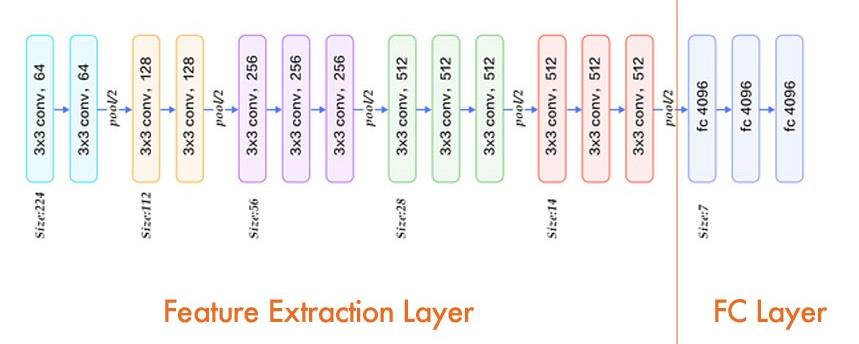


Gambar 2.7 *Fully Connected Layer*

### 2.6.5 Arsitektur Jaringan VGG16Net

Arsitektur Jaringan VGG16Net merupakan salah satu arsitektur jaringan *convolutional neural network* (CNN) yang dikembangkan oleh *Geometry Group* (VGG) dari University of Oxford. Arsitektur ini terdiri dari 16 *layer convolutional* dan layer-layer *fully connected*, sehingga memungkinkan VGG16Net untuk mengenali pola-pola kompleks dalam citra.

VGG16Net menggunakan konvolusi 3x3 dengan *stride* 1 dan *padding* sama untuk semua *layer convolutional*-nya, dan *max pooling* 2x2 dengan *stride* 2 setelah setiap dua layer *convolutional*. VGG16Net juga menggunakan *dropout* dan *regularization* untuk menghindari *overfitting*. VGG16Net memiliki kinerja yang sangat baik dalam pengenalan objek pada data set ImageNet, dengan menghasilkan tingkat akurasi yang lebih tinggi dibandingkan dengan arsitektur jaringan CNN lainnya.



Gambar 2.8 Arsitektur jaringan VGG16Net

VGG16Net yang disusulkan oleh Simoyan dan Zisserman dari Universitas Oxford dalam tulisan “*Very Deep Convolutional Networks for Large-Scale Image Recognition*” Arsitektur model VGG16 berhasil mencapai tingkat akurasi pengujian sebesar 92,7% pada data set ImageNet yang terdiri dari lebih dari 14 juta gambar dengan 1000 kelas. Model VGG16 termasuk salah satu model yang terkenal dan berhasil diajukan ke dalam kompetisi ILSVRC-2014.(Simonyan & Zisserman, 2014)

## 2.7 *Optimizer Adam*

Adam adalah sebuah algoritme optimasi gradien stokastik yang sangat popular di bidang *Deep learning*. Adam adalah singkatan dari “*Adaptive Moment Estimation*”. Adam sering digunakan untuk memperbaharui parameter model dalam pelatihan jaringan syarat yang beberapa kasus dapat memberikan konvergensi yang lebih cepat dan akurat dibanding algoritme optimisasi gradien stokastik lainnya.(Kingma & Ba, 2014)

## 2.8 *Keras*

Keras Merupakan *library* jaringan syaraf tiruan yang sangat populer dan tingkat tinggi yang dituliskan dalam bahasa python untuk membangun dan melatih sebuah model jaringan syaraf (*Neural Network*) Keras menyediakan antarmuka yang mudah digunakan untuk membangun model jaringan saraf yang kompleks dengan berbagai jenis lapisan (layer), seperti lapisan konvolusi, rekursif, dan *pooling*. Keras juga dapat bekerja dengan berbagai *backend* jaringan saraf seperti *TensorFlow, Theano*, dan *Microsoft* *Cognitive Toolkit* (CNTK). (Chollet, F., et al. 2015)

## 2.9 *TensorFlow*

*TensorFlow* merupakan *library open-source* untuk melakukan komputasi numerik dalam data yang berbentuk tensor. *Library* ini sangat banyak digunakan dalam pembelajaran mesin dan kecerdasan buatan dalam hal membangun sebuah model jaringan syaraf yang kompleks. *Tensorflow* menawarkan berbagai variasi *toolkit* yang memungkinkan untuk membangun model pada tingkat abstraksi yang kita inginkan(Nurfita & Ariyanto, 2018)

## 2.10 *Confusion Matrix*

Pengukuran performa *confusion matrix* adalah salah satu teknik evaluasi model klasifikasi yang digunakan untuk mengukur sejauh mana kinerja model dalam memprediksi label kelas dari suatu data. *Confusion matrix* atau disebut juga dengan *contingency table*, adalah tabel matriks yang menunjukkan jumlah prediksi yang benar atau salah oleh model (Yun, 2021)

Dalam menggunakan kinerja Confusion Matrix terdapat 4 istilah sebagai hasil proses klasifikasi, istilah tersebut adalah :

1. true positive (tp)
2. false positive (fp)
3. false negative (fn)
4. true negative (tn)

Pada jenis klasifikasi yang hanya memiliki 2 keluaran kelas seperti *WithMask* dan *WithoutMask* dapat disajikan pada tabel berikut:

**Tabel 2.1** *Confusion Matrix*

|  |  |  |
| --- | --- | --- |
| kelas | *withmask* | *withoutmask* |
| *withmask* | *true positive (tp)* | *false positive (fp)* |
| *withoutmask* | *false negative (fn)* | *true negative (tn)* |

Sehingga untuk mencari akurasi dalam suatu model berdasarkan tabel *confusion matrix* adalah sebagai berikut:

Akurasi *confusion matrix* juga dapat digunakan untuk mengitung presisi dan secara matematis dapat dirumuskan sebagai berikut:

Recall, secara definisi, mengukur seberapa banyak *True Positive* (TP) yang berhasil diidentifikasi oleh model, dibandingkan dengan total jumlah data yang sebenarnya positif. Secara matematis, recall dapat dituliskan dengan rumus sebagai berikut:

F1-Score adalah rata-rata harmonis antara presisi dan recall. Secara matematis, F1-Score dapat diungkapkan dengan rumus sebagai berikut: