

## BAB 2

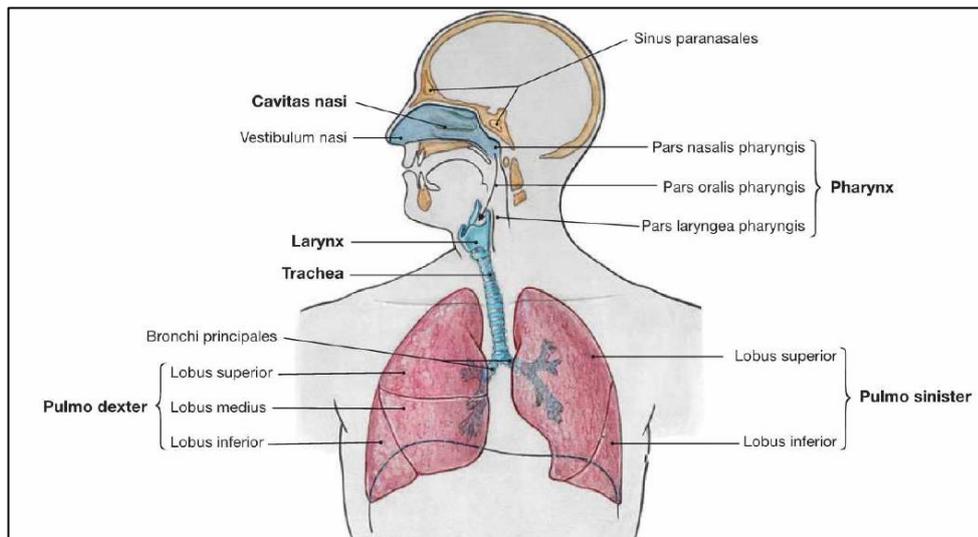
### LANDASAN TEORI

#### 2.1 Anatomi Paru

Paru-paru adalah salah satu organ yang termasuk ke dalam sistem pernapasan manusia. Paru-paru terbagi ke dalam dua bagian yaitu paru-paru kanan dengan tiga lobus dan paru-paru kiri dengan dua lobus. Paru kanan dipisahkan oleh Fissura obliqua dan Fissura horizontalis, sedangkan paru kiri dipisahkan oleh Fissura obliqua (Tomashefski *et al.*, 2008). Bagian kiri paru-paru lebih kecil dibandingkan kanan dikarenakan jantung membutuhkan ruang yang lebih di sisi kiri. Lapisan yang menyelubungi paru-paru adalah *pleura*. *Pleura* berfungsi sebagai pelumas paru-paru agar tidak mengiritasi saat paru-paru berkontraksi atau mengembang (Tomashefski *et al.*, 2008). Paru kanan memiliki volume 2-3 L saat inspirasi normal dan dapat mencapai 5-8 L saat inspirasi maksimal, sedangkan paru kanan memiliki volume yang lebih kecil 10-20% karena posisi jantung yang berada di kiri (Paulsen and Waschke, 2013).

Di dalam paru terdapat alveoli yang merupakan rongga yang dikelilingi oleh banyak kapiler yang berfungsi sebagai tempat pertukaran udara. Diameter alveoli dapat mencapai 150 sampai 500  $\mu\text{m}$ , dengan rata – rata sebesar 250  $\mu\text{m}$ . Terdapat sekitar 300 miliar alveoli pada pria dewasa dengan berat 70 kg dengan luas permukaan sebesar 143m<sup>2</sup> (Tomashefski *et al.*, 2008). Di atas paru terdapat cabang batang tenggorokan yang bercabang menjadi dua bagian. Cabang batang tenggorokan sebelah kanan lebih pendek, lebih lebar, dan lebih vertikal dari cabang batang tenggorokan kiri (Aung *et al.*, 2019). Sedangkan di bawah paru-paru

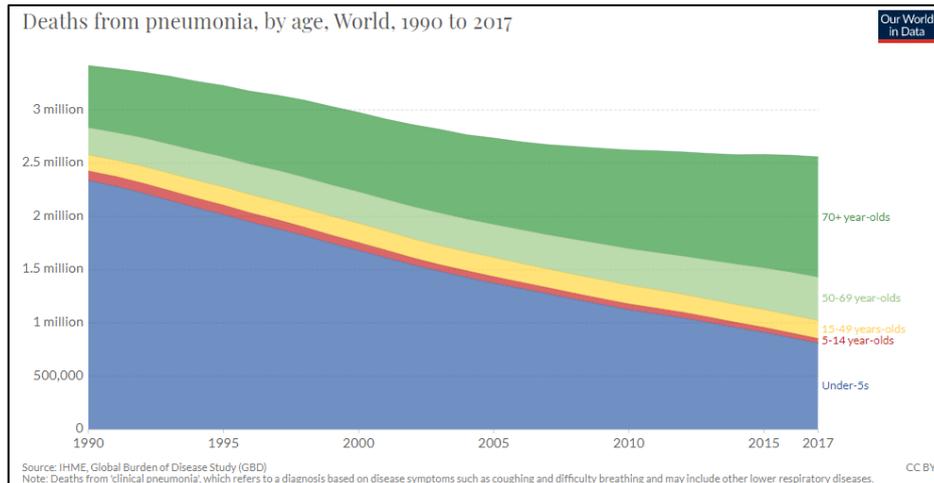
terdapat sebuah otot yang memisahkan dada dan perut, otot ini biasa disebut otot diafragma. Otot diafragma berfungsi dalam proses pernapasan, otot diafragma akan berkontraksi menjadi datar ketika seseorang menarik napas dan akan mengendur ketika menghembuskan napas.



Gambar 2.1 Anatomi Paru-paru (Paulsen and Waschke, 2013)

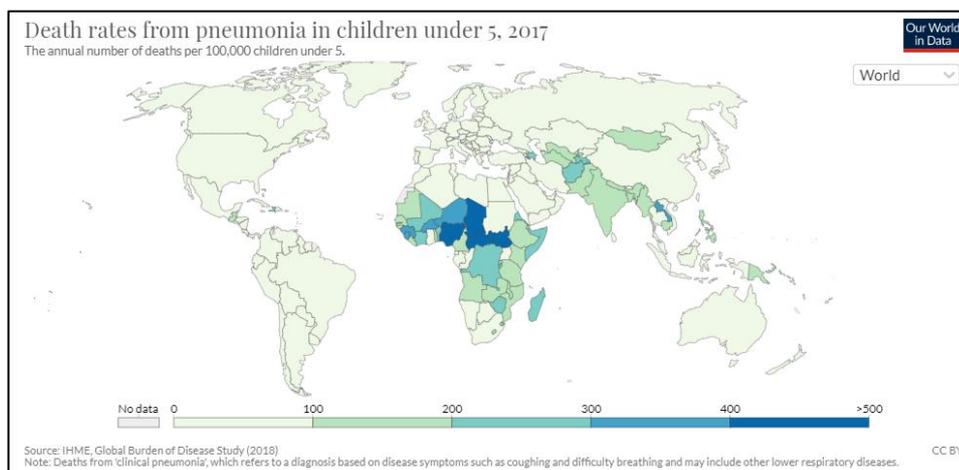
## 2.2 Pneumonia

Pneumonia atau radang paru-paru adalah penyakit infeksi yang menyerang paru, menyebabkan kantung udara di salah satu atau kedua paru-paru meradang dan membengkak. Akibatnya, penderita mengalami batuk berdahak, demam, sesak napas, dan menggigil. Penyebab pneumonia beragam, mulai dari virus, bakteri, dan jamur yang dapat menyebabkan pneumonia. Pneumonia dapat menyerang siapa pun. Kasus pneumonia terbanyak dialami oleh anak usia di bawah 5 tahun dan orang dewasa dengan usia lebih dari 70 tahun (Dadonaite and Roser, 2020). Pneumonia merupakan salah satu penyebab kematian pada anak kedua tertinggi di Indonesia (Anwar and Dharmayanti, 2014). Pada tahun 2017 setidaknya 15% anak meninggal karena pneumonia sedangkan 1.13 juta orang di atas 70 tahun meninggal dunia.



Gambar 2.2 Grafik Pneumonia Berdasarkan Umur (Dadonaite and Roser, 2020)

Kematian tertinggi akibat pneumonia berada di Afrika dan Asia Tenggara. Di Asia Tenggara, Filipina menduduki peringkat negara dengan kematian akibat pneumonia tertinggi. Kematian anak di bawah 5 tahun karena pneumonia tertinggi di seluruh Afrika dan Asia Tenggara berada di negara India, Nigeria, Pakistan, Republik Demokratik Kongo, dan Ethiopia (Dadonaite and Roser, 2020).



Gambar 2.3 Pemetaan Kematian Akibat Pneumonia untuk Anak di Bawah 5 Tahun (Dadonaite and Roser, 2020)

Gejala pneumonia sangat bervariasi mulai dari ringan hingga parah. Gejala ringan pneumonia sangat mirip dengan gejala flu atau pilek, namun berlangsung lama. Gejala lain yang dialami ketika seseorang terjangkit pneumonia adalah batuk kering atau berdahak dengan warna dahak yang kekuningan, hijau atau bahkan terkadang

terdapat bercak darah, kesulitan bernafas, detak jantung cepat, demam, berkeringat, menggigil dan nyeri pada dada (Lingkungan, 2011). Penyebab pneumonia sangat beragam dan dikelompokkan berdasarkan jenis kuman, sebagai berikut:

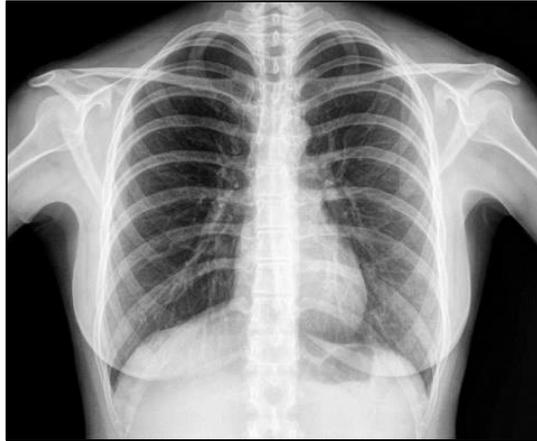
1. Pneumonia virus: Pneumonia jenis ini disebabkan oleh virus, seperti virus flue atau virus influenza serta coronavirus dapat menyebabkan penyakit ini. Pneumonia akibat coronavirus semakin meningkat semenjak pandemi Covid-19.
2. Pneumonia bakteri: Pneumonia jenis ini disebabkan oleh bakteri. Bakteri *Streptococcus pneumoniae* adalah bakteri yang sering menyebabkan pneumonia. Selain itu ada pula bakteri *Mycoplasma pneumoniae*, *Haemophilus influenzae*, dan *Legionella pneumophila*.
3. Pneumonia jamur: Pneumonia jenis ini disebabkan oleh jamur *Pneumocystis jirovecii*. Seseorang dengan sistem kekebalan tubuh yang rendah dapat terkena pneumonia yang disebabkan oleh jamur ini.

Pencegahan pneumonia dapat dilakukan melalui pemberian vaksin *Pneumococcal conjugate vaccine (PCV)* (Dadonaita and Roser, 2020), namun pemberian vaksin ini belum merata di seluruh dunia. Diagnosa terhadap pneumonia dapat dilihat dari gejala penderita, beberapa tes dapat dilakukan guna menyelidiki lebih lanjut penyakit pneumonia. Salah satunya adalah foto rontgen dada. Penyakit pneumonia dapat diidentifikasi dengan melihat adanya pergantian udara dalam alveoli oleh nanah, darah, sel dan cairan lainnya atau biasa disebut *consolidation*. *Consolidation* pada foto rontgen paru ditandai dengan adanya area berwarna abu-abu (Eldianto, 2019).

### 2.3 Foto Rontgen

Foto rontgen adalah pemeriksaan dengan menggunakan radiasi gelombang elektromagnetik untuk menampilkan gambar organ tubuh manusia. Foto rontgen digunakan untuk melihat kondisi organ tubuh atau mengamati perkembangan penyakit, dan mengetahui kemajuan dari pengobatan yang diberikan. Terdapat beberapa tipe dalam pemeriksaan rongga dada berdasarkan posisi tubuh. Untuk pemeriksaan tampilan depan paru dibagi menjadi dua jenis yaitu Posterior–anterior (PA), Anterior-posterior (AP). Sedangkan pemeriksaan tampilan samping biasa disebut lateral (Candemir and Antani, 2019).

Untuk mendapatkan citra PA, seseorang akan berdiri dengan posisi dada yang menempel pada mesin serta lengan yang direntangkan ke samping (Raof *et al.*, 2012). Sedangkan untuk mendapatkan citra AP, seseorang diharuskan berdiri atau terlentang di tempat tidur dengan punggung menempel pada mesin. Citra lateral digunakan untuk melengkapi tampilan depan paru-paru, sebab tampilan dari total volume paru secara keseluruhan pada citra lateral merupakan citra terbaik (Raof *et al.*, 2012). Gambaran dari organ tubuh seperti tulang atau besi ditampilkan dengan warna putih, sedangkan untuk organ yang memiliki ruang akan berwarna hitam. Contoh foto rontgen dapat dilihat pada Gambar 2.4.



Gambar 2.4 Foto Rontgen Dada  
(<https://alodokter.com>)

## 2.4 COVID-19

Virus *corona* atau *severe acute respiratory syndrome coronavirus 2* (SARS-CoV-2) atau dikenal sebagai Covid-19 merupakan virus dari keluarga *Coronaviridae* (Shereen *et al.*, 2020). Virus ini akan menyerang sistem pernapasan. Virus ini juga dapat menyebabkan gangguan pernapasan ringan hingga kematian yang disebabkan oleh kegagalan fungsi organ tubuh. Infeksi Covid-19 pertama kali ditemukan di kota Wuhan, China pada Desember 2019 dan kemudian menyebar ke hampir seluruh negara di dunia.

Covid-19 memiliki diameter 65 – 125 nm dan mengandung RNA untai tunggal (Shereen *et al.*, 2020). Selain Covid-19, virus ini juga termasuk ke dalam kelompok virus penyebab Severe Acute Respiratory Syndrome (SARS) dan Middle East Respiratory Syndrome (MERS). Jumlah kasus pada 14 April 2021 sejumlah 138.688.383 kasus dengan 2.978.935 kasus kematian di dunia. Kasus tertinggi pertama berada di negara Amerika sebanyak 59.053.949 kasus dengan 1.432.017 kasus kematian. Di urutan kedua, negara Eropa dengan 48.741.077 kasus dan 1.028.387 kasus kematian akibat Covid-19 (WHO, 2020a).

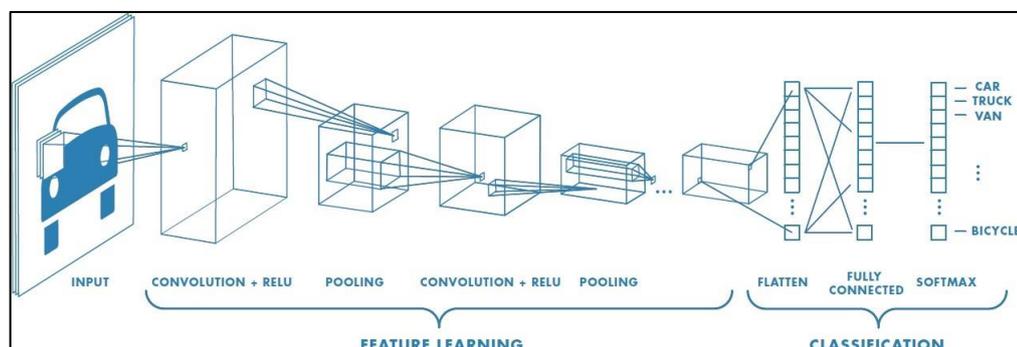
Gejala Covid-19 mirip dengan gejala flu yaitu pilek, sakit tenggorokan, demam, dan batuk kering. Namun gejala tersebut sulit hilang atau bahkan bertambah parah. Penderita yang mengalami gejala berat seperti demam tinggi, sesak napas, batuk berdahak, dan nyeri dada harus segera memeriksakan diri ke rumah sakit. Gejala berat tersebut dapat menyebabkan beberapa komplikasi seperti pneumonia, gagal ginjal, dan infeksi pada organ tubuh lainnya (Province *et al.*, 2020). Gejala ini muncul dalam waktu dua hari sampai dua minggu setelah terpapar covid-19. Virus ini menyebar lewat percikan ludah, batuk atau bersin (Phan *et al.*, 2020). Efek terpapar virus ini akan lebih berbahaya jika terjadi pada orang yang memiliki penyakit bawaan, ibu hamil, orang dengan usia lanjut, atau orang yang memiliki daya tahan tubuh rendah. Para tenaga kesehatan yang menangani Covid-19 diharuskan memakai alat perlindungan diri (APD). Untuk menentukan pasien terinfeksi Covid-19 dilakukan beberapa pengujian berupa *rapid test*, *swab test*, dan *CT scan*. Sampai sekarang Covid-19 masih belum dapat diobati, beberapa penelitian sedang mengembangkan vaksin Covid-19. Beberapa langkah yang dapat dilakukan untuk mencegah penularan Covid-19 antara lain:

1. Mencuci tangan dengan sabun dan air mengalir
2. Jika mengalami gejala Covid-19 lakukan isolasi mandiri dengan tinggal berpisah untuk sementara waktu
3. Gunakan masker saat beraktivitas di luar rumah terutama di tempat umum
4. Tidak menyentuh mata, hidung, dan mulut sebelum mencuci tangan atau ketika berada di tempat umum.
5. Tutup mulut dan hidung ketika bersin dan batuk dengan siku bagian dalam atau tisu, kemudian buang tisu atau masker ke tempat sampah

6. Jaga kebersihan benda sekitar dengan menggunakan *hand sanitizer* atau cairan desinfektan
7. Jaga jarak minimal 1m dan jangan ke luar rumah kecuali ada keperluan mendesak

## 2.5 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan neural network yang digunakan untuk mendeteksi atau mengenali objek pada sebuah *image*. Penelitian yang dilakukan oleh Hubel dan Wiesel menunjukkan bahwa sel saraf (neuron) yang terdapat di dalam otak mamalia tersusun menjadi sebuah *topographical map*, di mana setiap level nya fokus kepada karakteristik tertentu (Hubel and Wiesel, 1962). CNN bekerja dengan memanfaatkan proses konvolusi dengan menggerakkan kernel (filter) yang berukuran tertentu ke sebuah gambar, kemudian komputer mendapat informasi representatif dari hasil perkalian bagian gambar dengan filter yang digunakan (Eldianto, 2019). CNN terdiri dari beberapa tahapan. Secara garis besar arsitektur CNN terbagi menjadi dua bagian yaitu feature extraction layer dan fully connected layer. Ilustrasi arsitektur CNN dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur Convolutional Neural Network  
(<https://www.mathworks.com>)

Feature extraction berfungsi untuk mentranslasikan masukan menjadi *feature* berupa angka berdasarkan ciri dari masukan tersebut. Angka tersebut merepresentasikan masukan tersebut. Lapisan ini terdiri dari Convolutional dan Pooling Layer. Convolutional layer berfungsi untuk membuat *feature map* dengan cara menerapkan sebuah filter yang memindai seluruh masukan baik secara horizontal maupun vertikal. Proses ini biasa disebut operasi konvolusi (Indolia *et al.*, 2018). Operasi ini dapat dituliskan seperti

$$a_{ij} = \sigma((W * X)_{ij} + b) \quad (2.1)$$

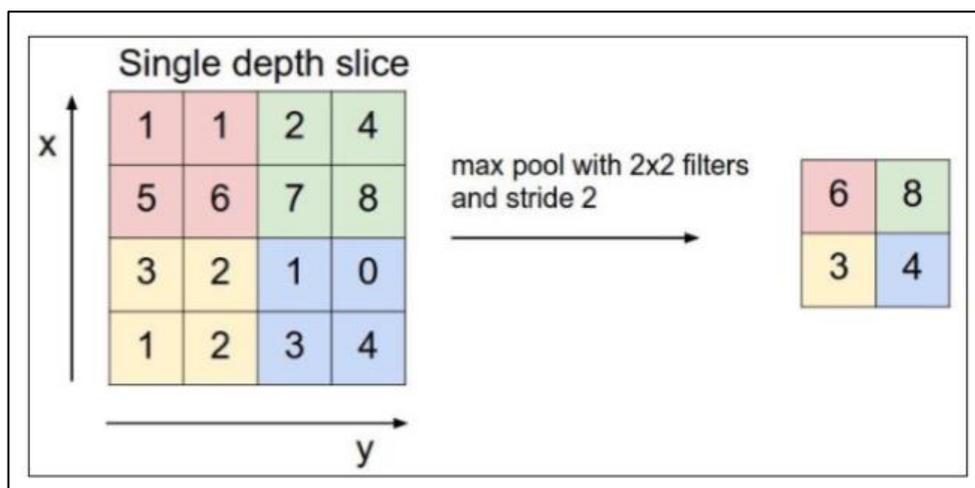
Di mana  $a_{ij}$  merupakan keluaran di lapisan berikutnya untuk lokasi (i,j) setelah menerapkan konvolusi. W merupakan kernel atau filter, sedangkan X merupakan masukan yang diberikan, dan b adalah bias. Operasi \* mewakili operasi konvolusi (Maggiori *et al.*, 2017). Untuk menghitung dimensi feature map dapat menggunakan rumus seperti di bawah ini:

$$O = \frac{w+N+2p}{s} + 1 \quad (2.2)$$

Berdasarkan persamaan di atas terdapat *stride*(s) dan *padding*(p). Stride merupakan parameter untuk menentukan jumlah pergeseran filter (Yamashita *et al.*, 2018). Sehingga convolutional filter akan bergeser sebanyak n pixel secara horizontal dan vertikal. Sedangkan padding merupakan parameter yang digunakan untuk menentukan jumlah pixel yang akan ditambahkan pada setiap sisi masukan (Yamashita *et al.*, 2018). Penggunaan padding berfungsi untuk mengukur dimensi keluaran agar tidak berkurang secara signifikan. Terdapat proses antara Convolutional layer dan Pooling layer yaitu fungsi aktivasi. Fungsi aktivasi yang sering digunakan adalah sigmoid atau ReLU karena memiliki sifat yang berfungsi lebih baik (Indolia *et al.*, 2018).

Lapisan selanjutnya adalah Pooling Layer. Pooling layer ini terdiri dari filter dengan stride dan ukuran tertentu. Penggunaan pooling bertujuan untuk mengurangi dimensi feature map sehingga mengurangi ukuran volume output dan perhitungan serta mengendalikan *overfitting*. Dalam pembuatan model CNN hal yang harus diperhatikan adalah memilih banyak jenis pooling, sehingga dapat menguntungkan kinerja model (Lee, Gallagher and Tu, 2016). Pooling yang sering digunakan adalah *average pooling* dan *max pooling* (Lee, Cheon and Kim, 2017). Average pooling yaitu pengambilan nilai rata-rata, sedangkan max pooling yaitu pengambilan nilai maksimal dari feature map.

Operasi *max pooling* dapat dilihat pada Gambar 2.6. Dimana dilakukan proses *max pooling* dengan *stride* 2 dan ukuran *filter* atau *kernel* sebesar 2x2. Posisi awal *filter* akan berada pada kiri atas kemudian *filter* akan mencari nilai terbesar. Selanjutnya *filter* akan bergerak sebanyak 2-pixel dan dilakukan kembali proses pengambilan nilai terbesar. Proses ini akan terus berlanjut dan berakhir ketika posisi *filter* berada di kanan bawah.



Gambar 2.6 Max Pooling  
(<https://medium.com>)

Tahapan selanjutnya adalah fully connected layer. Fully connected layer merupakan lapisan yang menghubungkan semua neuron dari lapisan sebelumnya dengan neuron di lapisan selanjutnya (Shamsaldin *et al.*, 2019). Pada lapisan sebelumnya setiap aktivitas akan diubah menjadi data satu dimensi kemudian dihubungkan ke semua neuron pada lapisan Fully Connected. Sebelum masuk ke tahap *fully connected* ini, perlu adanya proses flatten karena feature map yang dihasilkan pada *feature extraction layer* masih berbentuk multidimensional array, di mana proses ini mengubah multidimensional array menjadi sebuah vector agar dapat digunakan untuk proses selanjutnya (Yamashita *et al.*, 2018).

## 2.6 EfficientNet

Penelitian yang dilakukan oleh Tan dan Le (2019) membuat pengembangan baru tentang *neural networks*. Penelitian tersebut menjabarkan mengenai *model scaling* dan cara untuk mengidentifikasi hubungan antara depth, width, dan resolusi untuk mencapai performa yang lebih baik. *Scaling* biasanya digunakan untuk meningkatkan akurasi terhadap model yang telah dibuat. Terdapat tiga hal yang diperhatikan dalam *scaling* yaitu width, depth, dan resolution. Depth atau kedalaman adalah jumlah lapisan dalam jaringan, semakin dalam suatu jaringan maka dapat menyimpan informasi yang lebih banyak dan biasanya lebih akurat daripada model dengan jumlah lapisan yang sedikit. Sebagai contoh yaitu Residual Network (ResNet), ResNet-200 lebih dalam dan lebih akurat dibandingkan ResNet-18, namun terdapat masalah saat lapisan semakin dalam. Masalah tersebut antara lain data menjadi sulit dilatih dikarenakan hilangnya *gradient* kemudian pada jumlah lapisan tertentu akurasi menjadi menurun.

Sedangkan width adalah jumlah neuron di dalam lapisan atau jumlah filter dalam lapisan. Di dalam lapisan terdapat beberapa *channel* antara lain Red, Green, dan Blue. Jaringan yang memiliki lebih banyak *channel* di dalam lapisan dianggap lebih luas dibandingkan jaringan yang kecil. Namun tetap memiliki kelemahan yaitu sulit untuk mengambil fitur tingkat tinggi. Ketiga adalah resolusi, resolusi gambar yang semakin besar berarti memiliki lebih banyak informasi dibandingkan resolusi yang kecil. Resolusi gambar yang besar akan meningkatkan akurasi, tetapi membutuhkan pemrosesan yang lebih besar. Gambar yang memiliki resolusi 512x512 lebih banyak menampung informasi namun membutuhkan pemrosesan yang lebih besar dibandingkan dengan gambar yang memiliki resolusi 256x256.

Dengan permasalahan di atas Tan dan Le menemukan model *scaling* yang lebih baik, yaitu *compound scaling*. *Compound scaling* menggunakan koefisien *compound*  $\phi$  untuk mengukur skala *width*, *depth*, dan *resolution* secara bersamaan (Tan and Le, 2019).

$$\begin{aligned}
 \text{depth: } d &= \alpha^\phi \\
 \text{width: } w &= \beta^\phi \\
 \text{resolution: } r &= \gamma^\phi \\
 \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
 \alpha \geq 1, \beta \geq 1, \gamma \geq 1 & \tag{2.3}
 \end{aligned}$$

Di mana  $\phi$  merupakan koefisien yang telah ditentukan oleh pengguna, sedangkan  $\alpha, \beta, \gamma$  menentukan kedalaman, lebar, dan resolusi jaringan. Rumus 2.3 digunakan Tan dan Le untuk mencari *baseline* arsitektur. Kemudian untuk mengevaluasi metode ini, Tan dan Le menggunakan Neural Architecture Search (NAS) untuk mengoptimalkan akurasi dan FLOPs dan dinamakan EfficientNet-B0. Arsitektur ini mirip dengan MnasNet, yang membedakan adalah arsitektur

EfficientNet-B0 lebih besar dikarenakan target FLOPs yang besar (Tan and Le, 2019). Baseline arsitektur dapat dilihat pada Tabel 2.1.

Tabel 2.1 Baseline Arsitektur EfficientNet

| Stage<br>$i$ | Operator<br>$\hat{F}_i$ | Resolution<br>$\hat{H}_i \times \hat{W}_i$ | #Channels<br>$\hat{C}_i$ | # Layers<br>$\hat{L}_i$ |
|--------------|-------------------------|--|--------------------------|-------------------------|
| 1            | Conv 3x3                | 224x224                                    | 32                       | 1                       |
| 2            | MBCConv1, k3x3          | 112x112                                    | 16                       | 1                       |
| 3            | MBCConv6, k3x3          | 112x112                                    | 24                       | 2                       |
| 4            | MBCConv6, k5x5          | 56x56                                      | 40                       | 2                       |
| 5            | MBCConv6, k3x3          | 28x28                                      | 80                       | 3                       |
| 6            | MBCConv6, k5x5          | 14x14                                      | 112                      | 3                       |
| 7            | MBCConv6, k5x5          | 14x14                                      | 192                      | 4                       |
| 8            | MBCConv6, k3x3          | 7x7  | 320                      | 1                       |
| 9            | Conv1x1 & Pooling & FC  | 7x7  | 1280                     | 1                       |

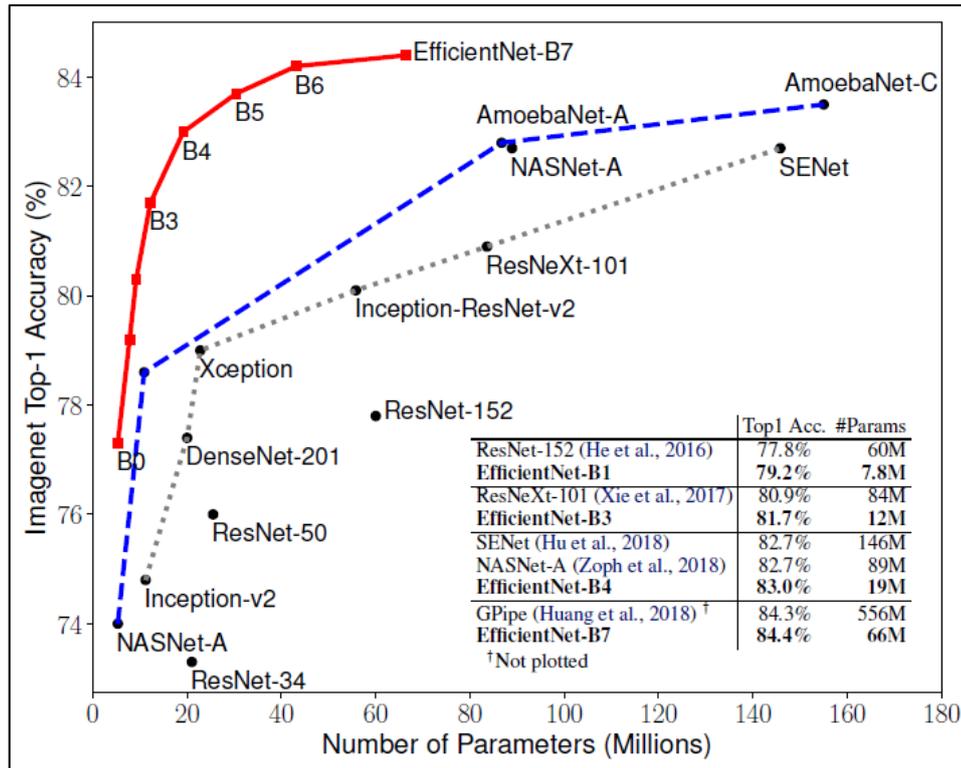
Untuk memperkecil pencarian dan mengurangi biaya operasi pencarian, Tan dan Le dengan cara menetapkan  $\phi = 1$ , dengan asumsi tersedia sumber daya dua kali lebih banyak dan dilakukan *small grid search* untuk  $\alpha, \beta, \gamma$ . Ditemukan bahwa nilai terbaik dari EfficientNet-B0 adalah  $\alpha = 1.2$ ,  $\beta = 1.1$ , dan  $\gamma = 1.15$ . Kemudian, tetapkan  $\alpha, \beta, \gamma$  sebagai konstanta dan lakukan eksperimen di mana nilai  $\phi$  adalah nilai yang berbeda. Dari penelitian tersebut dihasilkan EfficientNet-B1 sampai EfficientNet-B7. Hasil penelitian tersebut dan grafik performa EfficientNet dapat dilihat pada Tabel 2.2, Tabel 2.3 dan Gambar 2.7.

Tabel 2.2 Performa EfficientNet dengan Arsitektur Lain

| Model                            | Top-1 Acc | Top-5 Acc | #Params | Ratio-to-Efficient Net | #FLOPs | Ratio-to-Efficient Net |
|----------------------------------|-----------|-----------|---------|------------------------|--------|------------------------|
| <b>EfficientNet-B0</b>           | 77.1%     | 93.3%     | 5.3M    | 1x                     | 0.39B  | 1x                     |
| ResNet-50(He et al., 2016)       | 76.0%     | 93.0%     | 26M     | 4.9x                   | 4.1B   | 11x                    |
| DenseNet-169(Huang et al., 2017) | 76.2%     | 93.2%     | 14M     | 2.6x                   | 3.5B   | 8.9x                   |

Tabel 2.3 Performa EfficientNet dengan Arsitektur Lain (Lanjutan)

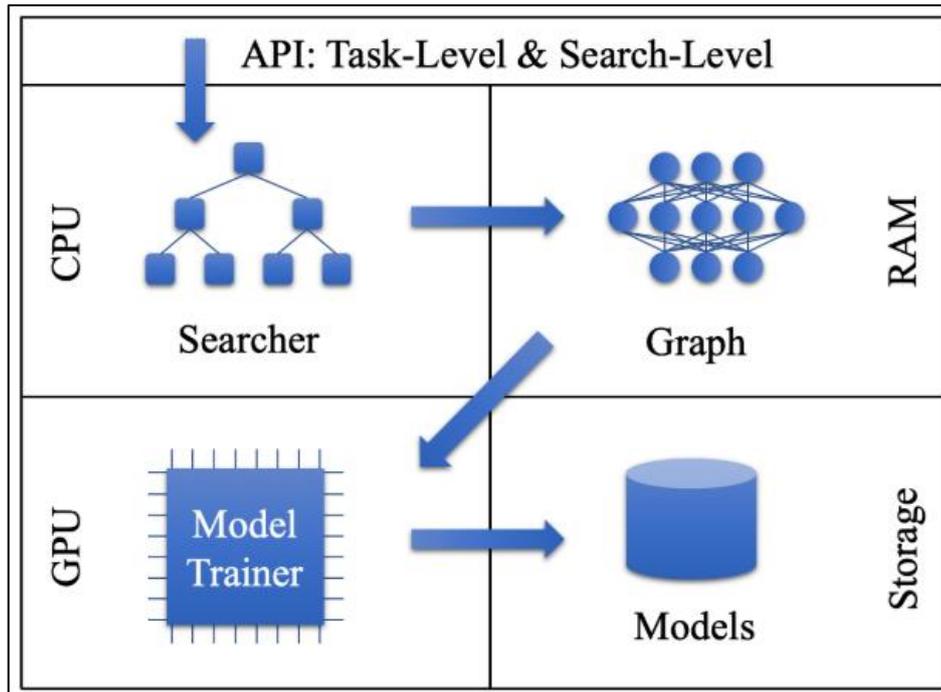
| Model                                    | Top-1 Acc | Top-5 Acc | #Params | Ratio-to-Efficient Net | #FLOPs | Ratio-to-Efficient Net |
|--|-----------|-----------|---------|------------------------|--------|------------------------|
| <b>EfficientNet-B1</b>                   | 79.1%     | 94.4%     | 7.8M    | 1x                     | 0.70B  | 1x                     |
| ResNet-152(He et al.,2016)               | 77.8%     | 93.8%     | 60M     | 7.6x                   | 11B    | 16x                    |
| DenseNet-264(Huang et al., 2017)         | 77.9%     | 93.9%     | 34M     | 4.3x                   | 6.0B   | 8.6x                   |
| Inception-v3(Szegedy et al.,2016)        | 78.8%     | 94.4%     | 24M     | 3.0x                   | 5.7B   | 8.1x                   |
| Xception(Chollet,2017)                   | 79.0%     | 94.5%     | 23M     | 3.0x                   | 8.4B   | 12x                    |
| <b>EfficientNet-B2</b>                   | 80.1%     | 94.9%     | 9.2M    | 1x                     | 1.0B   | 1x                     |
| Inception-v4(Szegedy et al.,2017)        | 80.0%     | 95.0%     | 48M     | 5.2x                   | 13B    | 13x                    |
| Inception-resnet-v2(Szegedy et al.,2017) | 80.1%     | 95.1%     | 56M     | 6.1x                   | 13B    | 13x                    |
| <b>EfficientNet-B3</b>                   | 81.6%     | 95.7%     | 12M     | 1x                     | 1.8B   | 1x                     |
| ResNeXt-101(Xie et al.,2017)             | 80.9%     | 95.6%     | 84M     | 7.0x                   | 32B    | 18x                    |
| PolyNet(Zhang et al.,2017)               | 81.3%     | 95.8%     | 92M     | 7.7x                   | 35B    | 19x                    |
| <b>EfficientNet-B4</b>                   | 82.9%     | 96.4%     | 19M     | 1x                     | 4.2B   | 1x                     |
| SENet(Hu et al.,2018)                    | 82.7%     | 96.2%     | 146M    | 7.7x                   | 42B    | 10x                    |
| NASNet-A(Zoph et al.,2018)               | 82.7%     | 96.2%     | 89M     | 4.7x                   | 24B    | 5.7x                   |
| AmoebaNet-A(Real et al.,2019)            | 82.8%     | 96.1%     | 87M     | 4.6x                   | 23B    | 5.5x                   |
| PNASNet(Liu et al.,2018)                 | 82.9%     | 96.2%     | 86M     | 4.5x                   | 23B    | 6.0x                   |
| <b>EfficientNet-B5</b>                   | 83.6%     | 96.7%     | 30M     | 1x                     | 9.9B   | 1x                     |
| AmoebaNet-C(Cubuk et al.,2019)           | 83.5%     | 96.5%     | 155M    | 5.2x                   | 41B    | 4.1x                   |
| <b>EfficientNet-B6</b>                   | 84.0%     | 96.8%     | 43M     | 1x                     | 19B    | 1x                     |
| <b>EfficientNet-B7</b>                   | 84.3%     | 97.0%     | 66M     | 1x                     | 37B    | 1x                     |
| GPipe(Huang et al., 2018)                | 84.3%     | 97.0%     | 557M    | 8.4x                   | -      | -                      |



Gambar 2.7 FLOPS vs Akurasi ImageNet  
(Tan and Le, 2019)

## 2.7 AutoKeras

AutoKeras adalah hasil dari penelitian Data Lab di Universitas A&M Texas (Jin, Song and Hu, 2019). Tujuan AutoKeras dibangun yaitu untuk membantu para peneliti untuk menggunakan *machine learning* walaupun tidak memahami *machine learning* secara mendalam. AutoKeras dirancang dengan memanfaatkan sumber dari CPU dan GPU secara maksimal. Arsitektur AutoKeras dibagi ke dalam lima bagian yaitu API, CPU, RAM, GPU, dan Storage. Arsitektur ini dapat dilihat pada Gambar 2.8.

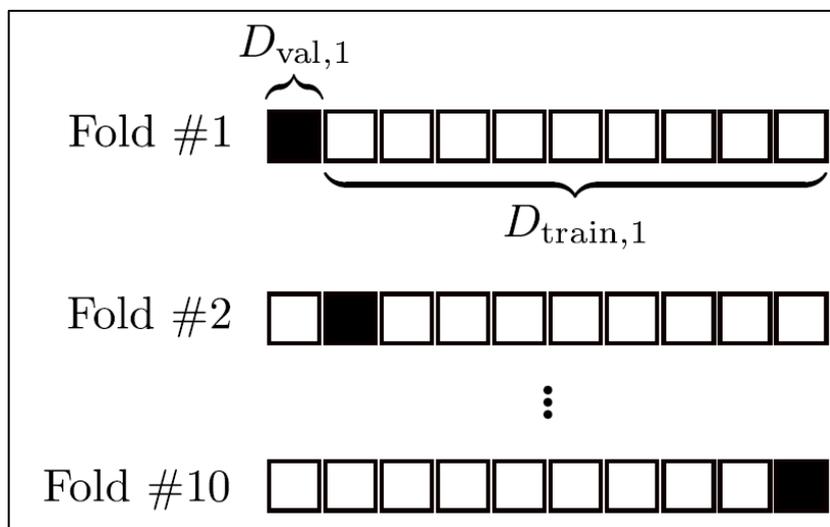


Gambar 2.8 Arsitektur AutoKeras  
(Jin, Song and Hu, 2019)

API berguna untuk melakukan *preprocessing* terhadap dataset. Hasil dari *preprocessing* data kemudian dikirim ke searcher yang berada di CPU, searcher merupakan modul yang berisi Bayesian Optimizer dan Gaussian Procces. Bayesian Optimizer pada searcher akan merancang arsitektur baru, kemudian arsitektur yang telah dirancang akan dibuat ke dalam arsitektur nyata dengan menggunakan graph modul yang terdapat di RAM. Arsitektur yang telah dibuat, kemudian dikirim ke GPU dan dilakukan training model. Hasil dari model yang telah dibangun kemudian disimpan ke dalam model storage, model storage berisi kumpulan model yang telah dilatih. Selama proses pencarian arsitektur dan training model, CPU dan GPU bekerja secara pararel guna meningkatkan efisiensi waktu dalam mencari model. Hasil dari performa model akan menjadi umpan balik untuk searcher, searcher akan melakukan perubahan dengan proses Gaussian.

## 2.8 K-Fold Cross Validation

*Cross Validation* merupakan sebuah metode yang bertujuan untuk memperoleh akurasi yang maksimal atau teknik pengambilan sampel secara berulang untuk memastikan model yang telah dibangun mendapatkan akurasi yang maksimal. *Cross Validation* hampir mirip dengan metode *random subsampling*, yang membedakan adalah dalam pengambilan sampel dilakukan dengan cara tidak membuat dua set pengujian tumpang tindih (Berrar, 2018). Dalam k-fold cross validation, *dataset* dibagi menjadi k bagian atau kelompok dengan pembagian data dilakukan secara acak serta jumlah iterasi yang akan dilakukan. Setiap bagian atau kelompok yang sudah ditentukan direpresentasikan sebagai *testing data*, sedangkan  $k - 1$  direpresentasikan sebagai *training data*. Prosedur tersebut akan terus diulang sampai k dan menghitung rata-rata jumlah hasil yang telah didapat tiap iterasi sebagai nilai akhir (Berrar, 2018).



Gambar 2.9 K-fold Cross Validation  
(Berrar, 2018)

## 2.9 Confusion Matrix

*Confusion Matrix* atau *error matrix* merupakan pengukuran yang populer dalam klasifikasi suatu masalah. *Confusion Matrix* pula dapat diterapkan pada *binary classification* dan klasifikasi yang menggunakan banyak kelas (Kulkarni, Chong and Batarseh, 2020). Baris dalam *confusion matrix* merepresentasikan kelas yang sebenarnya, sedangkan kolom dalam *confusion matrix* merepresentasikan kelas yang diprediksi.

Tabel 2.4 Confusion Matrix

|  | <b>Actual Values<br/>(Positive)</b> | <b>Actual Values<br/>(Negative)</b> |
|--|-------------------------------------|-------------------------------------|
| <b>Predicted Values<br/>(Positive)</b> | True Positive<br>(TP)               | False Positive<br>(FP)              |
| <b>Predicted Values<br/>(Negative)</b> | False Negative<br>(FN)              | True Negative<br>(TN)               |

Berdasarkan Tabel 2.4, terdapat 4 istilah pada *confusion matrix* yaitu True Positive (TP) yang merupakan data positif yang diprediksi sebagai data yang positif, False Positive (FP) merupakan data negatif yang diprediksi sebagai data positif, False Negative (FN) merupakan data positif yang diprediksi sebagai data negatif, dan True Negative (TN) yang merupakan data negatif yang diprediksi sebagai data yang negatif (Kulkarni, Chong and Batarseh, 2020).

*Confusion matrix* biasa digunakan untuk menghitung *performance metrics*. *Performance metrics* digunakan sebagai alat ukur dari kinerja model yang telah dibuat. *Performance metrics* yang biasa digunakan adalah akurasi, *precision*, *recall*, dan *f1-score*. Akurasi merupakan perhitungan antara jumlah data yang

diprediksi benar dan total semua data. *Precision* merupakan tingkat keakuratan model dalam memprediksi nilai yang benar, atau hasil dari perhitungan antara nilai data positif yang diprediksi benar dengan keseluruhan nilai yang diprediksi positif. *Recall* merupakan hasil dari perhitungan antara nilai data positif yang diprediksi benar dengan keseluruhan nilai yang benar positif. Sedangkan *F1-score* atau *f-value* adalah hasil dari gabungan perhitungan antara *recall* dan *precision* (Kulkarni, Chong and Batarseh, 2020). Persamaan 2.4, 2.5, 2.6, dan 2.7 merupakan rumus untuk akurasi, *precision*, *recall*, dan *f1-score*.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.4)$$

$$precision = \frac{TP}{TP+FP} \quad (2.5)$$

$$recall = \frac{TP}{TP+FN} \quad (2.6)$$

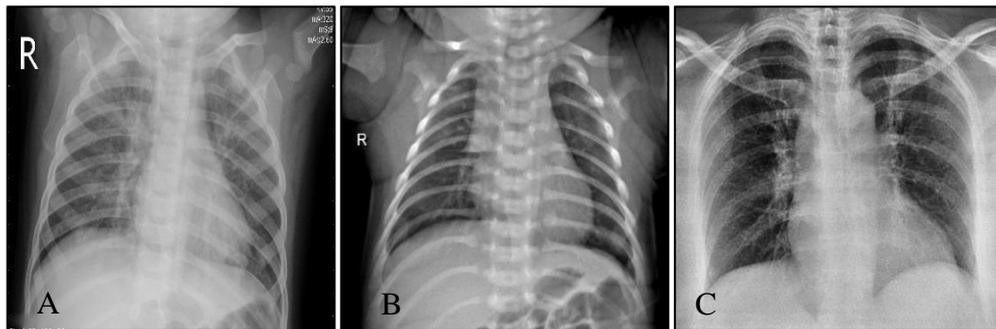
$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (2.7)$$

## 2.10 Dataset

*Dataset* yang digunakan adalah *dataset* yang telah dikumpulkan oleh Chowdhury, dkk (2020). Penelitian tersebut bertujuan untuk menguji model *deep learning* untuk mendeteksi Covid-19 berdasarkan foto rontgen dada sekaligus memaksimalkan akurasi dalam mendeteksi Covid-19. Foto rontgen yang digunakan sebagai dataset adalah foto posterior-anterior (PA) dan anterior-posterior (AP), gambar tersebut diambil dikarenakan banyak digunakan dalam diagnosa klinis.

Foto rontgen dada yang digunakan merupakan kumpulan dari berbagai sumber dengan rincian, *dataset* Covid-19 dikumpulkan dari berbagai sumber yang tersedia secara umum, *dataset* normal dan pneumonia dikumpulkan dari Kaggle database dan sumber lainnya (Chowdhury *et al.*, 2020). *Dataset* yang sudah

terkumpul dijadikan satu database yang digunakan untuk penelitian. Setidaknya terdapat 6 sumber *dataset* yang digunakan dalam mengumpulkan database foto rontgen dada. Database tersebut akan terus dikembangkan untuk kepentingan penelitian. Per 08 Februari 2021 setidaknya terdapat 1.341 gambar dengan label normal, 1.345 gambar dengan label pneumonia, dan 1.200 gambar Covid-19.



Gambar 2.10 Foto Rontgen Dada: Pneumonia (A), Normal (B), Covid-19 (C)  
(Chowdhury *et al.*, 2020)