

BAB II

DASAR TEORI

Pada bab 2 ini akan membahas terkait dengan kajian penelitian yang telah dilakukan sebelumnya dan dasar teori untuk menunjang penelitian ini.

2.1. KAJIAN PUSTAKA

Beberapa penelitian berkaitan dengan klasifikasi citra x-ray menggunakan teknik *computer vision* sudah pernah dilakukan. Metode yang digunakan juga bervariasi tergantung kebutuhan. Namun kebanyakan menggunakan metode *Convolutional Neural Network* (CNN) karena metode ini memang lebih maksimal dalam pengolahan citra. Selain menggunakan metode *deep learning*, beberapa penelitian juga menerapkan metode *machine learning* seperti *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN).

Pada penelitian Fahrizal et al (2021) yang berjudul “Klasifikasi Virus Pada Paru-Paru Dalam Gambar X-Ray Menggunakan *Convolution Neural Network*”[8], menggunakan metode CNN untuk mengklasifikasikan jenis penyakit paru-paru kedalam 3 kelas. Kelas-kelas ini dibedakan berdasarkan kondisi paru-paru, yaitu normal, pneumonia, dan covid-19 yang masing-masing terdiri atas 140 citra x-ray. Terdapat 2 jenis *optimizer* yang digunakan dalam penelitian ini, yaitu RMSprop dan Nadam. Hasil pengujian ini menunjukkan akurasi yang sangat baik, kondisi paru-paru normal menghasilkan akurasi sebesar 97%, paru-paru pneumonia menghasilkan akurasi 95%, dan paru-paru covid-19 menghasilkan akurasi maksimal, yaitu sebesar 100%.

Penelitian oleh Hariyani et al (2020) yang berjudul “Deteksi Penyakit Covid-19 Berdasarkan Citra X-Ray Menggunakan *Deep Residual Network*”[9], metode yang digunakan yaitu *deep learning* berbasis *deep residual network*. Penelitian tersebut mendeteksi infeksi virus covid-19 berdasarkan citra x-ray dada. Objek penelitian hanya terdiri atas dua kelas saja, yaitu normal dan covid-19 dengan total data sebanyak 339 citra. Untuk menambah variasi data peneliti juga melakukan proses augmentasi dengan cara rotasi, memanjangkan, melebarkan, dan memperbesar citra. Dalam penelitian ini menggunakan ResNet50 untuk *transfer learning* yang nantinya menghasilkan model baru. *Learning rate* yang digunakan

sebesar 0,001, dengan optimizer *Stochastic Gradient Descent* yang memiliki *weight* bernilai 0,0001 dengan *epoch* 30. Dengan 3 model yang dihasilkan dari proses *transfer learning*, model C mendapatkan nilai *precision*, *recall*, *F1 score*, dan akurasi yang cukup bagus. *Precision* bernilai 0,98, *recall* bernilai 0,95, *F1 score* bernilai 0,97, dan akurasinya sebesar 99%.

Ayumi dan Nurhaida (2021) dalam penelitiannya yang berjudul “Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan *Convolutional Neural Network*”[10], hanya fokus kepada dua kondisi atau kelas saja yaitu normal dan covid-19. Dataset yang digunakan sebanyak 1341 untuk paru-paru kondisi normal, dan 1200 untuk paru-paru terinfeksi covid-19. Penelitian ini menggunakan CNN dengan nilai *learning rate* sebesar 0,001 dan *epoch* sebanyak 50. Menghasilkan nilai *precision* sebesar 0,9650, *recall* sebesar 0,9650, *F1 score* sebesar 0,9650 dan nilai *cohens kappa* sebesar 0,9297. Hal ini menandakan bahwa model yang digunakan dapat mengenali dua kelas dengan sangat baik.

Naufal et al (2021) dalam penelitian yang berjudul “Analisis Perbandingan Algoritma Klasifikasi Citra Chest X-ray Untuk Deteksi Covid-19”[11], membandingkan beberapa algoritma klasifikasi untuk mendeteksi penyakit covid-19 pada paru-paru. Algoritma yang digunakan dalam penelitian ini adalah CNN, *K-Nearest Neighbor* (KNN), dan *Support Vector Machine* (SVM) yang tujuannya adalah untuk mengetahui *gap perform* dan waktu eksekusi yang dibutuhkan oleh ketiga algoritma tersebut. Dalam pengujian yang digunakan menggunakan 5 *cross validation* dengan algoritma CNN menghasilkan performa terbaik dengan nilai akurasi 0,9592, nilai *precision* 0,9592, nilai *recall* 0,959 dan nilai *F1 score* sebesar 0,959 dengan waktu rata-rata sebesar 3102,562 detik.

Hartato (2021) dalam penelitiannya berjudul “Penerapan *Convolutional Neural Network* pada Citra Rontgen Paru-Paru untuk Deteksi *SARS-CoV-2*”[12], menggunakan CNN yang terdiri atas 1 *Conv2D layer*, 2 *activation layer*, 1 *maxpooling layer*, 1 *dropout layer*, 1 *flatten layer*, dan 1 *dense layer*. Dataset yang digunakan adalah data *rontgen* yang terdiri atas 3 kelas, yaitu Normal dengan 1341 citra, Viral Pneumonia dengan 1345 citra, dan Covid-19 dengan 219 citra. Terdapat 4 skenario model penelitian dengan menggunakan platform Google Colab, model yang dibangun menghasilkan akurasi sebesar 98,69%, nilai sensitivitas mencapai

97,71%, dan spesifisitas mencapai 98,90%. Dari hasil yang didapatkan menunjukkan bahwa sistem yang dibuat mampu mendeteksi keadaan paru-paru berdasarkan citra *rontgen*.

Minarno et al (2021) dalam penelitiannya berjudul “Klasifikasi COVID-19 menggunakan *Filter Gabor* dan CNN dengan *Hyperparameter Tuning*”[13], menggunakan citra CT-Scan Covid dan Normal yang berjumlah sebanyak 2481 citra yang diambil dari rumah sakit Sao Paulo, Brazil. Metode yang digunakan dalam penelitian ini adalah *Filter Gabor* dan CNN dengan *Hyperparameter Tuning*. Pengujian dilakukan dengan 3 skenario berbeda, yaitu model usulan penulis, model usulan dengan *hyperparameter tuning*, dan model usulan dengan kombinasi *filter gabor* dan *hyperparameter tuning*. Skenario pertama menghasilkan nilai akurasi sebesar 95,1%, skenario kedua menghasilkan akurasi sebesar 96,4%, dan skenario ketiga menghasilkan akurasi mencapai 97,9%. Sehingga skenario terbaik adalah saat model ditambahkan *filter gabor* dan *hyperparameter tuning*. Namun dikatakan bahwa data citra yang digunakan masih kurang sehingga perlu mencoba teknik augmentasi lainnya untuk hasil klasifikasi yang lebih baik lagi.

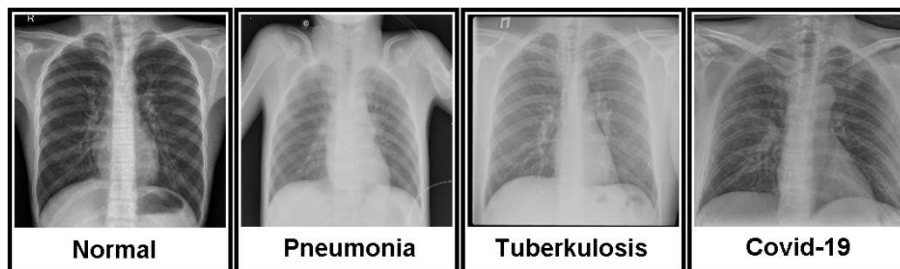
Penelitian lain oleh Yudistira et al (2020) yang berjudul “Deteksi Covid-19 Pada Citra Sinar-X Dada Menggunakan Deep Learning Yang Efisien”[14], menggunakan dataset sebanyak 1125 citra yang terdiri atas 3 kelas kondisi paru-paru, yaitu normal, pneumonia, dan covid-19. Metode yang digunakan penelitian ini adalah *deep learning* dengan CNN yang memiliki 4 model berbeda. Model pertama menggunakan Full CNN, model selanjutnya menggunakan parameter yang lebih kecil yaitu, ShuffleNet, EfficientNet, dan ResNet50 yang dibandingkan performanya. Hasilnya, model dengan ShuffleNet melakukan deteksi dengan cepat dengan membutuhkan waktu 0,0027 detik saja tetapi akurasi yang dihasilkan masih kurang, hanya sekitar 86,93% saja sama seperti model Full CNN. Sementara model dengan EfficientNet menghasilkan akurasi sebesar 87,5%, dan model dengan akurasi tertinggi adalah saat menggunakan ResNet50 yang mencapai akurasi 90,8%.

2.2. DASAR TEORI

Pada bagian dasar teori ini akan membahas tentang citra x-ray, pneumonia, tuberkulosis, covid-19, *deep learning*, *convolutional neural network*, model *sequential*, dan *confusion matrix*.

2.2.1. Citra X-Ray

Citra *X-Ray* merupakan hasil pemeriksaan dengan paparan radiasi sinar x dalam jumlah yang kecil terhadap tubuh untuk menghasilkan gambar bagian dalam tubuh manusia[7]. Hasil pemeriksaan ini menjadi salah satu pendukung bagi dokter dalam melakukan diagnosis terhadap keluhan pasien. Sehingga pembacaan hasil pemeriksaan harus dilakukan dengan tepat untuk menghindari hal-hal yang tidak diinginkan. Beberapa citra hasil pemeriksaan x-ray dengan kondisi normal, pneumonia, tuberkulosis, dan Covid-19 dapat dilihat pada gambar 2.1 dibawah ini.

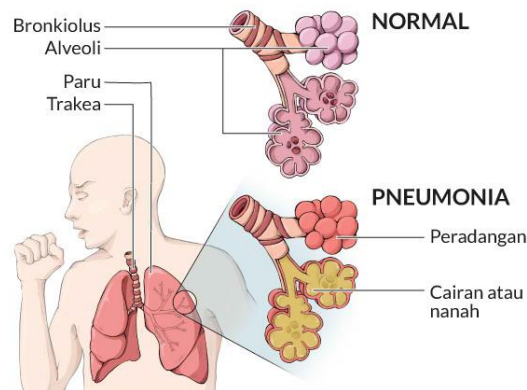


Gambar 2.1 Citra hasil pemeriksaan X-ray

Masing-masing kelas pada gambar 2.1 diatas memiliki perbedaan ciri yang dapat dilihat berdasarkan tingkat kekeruhan atau gelap terang (*opacity*). Pada citra normal, menunjukkan bahwa paru-paru berada dalam kondisi bersih tanpa kekeruhan. Kemudian pada citra pneumonia terlihat kekeruhan yang sangat tinggi pada bagian lobus fokal bagian atas. Pada citra tuberkulosis memiliki kekeruhan yang lebih tinggi pada kedua paru-paru. Sementara pada citra covid-19 terlihat kekeruhan dengan pola interstisial yang menyebar di kedua paru-paru[15]. Sehingga perbedaan ciri-ciri tersebut menjadi salah satu fitur ataupun pola yang akan dipelajari oleh sistem agar dapat memprediksi input gambar sesuai dengan kondisi atau kelas yang sebenarnya.

2.2.2. Pneumonia

Pneumonia (paru-paru basah) merupakan infeksi akut yang disebabkan oleh virus, jamur, bakteri yang menyerang jaringan paru-paru. Penderita penyakit ini kebanyakan adalah balita dan lansia[16]. Pasien pneumonia biasanya terdapat nanah dan cairan pada bagian alveolus, dapat dilihat juga pada gambar 2.2 dibawah ini.

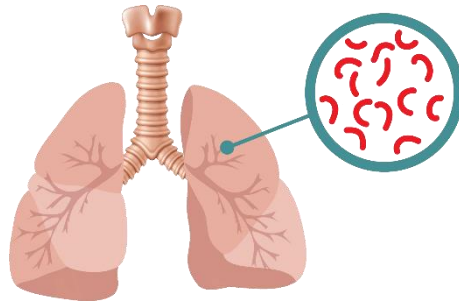


Gambar 2.2 Kondisi paru-paru terinfeksi pneumonia[17]

Kondisi ini menyebabkan pasien kesusahan bernapas dan semakin sedikit oksigen yang masuk ke paru-paru. Gejala penyakit ini cukup bervariasi, umumnya ditandai dengan batuk berdahak, demam, menggigil, sesak napas, nyeri dada ketika bernapas atau batuk, mual dan muntah, nafsu makan menghilang, serta tubuh yang mudah lelah.

2.2.3. Tuberkulosis (TBC)

TBC merupakan penyakit pada paru-paru yang disebabkan oleh infeksi bakteri bernama *Mycobacterium Tuberculosis*[6], diilustrasikan oleh gambar 2.3 dibawah. Penyakit ini merupakan penyakit yang bersifat menular dan termasuk kedalam 10 penyakit penyebab kematian di seluruh dunia. Apabila tidak cepat diatasi, penyakit ini akan berdampak buruk terhadap pasien yang menderitanya. Sejak dulu hingga kini, proses penanggulangan TBC di Indonesia dapat dikatakan belum mencapai maksimal[18].

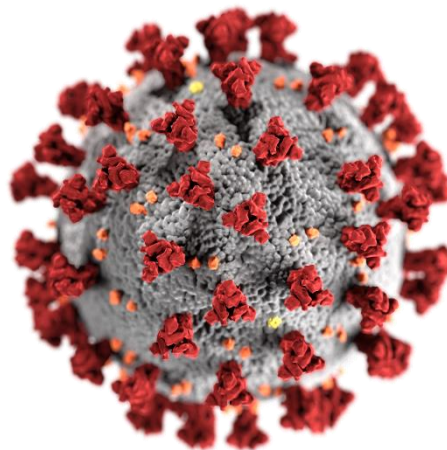


Gambar 2.3 *Mycobacterium Tuberculosis* menginfeksi paru-paru[19]

Penularan penyakit ini sangat cepat melalui udara, sehingga ketika seseorang tidak sengaja menghirup *droplet* (percikan cairan) saat seorang penderita TBC batuk dan bersin maka akan dengan sangat mudah terinfeksi. Gejala yang ditimbulkan pun beragam, seperti batuk disertai dahak dan darah yang berlangsung dalam waktu yang lama, demam, nyeri dada, serta berkeringat di malam hari.

2.2.4. Covid-19

Coronavirus disease (Covid-19) merupakan penyakit menular yang disebabkan oleh *Severe Acute Respiratory Syndrome-Coronavirus 2* (*SARS-Cov-2*)[20]. Bentuk dari virus tersebut dapat dilihat pada gambar 2.4 berikut ini.



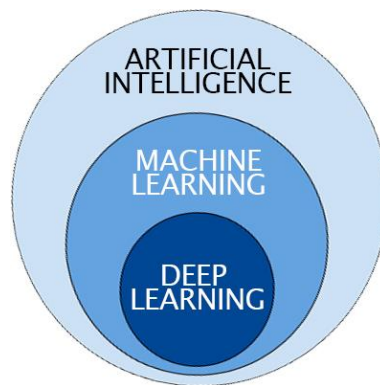
Gambar 2.4 Virus *SARS-Cov-2*[21]

Kasus pertama dilaporkan terjadi di kota Wuhan, Cina pada akhir 2019 dan mulai menyebar hingga ke seluruh dunia[4]. Kasus pertama di Indonesia

dilaporkan pada 2 Maret 2020 dan kasus positif selalu meningkat setiap harinya. Infeksi penyakit ini dapat menimbulkan gejala yang ringan, sedang, hingga berat. Bagi lansia dengan rekam medis yang kurang baik, gejala yang muncul bisa lebih parah daripada usia yang lebih muda. Penularan penyakit ini biasanya terjadi karena *droplet* secara langsung maupun yang menempel di permukaan benda. Hingga saat ini, varian dari virus tersebut sangat beragam, namun untuk penelitian ini citra paru-paru yang dapat dideteksi sebagai covid-19 hanya terbatas hingga ke varian delta saja.

2.2.5. *Deep Learning*

Deep Learning merupakan sebuah cabang dari *Artificial Intelligence* dan merupakan pengembangan dari *Artificial Neural Network* (ANN) namun dengan jaringan lebih kompleks[22]. *Deep learning* muncul ketika *Deep Belief Network* (DBN) membuat sebuah terobosan baru dengan menjadi model *handwritten digit recognition* yang paling akurat. *Deep learning* semakin populer seiring dengan diperkenalkannya algoritma *Convolutional Neural Network* (CNN)[23]. Selain kedua algoritma tersebut, *deep learning* juga memiliki algoritma lain seperti *Stacked Auto-Encoders*, *Recurrent Neural Network* (RNN) dan beberapa algoritma lainnya.



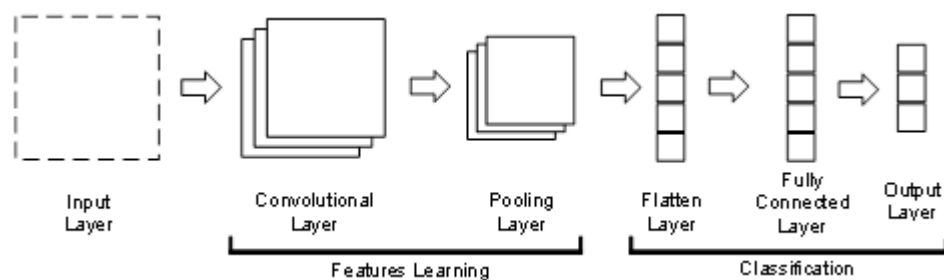
Gambar 2.5 Hubungan AI dengan *deep learning*[22]

Berdasarkan gambar 2.5 diatas, *deep learning* berada pada tingkatan yang lebih dalam dari pembelajaran mesin (*machine learning*). *Deep learning* sendiri dapat diterapkan untuk pengenalan gambar ataupun video, selain itu juga dapat

digunakan untuk mengenali suara manusia maupun suara objek atau benda. Selain itu juga dapat diterapkan pada *natural language processing* (NLP), dengan beberapa hasil yang dapat dirasakan saat ini seperti mesin penerjemah, *digital assistant*, mesin pencari, layanan *customer service* otomatis.

2.2.6. Convolutional Neural Network

Convolutional Neural Network adalah algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP)[22]. Pengembangan arsitektur ini dilakukan oleh satunya seorang peneliti asal Jepang bernama Kunihiro Fukushima pada tahun 1980, dan dinamakan sebagai NeoCognitron[24]. CNN sendiri memiliki arsitektur tertentu yang dapat dipadukan dengan berbagai trik seperti *dropout regularization*, pemanfaatan ReLU sebagai fungsi aktivasi, dan *data augmentation*. Sehingga algoritma ini mampu untuk mengklasifikasikan data gambar yang jumlahnya sangat besar[23]. CNN terdiri atas 2 bagian, yaitu *feature learning* dan *classification* yang masing-masing terdapat beberapa layer lagi didalamnya. Didalam bagian *feature learning* terdapat *convolutional* dan *pooling layer*, sedangkan pada *classification* terdapat *flatten*, *fully connected*, dan *output layer* seperti yang terlihat pada gambar 2.6 dibawah ini.



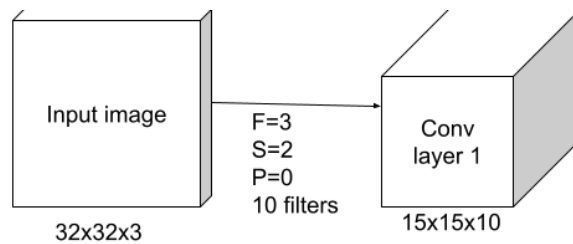
Gambar 2.6 Arsitektur CNN[25]

Pada setiap tahapan diatas, berlangsung proses-proses yang akan dijelaskan secara singkat dibawah ini.

1. Convolution Layer

Convolution merupakan proses pemberian filter terhadap input gambar yang dimasukkan. Filter yang digunakan adalah ukuran tinggi, lebar dan tebal

karena inputnya berupa gambar[25]. Nilai-nilai parameter filter tersebut dapat mempengaruhi hasil dari proses learning karena jika nilai yang diberikan semakin besar, maka parameter latihnya akan semakin besar juga. Hal tersebut akan berdampak terhadap lamanya waktu yang dibutuhkan sistem untuk melatih data yang ada. Gambaran proses ini dapat dilihat pada gambar 2.7 dibawah ini.



Gambar 2.7 Proses yang terjadi saat konvolusi[26]

Nilai *output* dari proses konvolusi juga dapat dihitung menggunakan persamaan matematis berikut ini[26].

$$Output = \left(\frac{(W-F+2P)}{s} + 1 \right) \quad \text{Pers. (2.1)}$$

Keterangan:

W = Panjang/lebar input

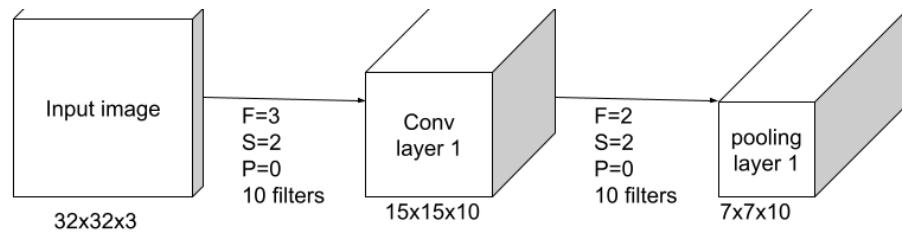
F= Panjang/lebar filter

P = Padding

S = Stride

2. Pooling Layer

Pada pooling layer terdapat proses *subsampling* yang dapat menggunakan *max pooling* ataupun *average pooling*. Dalam proses *pooling* ini, menggunakan keluaran *convolutional layer* sebagai input yang akan dibagi menjadi beberapa grid kecil lalu diambil nilai maksimalnya (*max pooling*) atau diambil nilai rata-rata (*average pooling*) untuk menyusun matriks citra yang telah direduksi[25]. Contoh proses *pooling* dapat dilihat pada gambar 2.8 berikut ini.



Gambar 2.8 Proses yang terjadi pada *pooling layer*[26]

Nilai *output* dari proses ini juga dapat dihitung menggunakan persamaan matematis berikut ini[26].

$$\mathbf{Output} = \left(\frac{W-F}{S} + 1 \right) \quad \text{Pers. (2.2)}$$

Keterangan:

W = Panjang/lebar input

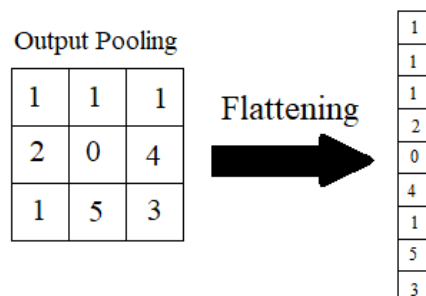
F= Panjang/lebar filter

P = Padding

S = Stride

3. *Flatten Layer*

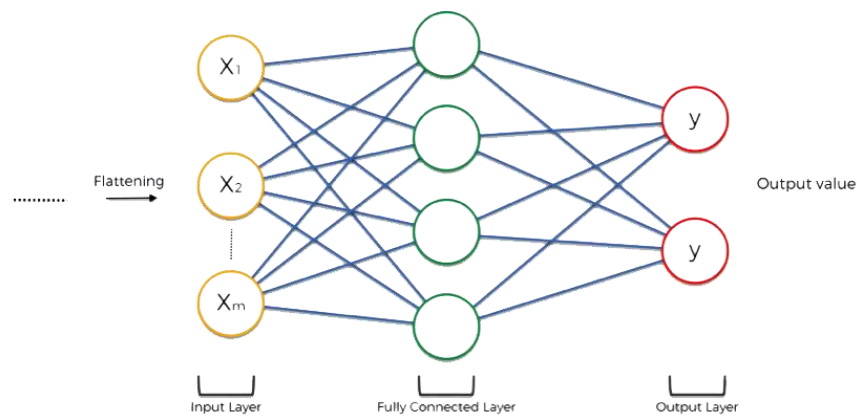
Proses *flatten* merupakan proses mengubah bentuk matriks dari *output pooling* menjadi vektor satu kolom saja atau satu dimensi. Vektor tersebut akan menjadi bagian *input* ANN untuk proses selanjutnya[27]. Cara merubah matriks tersebut adalah dengan mengambil tiap baris lalu mengubahnya menjadi 1 kolom, untuk lebih jelasnya dapat dilihat pada gambar 2.9 dibawah ini.



Gambar 2.9 Proses *flattening* terhadap *output pooling*[27]

4. *Fully Connected Layer*

Pada lapisan ini data satu dimensi dari proses sebelumnya digunakan sebagai *input* dan akan diolah sehingga dapat dilakukan klasifikasi terhadap data tersebut. Didalam lapisan ini terdapat neuron-neuron yang saling terhubung satu sama lain. Gambaran proses yang terjadi pada *layer* ini dapat dilihat pada gambar 2.10 berikut ini.



Gambar 2.10 Proses *Fully Connected Layer*[28]

2.2.7. *Model Sequential*

Model *Sequential* merupakan komposisi linier dari Keras layer yang mudah, minimal dan mampu mewakili hampir semua jaringan saraf yang ada. Model *Sequential* menampilkan kelas model untuk membuat model yang dapat disesuaikan kebutuhan penggunaannya. Pengguna bisa menggunakan konsep sub-kelas untuk membuat model kompleks secara individu[27]. *Sequential* atau sekuensial sendiri artinya adalah menggunakan *output* dari *layer* sebelumnya menjadi *input* untuk *layer* yang selanjutnya.

2.2.8. *Confusion Matrix*

Confusion Matrix merupakan sebuah alat bantu untuk melakukan analisis terhadap performa dari metode atau algoritma yang digunakan dalam proses klasifikasi[29]. *Confusion matrix* memiliki 4 istilah untuk menggambarkan hasil proses klasifikasi, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* seperti pada tabel 2.1 berikut.

Tabel 2.1 *Confusion Matrix*[25]

Class		Actual	
		Positive	Negative
Predicted	True	TP (<i>True Positive</i>)	TN (<i>True Negative</i>)
	False	FP (<i>False Positive</i>)	FN (<i>False Negative</i>)

TP dan TN digunakan memberikan informasi saat klasifikasi bernilai benar, sedangkan FP dan FN memberikan informasi saat hasil klasifikasi bernilai salah. Hasil dari *Confusion matrix* dapat digunakan untuk mencari nilai *precision*, *recall*, dan *accuracy*. *Precision* adalah kesesuaian antara data sebenarnya dengan hasil prediksi model. Nilai *precision* dapat dihitung menggunakan persamaan 2.1 dibawah. *Recall* adalah kemampuan model untuk menemukan kembali informasi yang pernah diterima sebelumnya. Nilai *recall* dapat dihitung menggunakan persamaan 2.2 dibawah. *Accuracy* adalah ketepatan sebuah model dalam mengklasifikasikan objek. Nilai *accuracy* dapat dihitung menggunakan persamaan 2.3 dibawah. Selain ketiga nilai tersebut, terdapat juga *F1-score* yang merupakan *harmonic mean* (rata-rata harmonik) dari *precision* dan *recall* yang dapat dihitung menggunakan persamaan 2.4 dibawah. *F1-score* merupakan indikator yang menunjukkan apakah model yang dilatih memiliki nilai *precision* dan *recall* yang baik atau tidak[25].

$$Precision = \frac{TP}{FP+TP} \times 100\% \quad \text{Pers. (2.3)}$$

$$Recall = \frac{TP}{FN+TP} \times 100\% \quad \text{Pers. (2.4)}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad \text{Pers. (2.5)}$$

$$F1\text{-score} = 2x \frac{Precision \times Recall}{Precision+Recall} \times 100\% \quad \text{Pers. (2.6)}$$