

MAKALAH

Deteksi COVID-19 pada Dataset X-Ray Dada dengan Metode
Capsule Neural Network (CapsNet)



Disusun oleh Tim **Yakuy 2** :

Ardacandra Subiantoro (18/427572/PA/18532)

Arief Pujo Arianto (18/430253/PA/18766)

Chrystian (18/430257/PA/18770)

**PROGRAM STUDI S1 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2020

Daftar Isi

1	Pendahuluan	2
1.1	Latar Belakang	2
1.2	Tujuan dan Manfaat	2
1.3	Batasan yang Digunakan	2
2	Metode Penambangan Data	4
2.1	Perangkat Uji Coba	4
2.2	Dataset	4
2.3	Metode Prapemrosesan	5
2.3.1	Konversi ke RGB	5
2.3.2	Resizing	5
2.3.3	Normalisasi	5
2.3.4	Konversi ke PyTorch Tensor	5
2.4	CapsNet	5
2.4.1	Introduksi CapsNet	5
2.4.2	Latar Belakang	6
2.4.3	Transformasi	6
2.4.4	Pooling	6
2.4.5	Capsules	7
2.4.6	Routing-by-agreement	7
2.4.7	Training	8
2.4.8	Kelebihan CapsNet	8
2.5	Evaluasi	8
3	Desain dan Implementasi Penambangan Data	10
3.1	Arsitektur CapsNet	11
4	Analisis	13
4.1	Analisis Exploratory Data Analysis	13
4.2	Analisis Hasil Model CapsNet	17
4.2.1	Analisis Metrik Penilaian	17
4.2.2	Hasil Rekonstruksi	20
5	Kesimpulan	22
6	Dokumentasi	22

1 Pendahuluan

1.1 Latar Belakang

Jumlah kasus COVID-19 terus meningkat secara eksponensial. Jumlah kasus total COVID-19 di seluruh dunia sudah mencapai lebih dari 19 juta kasus, dengan lebih dari 700 ribu kematian akibat COVID-19. Kasus COVID-19 di Indonesia sudah mencapai lebih dari 120 ribu dengan lebih dari 5 ribu kematian. Dampak negatif dari pandemi COVID-19 ini sangat terasa di Indonesia. Direktur Jenderal Pajak Kementerian Keuangan (Kemenkeu) Suryo Utomo membagi dampak pandemi COVID-19 menjadi tiga garis besar [11]. Dampak pertama adalah membuat konsumsi rumah tangga atau daya beli yang merupakan penopang 60 % terhadap ekonomi jatuh cukup dalam. Hal ini dibuktikan dengan data dari BPS yang mencatatkan bahwa konsumsi rumah tangga turun dari 5,02 % pada kuartal I 2019 ke 2,84 % pada kuartal I tahun ini. Dampak kedua yaitu pandemi menimbulkan adanya ketidakpastian yang berkepanjangan sehingga investasi ikut melemah dan berimplikasi pada terhentinya usaha. Dampak ketiga adalah seluruh dunia mengalami pelemahan ekonomi sehingga menyebabkan harga komoditas turun dan ekspor Indonesia ke beberapa negara juga terhenti.

Jumlah alat testing yang terbatas membuat tidak mungkin setiap pasien dengan penyakit pernafasan untuk dites dengan teknik konvensional (RT-PCR) untuk mendeteksi COVID-19. Jumlah ruang di rumah sakit, jumlah ventilator, dan jumlah alat protektif (PPE) bagi tenaga medis sangat terbatas, sehingga perlu digunakan dengan seefisien mungkin. Untuk meningkatkan efisiensi penggunaan alat medis, sangat penting untuk dapat membedakan pasien dengan *severe acute respiratory illness* (SARI) yang mungkin terinfeksi COVID-19. Oleh karena itu, kami mencoba menggunakan gambar X-Ray dada untuk mendeteksi infeksi COVID-19 dengan metode CapsNet. Penggunaan gambar X-Ray memiliki beberapa keuntungan dibandingkan tes konvensional [8] :

1. Pengambilan gambar X-Ray lebih banyak digunakan dan lebih murah dibandingkan tes konvensional.
2. Gambar X-Ray dapat dengan mudah ditransfer tanpa perlu ada alat transportasi dari tempat pengambilan gambar ke tempat analisis gambar, sehingga proses diagnostik dapat dilaksanakan dengan cepat.
3. Berbeda dengan CT Scan, terdapat alat-alat pengambil gambar X-Ray portable sehingga tes dapat dilakukan di ruangan-ruangan isolasi. Hal ini menyebabkan kurangnya penggunaan alat-alat protektif dan mengurangi risiko infeksi tersebar ke pasien-pasien lain di rumah sakit.

Dataset yang akan kami gunakan berasal dari dua sumber. Sumber pertama adalah Kaggle Chest X-Ray Images (Pneumonia) dataset (2018) milik Paul Mooney [9] yang berisi 5,863 gambar X-Ray dada yang dipilih dari pasien-pasien anak berumur satu sampai lima tahun dari Guangzhou Women and Children's Medical Center, Guangzhou. Paul Mooney mendapatkan data tersebut dari *paper* "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification" karya Kermany, D., Zhang, K., dan Goldbaum, M. [6]. Sumber kedua adalah dari *repository* Covid-19 Image Data Collection milik Cohen, J.P., Morrison, P., dan Dao, L. [1] yang berisi 454 gambar X-Ray dada pasien dengan penyakit paru-paru termasuk COVID-19 dan juga metadata yang mendeskripsikan informasi-informasi seperti jenis kelamin, umur, lokasi, dll dari pasien.

1.2 Tujuan dan Manfaat

Tujuan utama kami adalah untuk mengusulkan model berbasis *neural network* yang dapat secara akurat mendeteksi infeksi COVID-19 dari gambar X-Ray dada pasien. Model ini diharapkan dapat bertindak sebagai alat otomatis yang dapat memandu petugas kesehatan dalam mendiagnosis gambar X-Ray agar mengambil kesimpulan yang akurat. Manfaat yang diharapkan adalah peningkatan akurasi dan aksesibilitas tes deteksi infeksi COVID-19.

Perlu kami tekankan bahwa kami tidak mengusulkan penggunaan model ini untuk menggantikan metode tes diagnostik konvensional. Kami berharap bahwa model ini dapat bersifat saling melengkapi dengan tes konvensional, dimana hasil prediksi dari model dapat memastikan hasil dari tes konvensional atau menjadi metode alternatif bila tes konvensional tidak memungkinkan.

Tujuan kedua dari penambahan data yang kami lakukan adalah untuk mengeksplorasi metadata dari dataset yang tersedia melalui *Exploratory Data Analysis*. Kami mencoba untuk menelusuri apa pengaruh unsur-unsur seperti jenis kelamin dan umur terhadap infeksi COVID-19. Kami berharap informasi yang dapat kami ekstraksi dapat bermanfaat dalam upaya pendeteksi dan pengendalian kasus COVID-19.

1.3 Batasan yang Digunakan

Batasan masalah yang akan kami gunakan adalah sebagai berikut :

- Makalah ini akan membahas eksplorasi metadata dan penyusunan model untuk mendeteksi infeksi COVID-19 dari gambar X-Ray dada pasien[1].

- Dataset yang akan digunakan dibatasi pada Kaggle Chest X-Ray Images (Pneumonia) dataset (2018) milik Paul Mooney [9] dan *repository* Covid-19 Image Data Collection milik Cohen, J.P., Morrison, P, dan Dao, L. [1].
- Jenis penyakit yang akan diklasifikasi dibatasi pada COVID-19 dan Pneumonia saja.
- Metode penambangan data yang akan kami gunakan adalah Capsule Networks (CapsNets).

2 Metode Penambangan Data

2.1 Perangkat Uji Coba

Implementasi model kami dijalankan menggunakan dua *environment* Google Colaboratory. *Environment* pertama memiliki spesifikasi CPU Intel(R) Xeon(R) CPU @ 2.30GHz, RAM 13.3 GB, dan GPU Tesla K80. *Environment* kedua memiliki spesifikasi CPU Intel(R) Xeon(R) CPU @ 2.00GHz, RAM 13.3 GB, dan GPU Tesla T4. Bahasa yang digunakan adalah Python 3.6. *Library-library* yang kami gunakan adalah numpy, matplotlib, torch, torchvision, sklearn, seaborn, pandas, dan PIL.

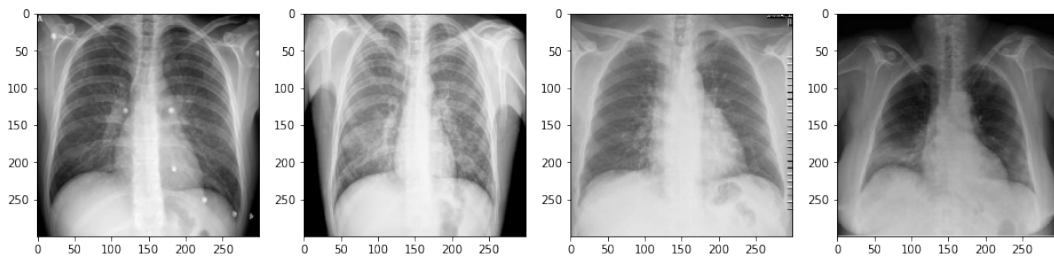
2.2 Dataset

Dataset yang akan kami gunakan adalah Kaggle Chest X-Ray Images (Pneumonia) dataset (2018) milik Paul Mooney [9] yang berisi 5,863 gambar X-Ray dada dan *repository* Covid-19 Image Data Collection milik Cohen, J.P., Morrison, P., dan Dao, L. [1] yang berisi 884 gambar X-Ray dada pasien COVID-19.



Figur 1: Gambar X-Ray pada dataset Kaggle Chest X-Ray Images (Pneumonia) dataset

Gambar-gambar X-Ray dada pada Kaggle Chest X-Ray Images (Pneumonia) dataset dipisah menjadi tiga kelas, yaitu *Normal*, *Bacterial Pneumonia*, dan *Viral Pneumonia*. Untuk memenuhi tujuan dari makalah ini, kami tidak perlu membedakan antara *Bacterial Pneumonia* dan *Viral Pneumonia*, sehingga kami menggabungkan kedua kelas tersebut menjadi satu kelas *Pneumonia*.



Figur 2: Gambar X-Ray pada dataset Covid-19 Image Data Collection

Gambar-gambar dari Kaggle Chest X-Ray Images (Pneumonia) dataset dan Covid-19 Image Data Collection disatukan menjadi dataset gabungan. Jumlah total gambar di dataset gabungan adalah 6310 : 1583 *Normal Case*, 4273 *Pneumonia Case*, dan 454 *Covid Case*. Dataset gabungan dipisah menjadi 3 bagian (*train*, *val*, *test*) dengan menggunakan metode *stratified sampling*. Metode *sampling* digunakan untuk memastikan adanya setiap kelas pada dataset baru yang akan diklasifikasi. Pembagian dataset dilakukan dengan perbandingan *train:val:test* = 6:2:2. *Training set* (*train*) adalah dataset untuk proses pembelajaran model, *Validation set* (*val*) adalah dataset yang digunakan untuk memberikan evaluasi tidak bias pada saat menyesuaikan hyperparameter dari model, dan *Test set* (*test*) adalah dataset terakhir dimana dataset ini hanya digunakan ketika melakukan pengujian akhir. Dataset-dataset tersebut dibuat dengan tujuan model tidak mengalami *overfit* dan model dapat menggeneralisasi data lain diluar dataset yang ada dengan baik.

	<i>Normal Case</i>	<i>Pneumonia Case</i>	<i>Covid Case</i>
<i>Train</i>	950	2564	272
<i>Val</i>	316	855	91
<i>Test</i>	317	854	91

2.3 Metode Prapemrosesan

Prapemrosesan adalah semua transformasi yang diaplikasikan ke data mentah sebagai bentuk persiapan untuk dimasukkan ke algoritma pembelajaran mesin. Prapemrosesan citra mungkin memiliki efek positif pada kualitas ekstraksi fitur dan hasil analisis gambar [7]. Pelatihan pada algoritma pembelajaran dengan citra mentah tanpa adanya prapemrosesan akan menyebabkan kinerja klasifikasi yang buruk [10].

2.3.1 Konversi ke RGB

RGB (*Red, Green, Blue*) adalah model warna aditif dimana warna merah, hijau, dan biru ditambahkan bersama dengan berbagai cara untuk mereproduksi sejumlah besar warna [4]. Tujuan utama model warna RGB adalah untuk mendeteksi, merepresentasikan, dan menampilkan gambar-gambar di sistem-sistem elektronik.

Transformasi pertama yang kami lakukan terhadap gambar-gambar masukan adalah mengkonversi model warna gambar menjadi RGB. Konversi tersebut perlu dilakukan model CapsNet mengambil masukan gambar dengan tiga *channel* warna : merah, hijau, dan biru. Transformasi dilakukan dengan fungsi *convert* pada *library* python PIL yang mengubah gambar dengan *channel* warna RGBA atau *greyscale* menjadi RGB.

2.3.2 Resizing

Resizing adalah proses dalam prapemrosesan citra untuk memastikan bahwa gambar memiliki ukuran yang sama. Hal ini perlu dilakukan sebab gambar-gambar pada data latihan dan data ujian memiliki ukuran yang berbeda-beda. Metode klasifikasi yang digunakan pada makalah ini mengharuskan gambar masukan memiliki ukuran yang sama. Gambar masukan dipotong ukurannya dengan ukuran panjang dan lebar sebesar (224, 224).

2.3.3 Normalisasi

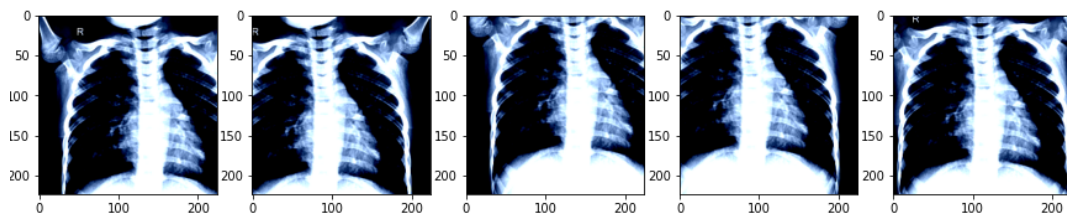
Normalisasi adalah proses yang mengubah kisaran intensitas pixel dan memastikan setiap parameter *input* memiliki distribusi yang serupa. Tujuan dari normalisasi pada gambar adalah untuk membuat gambar lebih akrab atau terlihat normal bagi indra manusia.

2.3.4 Konversi ke PyTorch Tensor

PyTorch *Tensor* adalah matriks berdimensi banyak yang mengandung elemen-elemen dengan tipe data yang sama. PyTorch *Tensor* memiliki banyak kemiripan dengan numpy array; keduanya adalah array n-dimensional generik untuk digunakan untuk komputasi numerik. Perbedaan utamanya adalah PyTorch *Tensor* dapat dijalankan pada CPU atau GPU.

Model yang digunakan pada makalah ini dijalankan menggunakan GPU, sehingga mengharuskan gambar masukan masuk dalam bentuk PyTorch *Tensor*. Gambar masukan yang masih dalam model warna RGB dikonversi menjadi PyTorch *Tensor* dengan fungsi `torchvision.transforms.ToTensor()`.

Berikut adalah hasil dari proses prapemrosesan :



Figur 3: Gambar X-Ray hasil proses prapemrosesan

2.4 CapsNet

2.4.1 Introduksi CapsNet

CapsNet (Capsule Neural Network) adalah sistem pembelajaran mesin bertipe *Artificial Neural Network* yang dapat memodelkan hubungan-hubungan bersifat hierarkis. Pendekatan ini dikembangkan dengan mengikuti organisasi syaraf biologis. Perbedaan CapsNet dengan *Convolutional Neural Network* (CNN) biasa adalah penambahan struktur *capsule*, dimana *capsule* dengan tingkat lebih tinggi akan menggunakan kembali keluaran dari beberapa *capsule* tingkat rendah untuk menghasilkan representasi informasi yang lebih stabil. Keluaran dari CapsNet adalah vektor berisi probabilitas dan *pose* (kombinasi dari posisi dan orientasi) dari sebuah observasi.

Salah satu keuntungan utama CapsNet adalah CapsNet dapat menjadi solusi “*Picasso Problem*” di bidang pengenalan gambar. Contoh “*Picasso Problem*” adalah ketika sebuah gambar wajah seseorang memiliki semua fitur-fitur seperti hidung dan mulut, namun posisi hidung ditukar dengan posisi mulut. *Convolutional Neural Network* biasa akan kesulitan mendeteksi gambar tersebut sebagai sebuah wajah. CapsNet dapat mengatasi masalah ini dengan mengeksploitasi fakta bahwa walaupun perubahan sudut pandang memiliki efek non-linear di tingkat pixel, efeknya linear di tingkat objek.

2.4.2 Latar Belakang

Awal mula perkembangan CapsNet terjadi pada tahun 2000, dimana Geoffrey Hinton mendeskripsikan sistem untuk merepresentasikan gambar dengan menggunakan gabungan teknik *segmentation* dan *parse trees*. Sistem ini terbukti berguna dalam mengklasifikasikan digit-digit tulisan tangan di dataset MNIST.

Pada tahun 2017, Hinton dan timnya memperkenalkan mekanisme *dynamic routing*[3] untuk *capsule networks*. Pendekatan ini berhasil meningkatkan performa model dalam mengolah data MNIST dan mengurangi ukuran data latihan. Hasilnya diklaim mengalahkan performa CNN terutama pada digit-digit yang *overlap*.

2.4.3 Transformasi

Dalam bidang citra komputer, terdapat tiga jenis sifat yang dapat dimiliki sebuah objek :

1. *Invariant* : sifat objek yang tidak berubah sebagai hasil dari suatu bentuk transformasi. Sebagai contoh, area dari lingkaran tidak berubah ketika lingkaran digeser ke kiri atau kanan.
2. *Equivariant* : sifat objek yang perubahannya dapat diprediksi bila dilakukan sebuah transformasi. Contohnya adalah titik pusat lingkaran berubah sesuai dengan arah gerak lingkaran tersebut.
3. *Non-equivariant* : sifat objek yang perubahannya tidak dapat diprediksi bila dilakukan sebuah transformasi. Contohnya adalah bila sebuah lingkaran ditransformasi menjadi sebuah oval, rumus untuk menghitung keliling objek tersebut sudah bukan $2 \pi r$.

Kelas sebuah objek dalam citra komputer diharapkan bersifat *invariant* ketika diterapkan banyak transformasi. Sebagai contoh, sebuah mobil harusnya tetap diklasifikasikan sebagai mobil walaupun gambarnya dibalik atau dkecilkan. Namun pada kenyataannya, sebagian besar karakteristik gambar bersifat *equivariant*.

Karakteristik-karakteristik *equivariant* seperti volume dan letak bagian-bagian objek disimpan dalam sebuah *pose*. *Pose* adalah data yang mendeskripsikan translasi, rotasi, skala, dan refleksi dari objek. Translasi adalah perubahan lokasi di satu atau lebih dimensi, rotasi adalah perubahan orientasi, skala adalah perubahan ukuran, dan refleksi adalah gambar yang dicerminkan.

CapsNet mempelajari *linear manifold* global antara objek dan pose objek tersebut dan merepresentasikan informasi dalam bentuk matriks. Dengan menggunakan matriks tersebut, CapsNet dapat mengidentifikasi objek walaupun objek tersebut sudah mengalami sejumlah transformasi. Informasi spasial dari objek dipisah dari informasi untuk mengklasifikasi objek itu sendiri, sehingga objek dapat diklasifikasi secara independen dari *pose*-nya.[2]

2.4.4 Pooling

Dalam CNN konvensional, digunakan teknik *pooling layer* yang berguna untuk mengurangi jumlah detail informasi yang diproses di lapisan yang lebih tinggi. *Pooling* mengizinkan sedikit *invariance translasional* (objek berada di lokasi yang berbeda) dan meningkatkan jumlah tipe fitur yang dapat direpresentasikan. CapsNet menolak penggunaan teknik *pooling*, dengan alasan :

1. *Pooling* melanggar persepsi bentuk biologis karena tidak ada bingkai koordinat intrinsik;
2. *Pooling* membuang informasi posisional, sehingga menghasilkan *invariance* dan bukan *equivariance*;
3. *Pooling* mengabaikan *linear manifold* yang mendasari banyak variasi dalam gambar-gambar;
4. *Pooling* membuat rute informasi antar lapisan secara statis, tidak mengkomunikasikan info potensial ke fitur-fitur secara dinamis;
5. *Pooling* merusak detektor fitur yang dengan dengan *pooling layer*, karena ada sejumlah informasi yang dihapus.

2.4.5 Capsules

Sebuah *capsule* adalah kumpulan neuron yang memiliki vektor aktivitas yang merepresentasikan berbagai properti-properti dari suatu tipe entitas yang terdapat pada gambar, seperti posisi, ukuran, orientasi, dll [3]. Kumpulan neuron tersebut secara kolektif menghasilkan vektor aktivitas yang diderivasi oleh CapsNet dari data masukan. Kemungkinan adanya suatu entitas dalam gambar direpresentasikan oleh panjang vektor, sedangkan orientasi vektor mengkuantifikasikan properti dari *capsule*.

Di struktur *Artificial Network* tradisional, keluaran dari neuron-neuron adalah sebuah nilai skalar yang secara longgar merepresentasikan probabilitas dari sebuah observasi. CapsNet menggantikan detektor fitur yang menghasilkan nilai skalar dengan *capsule-capsule* yang menghasilkan vektor dan *max-pooling* dengan metode *routing-by-agreement*.

Capsules bersifat mandiri dari satu sama lain, sehingga probabilitas deteksi benar meningkat drastis bila beberapa *capsule* setuju akan suatu prediksi. Dua *capsule* yang mengolah entitas berdimensi enam hanya akan sepakat akan suatu nilai dengan margin 10 % karena kebetulan hanya terjadi satu dalam satu juta kali percobaan. Bila dimensi entitas bertambah, maka kemungkinan kesepakatan karena kebetulan berkurang secara eksponensial.

Capsules di lapisan lebih tinggi mengambil keluaran dari *capsules* di lapisan lebih rendah, lalu menerima dari *capsules* yang keluarannya berkelompok. Sebuah kelompok akan menyebabkan *capsule* lebih tinggi untuk menghasilkan keluaran dengan probabilitas tinggi bahwa entitas terdapat dalam observasi tersebut. *Capsules* tingkat tinggi mengabaikan *outlier* karena mereka hanya berkonsentrasi pada keluaran yang berkelompok dengan keluaran-keluaran lain.

2.4.6 Routing-by-agreement

Keluaran dari suatu *capsule* (anak) akan dihubungkan ke *capsule* (orangtua) di lapisan berikutnya sesuai dengan kemampuan *capsule* anak untuk memprediksi keluaran dari *capsule* orangtua. Setelah melalui beberapa iterasi, keluaran setiap *capsule* orangtua dapat bersatu dengan prediksi beberapa *capsule* anak dan berpisah dengan beberapa *capsule* anak lain.

Untuk setiap *capsule* orangtua yang mungkin, setiap *capsule* anak menghitung vektor prediksi dengan mengalikan keluarannya dengan matriks beban yang dilatih dengan *backpropagation*. Lalu, keluaran dari *capsule* orangtua dihitung sebagai produk skalar dari sebuah prediksi dengan koefisien yang merepresentasikan probabilitas bahwa *capsule* anak adalah milik *capsule* orangtua tersebut. Sebuah *capsule* anak yang prediksinya relatif dekat dengan keluaran hasilnya akan meningkatkan koefisien antara *capsule* anak dan *capsule* orangtua, sedangkan prediksi yang relatif jauh akan mengurangi koefisien. Koefisien yang meningkat akan meningkatkan kontribusi dari *capsule* anak ke *capsule* orangtua, meningkatkan produk skalar dari prediksi *capsule* dengan keluaran *capsule* orangtua. Setelah beberapa iterasi, koefisien yang dihasilkan akan menghubungkan secara kuat *capsule* orangtua dengan *capsule-capsule* yang paling mungkin sebagai anak. Hal ini mengindikasikan bahwa keberadaan *capsule-capsule* anak mengimplikasikan keberadaan entitas yang direpresentasikan *capsule* orangtua. Semakin banyak *capsule* anak yang prediksinya mirip dengan keluaran *capsule* orangtua, semakin cepat koefisien bertambah dan mempercepat konvergensi. *Pose* dari *capsule* orangtua secara progresif menjadi lebih cocok dengan *pose* anak-anaknya.

Logit awal dari koefisien-koefisien adalah probabilitas log prior bahwa sebuah *capsule* anak adalah memiliki sebuah *capsule* orangtua. Prior-prior dapat dilatih bersama dengan beban-beban pada matriks beban. Prior tergantung lokasi dan tipe dari *capsule* orangtua dan anak, namun tidak tergantung masukan yang sedang diolah. Di setiap iterasi, koefisien akan disesuaikan melalui metode *softmax* sehingga jumlah setiap koefisien adalah 1. *Softmax* akan semakin memperbesar nilai-nilai yang besar dan semakin mengurangi nilai-nilai kecil.

Mekanisme pembuatan rute dinamis ini menyediakan sarana yang dibutuhkan untuk memisahkan objek-objek yang *overlap*.

Algoritma ini dapat direpresentasikan secara matematis sebagai berikut :

$$\hat{u}_{j|i} = W_{ij}u_i$$

Pose vektor u_i dirotasikan dan ditranslasi oleh matriks W_{ij} menjadi vektor $\hat{u}_{j|i}$ yang memprediksikan keluaran dari *capsule* orangtua.

$$s_j = \sum c_{ij}\hat{u}_{j|i}$$

Capsule s_j di lapisan yang lebih tinggi akan diberikan jumlahan dari prediksi-prediksi semua *capsule-capsule* di lapisan yang lebih rendah, dimana setiap prediksi dikalikan dengan koefisien c_{ij} .

2.4.7 Training

CapsNet dilatih secara *supervised* dengan cara meminimalisir jarak euclidean antara gambar dan keluaran dari sebuah CNN yang merekonstruksikan gambar masukan dari keluaran *capsule-capsule* terminal. Jaringan dilatih secara diskriminatif dengan menggunakan *routing-by-agreement* iteratif.

Terdapat dua tipe *loss* yang digunakan untuk melatih model. *Loss* tipe pertama adalah *margin loss*. Panjang dari sebuah vektor instansiasi merepresentasikan probabilitas bahwa entitas dari *capsule* terdapat pada sebuah gambar. Untuk membolehkan keberadaan entitas-entitas dengan jumlah banyak, *margin loss* terpisah harus dihitung untuk setiap *capsule*. Menurunkan bobot *loss* dari entitas-entitas yang absen menghentikan pembelajaran mengecilkan panjang vektor aktivitas untuk semua entitas. *Loss* total adalah jumlah dari *loss* setiap entitas. Hinton menggunakan *loss function* sebagai berikut :

$$L_k = \underbrace{T_k \max(0, m^+ - \|\mathbf{v}_k\|)^2}_{\text{class present}} + \underbrace{\lambda (1 - T_k) \max(0, \|\mathbf{v}_k\| - m^-)^2}_{\text{class not present}}, \quad T_k = \begin{cases} 1, & \text{digit of class } k \text{ present} \\ 0, & \text{otherwise} \end{cases}$$

Hinton menggunakan nilai 0.9 untuk m^+ dan nilai 0.1 untuk m^- supaya panjang tidak terlalu maksimal atau jatuh. Penurunan bobot dari bobot inisial kelas-kelas yang abses diatur dengan λ , dimana Hinton menggunakan nilai 0.5.

Loss tipe kedua adalah *reconstruction loss*. *Reconstruction loss* mendorong entitas-entitas untuk menyandikan parameter-parameter instansiasi dari masukan mereka. Vektor aktivitas final lalu digunakan untuk merekonstruksi gambar masukan menggunakan CNN *decoder* yang terdiri dari tiga lapisan yang terhubung penuh. Rekonstruksi meminimalisir jumlah dari kuadrat perbedaan antara keluaran unit logistik dan intensitas pixel. Hinton menggunakan nilai 0.0005 sebagai koefisien untuk memperkecil *reconstruction loss* supaya tidak mendominasi *margin loss* ketika latihan.

2.4.8 Kelebihan CapsNet

CapsNet diklaim memiliki empat kelebihan konseptual dibandingkan *convolutional neural networks* :

1. *Viewpoint invariance* : penggunaan matriks pose membolehkan jaringan CapsNet untuk mengenali objek-objek tanpa mempedulikan perspektif dari mana mereka dipandang.
2. Parameter lebih sedikit : *capsule* mengelompokkan neuron, sehingga koneksi antar lapisan membutuhkan lebih sedikit parameter.
3. Generalisasi lebih baik untuk sudut pandang baru : CapsNet lebih baik mengenali objek dari beberapa rotasi berbeda karena matriks *pose* menangkap karakteristik-karakteristik ini sebagai transformasi linear.
4. Dapat menangani *white-box adversarial attacks* : *Fast Gradient Sign Method* (FGSM) biasanya digunakan untuk menyerang CNN biasa dengan mengevaluasi gradien dari tiap pixel terhadap *loss* dari jaringan, lalu mengubah setiap pixel dengan maksimum epsilon (*error term*) untuk memaksimalkan *loss*. Metode ini dapat mengurangi akurasi CNN secara drastis (hingga di bawah 20 %), namun CapsNet dapat mempertahankan akurasi di atas 70 %.

2.5 Evaluasi

Evaluasi model adalah proses dimana performa model diukur untuk mengetahui kualitas dari model. Evaluasi pada masalah klasifikasi dapat dilakukan dengan melihat *confusion matrix*, *accuracy*, *precision*, *recall*, dan *F-measure*. *Confusion matrix* merangkum kinerja *classifier* terhadap data testing [5]. Contoh *confusion matrix* dapat dilihat di tabel di bawah:

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True Positive(TP)	False Positive(FP)
Predicted Negative Class	False Negative(FN)	True Negative(TN)

Tabel 1: Tabel *confusion matrix*

TP menunjukkan jumlah kelas positif yang diklasifikasikan dengan tepat, TN menunjukkan jumlah kelas negatif yang diklasifikasikan dengan tepat, FP menunjukkan jumlah kelas negatif yang diklasifikasikan secara salah, dan FN menunjukkan jumlah kelas positif yang diklasifikasikan secara salah.

Metrik-metrik yang digunakan untuk mengevaluasi performa model klasifikasi adalah *accuracy*, *precision*, *recall*, dan *F-measure*. Berikut adalah rumus-rumus untuk mengkalkulasi metrik-metrik tersebut :

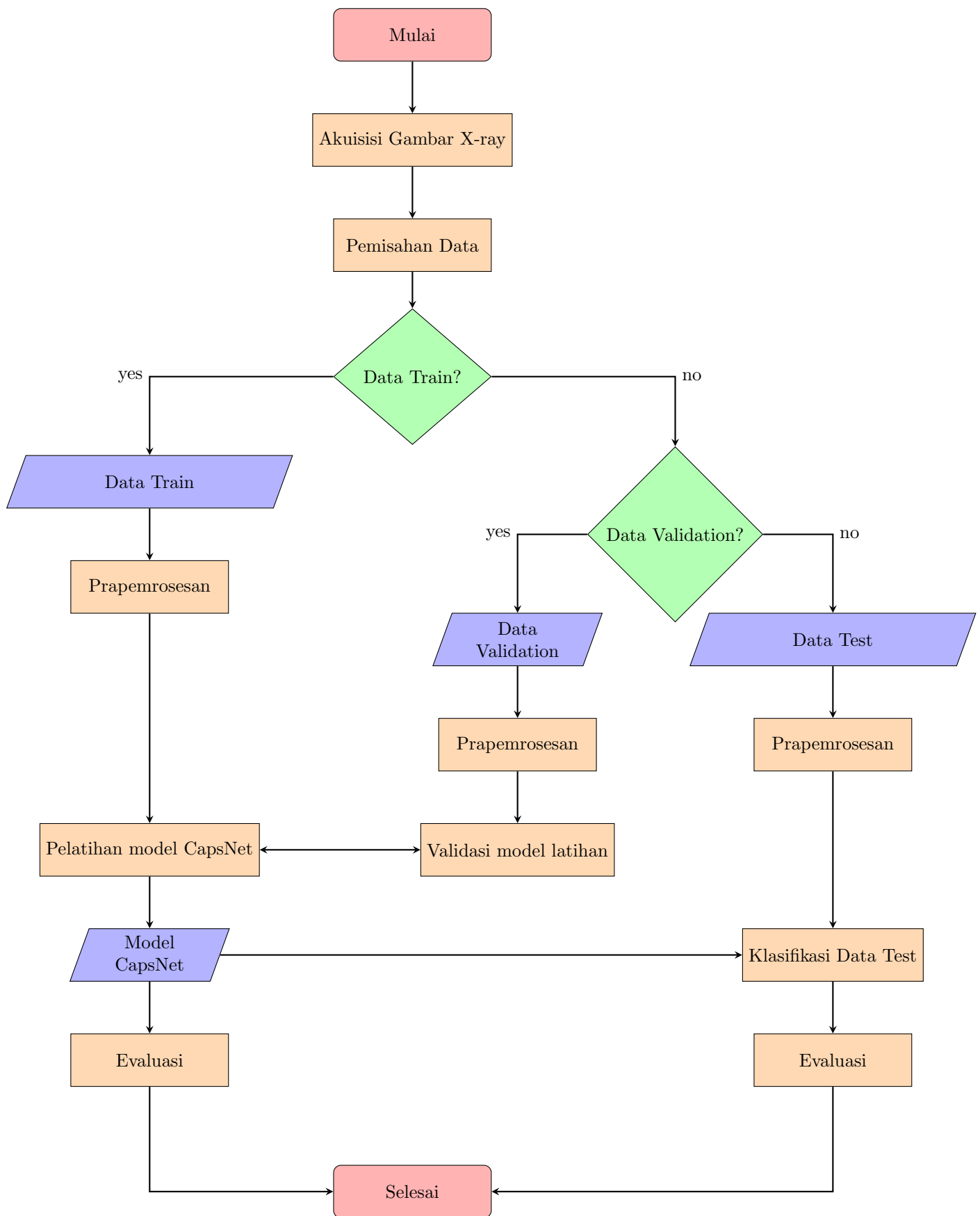
$$accuracy(acc) = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$precision(p) = \frac{TP}{TP + FP} \quad (2)$$

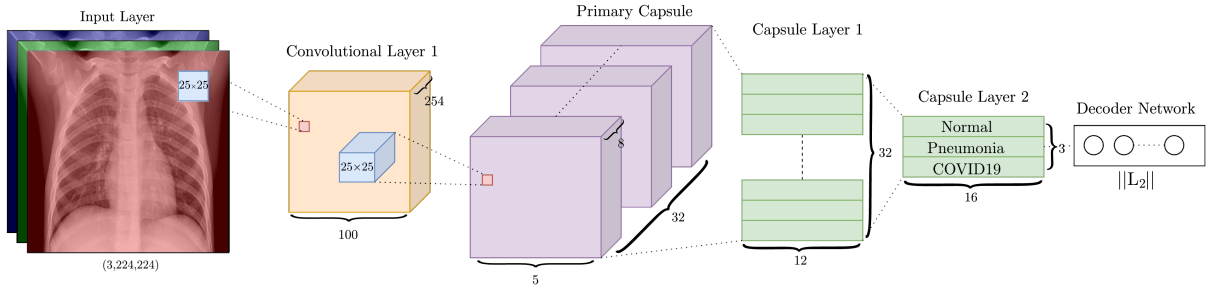
$$recall(r) = \frac{TP}{TP + FN} \quad (3)$$

$$F - measure(Fm) = \frac{2 * p * r}{p + r} \quad (4)$$

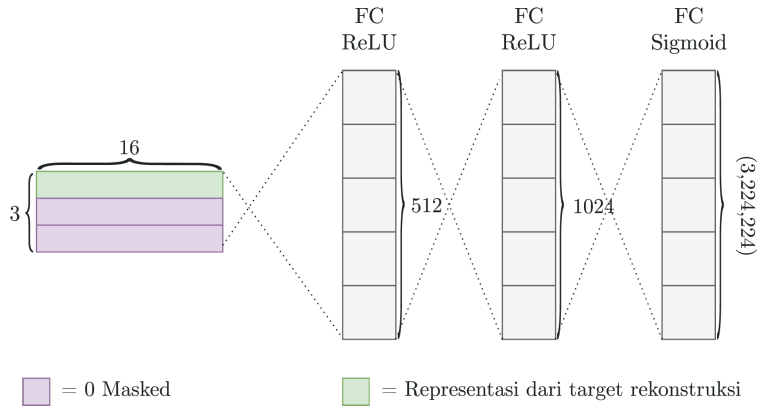
3 Desain dan Implementasi Penambahan Data



Figur 4: Diagram alur implementasi penambahan data



Figur 5: Arsitektur CapsNet



Figur 6: Arsitektur Decoder Network

3.1 Arsitektur CapsNet

Arsitektur CapsNet yang digunakan dalam makalah ini terdiri dari :

1. *Input Layer*
Input menggunakan gambar berukuran 224x224 dengan 3 *channel* yaitu *channel red*, *green*, dan *blue* atau biasa disebut dengan RGB.
2. *Convolutional Layer 1*
Layer ini akan melakukan operasi konvolusi pada gambar *input* menggunakan *out channel* atau *filter* sebanyak 256 dengan *kernel* berukuran 25x25 dan *stride* sebanyak 2. *Layer* ini akan menghasilkan *feature map*.
3. *Primary Capsule*
Layer ini akan mengubah hasil konvolusi menjadi *capsule-capsule*. *Layer* ini terdiri dari beberapa layer, yaitu:
 - *Convolution Layer 2*
Layer ini mendapatkan *input channel* sebanyak 256, lalu operasi konvolusi pada *layer* ini menggunakan *out channel* sebanyak 256 dengan *kernel* berukuran 25x25 dan *stride* sebanyak 16.
 - *Reshape*
Operasi *reshape* digunakan untuk mendapatkan 32x5x5 *capsule* berdimensi 8 setiap *capsule*-nya.
 - *Squash*
Layer squash adalah *layer* aktivasi dari primary capsule yang mengubah panjang vektor besar menjadi mendekati 1 dan vektor kecil menjadi 0.
4. *Capsule Layer 1*
Layer ini mendapatkan *input capsule* sebanyak 800 berdimensi 8 untuk setiap *capsule*. Lalu algoritma *Routing-by-agreement* akan dijalankan pada layer ini dan menghasilkan *output capsule* sebanyak 32 berdimensi 12 untuk setiap *capsule*.

5. *Capsule Layer 2*

Layer ini mendapatkan *input capsule* sebanyak 32 berdimensi 12 untuk setiap *capsule*. Lalu algoritma *Routing-by-agreement* akan dijalankan pada *layer* ini dan menghasilkan *output capsule* sebanyak jumlah *class* yaitu 3 dan berdimensi 16 untuk setiap *capsule*.

6. *Decoder Network*

Decoder Network adalah *layer* yang digunakan CapsNet untuk menghitung *loss* dan melakukan proses rekonstruksi gambar. *Decoder Network* terdiri dari:

- *Fully Connected Layer 1*

Layer ini mendapatkan *input* sebanyak jumlah *output capsule* dikali dimensi *capsule* dari *layer* sebelumnya yaitu 3×16 dan menghasilkan 512 *output neuron*. Fungsi aktivasi yang digunakan pada *layer* ini adalah ReLu.

- *Fully Connected Layer 2*

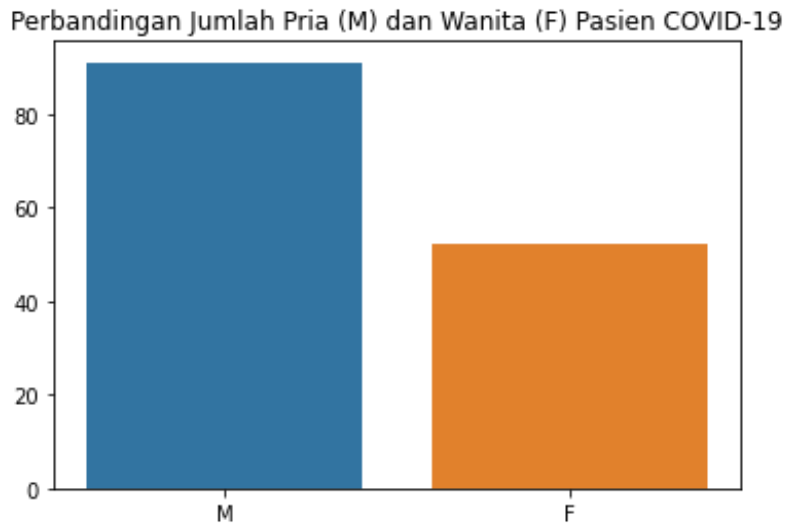
Layer ini mendapatkan *input* sebanyak jumlah *output neuron* dari *layer* sebelumnya yaitu 512 dan menghasilkan 1024 *output neuron*. Fungsi aktivasi yang digunakan pada *layer* ini adalah ReLu.

- *Fully Connected Layer 3*

Layer ini mendapatkan *input* sebanyak jumlah *output neuron* dari *layer* sebelumnya yaitu 1024 dan menghasilkan *output neuron* sebanyak ukuran gambar *input* yaitu $224 \times 224 \times 3$. Fungsi aktivasi yang digunakan pada *layer* ini adalah *sigmoid*.

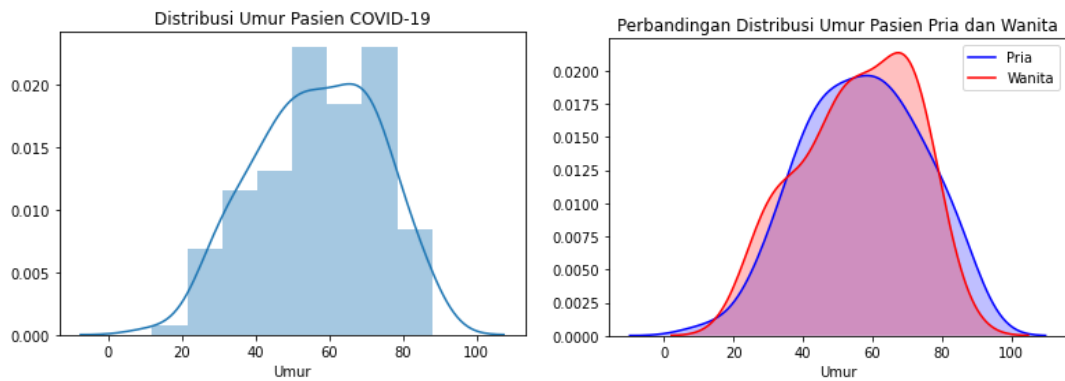
4 Analisis

4.1 Analisis Exploratory Data Analysis



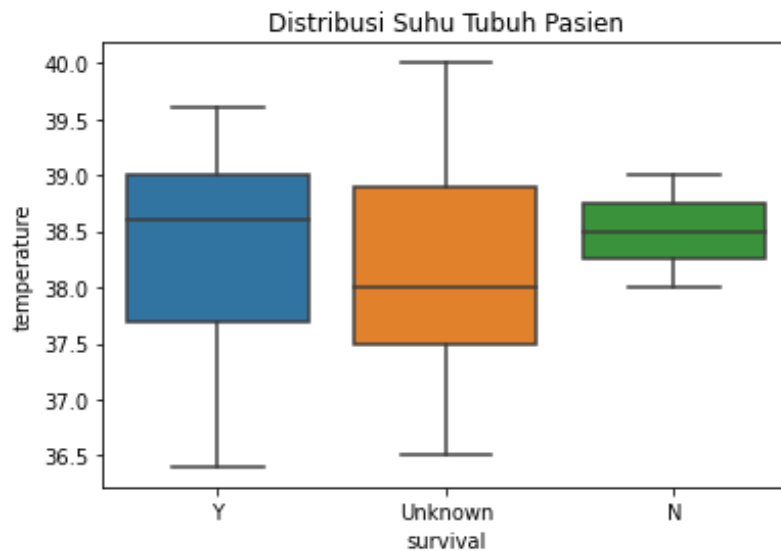
Figur 7: Grafik perbandingan jenis kelamin

Dalam dataset jumlah pasien COVID-19 pria jauh melebihi jumlah pasien wanita.



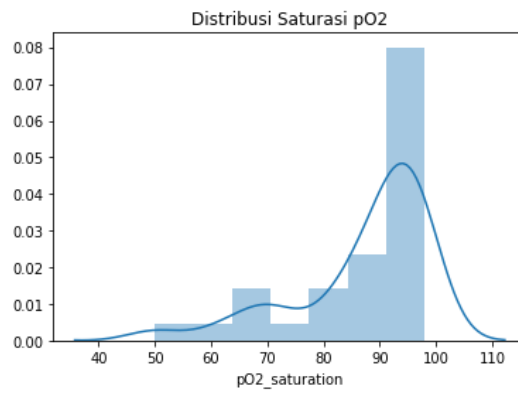
Figur 8: Grafik perbandingan umur

Distribusi umur menunjukkan bahwa sebagian besar pasien berusia paruh baya dengan terpusat pada kisaran umur 50-80. Tidak ada perbedaan signifikan antara distribusi umur pasien pria dan pasien wanita.

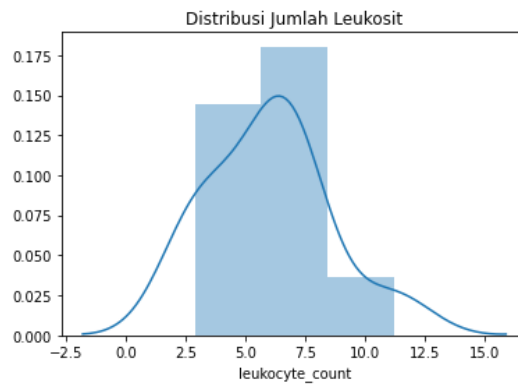


Figur 9: Grafik perbandingan suhu tubuh

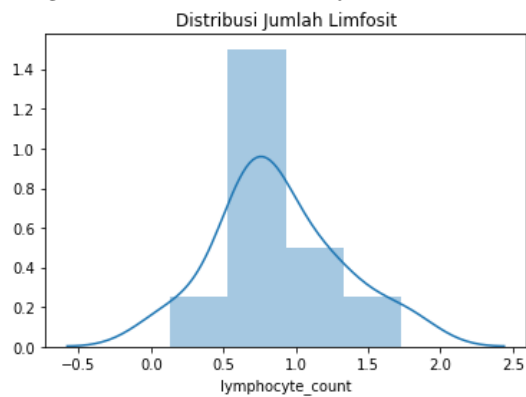
Grafik menunjukkan bahwa terdapat perbedaan jarak interkuartil antara pasien yang selamat dan pasien yang meninggal. Pasien yang selamat menunjukkan suhu tubuh sebagian besar di antara 37.75-39.0 derajat celsius, sedangkan pasien yang meninggal menunjukkan jarak yang lebih kecil antara 38.25-38.75 derajat celsius. Pasien-pasien yang masih dirawat sehingga kondisi keselamatannya masih belum diketahui memiliki suhu median yang lebih rendah dibandingkan pasien yang sudah dipastikan status keselamatannya. Hal ini mungkin menunjukkan bahwa pasien-pasien yang baru masuk dan masih dirawat sudah dapat ditangani dengan lebih baik sehingga suhu tubuh pasien secara keseluruhan menurun.



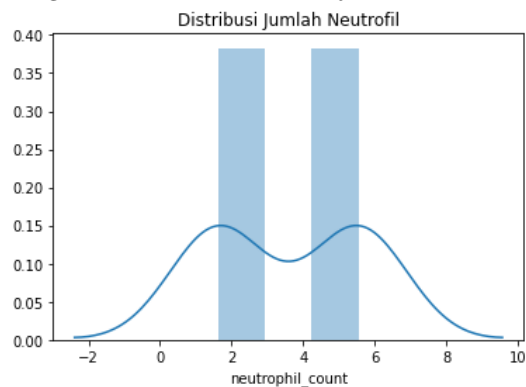
Figur 10: Grafik distribusi angka pO2



Figur 11: Grafik distribusi jumlah leukosit



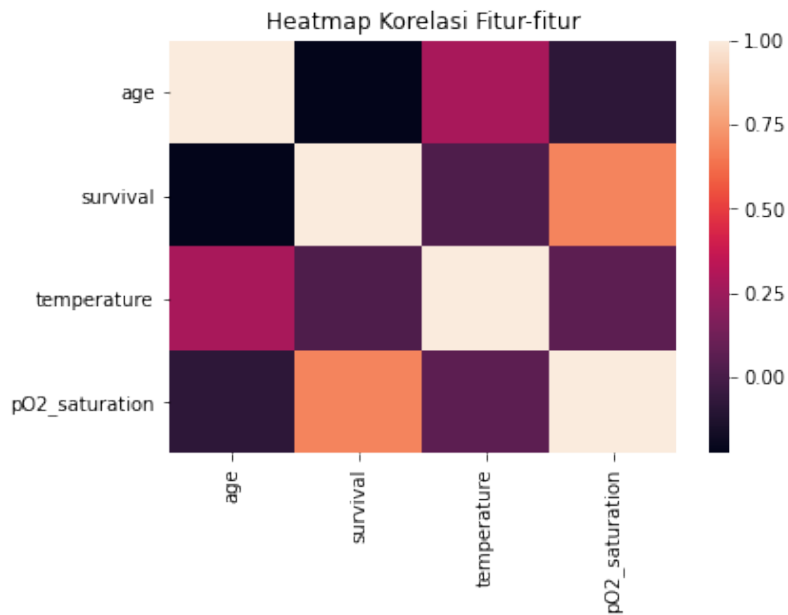
Figur 12: Grafik distribusi jumlah limfosit



Figur 13: Grafik distribusi jumlah neutrofil

Saturasi PO2 pasien COVID-19 berada di kisaran 50-100 dengan sebagian besar berada di kisaran 90-100. Angka leukosit pasien berada di kisaran 2.5-11.0 dengan terpusat di angka 7.5. Angka limfosit pasien berada di

kisaran 0.2-1.75 dengan sebagian besar berada di sekitar angka 0.75. Angka neutrofil pasien terpisah menjadi dua daerah, dimana sebagian pasien yang memiliki angka neutrofil diantara 1.8-3.0 dan sebagian pasien memiliki angka neutrofil diantara 4.2-5.8.

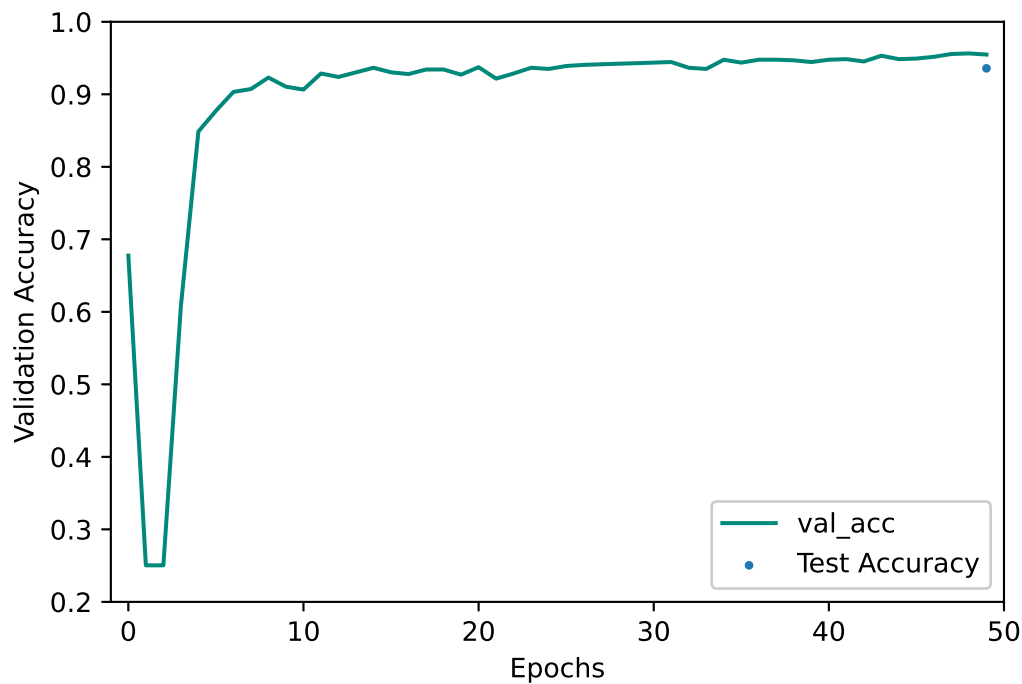


Figur 14: Grafik Heatmap korelasi fitur

Heatmap korelasi fitur menunjukkan bahwa keselamatan pasien sangat ditentukan oleh umur, dimana umur yang lebih tinggi akan menurunkan kemungkinan keselamatan pasien. Suhu tubuh terlihat tidak memiliki korelasi yang signifikan dengan keselamatan pasien, ditunjukkan dengan warna ungu yang memberi arti korelasi dekat dengan 0. Tingkat saturasi PO2 terlihat memiliki korelasi positif yang cukup signifikan dengan tingkat keselamatan pasien, dimana pasien dengan saturasi PO2 tinggi memiliki kemungkinan lebih tinggi untuk selamat.

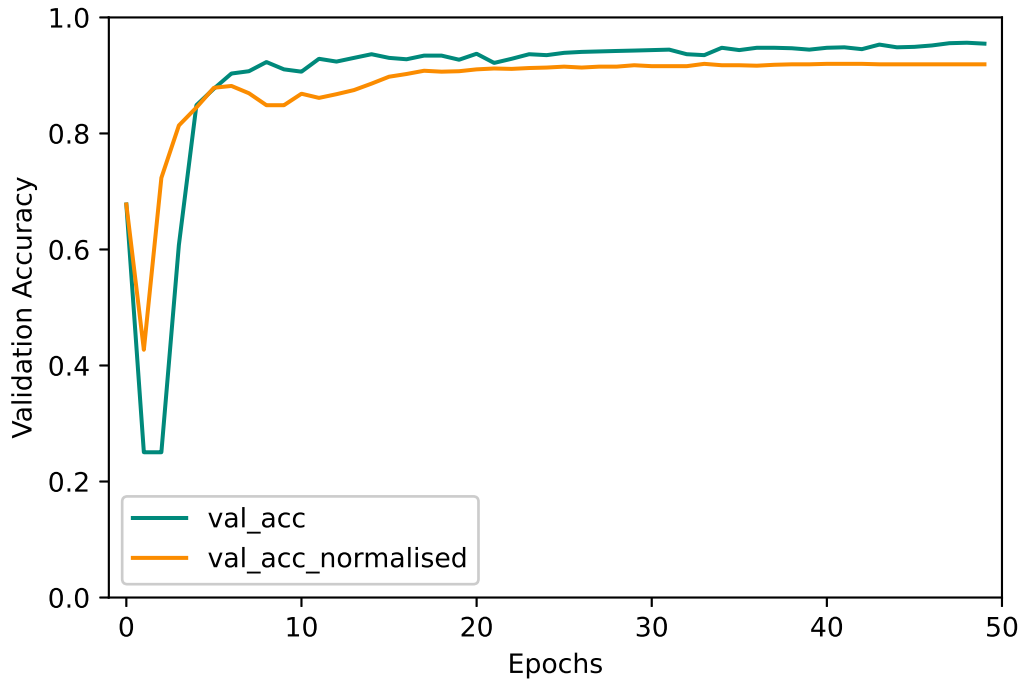
4.2 Analisis Hasil Model CapsNet

4.2.1 Analisis Metrik Penilaian



Figur 15: Grafik akurasi data validasi

Grafik akurasi data validasi menunjukkan fluktuasi besar di 5 *epoch* pertama, karena nilai-nilai awal pada matriks beban model bersifat acak sehingga model belum bisa membuat prediksi dengan akurat. Untuk *epoch* ke 10 dan seterusnya, akurasi validasi model sudah stabil dengan akurasi sekitar 90 %. Akurasi tes yang hanya sedikit kurang dari akurasi validasi akhir membuktikan bahwa model dapat generalisasi dengan baik pada data yang belum pernah dilihatnya. Hasil akurasi validasi di *epoch* terakhir latihan tercatat pada angka 95.56 %, sedangkan akurasi pada dataset *test* tercatat pada angka 94.29 %.

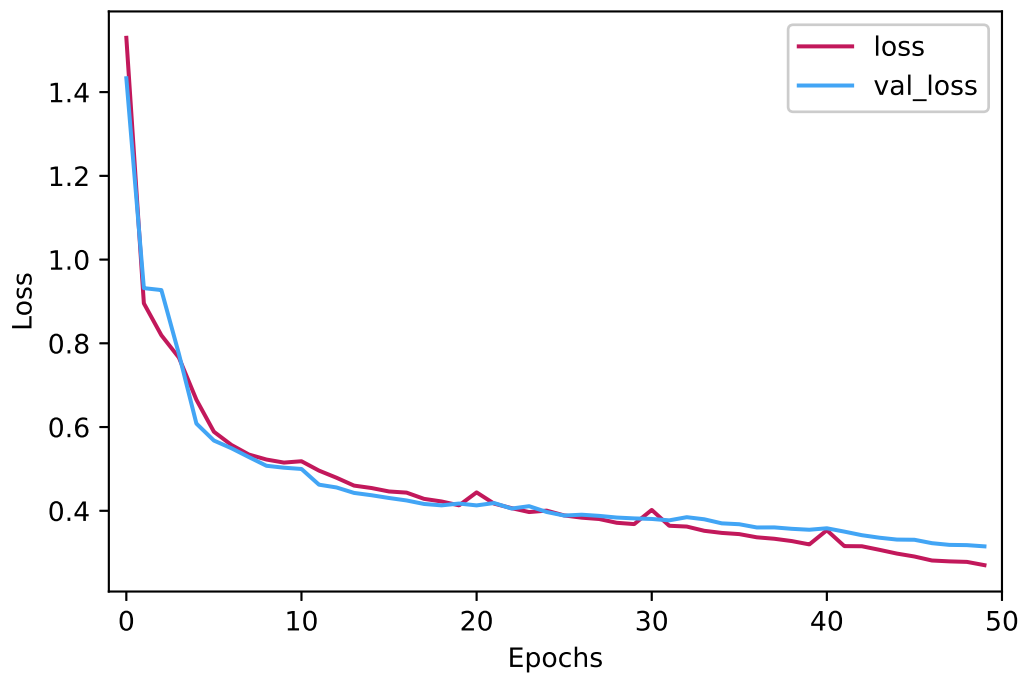


Figur 16: Grafik perbandingan akurasi data validasi dengan tambahan proses normalisasi

Bila proses normalisasi diaplikasikan ke gambar masukan, akurasi validasi model sedikit berkurang. Normalisasi pada gambar masukan berguna pada model CNN sebab model CNN belajar dengan secara terus-menerus menambahkan gradien *error vector* dikalikan dengan *learning rate* ke berbagai matriks beban. Bila vektor masukan tidak dinormalisasi, maka jarak distribusi nilai-nilai fitur akan berbeda untuk setiap fitur, sehingga hasil kurang maksimal. Namun, tampaknya normalisasi tidak membantu meningkatkan performa model CapsNet, melainkan sedikit menurunkan performanya.

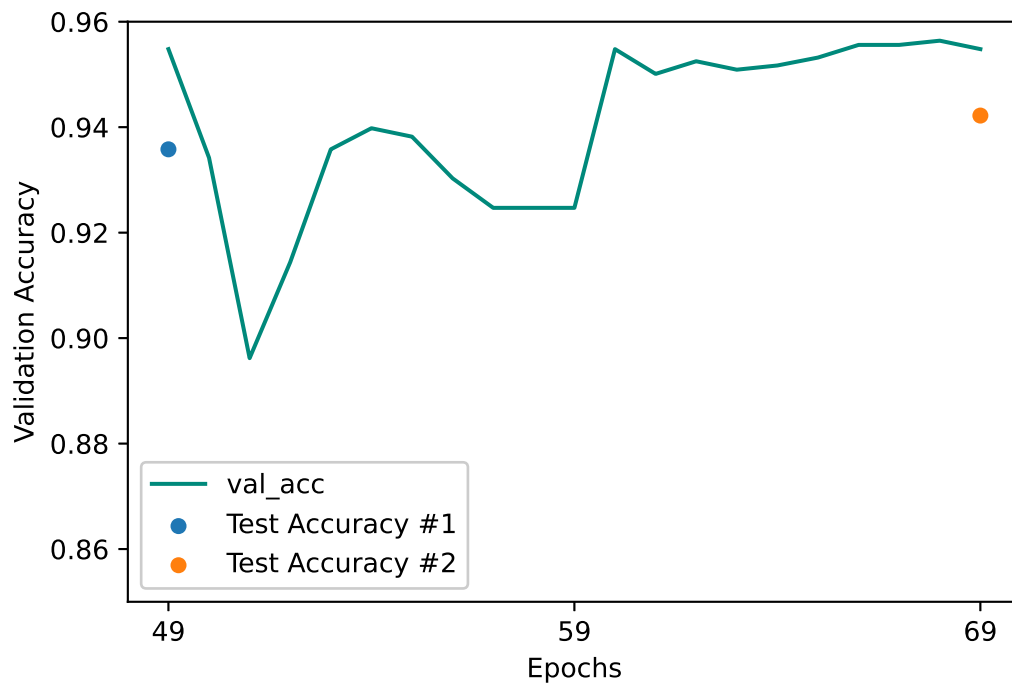
Hal ini terjadi karena CapsNet memiliki keunggulan dapat menyimpan data positional gambar, maka CapsNet memiliki kemampuan baik ketika gambar terkena transformasi affine atau transformasi yang dapat mengakibatkan perubahan.

Ini dapat ditunjukkan ketika terjadi transformasi normalisasi CNNs mendapatkan keuntungan sedangkan CapsNet tidak diuntungkan. Jadi kami berkesimpulan bahwa normalisasi pada model dalam makalah ini tidak dapat membantu.



Figur 17: Grafik penurunan fungsi *loss* model

Nilai fungsi *loss* dari model terlihat menurun drastis pada 10 *epoch* pertama, lalu menurun secara perlahan di *epoch-epoch* berikutnya. *Loss* pada model ketika latihan dan *loss* pada validasi terlihat mendekati satu sama lain, membuktikan bahwa model tidak mengalami *overfit*.



Figur 18: Grafik akurasi validasi kasus COVID-19

Akurasi validasi untuk khusus kasus-kasus COVID-19 terlihat fluktuatif sampai *epoch* ke 60. Pada *epoch* ke 61 dan seterusnya, kami mencoba memasukan lebih banyak gambar-gambar kasus COVID-19 untuk menam-

bahkan informasi yang didapat oleh model. Hasilnya adalah akurasi validasi model mulai lebih stabil di sekitar 95 %, dan akurasi tes model untuk kasus-kasus COVID-19 tercatat dengan akurasi sekitar 94 %.

	<i>Normal (Predicted)</i>	<i>Pneumonia (Predicted)</i>	<i>Covid (Predicted)</i>
<i>Normal (Actual)</i>	292	24	1
<i>Pneumonia (Actual)</i>	30	821	3
<i>Covid (Actual)</i>	4	10	77

Tabel 2: Tabel *confusion matrix* model

Hasil *confusion matrix* model akhir menunjukkan bahwa model dapat dengan tepat mengklasifikasikan 77 dari 91 kasus COVID-19. Dari 1171 kasus-kasus yang bukan COVID-19, model hanya mengklasifikasikan secara salah 4 kasus sebagai kasus COVID-19.

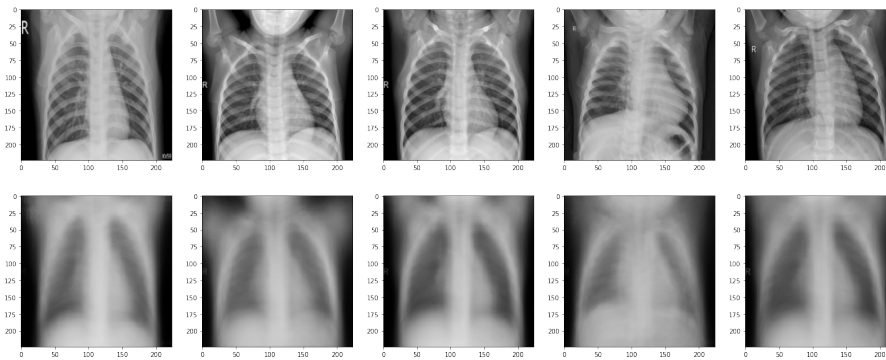
	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
<i>Normal</i>	0.90	0.92	0.91
<i>Pneumonia</i>	0.96	0.96	0.96
<i>Covid</i>	0.95	0.85	0.90
<i>Macro Avg</i>	0.94	0.91	0.92
<i>Weighted Avg</i>	0.94	0.94	0.94
<i>Accuracy</i>	0.94		

Tabel 3: Tabel metrik-metrik penilaian model

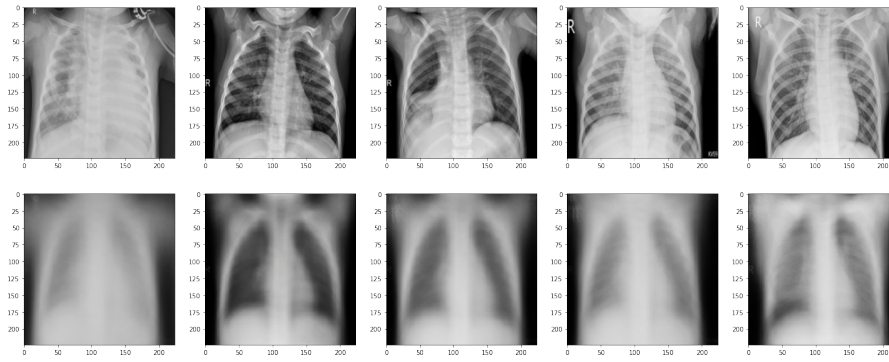
Akurasi keseluruhan model tercatat pada angka 94 %. *Macro average* menghitung nilai rata-rata *precision*, *recall*, dan *F1-score* dari tiap kelas. *Weighted average* menambahkan nilai *precision*, *recall*, dan *F1-score* masing-masing kelas dengan dikalikan beban yang tergantung dengan jumlah kasus benar dari tiap kelas. Terlihat bahwa *weighted average* dari *precision*, *recall*, dan *F1-score* bernilai konsisten dengan akurasi keseluruhan, yaitu 94 %.

4.2.2 Hasil Rekonstruksi

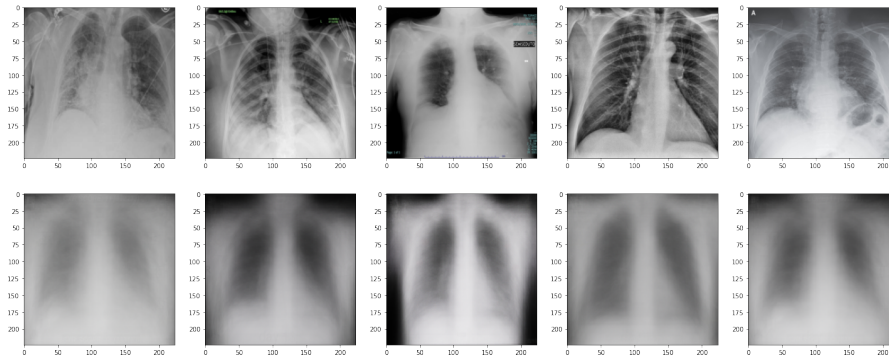
Hasil prediksi pada *Dense Capsule 2* diteruskan kepada *Decoder Network* menghasilkan rekonstruksi gambar dimana hasil ini akan dipakai sebagai $L_2/$ *Reconstruction Loss*. Metode hasil rekonstruksi sebagai *loss function* ini digunakan untuk menggeneralisasi data mencegah *overfitting* dengan mendorong *Capsules* untuk dapat *encode* fitur-fitur yang terdapat pada input gambar.



Figur 19: Gambar Chest X-Ray Normal (atas) dengan Rekonstruksi (bawah).



Figur 20: Gambar Chest X-Ray Pneumonia (atas) dengan Rekonstruksi (bawah).



Figur 21: Gambar Chest X-Ray COVID-19 (atas) dengan Rekonstruksi (bawah).

Dari hasil rekonstruksi gambar pada figur diatas, walaupun input gambar memiliki perbedaan seperti lebih besar, memiliki kemiringan atau transformasi affine lain. Hasil rekonstruksi akhir model dapat tetap menjaga struktur fitur input gambar awal. Hasil rekonstruksi secara samar memiliki bentuk, struktur dan fitur-fitur yang menyerupai input gambar awal.

5 Kesimpulan

Di era digital sekarang ini, kemampuan algoritma *Deep Learning* dapat menyaingi tingkat performa manusia. Tugas manusia memproses data dari suatu gambar sulit dikerjakan oleh komputer, tetapi sekarang dengan kemajuan algoritma dapat dilakukan dengan mesin. Berdasarkan percobaan yang telah kami lakukan, model kami dapat mengklasifikasi perbedaan antara gambar Chest X-Ray normal, pneumonia dan COVID-19 dengan akurasi validasi 0.9556 dan akurasi tes 0.9429.

6 Dokumentasi

Hasil dan program dapat ditemukan pada pranala berikut :<https://github.com/Yakuy/Gemastik13> (untuk sementara masih bersifat private)

Daftar Pustaka

- [1] J.P. Cohen, P. Morrison, and L. Dao. “Covid-19 image data collection”. In: *arXiv 2003.1159* (2020). URL: <https://github.com/ieee8023/covid-chestxray-dataset>.
- [2] G. Hinton, A. Krizhevsky, and S. Wang. “Transforming Auto-encoders”. In: ().
- [3] G. Hinton, S. Sabour, and N. Frosst. “Dynamic Routing Between Capsules”. In: (2017).
- [4] Robert Hirsch. *Exploring Colour Photography: A Complete Guide*. ISBN: ISBN 1-85669-420-8.
- [5] M. Hossin and M. Sulaiman. “A Review on Evaluation Metrics for Data Classification Evaluations”. In: *International Journal of Data Mining and Knowledge Management Process (IJDKP)* 5.2 (2015), pp. 1–11.
- [6] Daniel Kermany, Kang Zhang, and Michael Goldbaum. “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”. In: *Mendeley Data 2* (2018).
- [7] S. Krig. “Image pre-processing”. In: *Computer Vision Metrics* (1993), pp. 56–111. DOI: 10.1007/978-3-319-50490-23.
- [8] A. Mangal et al. “CovidAID: COVID-19 Detection Using Chest X-Ray”. In: (2020).
- [9] P. Mooney. *Kaggle chest x-ray images (pneumonia) dataset*. URL: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>. (2018).
- [10] K. K. Pal and K. S. Sudeep. *Preprocessing for Image Classification by Convolutional Neural Networks*, pp. 1778–1781. ISBN: 9781509007745.
- [11] N. Zuraya. *Tiga Dampak Besar Pandemi Covid-19 bagi Ekonomi RI*. URL: <https://republika.co.id/berita/qdgt5p383/tiga-dampak-besar-pandemi-covid19-bagi-ekonomi-ri>. (2020).