Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 7, No. 3, Maret 2023, hlm. 1116-1125 http://j-ptiik.ub.ac.id

Deteksi Covid-19 dari Citra *X-ray* menggunakan *Vision Transformer*

Javier Ardra Figo[1], Novanto Yudistira[2], Agus Wahyu Widodo[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]javier.a.figo@gmail.com, [2]yudistira@ub.ac.id, [3]a_wahyu_w@ub.ac.id

Abstrak

Virus Corona adalah virus beruntai tunggal RNA yang dapat menginfeksi manusia dan beberapa hewan. Pemanfaatan citra *x-ray* dada dapat menjadi salah satu sarana untuk pemeriksaan dan memonitor keadaan paru-paru seperti pada terduga penyakit *tuberculosis*, *pneumonia*, dan *hernia*. Pemanfaatan citra *x-ray* dengen *deep learning* menjadi salah satu solusi untuk pendeteksian virus covid-19. Arsitektur *Vision transformer* (ViT) merupakan arsitektur yang terinspirasi dari arsitektur *Transformer* yang merupakan *state-of-the-art* (SOTA) dalam pemrosesan bahasa alami. Salah satu *dataset* citra *x-ray* yang dapat digunakan oleh public adalah *CovidX*. *Dataset CovidX* terbagi menjadi 3 kelas yaitu *pneumonia, covid-19*, dan *normal* dengan jumlah total sebanyak 30.530 citra *x-ray*. *Dataset* akan menggunakan teknik augmentasi data berupa *gaussian blur* dan *colorjitter*. Percobaan arsitektur *vision transformer* akan dibagi menjadi 3 ukuran yaitu *base, large,* dan *huge.* Arsitektur ini akan dikombinasikan dengan *transfer learning* dan augmentasi data. Proses pelatihan akan berjalan sebanyak 40 *Epoch*, menggunakan *Optimizer Stochastic Gradient Descent*, *Scheduler WarmupCosine*, dan fungsi *Cross Entropy Loss*. Pengujian akan dilakukan dengan pengujian parameter ukuran arsitektur, pengujian pengaruh *transfer learning*, pengujian pengaruh augmentasi data, dan pengujian dengan arsitektur lain. Hasil pengujian terbaik dengan parameter akurasi adalah *ResNet50* yang dipretrained yang mencapai akurasi sebesar 0,9617006 pada data validasi dan 0,9548872 pada data tes. Berdasarkan nilai akurasi tersebut dapat disimpulkan bahwa terjadi *overfitting* pada arsitektur *ResNet*50.

Kata kunci: *Vision transformer*, Augmentasi Data, *Transfer Learning*, *X-ray*, Klasifikasi

Abstract

Corona Virus is a single stranded RNA virus that can infect human dan a few animal. X-ray Imaging can be one of the few way to check or monitor lungs condition such in the case of tuberculosis, pneumonia, and hernia. Combining X-ray Imaging with deep learning can be one of the solution to the covid-19 detection problem. Vision Tranformer is an architecture that inspired by transformer which is state of the art in the natural language processing realm. One of the few public dataset that contain x-ray image is covidX. CovidX can be breakdown into 3 classes which is pneumonia, covid-19, and normal with as few as 30,530 x-ray image available.the Dataset will processed with data augmentation gaussian blur and colorjitter. The vision transformer that will be used in this experiment is base, large, and huge. This architecture will be used with transfer learning and data augmentation. This experiment will use 40 Epoch, stochastic gradient descent Optimizer, WarmupCosine Scheduler, and Cross Entropy loss function. This experiment will test the effect of transfer learning toward accuracy, the effect of data augmentation toward accuracy, and then will be compared to other architecture. The best accuracy from this experiment is achieved by ResNet50 with transfer learning that achieve accuracy as high as 0.9617006 with validation data and 0.9548872 in test data. Based on this result, the model is overfitting.

Keywords: *Vision transformer, Data Augmentation, Transfer Learning, X-ray, Classification*

1. PENDAHULUAN

Virus Corona adalah virus beruntai tunggal RNA yang dapat menginfeksi manusia dan beberapa hewan (Velavan and Meyer, 2020).Virus ini pertama kali diidentifikasi di kota Wuhan, Huber, Tiongkok pada tanggal 1 Desember 2019. Dan pada tanggal 11 Maret 2020, Organisasi Kesehatan Dunia atau *World Health Organization* (WHO) menetapkan penyakit yang disebabkan oleh virus SARS-

CoV-2 ditetapkan sebagai pandemi. Hingga tanggal 6 Mei 2022 di website resmi WHO tercatat seluruh dunia memiliki kasus positif terkonfirmasi sebanyak 513.955.910 kasus dan kematian yang telah terkonfirmasi sebagai akibat dari virus Corona sendiri sebanyak 6.249.700 kasus. Sedangkan untuk negara Indonesia sendiri kasus positif terkonfirmasi yang tercatat adalah sebanyak 6.047.986 dan kasus kematian sebanyak 156.357 kasus. Salah satu metode yang digunakan untuk mendeteksi virus Corona adalah menggunakan *Reverse Transcription-Polymerase Chain Reaction* atau RT-PCR (Li et al., 2020). Metode ini digunakan sebagai metode utama dalam pendeteksian dini untuk pasien yang terduga terinfeksi virus Corona. Pada beberapa kasus, metode ini dapat menimbulkan kebingungan apabila hasil metode tersebut negatif namun terdapat beberapa gejala dini virus Corona pada pasien tersebut. Selain itu diperlukan juga metode deteksi covid-19 yang lebih cepat dari RT-PCR untuk meringankan beban fasilitas kesehatan yang terdampak akibat pandemi virus corona ini. Karena alasan itulah diperlukan metode yang dapat digunakan sebagai metode yang dapat memastikan apakah gejala yang dirasakan tersebut diakibatkan oleh virus corona dan tidak memakan waktu yang banyak.

Pemanfaatan citra *x-ray* dada dapat menjadi salah satu sarana untuk pemeriksaan dan memonitor keadaan paru-paru seperti pada terduga penyakit tuberculosis, pneumonia, dan hernia. Pemanfaatan citra *x-ray* dengan deep learning menjadi salah satu solusi terhadap permasalahan diatas. Salah satu masalah yang *deep learning* dapat selesaikan adalah Klasifikasi. Data yang nanti akan digunakan untuk *deep learning* dapat berupa citra *x-ray* atau citra *computed tomography* (CT). Walaupun secara kualitas citra CT melebihi citra *x-ray*, ada beberapa kekurangan dari citra CT seperti ketersediaan jumlah citra dan aksesbilitas *dataset* yang berisi citra CT (Wang, Lin and Wong, 2020). Salah satu *dataset* yang menyediakan citra *x-ray* yang dianotasi ahli mengenai covid-19 terbanyak dan dapat diakses publik adalah *COVIDx*.

Dataset COVIDxV9 terdapat citra *x-ray* berjumlah 30.530 citra yang dibagi menjadi 3 label yaitu normal, *covid-19*, dan *pneumonia*. *Dataset* ini akan terus diperbarui selama pandemi covid-19 ini berlangsung. *Dataset* ini merupakan kumpulan dari beberapa *dataset-dataset* yang ada lalu digabung menjadi satu *dataset* besar. Penelitian pertama yang menggunakan *dataset* ini pertama kali ada lah penelitian mengenai deteksi covid-19 yang menggunakan arsitektur *CovidNet*. Akurasi tertinggi yang dicapai *CovidNet* mencapai 93,3% (Wang, Lin and Wong, 2020). Penelitian lainnya yang menggunakan *dataset* ini menggunakan arsitektur *Resnet-50* memiliki akurasi tertinggi sebesar 96,2% (Farooq and Hafeez, 2020).

Arsitektur *Vision transformer* (ViT) merupakan arsitektur yang terinspirasi dari arsitektur *Transformer* yang merupakan *state-of-the-art* (SOTA) dalam pemrosesan bahasa alami. Arsitektur *Vision transformer* terdiri dari beberapa proses yang dapat dibagi menjadi *patch embeddings*, *multi-head attention*, dan *multi layer perceptron*. Pada proses *patch embeddings* gambar masukan di bagi menjadi beberapa patch yang jumlahnya disesuaikan dengan ukuran arsitekturnya. Kemudian gambar tersebut diberi *position embeddings* yang berfungsi untuk memberi informasi lokasi dari patch tersebut pada gambar tersebut. Misalkan patch paling kiri atas diberi nomor 1 dan patch paling kanan bawah diberi nomor 16 berturut-turut. Setelah itu patch tersebut akan dimasukkan ke *multi-head attention*. Pada proses ini terjadi proses pengolahan yang akan mengakibatkan data dari patch yang telah dibagi pada proses sebelumnya untuk diambil informasi pentingnya menggunakan mekanisme *attention*. Mekanisme ini mengamplifikasi informasi yang penting dan akan menghilangkan informasi yang tidak penting berupa pixel. Hasil dari proses ini kemudian akan diproses lebih lanjut pada *multi layer perceptron*. Pada proses ini terjadi fungsi aktivasi menggunakan *gaussian error linear unit*. Hasil keluarannya akan dilanjutkan ke layer linear agar hasilnya dapat disesuaikan dengan jumlah prediksi kelas yang tersedia dan diaktivasi menggunakan fungsi aktivasi *softmax* dan menghasilkan hasil prediksi dari gambar masukan. Menggunakan *Dataset ImageNet*, ViT diuji dengan arsitektur SOTA dalam visi computer seperti *ResNet152x4* dan *EfficientNet-L2* dan memperoleh akurasi paling besar diantara ketiga arsitektur tersebut dengan akurasi sebesar 88,55% (Dosovitskiy et al., 2020).

Overfitting adalah kondisi dimana arsitektur memiliki akurasi yang tinggi pada data latih, namun pada data tes akurasinya rendah. *Overfitting* mengakibatkan arsitektur untuk menghapal seluruh pola bahkan *noise* yang ada pada data latih yang kemudian kesulitan untuk

mengenali pola yang terdapat pada data tes (Ying, 2019). Karena itu, digunakan augmentasi data yang berfungsi untuk menambah jumlah dan/atau kualitas dari data latih sehingga arsitektur dapat lebih mudah untuk menggeneralisasi data pada data tes. Selain augmentasi data, ada metode lain untuk menghindari *overfitting* yaitu dengan *transfer learning*. *Transfer learning* adalah metode melatih arsitektur pada *dataset* yang jumlahnya lebih banyak terlebih dahulu lalu menggunakan arsitektur yang sudah dilatih tersebut untuk digunakan pada *dataset* yang ingin diuji, dalam kasus ini adalah *dataset COVIDx*.

2. KAJIAN PUSTAKA

2.1 Dataset

Dataset yang digunakan adalah *dataset COVIDxV9*. *Dataset* ini sudah dikum-pulkan dari berbagai macam *dataset* umum yang memiliki foto *x-ray* dada untuk pasien yang positif *covid-19, pneumonia* dan *normal*. Untuk distribusi data dari dataset CovidXV9 dapat dilihat pada tabel 1 dan contoh dataset CovidXV9 pada gambar 1

Tabel 1 Distribusi Kelas CovidXV9

Tipe	Normal	Pneumo -nia	Covid-19	Total
Latih	8.085	5.555	16.940	30.130
Uji	100	100	200	400

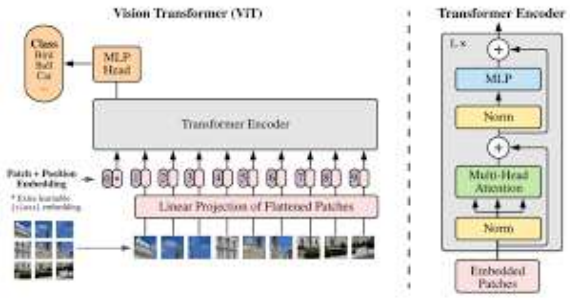
Gambar 1 Dari kiri ke kanan : pasien terpapar covid-19, kondisi normal, dan pneumonia.

2.2 Augmentasi Data

Kekurangan dari dataset yang kecil adalah arsitektur yang dilatih menggunakan dataset tersebut akan sulit untuk menggeneralisasi data yang ada pada data latih (Perez and Wang, 2017). Karena kekurangan tersebut, dataset yang kecil akan mengakibatkan masalah overfitting. Untuk menyelesaikan masalah tersebut ada beberapa cara yang dapat digunakan seperti Augmentasi data. Augmentasi data adalah proses menambah jumlah dataset melalui manipulasi data pada dataset yang ingin digunakan. Bentuk manipulasi pada data yang dalam hal ini adalah citra dengan mengubah piksel pada citra yang mengakibatkan terbentuknya citra baru tanpa menghilangkan informasi-informasi penting yang ada dalam citra tersebut. Beberapa augmentasi data yang akan digunakan adalah Gaussian Blur dan ColorJitter

2.3 Vision Transformer

Vision Transformer adalah salah satu arsitektur pengolahan gambar yang menggunakan dasar dari arsitektur Transformer. Sebelumnya Transformer merupakan state-of-the-art dalam bidang pemrosesan Bahasa alami. Sedangkan sebelum arsitektur ini ada, state-of-the-art dari visi komputer menggunakan Convolutional Neural Network (CNN). Pertama kali muncul istilah Vision Transformer adalah pada jurnal yang berjudul An Image is Worth 16x16 Words: Tranformer For Image Recognition at Scale oleh (Dosovitskiy et al., 2020). Di jurnal tersebut menjelaskan bagaimana arsitektur Vision Transformer dibandingkan dengan arsitektur state-of-the-art pada bidang klasifikasi citra. Arsitektur Vision Transformer diilustrasikan pada gambar 2



Gambar 2 Ilustrasi arsitektur vision transformer

2.3.1 Patch Embeddings

Syarat data masukan untuk arsitektur *vision transformer* adalah citra tersebut dapat dibagi menjadi *patch* berukuran nxn. *Patch embeddings* adalah proses dimana citra tersebut akan dibagi menjadi beberapa *patch* yang kemudian akan dijadikan vector satu dimensi berisi nilai piksel citra tersebut. Persamaan 1 akan menjelaskan rumus perubahan vector dua dimensi menjadi satu dimensi pada proses *patch embeddings*.

$$X \in R^{H\,x\,W\,x\,C} \rightarrow X_P \in R^{N\,x\,(P^2.C)} \quad (1)$$

Keterangan :

X = Citra masukan
X_p = Citra masukan setelah transformasi
H = Tinggi citra masukan
W = Lebar citra masukan
C = Jumlah *channel* citra masukan
P = Ukuran Patch
N = $\frac{HW}{P^2}$

2.3.2 Layer Normalization

Proses pelatihan arsitektur *deep learning* pada umumnya menghabiskan sumber daya komputasi yang sangat banyak, sehingga dibutuhkan suatu cara yang dapat mengurangi penggunaan daya dan waktu yang dibutuhkan untuk menyelesaikan proses pelatihan tersebut. Salah satu langkah yang dapat ditempuh adalah *Layer Normalization*. *Layer Normalization* mengurangi penggunaan sumber daya dan waktu pelatihan dengan mengurangi nilai inputan menggunakan rata-rata dan standar deviasi. Semakin nilai inputan mendekati rata-rata dan standar deviasi, maka nilai inputan tersebut akan mendekati 0. Persamaan 2 berisi rumus menghitung *Layer Normalization*.

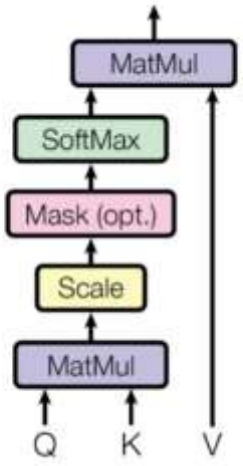
$$x'_{i,k} = \frac{x_{i,k} - \mu^i}{\sqrt{\sigma_i^2 + \epsilon}} \quad (2)$$

Keterangan :
$x_{i,k}$ = Nilai inputan / Vektor
μ^i = Rata-rata nilai inputan / vektor
σ_i^2 = Standar deviasi nilai inputan / vektor
ϵ = Denominator

2.3.3 Multi-Head Attention

Setelah *Layer Normalization*, *Layer* selanjutnya adalah *Multi Head Attention*. Di lapisan ini terjadi proses *attention* yang berfungsi untuk mencari informasi jangka Panjang yang berguna dari vektor masukan. *multi head attention* bekerja dengan cara menduplikasi vektor masukan menjadi *query*, *key*, dan *value*. Kemudian *query*, *key*, dan *value* tersebut akan diproses menggunakan *scaled dot product.* Proses tersebut akan dilakukan sesuai jumlah *head* dan kemudian hasilnya akan di*concat* lalu dikompres menjadi ukuran vektor yang telah ditentukan. Gambar 3 akan mengilustrasikan proses yang terjadi selama proses *scaled-dot product.*

Gambar 3 Ilustrasi proses *scaled-dot product*



2.3 Multi Layer Perceptron

Multi Layer Perceptron (MLP) adalah model jaringan syaraf tiruan yang terinspirasi dari system syaraf pada tubuh manusia. MLP terbagi menjadi 3 bagian yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Seperti system syaraf manusia, MLP memiliki neuron yang memiliki bobot. disetiap neuron tersebut ada fungsi aktivasi non-linear kecuali pada lapisan masukan.

2.3.5 Gaussian Error Linear Units

GELU atau Gaussian Error Linear Units merupakan fungsi aktivasi nonlinear. *GELU* Ketika dibandingkan fungsi aktivasi yang lain seperti ReLU dan ELU memiliki peningkatan performa dibidang visi computer, pemrosesan bahasa alami, dan suara (Hendrycks and Gimpel, 2020). GELU banyak digunakan pada arsitektur Transformer. Persamaan 3 berisi rumus menghitung fungsi aktivasi GELU.

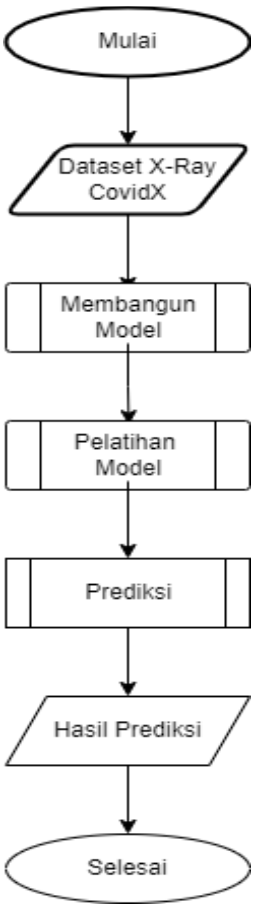
$$0{,}5x(1 + tanh[\sqrt{2/\pi}\,(x + 0{,}044715x^3)] \quad (3)$$

2.4 gradCAM

Ketika sebuah arsitektur melakukan propagasi balik, gradien akan menjadi salah satu parameter bagaimana penambahan atau pengurangan bobot berdasarkan kontribusi bobot tersebut penting atau tidak. Hal ini membuktikan bahwasanya gradien menyimpan informasi mengenai kondisi dari suatu layer mengenai dimana neuron yang bekerja keras dan mana neuron yang kurang bekerja (Selvaraju et al., 2017). Hal ini dimanfaatkan oleh gradCAM untuk melakukan visualisasi terhadap gradien tersebut.

3. METODOLOGI PENELITIAN

Perancangan sistem deteksi covid-19 dibentuk berdasarkan bentuk dataset yang berupa citra. Pelatihan dimulai dari preprocessing dataset CovidX seperti merubah bentuk citra menjadi 224 x 224 untuk menyesuaikan arsitektur Vision Transformer. Setelah itu arsitektur Vision Transformer akan dibangun dan data latih akan melatih arsitektur tersebut. Setelah model dilatih, akan dilakukan prediksi menggunakan data latih yang akan menghasilkan hasil prediksi. Diagram alur sistem selengkapnya dapat dilihat pada Gambar 4.



Gambar 4 Diagram Alir Penelitian

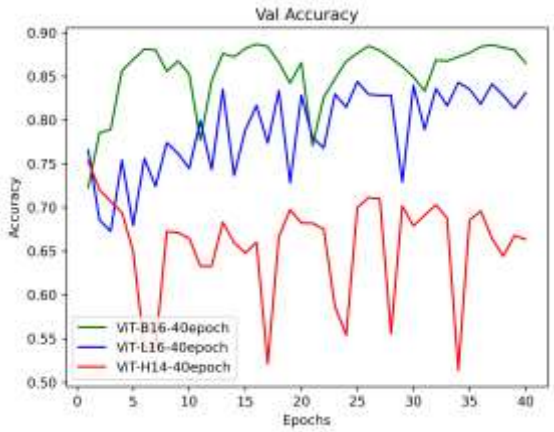
4. HASIL DAN PEMBAHASAN

4.1 Pengujian dan Analisis Pengaruh Ukuran Arsitektur

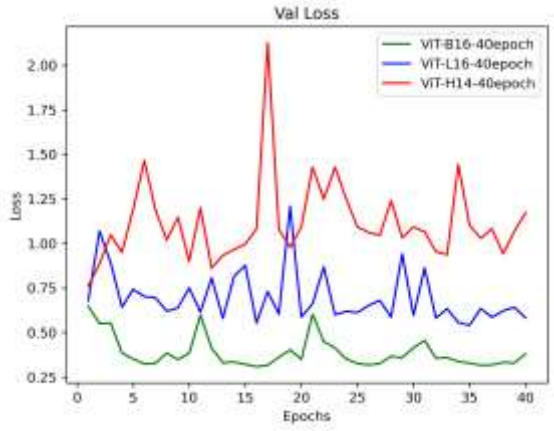
Untuk Pengujian berdasarkan ukuran arsitektur akan menggunakan parameter nilai akurasi dan loss pada data validasi dan data testing. Akan diambil nilai tertinggi dari masing-masing arsitektur selama 40 epoch. Untuk jenis arsitektur yang akan diujikan adalah Vision Transformer Base-16, Vision Transformer Large-16, dan Vision Transformer Huge-14.

Tabeel 2 Hasil pengujian akurasi dan nilai loss berdasarkan ukuran arsitektur

	Akurasi		Loss	
Tipe	**Val**	**Test**	**Val**	**Test**
Base-16	0,88	0,72	0,3	0,62
Large-16	0,84	0,68	0,54	1,04
Huge-14	0,75	0,48	0,76	1,69



Gambar 5 Hasil pengujian akurasi berdasarkan ukuran arsitektur



Gambar 6 Hasil pengujian nilai *loss* berdasarkan ukuran arsitektur

Dari hasil pengujian nilai akurasi dan loss berdasarkan ukuran arsitektur, dapat dilihat bahwa akurasi tertinggi diantara Vision Transformer base-16, large-16, dan Huge-14 adalah Vision Transformer Base-16 dengan nilai akurasi sebesar 0.88 pada data validasi di epoch ke-16. Jika diukur dari nilai loss maka nilai loss terbaik dicapai oleh arsitektur Vision Transformer base-16 sebesar 0.30 pada data validasi di epoch ke-16.Selisih nilai loss antara data validasi dan data tes menunjukkan adanya kecerundungan model mengalami overfit.

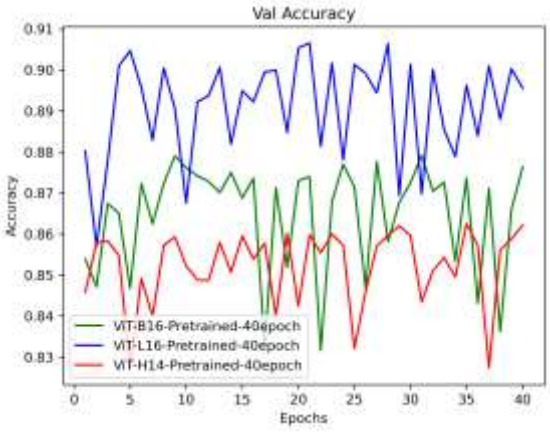
4.2 Pengujian dan Analisis Pengaruh Pretrained

Untuk Pengujian pengaruh pretrained

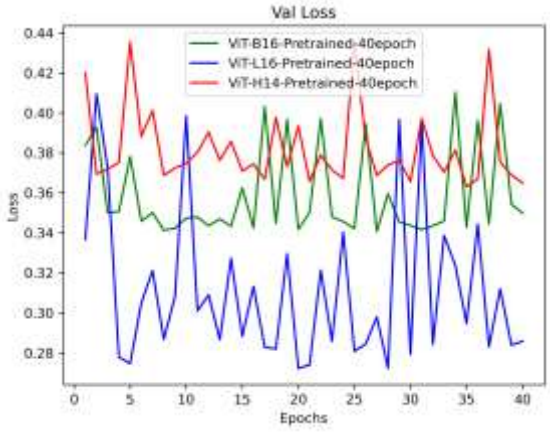
terhadap akan dilakukan dengan mem bandingkan nilai akurasi dan loss antar model Vision Transformer yang dipretrain-ed lalu setelah itu arsitektur tersebut akan dibandingkan dengan model arsitektur yang tidak di-pretrained. Nilai akurasi dan loss akan diambil pada data validasi dan tes. Dan untuk pengujian ini arsitektur yang akan diujikan ada Vision Transformer base-16, large-16, dan huge-14.

Tabel 3 Hasil penegujian Akurasi dan nilai loss berdasarkan ukuran arsitketur yang dipretrained

Tipe	Akurasi		Loss	
	Val	Test	Val	Test
Base-16	0,87	0,71	0,34	0,66
Large-16	0,90	0,75	0,27	0,60
Huge-14	0,86	0,71	0,36	0,64



Gambar 7 Hasil akurasi dari pengujian berdasarkan ukuran arsitektur yang dipretrained



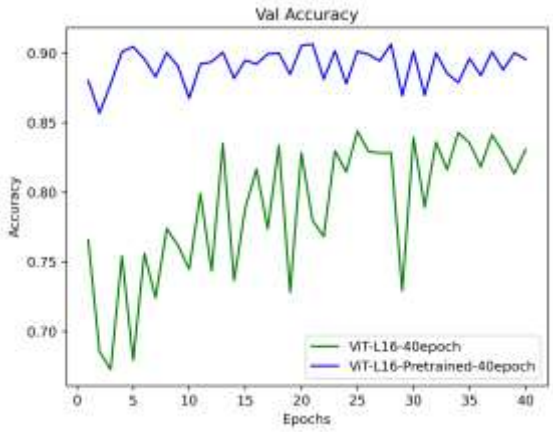
Gambar 8 hasil nilai loss dari pengujian berdasarkan ukuran arsitktur yang dipretrained

Dari pengujian pengaruh pretrained, dapat dilihat bahwa arsitektur yang memiliki akurasi tertinggi adalah Large-16 sebesar 0.90 yang didapat dari pengujian pada data validasi pada epoch ke-21. Pada epoch ke 21 pula nilai loss terbaik didapatkan oleh arsitektur Large-16 sebesar 0.27. Sesuai hasil dari pengujian diatas, maka pengujian selanjutnya untuk mengetahui efek dari pretrained ke nilai akurasi dan loss akan menggunakan arsitektur Vision Transformer Large-16.

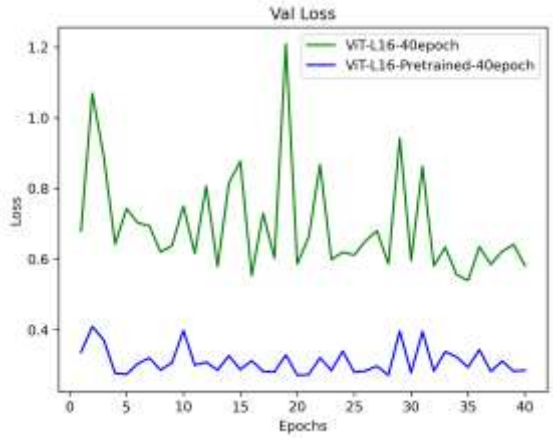
Untuk Pengujian selanjutnya mengenai pengaruh pretrained adalah antara arsitektur Vision Transformer Large-16 yang dipretrained dan tidak dipretained yang akan menggunakan parameter yang sama yaitu nilai akurasi dan loss pada data validasi dan tes.

Tabel 4 Hasil penegujian akurasi dan nilai loss berdasarkan arsitektur large-16 yang dipretrained dan tidak dipretrained

Large-16	Akurasi		Loss	
	Val	Test	Val	Test
Pretrained	0,90	0,75	0,27	0,60
Tanpa Pretrained	0,84	0,68	0,54	1,04



Gambar 9 hasil pengujian akurasi berdasarkan pretrained atau tanpa pretrained



Gambar 10 hasil pengujian nilai loss berdasarkan pretrained atau tanpa pretrained

Dari gambar 9 dan 10 dapat dilihat bahwa ada selisih yang lumayan besar antara arsitektur Vision Transformer large-16 yang di pretrained

dan tidak dipretrained dimana arsitektur yang lebih bagus secara nilai akurasi dan loss adalah Large-16 yang dipretrained. Akurasi tertinggi yang diraih oleh arsitektur Vision Transformer Large-16 yang dipretrained adalah 0.90 pada epoch 21 dibandingkan dengan yang tidak dipretrained sebesar 0.84 yang didapat pada epoch ke 25. Kemudian ketika dilihat dari nilai loss maka nilai loss terbaik yang diraih oleh arsitektur Vision Transformer large-16 yang menggunakan pretrained adalah 0.27 pada epoch ke 21 dibandingkan dengan yang tidak dipretrained sebesar 0.54 pada epoch ke 25.

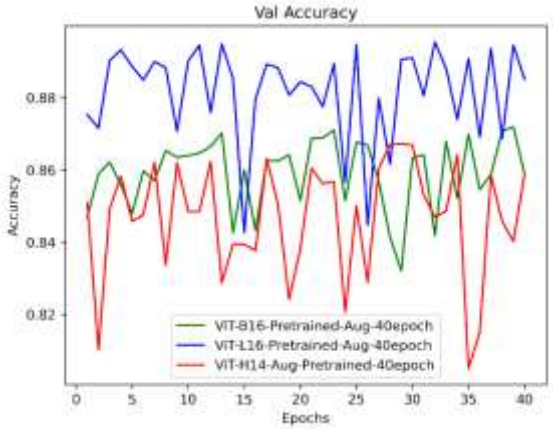
Dari pengujian ini terlihat pengaruh menggunakan pretrained dapat meningkatkan nilai akurasi dan mengurangi loss pada arsitektur Vision Transformer large-16. Namun jika kedua arsitektur dibandingkan nilai lossnya antara data validasi dan data tes, dapat diambil kesimpulan bahwa kedua model ini mengalami overfit dikarenakan nilai loss dari data validasi dan data tes terdapat selisih yang besar dan ini mempengaruhi ke nilai akurasi yang dapat diraih oleh arsitektur Vision Transformer large-16 baik yang dipretained dan yang tidak dipretained.

4.3. Pengujian dan Analisis Pengaruh Augmentasi data

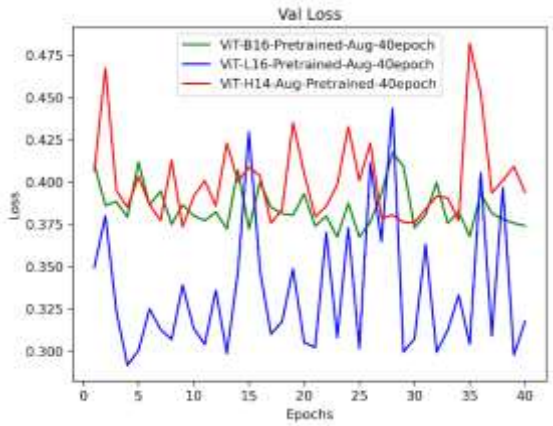
Berdasarkan pengujian sebelumnya, pretrained memberikan hasil yang lebih baik ketimbang model yang tidak dipretrained. Oleh karena itu untuk penelitian pengaruh augmentasi data akan menggunakan kombinasi dari augmentasi dan pretrained. Untuk Pengujian pengaruh augmentasi data akan dilakukan dengan membandingkan nilai akurasi dan loss antar model Vision Transformer yang diaugmentasi setelah itu arsitektur tersebut akan dibandingkan dengan model arsitektur yang sama namun tidak diaugmentasi. Nilai akurasi dan loss akan diambil pada data validasi dan tes. Dan untuk pengujian ini arsitektur yang akan diujikan ada Vision Transformer adalah base-16, large-16, dan huge-14.

Tabel 5 Pengujian hasil akurasi dan nilai loss berdasarkan ukuran arsitektur

	Akurasi		Loss	
Tipe	**Val**	**Test**	**Val**	**Test**
Base-16	0,87	0,66	0,36	0,72
Large-16	0,89	0,75	0,29	0,57
Huge-14	0,86	0,68	0,37	0,70



Gambar 11 Hasil pengujian akurasi berdasarkan ukuran arsitektur yang diaugmentasi



Gambar 12 Hasil pengujian nilai loss berdasarkan ukuran arsitktur yang diaugmentasi

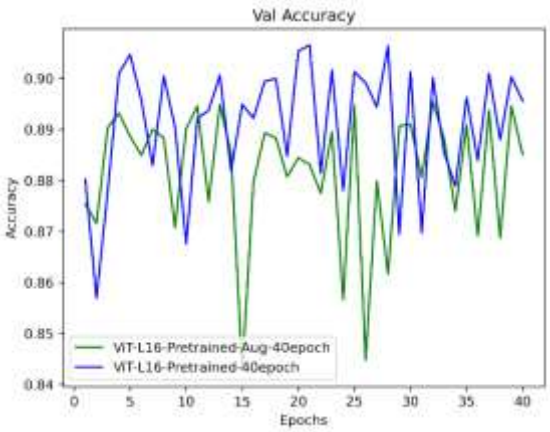
Dari pengujian pengaruh augmentasi data, dapat dilihat bahwa arsitektur dipretrained dan diaugmentasi yang memiliki akurasi tertinggi adalah Large-16 sebesar 0.89 yang didapat dari pengujian pada data validasi pada epoch ke-32. Pada epoch ke 32 pula nilai loss terbaik didapatkan oleh arsitektur Large-16 sebesar 0.29. Sesuai hasil dari pengujian diatas, maka pengujian selanjutnya untuk mengetahui efek dari augmentasi ke nilai akurasi dan loss akan menggunakan arsitektur Vision Transformer Large-16.

Untuk Pengujian selanjutnya mengenai pengaruh augmentasi adalah antara arsitektur Vision Transformer Large-16 yang diaugmentasi dan tidak diaugmentasi. Pengujiannya akan menggunakan parameter yang sama yaitu nilai akurasi dan loss pada data validasi dan tes.

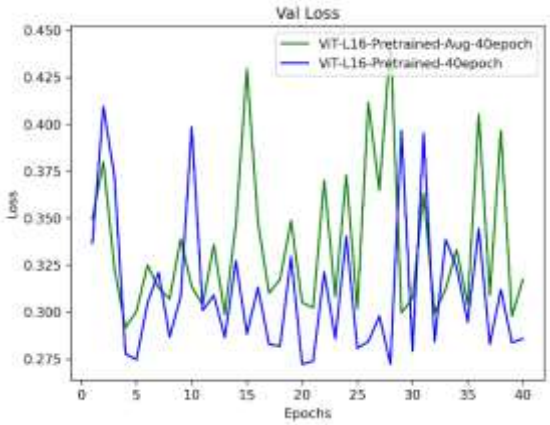
Tabel 6 Hasil pengujian akurasi dan nilai loss pada

arsiterktur yang diaugmentasi

Large-16	Akurasi		Loss	
	Val	Test	Val	Test
Augmenetasi	0,90	0,75	0,27	0,60
Tanpa Augmnetasi	0,89	0,75	0,29	0,57



Gambar 13 Hasil pengujian nilai akurasi pada arsitektur yang diaugmentasi dan tidak diaugmentasi



Gambar 14 Hasil pengujian nilai loss pada arsitektur yang diaugmentasi dan tidak diaugmentasi

Dari gambar 13 dan 14 dapat dilihat bahwa untuk akurasi dan loss, arsitektur *vision transformer Large-16* yang tanpa di augmentasi memiliki hasil yang lebih baik. Akurasi model yang tidak di augmentasi paling tinggi didapatkan sebesar 0,90 sedangkan yang diaugmentasi justru akurasinya dibawah model yang tidak diaugmentasi sebesar 0,89. Apabila diperhatikan dari nilai loss, model yang tidak diaugmentasi memiliki nilai loss yang lebih rendah dibanding dari model yang diaugmentasi. Hasil ini kemungkinan diperoleh dikarenakan salah satu metode augmentasi yaitu *gaussian blur* semakin mengaburkan gambar yang sudah minim fitur, setelah diaugmentasi fiturnya semakin sulit dikenali oleh model.

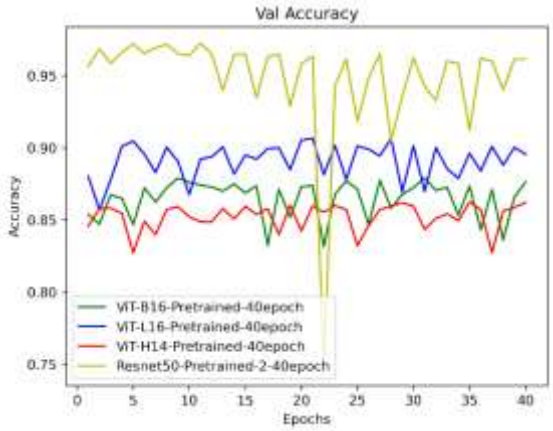
Dari pengujian pengaruh augmentasi data terlihat pengaruh menggunakan augmentasi dapat menurunkan nilai akurasi dan *loss* pada arsitektur *Vision transformer large-16*. Dan ketika kedua arsitektur dibandingkan nilai *loss*nya antara data validasi dan data tes, dapat dilihat bahwa kedua model ini mengalami overfit dikarenakan nilai *loss* dari data validasi dan data tes terdapat selisih yang besar dan ini mempengaruhi ke nilai akurasi yang dapat diraih oleh arsitektur *Vision transformer large-16* baik yang diaugmentasi dan yang tidak diaugmentasi.

4.4. Pengujian Seluruh Arsitktur

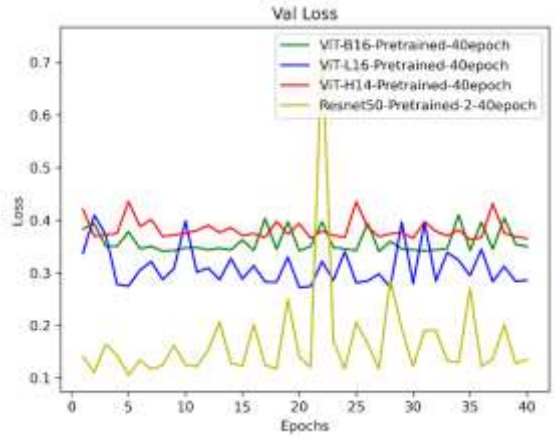
Berdasarkan pengujian sebelumnya, didapatkan hasil bahwa pretrained meningkatkan akurasi dan mengurangi nilai loss dari arsitktur tersebut. Maka dari itu, Pengujian seluruh arsitktur akan dilakukan menggunakan pretrained dan tanpa augmentasi data. Tabel 7 menunjukkan hasil akurasi dan loss dari setiap arsitektur yang dipretrained sebanyak 40 *Epoch*. Untuk gambar 6.11 menggambarkan nilai akurasi semua model selama 40 *Epoch*. Untuk gambar 6.12 menggambarkan nilai loss semua model selama 40 *Epoch*.

Tabel 7 hasil pengujian seluruh arsitektur

Tipe	Akurasi		Loss	
	Val	Test	Val	Test
Base-16	0,87	0,71	0,34	0,66
Large-16	0,90	0,75	0,27	0,60
Huge-14	0,86	0,71	0,36	0,64
Resnet50	0,96	0,95	0,12	0,19



Gambar 15 Hasil pengujian akurasi seluruh arsitektur

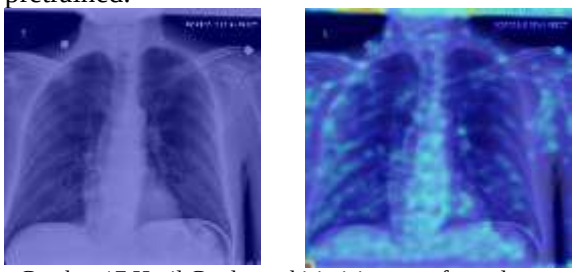


Gambar 16 Hasil pengujian nilai loss seluruh arsitektur

Dari pengujian seluruh arsitektur dapat dilihat arsitektur *vision transformer* yang terbaik masih eeebelum bisa mendekati akurasi dari RessNet50, Keterbatasan jumlah data yang kurang banyak mungkin adalah salah satu alasan mengapa pada penelitian ini *vision transformer* kurang bersaing. Akurasi paling tinggi didapati oleh arsitektur Resnet50 sebesar **0,96** dan nilai loss sebesar 0,12. terlepas dari hasil yang baik dari model Resnet50, didapati perbedaan nilai loss pada data latih dan data tes. Ini menunjukkan adanya indikasi overfitting yang dialami oleh model Resnet50.

4.5. Hasil GracCAM

Dalam pengujian hasil *gradCAM* ini akan membandingkan bagaimana visualisasi pada arsitektur *vision transformer large*-16 *dipretrained* dan *resnet50 dipretrained*. Gambar 17 merupakan hasil *gradCAM* Ketika dijalankan pada arsitektur *Vision transformer Large*-16 dan hasil *gradCAM* Ketika dijalankan pada arsitektur *ResNet50* yang keduanya menggunakan pretrained.



Gambar 17 Hasil Gradcam: kiri vision transformr large-16, kanan resnet50

5. Kesimpulan dan Saran

Pengujian dilakukan terhadap pengaruh pretrained dan pengaruh augmentasi terhadap hasil pengujian. Dari pengujian tersebut didapatkan bahwa pada penelitian kali ini pretrained atau transfer learning berhasil meningkatkan akurasi dari arsitektur vision transformer. Sedangkan augmentasi data yang menggunakan gaussian blur dan colorjitter belum berhasil meningkatkan akurasi dari arsitektur vision transformer. hal ini dapat disebabkan dari kurang cocoknya augmentasi yang digunakan terhadap model data yang berupa foto x-ray. Hasil terbaik didapatkan dari arsitektur vision transformer large-16 yang di pretrained dengan akurasi 0.906 pada data validasi dan 0.759 pada data tes. Perbedaan yang signifikan antara akurasi pada data validasi dan data tes menandakan bahwa ada overfitting model yang terjadi. Beberapa Saran dari penelitian ini yaitu perlunya mencari cara lainnya agar kendala overfit ini dapat teratasi, perlu juga pemilihan metode augmentasi data yang lebih baik lagi sehingga memungkinkan untuk meningkatkan akurasi dari arsitektur vision transformer. dataset dengan kelas yang lebih terbagi rata jumlah datanya mungkin juga akan meningkatkan akurasi dari arsitektur vision transformer.

6. DAFTAR PUSTAKA

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Farooq, M., Hafeez, A., 2020, COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs. https://doi.org/10,48550/arXiv.2003.14395

Hendrycks, D., Gimpel, K., 2020, Gaussian Error Linear Units (GELUs). https://doi.org/10,48550/arXiv.1606.08415

Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., Ren, R., Leung, K.S.M., Lau, E.H.Y., Wong, J.Y., Xing, X., Xiang, N., Wu, Y., Li, C., Chen, Q., Li, D., Liu, T., Zhao, J., Liu, M., Tu, W., Chen, C., Jin, L., Yang, R., Wang, Q., Zhou, S., Wang, R., Liu, H., Luo, Y., Liu, Y., Shao, G., Li, H., Tao, Z., Yang, Y., Deng, Z., Liu, B., Ma, Z., Zhang, Y., Shi, G., Lam, T.T.Y., Wu, J.T., Gao, G.F., Cowling, B.J., Yang,

B., Leung, G.M., Feng, Z., 2020, Early Transmission Dynamics in Wuhan, China, of Novel Coronavirus–Infected Pneumonia. N. Engl. J. Med. 382, 1199–1207. https://doi.org/10,1056/NEJMoa2001316

Perez, L., Wang, J., 2017. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. https://doi.org/10,48550/arXiv.1712.04621

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. Presented at the Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626.

Velavan, T.P., Meyer, C.G., 2020, The COVID-19 epidemic. Trop. Med. Int. Health 25, 278–280, https://doi.org/10,1111/tmi.13383

Wang, L., Lin, Z.Q., Wong, A., 2020, COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci. Rep. 10, 19549. https://doi.org/10,1038/s41598-020-76550-z

Ying, X., 2019. An Overview of Overfitting and its Solutions 1168, 022022. https://doi.org/10,1088/1742-6596/1168/2/022022