

# Perbandingan Deteksi Letak Polip pada Citra *Colonoscopy* menggunakan CNN dengan Arsitektur RetinaNet

RONALDO DAVE JONATHAN, MEILAN JIMMY HASUGIAN, ERWANI MERRY  
SARTIKA\*

Program Studi Teknik Elektro, Universitas Kristen Maranatha, Indonesia  
Email: [erwanimerry@gmail.com](mailto:erwanimerry@gmail.com)\*

*Received* 21 Juli 2022 | *Revised* 30 Agustus 2022 | *Accepted* 22 September 2022

## ABSTRAK

*Penyakit kanker kolorektal diawali munculnya polip pada usus besar yang dapat berubah menjadi tumor ganas dan menimbulkan kanker. Sehingga diperlukan screening terhadap usus besar menggunakan colonoscopy. Menurut penelitian sekitar 26% polip terlewat saat prosedur colonoscopy. Pada penelitian ini dilakukan implementasi Convolutional Neural Network (CNN) dengan arsitektur RetinaNet untuk mendeteksi letak polip pada citra colonoscopy. Perbandingan dilakukan pada 3 jenis arsitektur yaitu ResNet-50, ResNet-101, dan ResNet-152 sebagai backbone pada arsitektur RetinaNet. Model yang terbaik berdasarkan metrik Intersection over Union (IoU) adalah model RetinaNet (Backbone = ResNet-50) tanpa data augmentation dengan nilai 0.8415. Sedangkan model yang terbaik berdasarkan metrik Average Precision (AP) adalah RetinaNet (Backbone = ResNet-101) dengan data augmentation dengan nilai  $AP_{25} = 0.9308$ ,  $AP_{50} = 0.9039$ ,  $AP_{75} = 0.6985$ .*

**Kata kunci:** *polip, colonoscopy, Convolutional Neural Network (CNN), RetinaNet*

## ABSTRACT

*Colorectal cancer always begins with the appearance of polyps in the colon which can turn into malignant tumors and cause cancer. Therefore, it is necessary to screen the large intestine using colonoscopy. However, according to studies, about 26% of polyps are missed during colonoscopy procedures. In this study, a Convolutional Neural Network (CNN) with RetinaNet architecture was implemented to detect the location of polyps in colonoscopy images. Comparisons were made on 3 types of architecture, namely ResNet-50, ResNet-101, and ResNet-152. From the evaluation results, the best model based on the Intersection over Union (IoU) metric is the RetinaNet model (Backbone = ResNet-50) without augmentation data with a value of 0.8415. While the best model based on the Average Precision (AP) metric is RetinaNet (Backbone = ResNet-101) with data augmentation with values  $AP_{25} = 0.9308$ ,  $AP_{50} = 0.9039$ ,  $AP_{75} = 0.6985$ .*

**Keywords:** *polyp, colonoscopy, Convolutional Neural Network (CNN), RetinaNet*

## 1. PENDAHULUAN

Kanker kolorektal adalah penyakit yang disebabkan karena perkembangan sel yang tidak terkendali pada usus besar. Di Amerika Serikat, urutan kedua penyebab kematian akibat kanker disebabkan oleh kanker kolorektal. Pada tahun 2020, terdapat 147.950 kasus kanker kolorektal dengan kematian mencapai 53.200 jiwa (**Siegel, dkk, 2020**). Kanker kolorektal selalu diawali dengan munculnya polip yang merupakan pertumbuhan jaringan non-kanker yang muncul pada lapisan *musocal* pada usus besar (**ACS, 2020**). Akan tetapi, jika tidak ditangani lebih lanjut, polip dapat berubah menjadi tumor ganas dan menimbulkan kanker. Oleh sebab itu, polip harus segera dibuang agar meminimalkan resiko kanker kolorektal (**Pidala & Cusick, 2017**). Proses *screening* terhadap usus besar diperlukan untuk memeriksa adanya polip yang berpotensi menjadi kanker.

Metode paling umum untuk melakukan *screening* terhadap usus besar adalah dengan menggunakan *colonoscopy*. *Colonoscopy* adalah sebuah prosedur medis untuk melihat kondisi di dalam usus besar yang dilakukan dengan cara memasukkan sebuah selang fleksibel yang dilengkapi dengan kamera melalui anus (**Dobie, dkk, 2012**). Salah satu masalah yang dialami saat *screening* usus besar menggunakan *colonoscopy* adalah polip yang secara tidak sengaja terlewat oleh dokter. Menurut penelitian, sekitar 26% polip terlewat saat *colonoscopy* dilakukan (**Cheng, dkk, 2021**). Hal ini disebabkan oleh beberapa faktor, salah satunya yaitu ketidaktepatan dokter saat melakukan prosedur *colonoscopy* pada pasien (**Choi, dkk, 2014**). Untuk menghindari permasalahan tersebut, diperlukan sebuah sistem berbasis komputer yang dapat mendeteksi adanya polip secara otomatis.

Deteksi polip secara otomatis menggunakan sistem berbasis komputer sudah pernah dilaksanakan pada kompetisi "*The Automatic Polyp Detection subchallenge*" yang diselenggarakan oleh *Medical Image Computing and Computer Assisted Intervention (MICCAI)* pada tahun 2015. Kompetisi ini memiliki tujuan utama, yaitu melakukan deteksi lokasi polip pada citra yang diambil dari video *colonoscopy* sehingga diharapkan dapat mengurangi *miss-rate* saat melakukan prosedur *colonoscopy* pada pasien. Laporan hasil kompetisi menunjukkan metode yang paling baik adalah menggunakan *Convolutional Neural Network (CNN)* karena memiliki nilai metrik yang paling unggul dibandingkan metode yang lain, seperti metode *hand-crafted* (**Bernal, dkk, 2017**).

*Convolutional Neural Network (CNN)* adalah jenis *neural network* yang umum digunakan dalam *computer vision*, dan *Natural Language Processing (NLP)* (**Albawi, dkk, 2017**). CNN umumnya tersusun atas 3 lapisan utama yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. *Convolution layer* dan *pooling layer* digunakan untuk melakukan *feature extraction*, sedangkan *fully connected layer* digunakan untuk memetakan *feature* menjadi keluaran akhir (**Patil & Rane, 2021**). Salah satu arsitektur CNN yang digunakan untuk mendeteksi objek pada sebuah citra adalah RetinaNet.

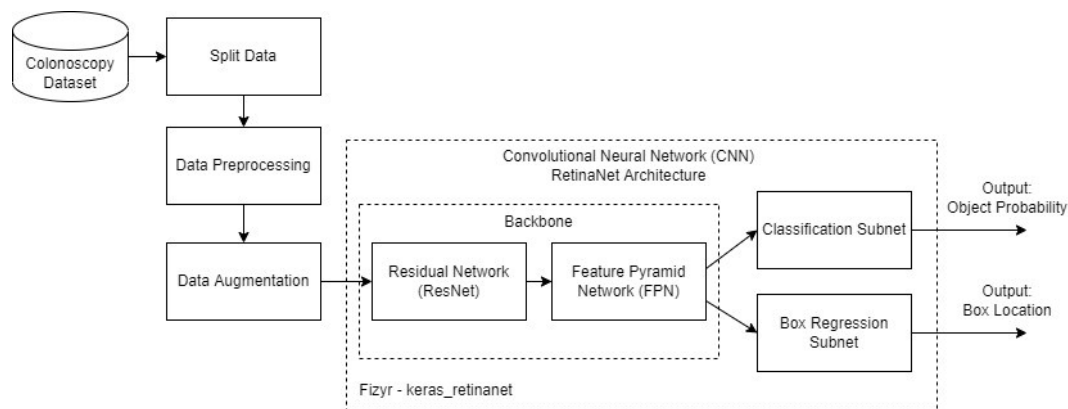
Arsitektur RetinaNet merupakan *one stage detector* yang menggunakan 2 arsitektur lain yang menjadi *backbone* yaitu *Residual Network (ResNet)* yang merupakan pemenang kompetisi *ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC 2015)* dan *Feature Pyramid Network (FPN)* (**He, dkk, 2016**). RetinaNet adalah arsitektur CNN yang dinilai sangat baik untuk menyelesaikan masalah deteksi objek. Hal tersebut dibuktikan dengan fakta bahwa RetinaNet mempunyai performa yang lebih baik jika dibandingkan dengan *two stage detector* seperti arsitektur *Region-based Convolutional Neural Network (RCNN)* saat di tes menggunakan dataset *Microsoft Common Objects in Context (MS COCO)* (**Lin, dkk, 2020**). Maka berdasarkan latar belakang yang telah dijelaskan sebelumnya, pada penelitian ini akan

dilakukan penerapan CNN dengan membandingkan arsitektur RetinaNet (ResNet-50, ResNet-101, dan ResNet-152) untuk deteksi terbaik letak polip pada citra colonoscopy.

## 2. METODE PENELITIAN

### 2.1 Diagram Blok Sistem

Pada Penelitian ini, dilakukan penerapan *Convolutional Neural Network* (CNN) dengan membandingkan arsitektur RetinaNet yang digunakan (ResNet-50, ResNet-101, dan ResNet-152) untuk memperoleh deteksi letak polip pada citra *colonoscopy* yang terbaik. Diagram blok dari proses deteksi polip pada penelitian ini diperlihatkan pada Gambar 1.



**Gambar 1. Diagram Blok Sistem Deteksi Letak Polip**

Dataset yang digunakan didapatkan dari tiga sumber yang berbeda. Dataset yang digunakan pertama-tama dilakukan proses *split* data untuk memisahkan antara data untuk pelatihan, validasi dan test. Setelah itu dilakukan *data preprocessing* untuk mengolah semua data sebelum masuk ke dalam model CNN. Setelah semua data diolah, data *training* akan diperbanyak pada proses *data augmentation*. Setelah itu, dilakukan pelatihan model menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur RetinaNet. Keluaran dari model CNN yang dibuat adalah probabilitas kelas objek dan lokasi *bounding box* objek.

### 2.2 Dataset Colonoscopy

Dataset yang digunakan yaitu dataset CVC-ClinicDB (Bernal, dkk, 2015), dataset ETIS-Larib (Silva, dkk, 2014), dan dataset Kvasir SEG (Jha, dkk, 2020). Ketiga dataset tersebut berisikan sampel citra polip pada usus besar yang diambil dari video *colonoscopy*. Banyaknya jumlah sampel data yang ada pada tiap dataset diperlihatkan pada Tabel 1.

**Tabel 1. Banyak Sampel Data pada Setiap Dataset**

Dataset	Banyak Data	Total	Resolusi	Format file
CVC-ClinicDB	612 buah citra	1808 buah citra	384 x 288	.tif
ETIS-Larib	1000 buah citra		1225 x 966	.tif
Kvasir SEG	196 buah citra		332x487 sampai 1920x1072	.jpg

Setiap citra baik dari semua dataset tersebut dilengkapi dengan *ground truth* berbentuk *mask* yang menunjukkan letak polip pada usus yang sebenarnya. Ketiga dataset tersebut memiliki resolusi citra dan format file yang berbeda satu dengan yang lainnya.

### 2.3 Split Data

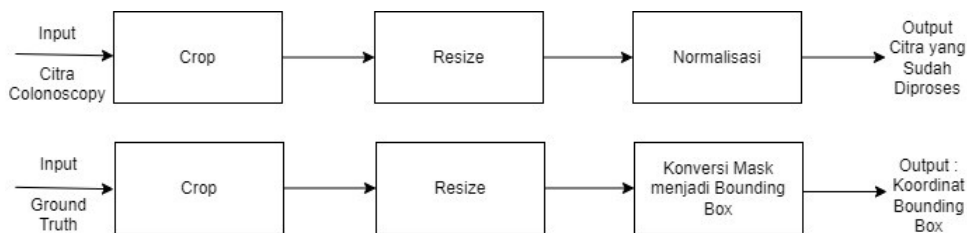
*Split Data* adalah proses yang digunakan untuk memisahkan antara *training data*, *validation data*, dan *test data*. *Training data* adalah data yang akan dipelajari oleh model. *Validation data* adalah data yang digunakan untuk melakukan validasi terhadap model sebagai acuan untuk mengubah parameter dari model yang dibuat. Sedangkan *test data* adalah data yang digunakan untuk evaluasi dan menguji performa dari model CNN yang dibuat. Rasio dari *training data*, *validation data*, dan *test data* yang dipakai diperlihatkan pada Gambar 2. Digunakan rasio 80% untuk *training data*, 10% untuk *validation data*, dan 10% untuk *test data* dari 1808 buah citra. Setelah dilakukan *split data*, diperoleh *training data* sebanyak 1448 buah citra, *validation data* sebanyak 180 buah citra, dan *test data* sebanyak 180 buah citra.



Gambar 2. Rasio *Split Data*

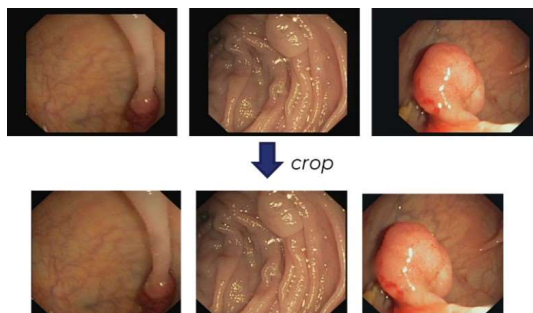
### 2.4 Data Preprocessing

*Data preprocessing* adalah sebuah proses yang digunakan untuk mengolah data yang akan dimasukkan kedalam model CNN. Diagram blok dari *data preprocessing* diperlihatkan pada Gambar 3.



Gambar 3. Diagram Blok *Data Preprocessing*

Pertama-tama dilakukan proses *crop* pada citra dan *ground truth*. Proses *crop* dilakukan untuk membuang *border* hitam disekitar citra. Proses *crop* hanya akan dilakukan terhadap dataset CVC-ClinicDB dikarenakan terdapat *border* hitam yang cukup tebal disekitar citra. Contoh sampel data dari CVC-Clinic DB sebelum dilakukan *crop* diperlihatkan pada Gambar 4.



Gambar 4. Contoh Sampel Citra Sebelum dan Sesudah Proses *Crop*

Setelah itu dilakukan proses *resize* untuk mengubah ukuran data agar seragam. Semua citra dan *ground truth* akan diubah ukurannya menjadi 512x512 piksel. Tahapan selanjutnya adalah melakukan normalisasi pada citra untuk mengubah rentang nilai piksel dari 0-255 menjadi 0-1. Normalisasi dilakukan dengan menggunakan Persamaan (1).

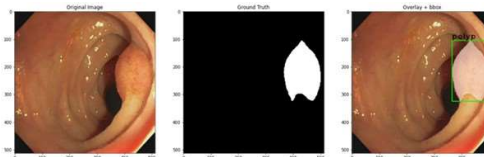
$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Selanjutnya dilakukan konversi *ground truth* yang berbentuk *mask* menjadi koordinat *bounding box*. Semua *ground truth* harus dikonversi menjadi bentuk koordinat *bounding box* supaya sesuai dengan keluaran dari model RetinaNet untuk melakukan deteksi objek pada citra. Koordinat *bounding box* yang dihasilkan mempunyai format x1, y1, x2, y2. Nilai x1 menyatakan titik sumbu x minimum, nilai x2 menyatakan titik sumbu x maksimum, nilai y1 menyatakan titik sumbu y minimum, dan nilai y2 menyatakan titik sumbu y maksimum pada *bounding box* yang dihasilkan. Selanjutnya, koordinat tersebut di simpan dalam bentuk *dataframe*. Lima baris pertama *dataframe* dari *training data*, *validation data*, dan *test data* ditampilkan pada Gambar 5.

Training Data						Validation Data						Test Data								
img_path	x1	y1	x2	y2	class_name	img_path	x1	y1	x2	y2	class_name	img_path	x1	y1	x2	y2	class_name			
0	./train/0.png	167	110	470	512	polyp	0	./validation/0.png	24	303	401	498	polyp	0	./test/0.png	118	89	308	267	polyp
1	./train/1.png	82	82	289	384	polyp	1	./validation/1.png	220	202	365	379	polyp	1	./test/1.png	97	307	290	463	polyp
2	./train/2.png	182	33	400	277	polyp	2	./validation/2.png	265	185	506	512	polyp	2	./test/2.png	22	105	380	512	polyp
3	./train/3.png	190	122	473	470	polyp	3	./validation/3.png	0	46	512	512	polyp	3	./test/3.png	190	240	386	426	polyp
4	./train/4.png	159	149	440	471	polyp	4	./validation/3.png	350	340	512	512	polyp	4	./test/4.png	283	210	357	277	polyp

**Gambar 5. Dataframe dari Training Data, Validation Data, dan Test Data**

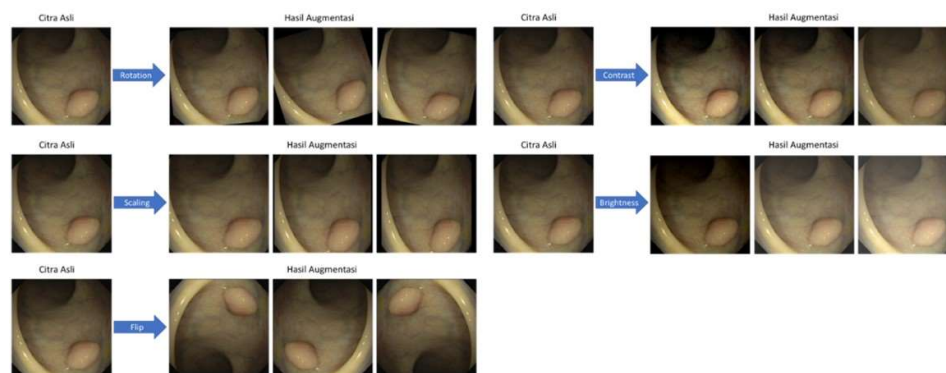
*Dataframe* yang dibuat adalah sebuah tabel yang berisikan kolom yang meliputi *img\_id*, *x1*, *y1*, *x2*, *y2*, dan *class\_name*. Kolom *img\_path* berisikan lokasi data citra disimpan. Sedangkan kolom *class\_name* berisikan nama kelas dari objek yaitu *polyp*. Contoh visualisasi dari *bounding box* yang dihasilkan ditampilkan pada Gambar 6.



**Gambar 6. Contoh Visualisasi Bounding Box**

## 2.5 Data Augmentation

Untuk menambah variasi data yang akan di pelajari oleh model, dilakukan proses *data augmentation*. Teknik *data augmentation* yang akan dilakukan meliputi *rotation*, *scaling*, *flip*, *contrast*, dan *brightness*. Contoh hasil dari data augmentation dengan teknik *rotation*, *scaling*, *flip*, *contrast*, dan *brightness* diperlihatkan pada Gambar 7.

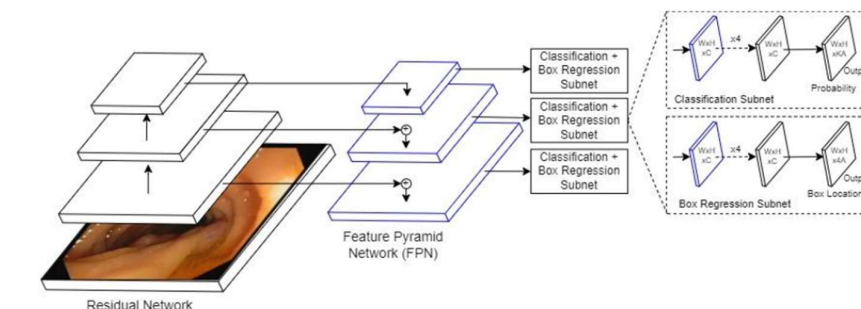


**Gambar 7. Contoh Hasil Data Augmentation**

*Rotation* dilakukan dengan cara memutar citra secara acak. Besarnya putaran citra yang dipakai berkisar antara  $-0,1$  sampai  $0,1$  radian. *Scaling* dilakukan dengan cara memperbesar atau memperkecil citra secara acak. Besarnya *scaling* pada citra yang dipakai berkisar antara  $0,9$  sampai  $1,1$  kali. *Flip* dilakukan dengan cara membalikan citra secara horizontal maupun vertikal. Peluang yang digunakan untuk sebuah citra mengalami proses *flip* secara vertikal atau horizontal yaitu  $0.5$ . *Contrast* dilakukan dengan menurunkan atau menaikkan kontras dari citra. Banyaknya perubahan *contrast* pada citra berkisar antara nilai  $0,9$  sampai  $1,1$ . *Brightness* dilakukan dengan cara menurunkan atau menaikkan tingkat kecerahan pada citra secara acak. Banyaknya perubahan kecerahan pada citra berkisar antara nilai  $-0,1$  sampai  $0,1$ .

## 2.6 Pelatihan Model CNN

Pada tahap ini, dilakukan pelatihan model *Convolutional Neural Network* (CNN). Digunakan Tensorflow versi 2.3.1 untuk melatih model CNN. Arsitektur CNN yang dipakai adalah RetinaNet. Digunakan tiga jenis arsitektur *Residual Network* (ResNet) yang menjadi *backbone* pada arsitektur RetinaNet. Ketiga jenis arsitektur tersebut adalah ResNet-50, ResNet-101, dan ResNet-152. Ketiga model tersebut berperan sebagai *feature extractor* yang berfungsi untuk melakukan ekstraksi *feature* dari citra masukan pada model RetinaNet. Model ResNet yang dipakai merupakan model *pretrained* yang sudah dilatih dengan dataset ImageNet. Arsitektur RetinaNet ditampilkan pada Gambar 8 (Lin, dkk, 2020).



Gambar 8. Arsitektur RetinaNet

Proses dari RetinaNet diawali dengan mengumpukan citra input ke *Residual Network* (ResNet) dan *Feature Pyramid Network* (FPN) yang menjadi *backbone* dari arsitektur RetinaNet. ResNet dan FPN berfungsi sebagai *feature extractor* yang akan menghasilkan *feature map*. Dengan adanya FPN, *feature map* yang dihasilkan mempunyai skala yang berbeda sehingga dapat digunakan untuk mendeteksi objek dengan skala yang berbeda. Setiap *feature map* yang telah dihasilkan selanjutnya akan diumpukan ke dalam *classification subnet* untuk memprediksi kelas objek dan *box regression subnet* untuk memprediksi koordinat *bounding box*. Pada *classification subnet* digunakan *loss function* berupa *Focal Loss*. Sedangkan pada *box regression subnet* digunakan *loss function* berupa *Smooth-L1 Loss*. Untuk *anchor box* yang dipakai mempunyai luas berkisar antara  $32^2$  sampai  $512^2$  yang dimulai dari level piramida yang terkecil sampai yang terbesar. Pada setiap level piramida, digunakan *anchor box* dengan rasio  $\{1:2, 1:1, 2:1\}$ . Selain itu, pada setiap level piramida juga dipakai *anchor box* dengan ukuran  $\{2^0, 2^{1/3}, 2^{2/3}\}$  pada setiap rasio *anchor box*. Konfigurasi dari model CNN yang digunakan saat proses pelatihan ditunjukkan pada Tabel 2.

*Learning\_rate* adalah sebuah nilai yang menyatakan seberapa banyak perubahan bobot pada model saat proses pelatihan setiap pengulangan *epoch*. *Epoch* merupakan seberapa banyak pengulangan yang dilakukan pada saat proses pelatihan. *Steps\_per\_epoch* menyatakan seberapa banyak batch data yang dipelajari oleh model setiap pengulangan *epoch*. *Verbose* berfungsi untuk menampilkan informasi yang berkaitan pada saat proses pelatihan. *Callback* adalah sebuah konfigurasi dari model yang berfungsi untuk menyimpan bobot parameter

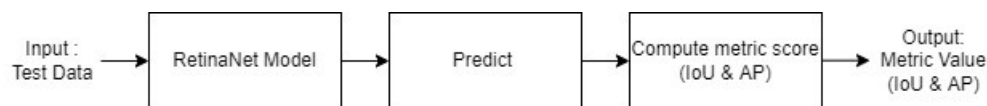
model yang terbaik pada saat proses pelatihan. Algoritma optimasi atau *optimizer* yang dipakai saat pelatihan model yaitu *Stochastic Gradient Descent* (SGD).

**Tabel 2. Konfigurasi Pelatihan pada Model CNN**

Konfigurasi Pelatihan	Nilai
learning_rate	$10^{-3}$
epochs	50
Steps_per_epoch	400
verbose	1
callbacks	ModelCheckpoint, CSVLogger, ReduceLROnPlateau
Optimizer	<i>Stochastic Gradient Descent</i> (SGD)

## 2.6 Evaluasi Model CNN

Evaluasi model CNN digunakan untuk mengetahui seberapa baik model dalam memprediksi polip pada citra *colonoscopy*. Evaluasi terhadap model CNN yang sudah dilatih dilakukan menggunakan *test data* yang berjumlah 180 buah citra. Ukuran metrik yang digunakan untuk menilai model dalam mendeteksi objek yaitu *Intersection over Union* (IoU) dan *Average Precision* (AP). Diagram blok evaluasi model CNN ditunjukkan pada Gambar 9.



**Gambar 9. Diagram Blok Evaluasi Model CNN**

Nilai IoU menyatakan seberapa overlap hasil prediksi dengan ground truth. Semakin banyak area yang *overlap* antara prediksi model dan ground truth, maka semakin besar juga nilai IoU yang dihasilkan. Nilai metrik IoU dinyatakan pada Persamaan (2).

$$Intersection\ over\ Union\ (IoU) = \frac{Prediction \cap GroundTruth}{Prediction \cup GroundTruth} \quad (2)$$

Sedangkan nilai AP menyatakan seberapa presisi model dalam melakukan deteksi objek. Pada penelitian ini akan dipakai perhitungan AP pada pada threshold IoU 0,25 (AP<sub>25</sub>), 0,50 (AP<sub>50</sub>), 0,75 (AP<sub>75</sub>). Nilai AP didapatkan dengan menghitung area dari kurva *precision-recall*. *Recall* dan *precision* sendiri dihitung melalui Persamaan (3) dan Persamaan (4)

$$recall = \frac{T_p}{T_p + F_n} \quad (3)$$

$$precision = \frac{T_p}{T_p + F_p} \quad (4)$$

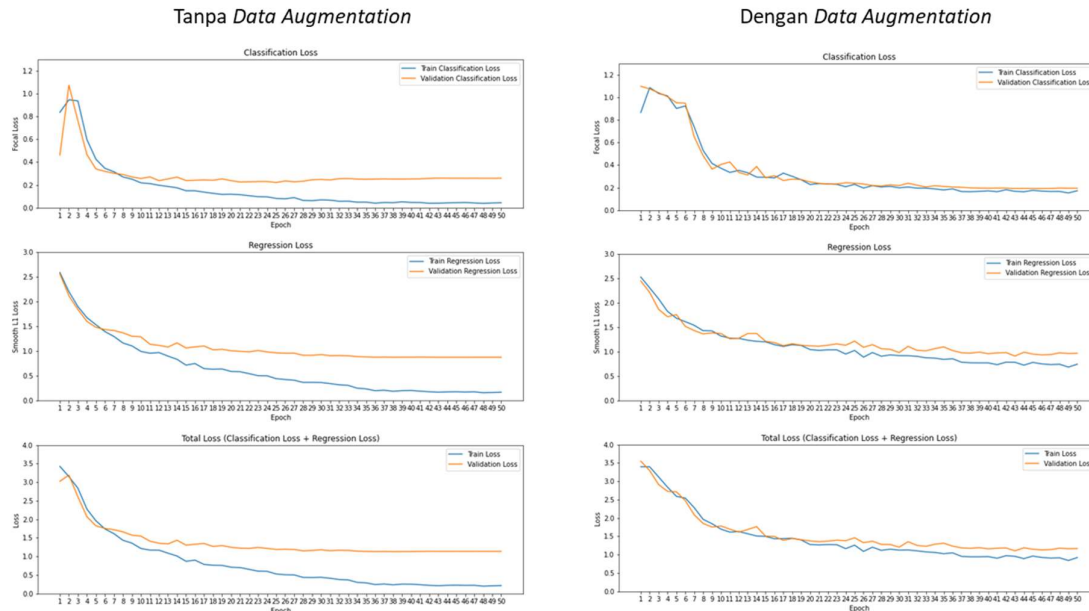
## 3. HASIL DAN PEMBAHASAN

### 3.1 Kurva Pelatihan Model

Pada proses pelatihan model RetinaNet menggunakan *backbone* ResNet-50, dicari nilai dari *classification loss*, *regression loss*, dan *total loss*. Kurva dari *classification loss*, *regression loss*, dan *total loss* pada saat proses pelatihan model RetinaNet dengan *backbone* ResNet-50 ditampilkan pada Gambar 10.

# Perbandingan Deteksi Letak Polip pada Citra *Colonoscopy* menggunakan CNN dengan Arsitektur RetinaNet

Backbone : ResNet-50

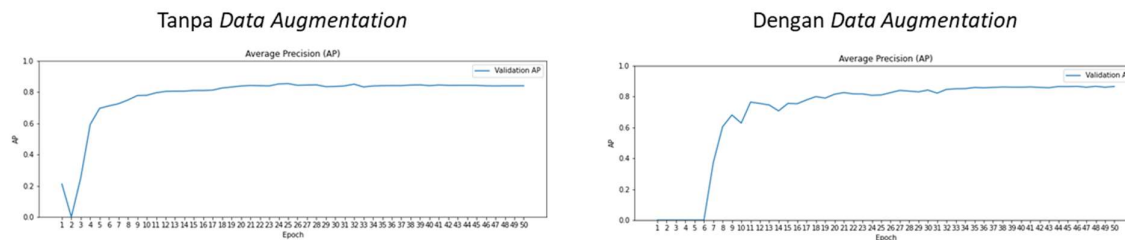


**Gambar 10. Kurva Loss Pelatihan RetinaNet (Backbone: ResNet-50)**

Berdasarkan kurva loss yang ditunjukkan pada Gambar 10, terlihat model yang dilatih tanpa *data augmentation* cenderung mengalami *overfitting*. *Overfitting* merupakan kondisi saat model mempelajari data latih terlalu detail, sehingga performa model hanya baik terhadap data yang dipelajari oleh model tersebut. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang perbedaannya semakin menjauh saat *epoch* bertambah (mulai epoch ke 6). Sedangkan model dengan *data augmentation* mempunyai kecenderungan *overfitting* yang lebih rendah. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang memiliki perbedaan yang tidak terlalu jauh saat *epoch* bertambah (epoch ke 20 mulai tampak ada perbedaan namun tidak terlalu jauh).

Setelah itu dilakukan validasi terhadap model yang dibuat menggunakan *validation* data yang berjumlah 180 buah citra. Bobot parameter dari model yang dilatih akan disimpan berdasarkan nilai *validation AP* yang terbaik. Pada pelatihan menggunakan *backbone* ResNet-50 tanpa *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-25 dengan nilai 0.8550. Sedangkan pada pelatihan menggunakan *backbone* ResNet-50 dengan *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-48 dengan nilai 0.8675. Kurva *validation AP* model RetinaNet dengan *backbone* ResNet-50 ditunjukkan pada Gambar 11.

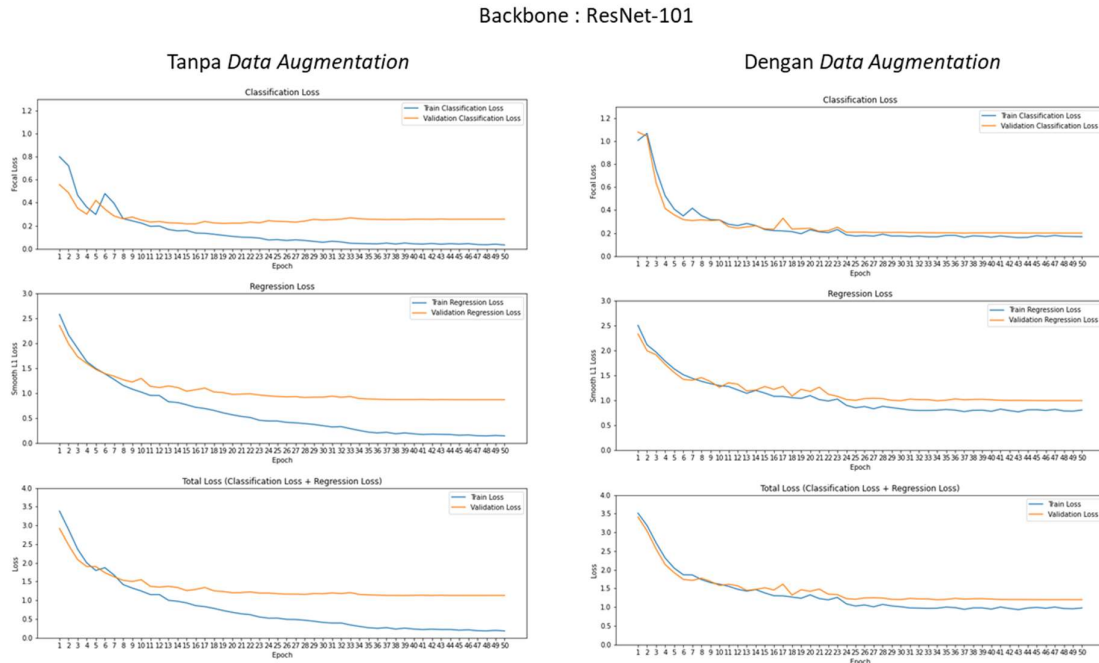
Backbone : ResNet-50



**Gambar 11. Kurva Validasi Pelatihan RetinaNet (Backbone: ResNet-50)**



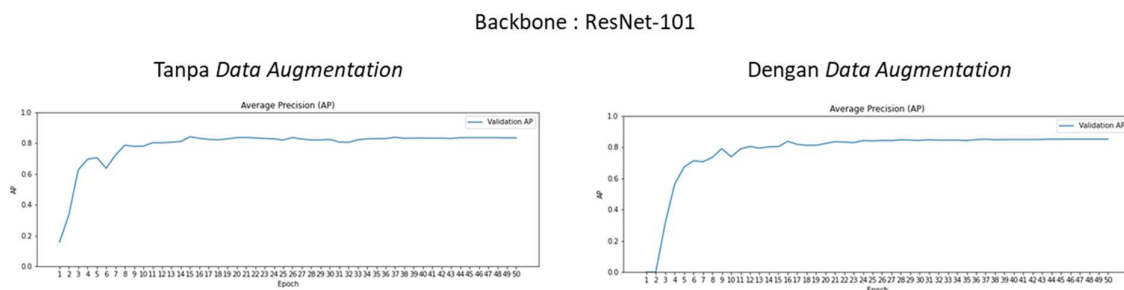
Pada proses pelatihan model RetinaNet menggunakan *backbone* ResNet-101, dicari nilai dari *classification loss*, *regression loss*, dan *total loss*. Kurva dari *classification loss*, *regression loss*, dan *total loss* pada saat proses pelatihan model RetinaNet dengan *backbone* ResNet-101 ditampilkan pada Gambar 12.



**Gambar 12. Kurva *Loss* Pelatihan RetinaNet (*Backbone*: ResNet-101)**

Berdasarkan kurva *loss* model RetinaNet dengan *backbone* ResNet-101 yang ditunjukkan pada Gambar 12, model yang dilatih tanpa *data augmentation* cenderung mengalami *overfitting*. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang perbedaannya semakin menjauh saat *epoch* bertambah. Sedangkan model yang memakai *data augmentation* mempunyai kecenderungan *overfitting* yang lebih rendah. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang memiliki perbedaan yang tidak terlalu jauh saat *epoch* bertambah.

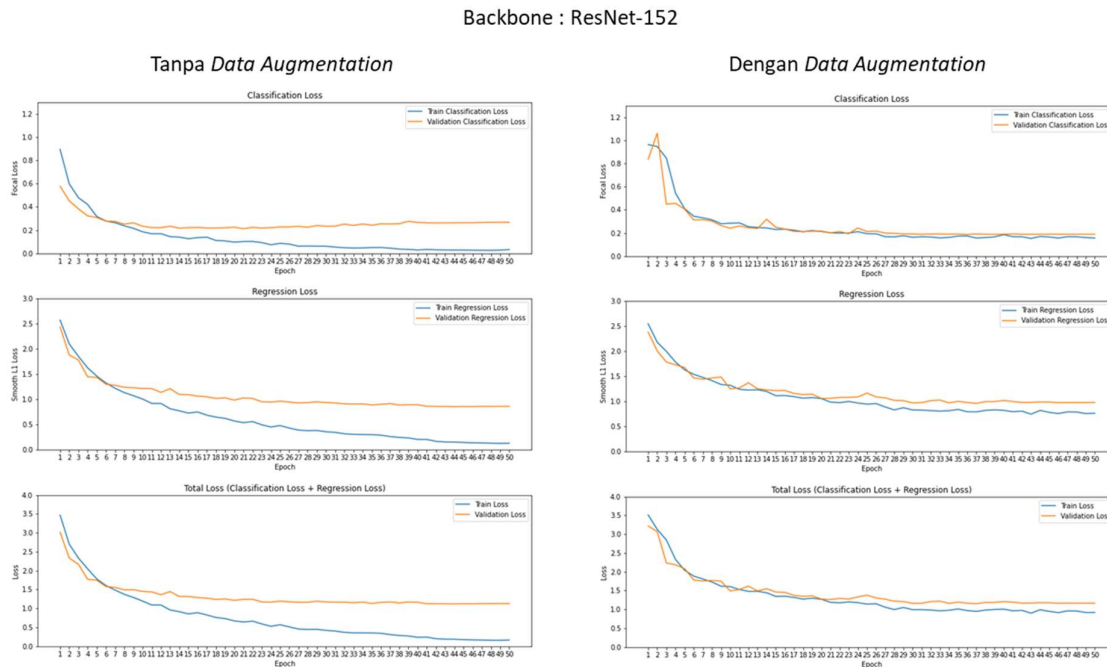
Pada pelatihan menggunakan *backbone* ResNet-101 tanpa *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-15 dengan nilai 0.8420. Sedangkan pada pelatihan menggunakan *backbone* ResNet-101 dengan *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-46 dengan nilai 0.8525. Kurva *validation AP* model RetinaNet dengan *backbone* ResNet-101 ditunjukkan pada Gambar 13.



**Gambar 13. Kurva Validasi Pelatihan RetinaNet (*Backbone*: ResNet-101)**

Perbandingan Deteksi Letak Polip pada Citra *Colonoscopy* menggunakan CNN dengan Arsitektur RetinaNet

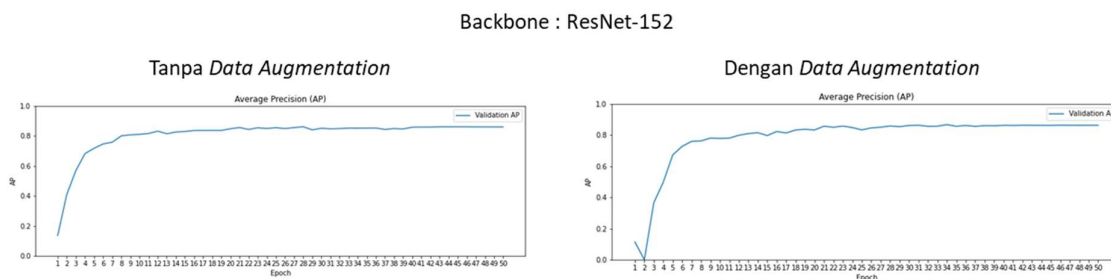
Pada proses pelatihan model RetinaNet menggunakan *backbone* ResNet-152, dicari nilai dari *classification loss*, *regression loss*, dan *total loss*. Kurva dari *classification loss*, *regression loss*, dan *total loss* pada saat proses pelatihan model RetinaNet dengan *backbone* ResNet-152 ditampilkan pada Gambar 14.



**Gambar 14. Kurva Loss Pelatihan RetinaNet (Backbone: ResNet-101)**

Berdasarkan kurva *loss* yang ditunjukkan pada Gambar 14, terlihat model yang dilatih tanpa *data augmentation* cenderung mengalami *overfitting*. Kondisi saat model mempelajari data latih terlalu detail, sehingga performa model hanya baik terhadap data yang dipelajari oleh model tersebut disebut *overfitting*. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang perbedaannya semakin menjauh saat *epoch* bertambah. Sedangkan model dengan *data augmentation* mempunyai kecenderungan *overfitting* yang lebih rendah. Hal ini ditunjukkan dari nilai *train loss* dan *validation loss* yang memiliki perbedaan yang tidak terlalu jauh saat *epoch* bertambah

Pada pelatihan menggunakan *backbone* ResNet-152 tanpa *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-28 dengan nilai 0.8624. Sedangkan pada pelatihan menggunakan *backbone* ResNet-101 dengan *data augmentation*, nilai *validation AP* yang terbaik ada pada *epoch* ke-34 dengan nilai 0.8680. Kurva *validation AP* model RetinaNet dengan *backbone* ResNet-152 ditunjukkan pada Gambar 15.



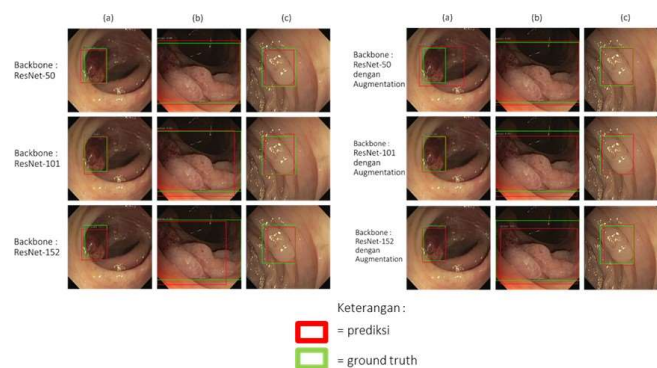
**Gambar 15. Kurva Validasi Pelatihan RetinaNet (*Backbone*: ResNet-152)****3.2 Hasil Evaluasi Model**

Setelah model RetinaNet selesai dilatih, dilakukan evaluasi model menggunakan *test data* yang berjumlah 180 buah citra. Digunakan dua ukuran metrik yang digunakan untuk mengevaluasi model yang dibuat yaitu *Intersection over Union* (IoU) dan *Average Precision* (AP). Hasil Evaluasi model CNN yang dibuat ditunjukkan pada Tabel 3.

**Tabel 3. Hasil Evaluasi Model Menggunakan *Test Data***

Arsitektur	<i>Backbone</i>	<i>Data Augmentation</i>	Semua Test Data			
			Mean IoU	AP <sub>25</sub>	AP <sub>50</sub>	AP <sub>75</sub>
RetinaNet	ResNet-50	-	<b>0.8415</b>	0.9193	0.8843	0.6531
		+	0.7678	0.8326	0.8893	0.6317
RetinaNet	ResNet-101	-	0.7761	0.9020	0.8724	0.5112
		+	0.7547	<b>0.9308</b>	<b>0.9039</b>	<b>0.6985</b>
RetinaNet	ResNet-152	-	0.8134	0.9039	0.8784	0.6167
		+	0.7966	0.9228	0.8903	0.6902

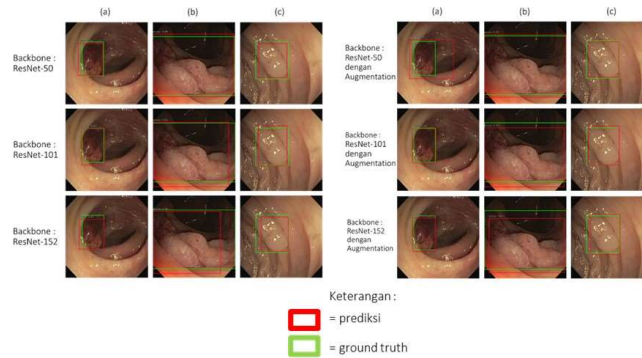
Berdasarkan hasil evaluasi dari menggunakan *test data* yang ditampilkan pada Tabel 3, model yang terbaik berdasarkan nilai metrik *Intersection over Union* (IoU) adalah RetinaNet dengan *backbone* ResNet-50 tanpa *data augmentation* dengan nilai 0.8415. Model tanpa *data augmentation* menunjukkan skor IoU yang lebih tinggi jika dibandingkan dengan model yang menggunakan *data augmentation* sehingga model tanpa *data augmentation* memiliki prediksi yang letaknya lebih mendekati dengan *ground truth* yang merupakan letak polip yang sesungguhnya pada citra. Sedangkan model yang terbaik berdasarkan nilai metrik *Average Precision* (AP) adalah RetinaNet dengan *backbone* ResNet-101 yang menggunakan *data augmentation* dengan nilai AP<sub>25</sub> = 0.9308, AP<sub>50</sub> = 0.9039, dan AP<sub>75</sub> = 0.6985. Hal ini bahwa model yang menggunakan *data augmentation* memiliki skor AP yang lebih tinggi jika dibandingkan dengan model tanpa *data augmentation* sehingga model dengan *data augmentation* lebih presisi dalam memprediksi polip

**Gambar 16. Contoh Prediksi Dataset CVC ClinicDB yang Berhasil**

Contoh sampel hasil prediksi dari model RetinaNet berhasil pada dataset CVC-ClinicDB ditunjukkan pada Gambar 16. Baik model RetinaNet dengan *backbone* ResNet-50, ResNet-101, maupun ResNet-152 berhasil memprediksi letak polip dengan benar. Hasil prediksi dari model ditunjukkan dengan kotak berwarna merah. Sedangkan *ground truth* yang menunjukkan posisi objek sebenarnya ditunjukkan dengan kotak berwarna hijau. Prediksi

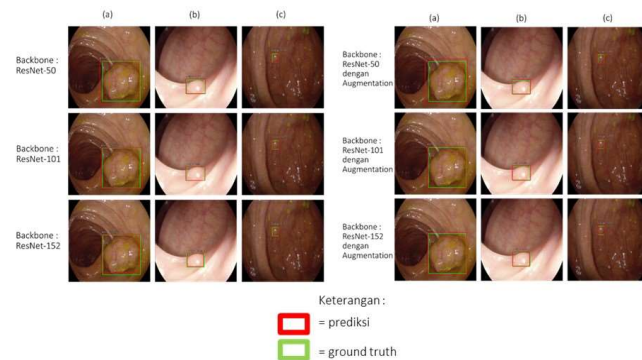
dikatakan berhasil karena hasil prediksi model (kotak hijau) sesuai dengan *ground truth* (kotak merah).

Terdapat juga prediksi yang gagal pada citra yang berasal dari dataset CVC-ClinicDB. Pada pengujian menggunakan dataset CVC-Clinic-DB terjadi kegagalan pada citra dengan polip yang sulit dibedakan oleh mata karena terdapat tonjolan lain yang menyerupai polip. Tonjolan lain pada citra yang sebenarnya bukan polip, diprediksi sebagai polip oleh model. Contoh sampel hasil prediksi model RetinaNet yang gagal pada dataset CVC-ClinicDB ditunjukkan pada Gambar 17.



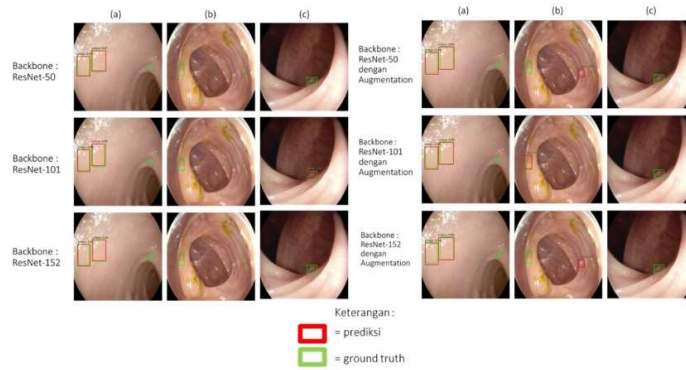
**Gambar 17. Contoh Prediksi Dataset CVC ClinicDB yang Gagal**

Contoh sampel hasil prediksi dari model RetinaNet berhasil pada dataset Etis-Larib ditunjukkan pada Gambar 18. Baik model RetinaNet dengan *backbone* ResNet-50, ResNet-101, maupun ResNet-152 berhasil memprediksi letak polip dengan benar. Hasil prediksi dari model ditunjukkan dengan kotak berwarna merah.



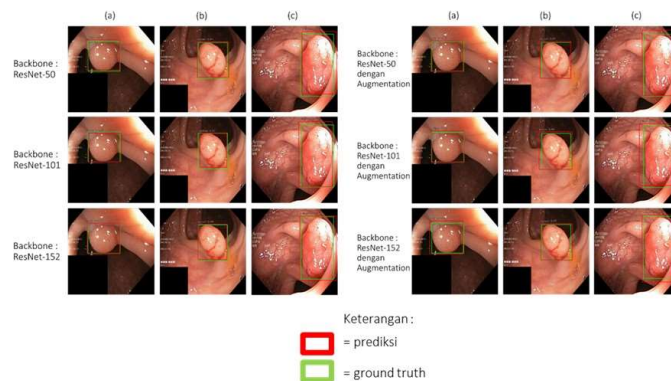
**Gambar 18. Contoh Prediksi Dataset Etis-Larib yang Berhasil**

Terdapat juga prediksi yang gagal pada citra yang berasal dari dataset Etis-Larib. Pada pengujian menggunakan dataset Etis-Larib terjadi kegagalan pada citra dengan polip yang berukuran kecil. Contoh sampel hasil prediksi model RetinaNet yang gagal pada dataset Etis-Larib ditunjukkan pada Gambar 19.



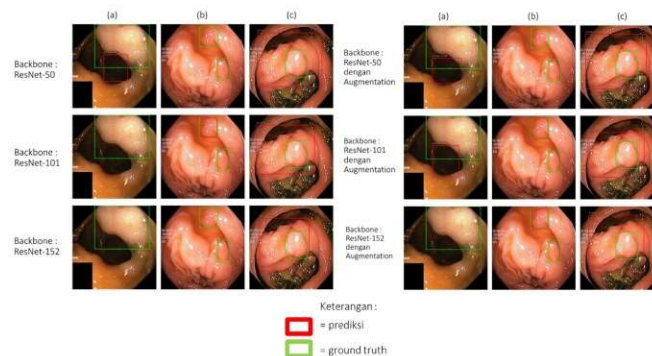
**Gambar 19. Contoh Prediksi Dataset Etis-Larib yang Gagal**

Contoh sampel hasil prediksi dari model RetinaNet berhasil pada dataset Kvasir SEG ditunjukkan pada Gambar 20. Baik model RetinaNet dengan *backbone* ResNet-50, ResNet-101, maupun ResNet-152 berhasil memprediksi letak polip dengan benar. Hasil prediksi dari model ditunjukkan dengan kotak berwarna merah.



**Gambar 20. Contoh Prediksi Dataset Kvasir SEG yang Berhasil**

Terdapat juga prediksi yang gagal pada citra yang berasal dari dataset CVC-ClinicDB. Pada pengetesan menggunakan dataset CVC-Clinic-DB terjadi kegagalan pada citra dengan polip yang sulit dibedakan oleh mata karena terdapat tonjolan lain yang menyerupai polip. Contoh sampel hasil prediksi model RetinaNet yang gagal pada dataset CVC-ClinicDB ditunjukkan pada Gambar 21.



**Gambar 21. Contoh Prediksi Dataset Kvasir SEG yang Gagal**

Maka dari hasil evaluasi model keseluruhan, model tanpa *data augmentation* cenderung mempunyai skor *Intersection over Union* (IoU) yang lebih tinggi jika dibandingkan dengan model yang menggunakan *data augmentation*. Namun model dengan *data augmentation* cenderung mempunyai skor *Average Precision* (AP) yang lebih tinggi jika dibandingkan dengan model yang tidak menggunakan *data augmentation*. Pada penelitian ini menunjukkan bahwa model RetinaNet yang dibuat masih memiliki kekurangan dalam memprediksi citra polip karena terdapat beberapa kondisi citra yang gagal di prediksi. Kemungkinan diperlukan penambahan dataset dari sumber lain, sehingga variasi data yang dipelajari model CNN lebih banyak, atau dapat dicoba menggunakan backbone lainnya sehingga performanya dapat menjadi lebih baik.

#### 4. KESIMPULAN

Deteksi polip pada citra *colonoscopy* berhasil diimplementasikan menggunakan model *Convolutional Neural Network* (CNN) dengan arsitektur RetinaNet. Dari hasil evaluasi, model yang terbaik berdasarkan metrik *Intersection over Union* (IoU) adalah model RetinaNet (*Backbone* = ResNet-50) tanpa *data augmentation* dengan nilai 0.8415. Sedangkan model yang terbaik berdasarkan metrik *Average Precision* (AP) adalah RetinaNet (*Backbone* = ResNet-101) dengan *data augmentation* dengan nilai  $AP_{25} = 0.9308$ ,  $AP_{50} = 0.9039$ ,  $AP_{75} = 0.6985$ . Model tanpa *data augmentation* cenderung mempunyai skor *Intersection over Union* (IoU) yang lebih tinggi jika dibandingkan dengan model yang menggunakan *data augmentation*, tetapi model dengan *data augmentation* cenderung mempunyai skor *Average Precision* (AP) yang lebih tinggi jika dibandingkan dengan model yang tidak menggunakan *data augmentation*. Model RetinaNet yang dibuat masih memiliki kekurangan dalam memprediksi citra polip karena terdapat beberapa kondisi citra yang gagal di prediksi.

#### DAFTAR RUJUKAN

- ACS. (2020). Colorectal Cancer Facts and Figures 2020-2022. *American cancer society*, 66(11), 1–41. <https://www.cancer.org>
- Albawi, S., Mohammed, T. A. M., & Alzawi, S. (2017). Layers of a Convolutional Neural Network. *Ieee*, 16.
- Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., & Vilariño, F. (2015). WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics: The Official Journal of the Computerized Medical Imaging Society*, 43, 99–111. <https://doi.org/10.1016/j.compmedimag.2015.02.007>
- Bernal, J., Tajkbaksh, N., Sanchez, F. J., Matuszewski, B. J., Chen, H., Yu, L., Angermann, Q., Romain, O., Rustad, B., Balasingham, I., Pogorelov, K., Choi, S., Debard, Q., Maier-Hein, L., Speidel, S., Stoyanov, D., Brandao, P., Cordova, H., Sanchez-Montes, C., ... Histace, A. (2017). Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge. *IEEE Transactions on Medical Imaging*, 36(6), 1231–1249. <https://doi.org/10.1109/TMI.2017.2664042>

- Cheng, C. L., Kuo, Y. L., Hsieh, Y. H., Tang, J. H., & Leung, F. W. (2021). Comparison of Right Colon Adenoma Miss Rates Between Water Exchange and Carbon Dioxide Insufflation: A Prospective Randomized Controlled Trial. *Journal of clinical gastroenterology*, *55*(10), 869–875. <https://doi.org/10.1097/MCG.0000000000001454>
- Choi, H. N. a., Kim, H. H. e., Oh, J. S. eo., Jang, H. S. an., Hwang, H. S. i., Kim, E. Y. oun., Kwon, J. G. o., & Jung, J. T. a. (2014). Factors influencing the miss rate of polyps in a tandem colonoscopy study. *The Korean journal of gastroenterology = Taehan Sohwagi Hakhoe chi*, *64*(1), 24–30. <https://doi.org/10.4166/kjg.2014.64.1.24>
- Dobie, K. A., Burke, C., & Editor, T. (2012). *Colonoscopy For Dummies, Special Edition*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, (pp. 770–778). <https://doi.org/10.1109/CVPR.2016.90>
- Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., de Lange, T., Johansen, D., & Johansen, H. D. (2020). Kvasir-SEG: A Segmented Polyp Dataset. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11962 LNCS*, 451–462. [https://doi.org/10.1007/978-3-030-37734-2\\_37](https://doi.org/10.1007/978-3-030-37734-2_37)
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- Patil, A., & Rane, M. (2021). Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. *Smart Innovation, Systems and Technologies*, *195*, 21–30. [https://doi.org/10.1007/978-981-15-7078-0\\_3](https://doi.org/10.1007/978-981-15-7078-0_3)
- Pidala, M. J., & Cusick, M. V. (2017). The Difficult Colorectal Polyp. *Surgical Clinics of North America*, *97*(3), 515–527. <https://doi.org/10.1016/j.suc.2017.01.003>
- Siegel, R. L., Miller, K. D., Goding Sauer, A., Fedewa, S. A., Butterly, L. F., Anderson, J. C., Cercek, A., Smith, R. A., & Jemal, A. (2020). Colorectal cancer statistics, 2020. *CA: A Cancer Journal for Clinicians*, *70*(3), 145–164. <https://doi.org/10.3322/caac.21601>
- Silva, J., Histace, A., Romain, O., Dray, X., & Granado, B. (2014). Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer. *International Journal of Computer Assisted Radiology and Surgery*, *9*(2), 283–293. <https://doi.org/10.1007/s11548-013-0926-3>