



TUGAS AKHIR - EF234801

SEGMENTASI DARAH PADA CITRA UJI SILANG SERASI DENGAN PENDEKATAN DEEP LEARNING

JANUAR EVAN ZURIEL BANJARNAHOR

NRP 5025201210

Dosen Pembimbing I

Dini Adni Navastara, S.Kom., M.Sc.

NIP 19851017 201504 2 001

Dosen Pembimbing II

Prof. Dr. Eng. Chastine Faticah, S.Kom., M.Kom.

NIP 19751220 200112 2 002

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



TUGAS AKHIR - EF234801

SEGMENTASI DARAH PADA CITRA UJI SILANG SERASI DENGAN PENDEKATAN DEEP LEARNING

JANUAR EVAN ZURIEL BANJARNAHOR

NRP 5025201210

Dosen Pembimbing I

Dini Adni Navastara, S.Kom., M.Sc.

NIP 19851017 201504 2 001

Dosen Pembimbing II

Prof. Dr. Eng. Chastine Faticahah, S.Kom., M.Kom.

NIP 19751220 200112 2 002

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - EF234801

BLOOD SEGMENTATION IN CROSMATCH TESTING IMAGE WITH DEEP LEARNING APPROACH

JANUAR EVAN ZURIEL BANJARNAHOR

NRP 5025201210

Advisor I

Dini Adni Navastara, S.Kom., M.Sc.

NIP 19851017 201504 2 001

Advisor II

Prof. Dr. Eng. Chastine Faticahah, S.Kom., M.Kom.

NIP 19751220 200112 2 002

Study Program Bachelor of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2024

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SEGMENTASI DARAH PADA CITRA UJI SILANG SERASI DENGAN PENDEKATAN DEEP LEARNING

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : **JANUAR EVAN ZURIEL BANJARNAHOR**

NRP. 5025201210

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dini Adni Navastara, S.Kom., M.Sc.  Pembimbing
2. Prof. Dr. Eng. Chistine Faticah, S.Kom., M.Kom.  Ko-pembimbing
3. Prof. Dr. Diana Purwitasari, S.Kom., M.Sc.  Penguji
4. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D  Penguji

SURABAYA
Januari, 2024

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

BLOOD SEGMENTATION IN CROSMATCH TESTING IMAGE WITH DEEP LEARNING APPROACH

FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a degree in Computer Science at
Undergraduate Study Program of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

By: JANUAR EVAN ZURIEL BANJARNAHOR

NRP. 5025201210

Approved by Final Project Examiner Team:

1. Dini Adni Navastara, S.Kom., M.Sc.



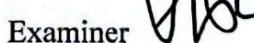
Advisor

2. Prof. Dr. Eng. Chistine Faticahah, S.Kom., M.Kom.



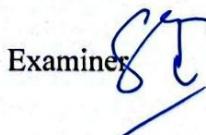
Co-Advisor

3. Prof. Dr. Diana Purwitasari, S.Kom., M.Sc.



Examiner

4. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D



Examiner

SURABAYA
January, 2024

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Januar Evan Zuriel Banjarnahor / 5025201210

Departemen : Teknik Informatika

Dosen Pembimbing I / NIP : Dini Adni Navastara, S.Kom., M.Sc. /
198510172015042001

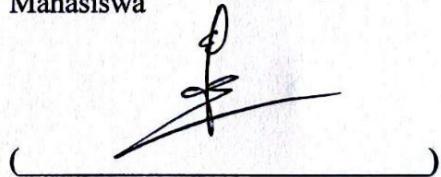
Dosen Pembimbing II / NIP : Prof. Dr. Eng. Chastine Faticah, S.Kom., M.Kom. /
197512202001122002

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Segmentasi Darah pada Citra Uji Silang Serasi dengan Pendekatan *Deep Learning*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari di temukan ketidak sesuaian dengan pernyataan ini, maka saya bersedia menerima saksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 7 Februari 2024

Mahasiswa

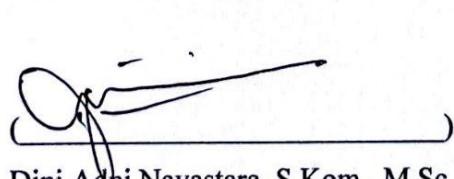


Januar Evan Zuriel Banjarnahor

NRP. 5025201210

Mengetahui

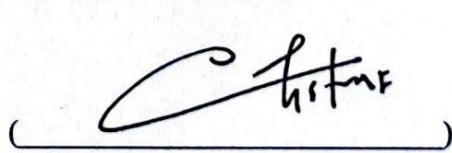
Dosen Pembimbing I



Dini Adni Navastara, S.Kom., M.Sc.

NIP. 198510172015042001

Dosen Pembimbing II



Prof. Dr. Eng. Chastine Faticah,
S.Kom., M.Kom.

NIP. 197512202001122002

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Januar Evan Zuriel Banjarnahor / 5025201210
Department : Teknik Informatika
Advisor I / NIP : Dini Adni Navastara, S.Kom., M.Sc. /
198510172015042001
Advisor II / NIP : Prof. Dr. Eng. Chastine Faticahah, S.Kom., M.Kom. /
197512202001122002

Hereby declare that Final Project with the title of “Blood Segmentation In Crossmatch Testing Image With Deep Learning Approach” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, February 7th 2024

Mahasiswa

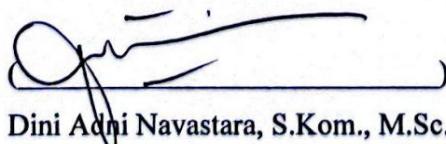


Januar Evan Zuriel Banjarnahor

NRP. 5025201210

Acknowledge

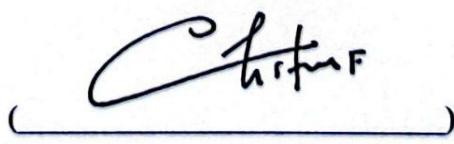
Advisor I



Dini Adni Navastara, S.Kom., M.Sc.

NIP. 198510172015042001

Advisor II



Prof. Dr. Eng. Chastine Faticahah,
S.Kom., M.Kom.

NIP. 197512202001122002

[Halaman ini sengaja dikosongkan]

SEGMENTASI DARAH PADA CITRA UJI SILANG SERASI DENGAN PENDEKATAN DEEP LEARNING

Nama Mahasiswa / NRP : Januar Evan Zuriel Banjarnahor / 5025201210

Departemen : Teknik Informatika FTEIC - ITS

Dosen Pembimbing : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Kecepatan dan ketepatan dalam melayani pasien adalah faktor penting dalam pelayanan kepada masyarakat. Pelayanan kesehatan, termasuk pelayanan transfusi darah, juga memiliki masalah unik yang memerlukan penyelesaian yang efektif. Evaluasi kepuasan pasien merupakan indikator utama untuk menilai kualitas pelayanan kesehatan. Dalam pelayanan transfusi darah, pemeriksaan tes silang serasi (pratransfusi) sangat penting untuk mencegah komplikasi pada pasien. Namun, masalah utama yang dihadapi adalah pembacaan dan dokumentasi hasil uji silang serasi yang masih dilakukan secara manual. Hal ini dapat menghambat pelayanan rumah sakit jika tidak didukung oleh jumlah tenaga kesehatan yang memadai. Oleh karena itu, penelitian ini bertujuan untuk membantu pelayanan rumah sakit sehingga pekerjaan pencatatan rekam medis sudah tidak manual sehingga dapat mengalokasikan tenaga kesehatan dengan lebih efisien melalui otomatisasi pembacaan dan penyimpanan hasil uji silang serasi dengan model *deep learning* segmentasi citra yang sudah dibuat. Metode yang dilakukan pada penelitian yang telah dilakukan mencakup pengumpulan data, *preprocessing* data, pemisahan data, augmentasi data, pembuatan model, pelatihan model, dan evaluasi model. Pengumpulan data dilakukan dengan *smartphone*. Pemisahan data dibagi menjadi pelatihan, validasi, dan pengujian. Jenis augmentasi yang digunakan adalah kontras, *brightness*, dan *noise*. Evaluasi model yang digunakan adalah skor IoU, skor F1, presisi, *recall*, dan akurasi. Model terbaik berdasarkan uji coba skenario adalah *encoder* ResNeXt50, *decoder* Pyramid Attention Network (PAN) dengan *hyperparameter batch size* 16 dan *optimizer* Adam dengan *learning rate* 0.001. Performa model terbaik yang didapatkan adalah skor IoU sebesar 0.8636, skor F1 sebesar 0.9238, presisi sebesar 0.9371, *recall* sebesar 0.9152, dan akurasi sebesar 0.9811. Hasil pada model mampu merepresentasikan pola reaksi darah pada citra orisinal meskipun terdapat beberapa hasil prediksi model kurang detil sehingga representasi pola reaksi darah pada citra tersebut berubah.

Kata kunci: *Pelayanan transfusi darah, Uji Silang Serasi, Segmentasi Citra, Pyramid Attention Network (PAN), ResNeXt50.*

[Halaman ini sengaja dikosongkan]

BLOOD SEGMENTATION IN CROSMATCH TESTING IMAGE WITH DEEP LEARNING APPROACH

Student Name / NRP: Januar Evan Zuriel Banjarnahor / 5025201210

Department : Informatics Engineering FTEIC - ITS

Advisor : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

Speed and accuracy in serving patients are important factors in service to the community. Health services, including blood transfusion services, also have unique problems that require effective solutions. Evaluation of patient satisfaction is a key indicator to assess the quality of health services. In blood transfusion services, cross-match testing (pretransfusion) is essential to prevent complications in patients. However, the main problem faced is the reading and documentation of the cross-matched test results which are still done manually. This can hamper hospital services if not supported by an adequate number of health workers. Therefore, this research aims to help hospital services so that the work of recording medical records is no longer manual so that it can allocate health workers more efficiently through the automation of reading and storing the results of the cross-match test with the deep learning model of image segmentation that has been made. The methods used in the research include data collection, data preprocessing, data separation, data augmentation, model building, model training, and model evaluation. Data collection is done with a smartphone. Data separation is divided into training, validation, and testing. The types of augmentation used are contrast, brightness, and noise. The model evaluation used is IoU score, F1 score, precision, recall, and accuracy. The best model based on the test scenario is ResNeXt50 encoder, Pyramid Attention Network (PAN) decoder with hyperparameter batch size 16 and Adam optimizer with learning rate 0.001. The best model performance obtained is the IoU score of 0.8636, F1 score of 0.9238, precision of 0.9371, recall of 0.9152, and accuracy of 0.9811. The results in the model are able to represent the blood reaction pattern in the original image even though there are some prediction results that are less detailed and representation of the blood reaction pattern in the image changes.

Keyword: *Blood Transfusion Service, Crossmatch Testing, Image Segmentation, Pyramid Attention Network (PAN), ResNeXt50.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Esa atas segala berkat, karunia, dan Rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul : **“SEGMENTASI DARAH PADA UJI SILANG SERASI DENGAN PENDEKATAN DEEP LEARNING”**

Tugas Akhir ini dibuat sebagai salah satu syarat kelulusan untuk meraih gelar Sarjana Komputer program Strata Satu (S-1) di Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember. Pembuatan Tugas Akhir ini tidak luput dari kesalahan dan masalah-masalah. Penulis berharap bahwa penulisan Tugas Akhir ini dapat berkontribusi pada masyarakat secara nyata dan pengembangan ilmu pengetahuan untuk saat ini dan kedepannya. Pembuatan Tugas Akhir ini tidak akan mungkin tanpa dukungan dan bantuan dari berbagai pihak yang terlibat. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, doa, dan bantuan selama proses penyelesaian Tugas Akhir ini, antara lain :

1. Tuhan Yesus yang selalu memberikan berkat dan perlindungan selama proses penyelesaian Tugas Akhir ini berjalan.
2. Keluarga penulis yang selalu berdoa dan memberikan bantuan secara moral dan material selama proses penyelesaian Tugas Akhir ini berjalan.
3. Ibu Dini Adni Navastara, S.Kom, M.Sc. dan Ibu Prof. Dr. Eng. Chistine Faticahah, S.Kom., M.Kom. selaku dosen pembimbing satu dan dosen pembimbing dua yang telah membimbing penulis, memberikan motivasi, dan nasihat selama proses penyelesaian Tugas Akhir ini berjalan
4. Ibu Dr.Eng.Chistine Faticahah, S.Kom., M.Kom. selaku kepala Departemen Teknik Informatika ITS.
5. Semua Bapak dan Ibu Dosen yang sudah memberikan ilmu dan nasihat selama perkuliahan penulis di Teknik Informatika ITS.
6. Teman-teman Angkatan 2020 yang sudah membantu selama penulis berkuliah di Teknik Informatika ITS.
7. Teman-teman PMK ITS 2023, terutama *Media Production* yang selalu membantu dalam doa selama penulisan Tugas Akhir berjalan.
8. Sahabat penulis dalam belajar bersama selama berkuliah di Teknik Informatika ITS yaitu, Estha, Ravindra, Bagas, Naufal, Hilmi, Nadya, Vikko, Haniif, Kurnia, Tavi, Amanda, dan Jessica.
9. Semua Staff dan Karyawan pada Unit Transfusi Darah RSUD Dr. Soetomo membantu dalam penelitian, terutama Bu Dewi pada pengajuan laik etik dan Bu Indarwati sebagai koordinator penelitian.
10. Prof. Dr. Jusak Nugraha, dr, MS, SpPK(K). sebagai pembimbing klinis pada penelitian yang dilakukan pada RSUD Dr. Soetomo.
11. Semua Staff Komite Etik RSUD Dr. Soetomo yang sudah membantu menjelaskan persyaratan pengajuan laik etik dengan baik.
12. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Surabaya, 2 Februari 2024

Januar Evan Zuriel Banjarnahor

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
PERNYATAAN ORISINALITAS.....	ix
ABSTRAK.....	xiii
KATA PENGANTAR.....	xvii
DAFTAR ISI.....	xix
DAFTAR GAMBAR/GRAFIK/DIAGRAM.....	xxi
DAFTAR TABEL.....	xxiii
DAFTAR KODE SEMU.....	xxv
DAFTAR PERSAMAAN.....	xxvii
BAB I PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Penelitian Terkait.....	3
2.2 Dasar Teori.....	5
2.2.1 Golongan Darah.....	5
2.2.2 Uji Silang Serasi.....	5
2.2.3 Augmentasi Citra.....	7
2.2.4 Segmentasi Citra.....	7
2.2.5 CNN (Convolutional Neural Network).....	8
2.2.6 Arsitektur Klasifikasi CNN.....	8
2.2.7 Arsitektur Segmentasi CNN.....	12
2.2.8 Fungsi Kerugian.....	15
2.2.9 Evaluasi Performa.....	16
BAB III METODOLOGI.....	19
3.1 Deskripsi Umum Sistem.....	19
3.2 Dataset yang Digunakan.....	19
3.3 Peralatan pendukung.....	20

3.4	Perancangan Sistem.....	21
3.4.1	Koleksi data.....	21
3.4.2	Data Preprocessing.....	22
3.4.3	Pemisahan Dataset dengan Normalisasi.....	22
3.4.4	Augmentasi Citra.....	23
3.4.5	Pembuatan Model.....	23
3.4.6	Model Fitting.....	24
3.4.7	Evaluasi Performa.....	26
3.5	Implementasi.....	26
3.5.1	Implementasi Data Preprocessing.....	26
3.5.2	Implementasi Pemisahan Dataset dengan Normalisasi.....	28
3.5.3	Implementasi Augmentasi Citra.....	29
3.5.4	Implementasi Pembuatan Model.....	30
3.5.5	Implementasi Model Fitting.....	31
3.5.6	Implementasi Evaluasi Performa Model.....	31
	BAB IV HASIL DAN PEMBAHASAN.....	33
4.1	Skenario Uji Coba dan Analisis.....	33
4.1.1	Skenario Perbandingan Encoder.....	33
4.1.2	Skenario Perbandingan Hyperparameter.....	37
4.1.3	Skenario Perbandingan Metode Decoder.....	44
4.1.4	Skenario Perbandingan Tanpa Augmentasi dan Augmentasi pada Latihan	53
4.1.5	Hasil Performa Model Terbaik pada Tiap Kelas.....	56
4.2	Pembahasan/Diskusi.....	56
4.2.1	Pembahasan Skenario Perbandingan Metode Encoder.....	56
4.2.2	Pembahasan Skenario Perbandingan Metode Hyperparameter.....	57
4.2.3	Pembahasan Skenario Perbandingan Metode Decoder.....	58
4.2.4	Pembahasan Skenario Perbandingan Augmentasi dan Tanpa Augmentasi	59
4.2.5	Pembahasan Performa Model Terbaik pada Tiap Kelas.....	59
	BAB V KESIMPULAN DAN SARAN.....	61
5.1	Kesimpulan.....	61
5.2	Saran.....	61
	DAFTAR PUSTAKA.....	63
	LAMPIRAN.....	67

BIODATA PENULIS.....115

DAFTAR GAMBAR/GRAFIK/DIAGRAM

Gambar 2.1 Petunjuk Kategori Penggumpalan Darah	
Gambar 2.2 Contoh Augmentasi Citra Warna Klasifikasi Melanoma	
Gambar 2.3 Hasil Segmentasi Sel Leukimia Sel Orisinal, Segmentasi Sitoplasma, Segmentasi Area dalam	
Gambar 2.4 Ilustrasi Komponen pada CNN	
Gambar 2.5 Contoh Visualisasi dari Arsitektur FCN	
Gambar 2.6 Arsitektur dari R-CNN	
Gambar 2.7 Arsitektur dari Fast R-CNN	
Gambar 2.8 Arsitektur dari Faster R-CNN	
Gambar 2.9 Ilustrasi dari RPN	
Gambar 2.10 Arsitektur dari Mask R-CNN	
Gambar 2.11 Ilustrasi Arsitektur dari U-Net	
Gambar 2.12 Ilustrasi Arsitektur dari FPN	
Gambar 2.13 Ilustrasi dari <i>Feature Pyramid Attention</i>	
Gambar 2.14 Ilustrasi GAU pada Arsitektur PAN	
Gambar 2.15 Ilustrasi dari DeepLabV3+14	
Gambar 3.1 Kartu Gel untuk Uji Silang Serasi	
Gambar 3.2 Contoh Pemotongan Gambar untuk Tabung dengan Kelas Negatif (a), Positif Satu (b), Positif dua (c), Positif Tiga (d), dan Positif Empat (d).	
Gambar 3.3 Diagram Alur Proses Penelitian	
Gambar 3.4 Contoh Gambar Tabung dengan <i>Ground Truth</i> Segmentasi Darah	
Gambar 3.5 <i>User Interface</i> dari Label Studio	
Gambar 3.6 Hasil Augmentasi dari Gambar Normal (a) dengan Kontras (b), Kecerahan (c), Noise (d), dan Kombinasi Ketiga Jenis Augmentasi (e).....	30
Gambar 4.1 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr0001-Batch8.....	34
Gambar 4.2 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr0001-Batch8.....	35
Gambar 4.3 Grafik Pelatihan dan Validasi Model FPN-EfficientNet-B5-Adam-lr0001-Batch8	35
Gambar 4.4 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr0001-Batch32.....	38
Gambar 4.5 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr001-Batch32.....	38
Gambar 4.6 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr0001-Batch16.....	39
Gambar 4.7 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr001-Batch16.....	40
Gambar 4.8 Grafik Pelatihan dan Validasi Model FPN-EfficientNet-B5-Adam-lr001-Batch8	42
Gambar 4.9 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-ResNet34-Adam-lr001-Batch32.....	45

Gambar 4.10 Grafik Pelatihan dan Validasi Model PAN-ResNet34-Adam-lr001-Batch32.....	45
Gambar 4.11 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-ResNeXt50-Adam-lr001-Batch16.....	48
Gambar 4.12 Grafik Pelatihan dan Validasi Model PAN-ResNeXt50-Adam-lr001-Batch16.....	48
Gambar 4.13 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-EfficientNet-B5-Adam-lr001-Batch8.....	51
Gambar 4.14 Grafik Pelatihan dan Validasi Model PAN-EfficientNet-B5-Adam-lr001-Batch8.....	51
Gambar 4.15 Grafik Pelatihan dan Validasi Model Augmentasi.....	54

DAFTAR TABEL

Tabel 2.1 Karakteristik Penggolongan Pola Reaksi Darah pada Uji Silang Serasi.....	6
Tabel 3.1 Spesifikasi Peralatan Pendukung.....	20
Tabel 3.2 Deskripsi <i>Layer</i> dalam Arsitektur CNN Segmentasi.....	24
Tabel 3.3 Deskripsi <i>Hyperparameter</i> yang Digunakan.....	25
Tabel 4.1 Rangkuman Encoder, Decoder, dan Hyperparameter yang Digunakan.....	33
Tabel 4.2 Hasil Uji Coba Model pada Skenario Perbandingan Encoder.....	34
Tabel 4.3 Hasil Model Terbaik pada Skenario Perbandingan Encoder.....	36
Tabel 4.4 Metrik dari Citra dengan Skor IoU Terendah pada Skenario Perbandingan Encoder	37
Tabel 4.5 Hasil Uji Coba Model ResNet34 pada Skenario Perbandingan Hyperparameter....	37
Tabel 4.6 Hasil Uji Coba Model ResNeXt50 pada Skenario Perbandingan Hyperparameter	39
Tabel 4.7 Metrik dari Citra dengan Skor IoU Terendah Model ResNeXt50 Skenario Perbandingan Hyperparameter.....	40
Tabel 4.8 Hasil Model ResNeXt50 Terbaik pada Skenario Perbandingan Hyperparameter....	41
Tabel 4.9 Hasil Uji Coba Model EfficientNet-B5 pada Skenario Perbandingan Hyperparameter.....	42
Tabel 4.10 Hasil Model EfficientNet-B5 Terbaik pada Skenario Perbandingan Hyperparameter.....	43
Tabel 4.11 Metrik dari Citra dengan Skor IoU Terendah Model EfficientNet-B5 Skenario Perbandingan Hyperparameter.....	44
Tabel 4.12 Hasil Uji Coba Model ResNet34 pada Skenario Perbandingan Decoder.....	44
Tabel 4.13 Hasil Model ResNet34 Terbaik pada Skenario Perbandingan Decoder.....	46
Tabel 4.14 Metrik dari Citra dengan Skor IoU Terendah Model ResNet34 Skenario Perbandingan Decoder.....	47
Tabel 4.15 Hasil Uji Coba Model ResNeXt50 pada Skenario Perbandingan Decoder.....	47
Tabel 4.16 Hasil Model ResNeXt50 Terbaik pada Skenario Perbandingan Decoder.....	49
Tabel 4.17 Metrik dari Citra dengan Skor IoU Terendah Model ResNeXt50 Skenario Perbandingan Decoder.....	50
Tabel 4.18 Hasil Uji Coba Model EfficientNet-B5 pada Skenario Perbandingan Decoder....	50
Tabel 4.19 Hasil Model EfficientNet-B5 Terbaik pada Skenario Perbandingan Decoder.....	52
Tabel 4.20 Metrik dari Citra dengan Skor IoU Terendah Model EfficientNet-B5 Skenario Perbandingan Decoder.....	53
Tabel 4.21 Hasil Uji Coba Model pada Skenario Perbandingan Augmentasi dan Tanpa Augmentasi.....	53
Tabel 4.22 Metrik dari Citra dengan Skor IoU Terendah Model Augmentasi.....	54
Tabel 4.23 Hasil Model Augmentasi.....	55
Tabel 4.24 Metrik dari Citra dengan Skor IoU Terendah Model Augmentasi.....	56
Tabel 4.25 Hasil Model Terbaik dari Skenario Satu hingga Tiga.....	59

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SEMU

Kode Semu 3.1 Pemindahan Citra sesuai JSON dari Label Studio.....	27
Kode Semu 3.2 Pembuatan JSON Citra dengan Kelas Pola Reaksi.....	28
Kode Semu 3.3 Pembagian Latihan, Validasi, dan Tes sesuai Rasio.....	28
Kode Semu 3.4 Loading Citra sesuai Pembagian.....	29
Kode Semu 3.5 Implementasi Augmentasi Gambar.....	29
Kode Semu 3.6 Pendefinisian Model.....	30
Kode Semu 3.7 Iterasi pada Pelatihan Model.....	31
Kode Semu 3.8 Kalkulasi Tiap Metrik.....	32
Kode Semu 3.9 Pengujian Model dengan Dataset Tes.....	32

[Halaman ini sengaja dikosongkan]

DAFTAR PERSAMAAN

Persamaan 2.1 Jaccard Index.....	15
Persamaan 2.2 <i>Jaccard Loss</i>	15
Persamaan 2.3 Koefisien Dice.....	16
Persamaan 2.4 <i>Dice Loss</i>	16
Persamaan 2.5 <i>F1-Score</i>	16
Persamaan 2.6 <i>IoU Score</i>	17
Persamaan 2.7 <i>Sensitivity</i>	17
Persamaan 2.8 <i>Spesificity</i>	17
Persamaan 2.9 <i>Accuracy</i>	17

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

1.1 Latar belakang

Dalam pelayanan kepada masyarakat, kecepatan dan ketepatan dalam melayani pasien adalah beberapa kunci dari kesuksesan sebuah layanan. Setiap pelayanan yang ada mempunyai masalah – masalah yang unik dan dapat diselesaikan dengan cara – cara yang berbeda. Salah satu bidang pelayanan kepada masyarakat adalah pelayanan kesehatan. Menurut Gülfem Tuzkaya, menilai kualitas dari pelayanan kesehatan berdasarkan kepuasan dari pihak pasien sangat berguna untuk mengindikasi faktor utama yang mempengaruhi operasional pelayanan (Tuzkaya et al., 2019).

Salah satu pelayanan kesehatan dalam rumah sakit adalah pelayanan transfusi darah. Setiap permintaan kantong darah memerlukan uji silang serasi antara darah resipien dengan darah pendonor. Tujuan dilakukannya pemeriksaan pratransfusi darah adalah mencegah terjadinya aloantibodi pada darah resipien terhadap darah pendonor maupun sebaliknya. Pengujian yang dilakukan adalah uji silang serasi atau *crossmatch test*. Pemeriksaan uji silang serasi dilakukan dengan awal pembuatan suspensi sel darah 1% pasien dan donor. Setelah itu, suspensi sel darah pasien atau donor akan dicampurkan pada serum yang berlawanan tergantung pada jenis tes yang dicoba ke dalam kartu gel. Kartu gel tersebut diberikan identitas penggerjaan. Kartu gel akan dimasukkan ke dalam inkubasi untuk didiamkan pada suhu tertentu. Tahap selanjutnya adalah memasukkan kartu gel ke dalam mesin *centrifuge* untuk mendapatkan hasil dari uji silang serasi. Hasil dari uji silang serasi ini akan dicatat sebagai rekam medis dari pasien tersebut.

Salah satu permasalahan terbesar yang dihadapi BDRS (bank darah rumah sakit), terutama untuk rumah sakit dengan pasien dalam jumlah banyak, adalah membaca hasil dan dokumentasi hasil pengujian uji silang serasi masih dilakukan secara manual. Hal ini dapat menjadi penghambat dalam pelayanan rumah sakit apabila permintaan kantong darah tidak diimbangi dengan ketersediaan tenaga kesehatan. Meskipun pengkategorian kelas pada uji silang serasi dapat dilakukan pada manusia, tentunya ada saatnya model dapat membantu para tenaga kesehatan ketika beban pekerjaan dengan prioritas tinggi sedang tinggi. Selain itu, digitalisasi rekam medis dapat membantu pelayanan unit transfusi darah kedepannya dan terintegrasi dengan rekam medis digital lainnya. Oleh karena itu, penelitian ini dilakukan untuk membantu mempercepat pelayanan rumah sakit dan mengalokasikan tenaga kesehatan ke tugas dengan prioritas lebih tinggi. Solusi yang kami ajukan adalah otomatisasi pembacaan dan penyimpanan hasil uji silang serasi menggunakan *machine learning* dengan teknik *deep learning*. Model yang akan dibuat akan melakukan segmentasi darah pada tabung kartu gel uji silang serasi. Teknik ini sudah banyak digunakan pada citra medis, terutama pada sel darah merah. Namun, penelitian sebelumnya yang sudah dilakukan sebelumnya, seperti penelitian yang dilakukan oleh Delgado-Ortet et al. dan Zhang et al, melakukan segmentasi citra medis sel darah merah pada mikroskop, bukan citra medis yang diambil tanpa mikroskop. Tentunya hal ini memiliki tantangan tersendiri untuk mendapatkan hasil model yang baik dalam melakukan tugas segmentasi darah (Delgado-Ortet et al., 2020) (Zhang et al., 2020).

Dalam pembuatan aplikasi tersebut, terdapat beberapa tahap yang harus dilakukan untuk dapat memasukkan hasil uji silang serasi. Salah satu tahap yang dilalui adalah tahap klasifikasi hasil uji silang serasi menggunakan model *machine learning* untuk otomatisasi. Tahapan yang dilalui pada pembuatan model *machine learning* adalah *preprocessing* citra, *processing* citra, pelatihan model *machine learning*, dan evaluasi model. Penelitian ini fokus pada eksplorasi segmentasi darah pada model yang sesuai dengan representasi kelas pola

reaksi yang sudah ditentukan. Hasil yang terbaik pada evaluasi model yang akan digunakan pada kasus nyata.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir adalah sebagai berikut :

1. Bagaimana proses pengumpulan dan persiapan data uji silang serasi yang akan digunakan?
2. Bagaimana proses pembuatan model dalam segmentasi darah pada citra hasil uji silang serasi?
3. Bagaimana mengevaluasi kinerja model segmentasi terhadap citra hasil uji silang serasi?
4. Bagaimana hasil segmentasi dari model deep learning terhadap citra *ground truth*?

1.3 Batasan Masalah

Batasan masalah yang dibahas dalam Tugas Akhir ini adalah sebagai berikut :

1. Hasil klasifikasi hanya mencakup - (negatif), +1 (positif satu), +2 (positif dua), +3 (positif tiga), dan +4 (positif empat).
2. Dataset citra yang dipakai diambil hanya menggunakan hasil kartu gel uji silang serasi.
3. Permasalahan hanya sebatas segmentasi darah dari citra tabung kartu gel uji silang serasi.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah :

1. Membuat model *deep learning* yang mampu melakukan tugas dalam segmentasi pada citra uji silang serasi dari masukan citra sistem.
2. Melakukan evaluasi berbagai model *deep learning* untuk mendapatkan hasil model terbaik berdasarkan metrik evaluasi yang ditentukan untuk digunakan pada sistem.

1.5 Manfaat

Penelitian ini diharapkan menghasilkan suatu model *deep learning* yang bermanfaat untuk pembacaan uji silang serasi secara otomatis dan akurat. Selain itu, penelitian ini dapat digunakan dengan integrasi rekam medis lainnya.

BAB II TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang digunakan sebagai landasan dalam pembuatan model segmentasi dengan teknik *deep learning* yang dibuat.

2.1 Penelitian Terkait

Penyusunan Tugas Akhir mengenai segmentasi darah ini melibatkan penelitian-penelitian terdahulu yang berkaitan dengan segmentasi citra. Penelitian dengan judul *Segmentation of the Blood Vessels and Optic Disk in Retinal Images* yang dilakukan oleh Ana Salazar-Gonzalez, Djibril Kaba, Yongmin Li, dan Xiaohui Liu pada tahun 2014 dengan menggunakan 3 buah dataset, yaitu DIARETDB1, DRIVE, dan STARE. Proses segmentasi pembuluh darah (*blood vessel*) dilakukan dengan metode *Graph Cut Segmentation* yang menggunakan prinsip “energi” untuk menentukan hubungan antar *pixel* pada citra pembuluh darah. Proses segmentasi pusat syaraf mata dilakukan berdasarkan hasil segementasi pembuluh darah dengan gabungan metode *Markov Random Field* (MRF) dan *Compensation Factor*. Hasil dari penelitian ini mampu mengungguli 3 metode lain pada data STARE dan 6 metode lain pada data DRIVE berdasarkan rata-rata akurasi (Salazar-Gonzalez, 2014).

Penelitian dengan judul *Deep Learning-Based Detection and Correction of Cardiac MR Motion Artefacts During Reconstruction for High-Quality Segmentation* yang dilakukan oleh Ilkay Oksuz, James R. Clough, Bram Ruijsink, Esther Puyol Anton, Aurelien Bustin, Gastao Cruz, Claudia Prieto, Andrew P.King, dan Julia A.Schnabel pada tahun 2020, membahas tentang segmentasi rongga jantung (*cardiac cavity*) dengan menggunakan dataset hasil CMR (*Cardiovascular Magnetic Resonance*) dari UK Biobank. Proses segmentasi ini dengan model U-net segmentation dan diawali dengan proses rekonstruksi gambar menggunakan arsitektur *Convolutional Recurrence Neural Network* (CRNN). Proses rekonstruksi dan segmentasi dilakukan pada satu *network* yang sama. Hasil dari penelitian ini menunjukkan bahwa metode yang diusulkan meningkatkan performa segmentasi. Kekurangan dari penelitian ini adalah data untuk gambar rusak merupakan hasil dari kecerdasan buatan (Ilkay Oksuz, 2020).

Penelitian dengan judul *Image Segmentation of Blood Cells in Leukemia Patients* yang dilakukan oleh Adnan Khashman dan Esam Al-zgoul pada tahun 2010 dengan dataset sel leukemia dari Health Center University of Virginia. Penelitian ini membahas proses identifikasi tipe leukemia dengan menggunakan *image segmentation*. Proses segmentasi dilakukan dengan metode *Bi-Modal Thresholding* yang memisahkan antara daerah sitoplasma sel dan daerah nucleus sel. Hasil dari penelitian ini menunjukkan bahwa metode *Bi-Modal Thresholding* mampu dengan efisien mensegmentasi sel darah yang terkena leukemia. Kekurangan dari penelitian ini ialah data set yang digunakan masih relatif kecil, yaitu 120 gambar, serta uji coba hasil segmentasi masih dilakukan pada dataset *train* (Khashman & Al-Zgoul, 2010).

Penelitian dengan judul *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* yang dilakukan oleh Vijay Badrinarayanan, Alex Kendall, dan Roberto Cipolla pada tahun 2017 dengan dataset *road scene* dan *indoor scene*. Proses segmentasi pada penelitian ini melibatkan tiga proses utama, yaitu *encoding*, *decoding*, dan segmentasi *pixel-wise*. Pada jaringan *encoder*, model melakukan proses *convolution* untuk menghasilkan fitur *maps*. Fitur *maps* yang memiliki resolusi rendah akan di-*map* ulang menjadi resolusi penuh *maps* yang nantinya akan digunakan untuk proses *pixel-wise* segmentasi. Pada kasus *road scene*, hasil dari penelitian ini menunjukkan bahwa model SegNet mampu bersaing dengan model-model lainnya, seperti DeepLab-LargeFOV, FCN, dan lain sebagainya. Pada kasus *indoor scene*, SegNet mampu mensegmentasi objek-objek yang berukuran besar dengan baik,

tetapi kurang baik untuk objek-objek yang berukuran lebih kecil (Badrinarayanan, 2017).

Penelitian dengan judul *Neural Network Segmentation Methods for Fatigue Crack Images Obtained with X-ray Tomography* yang dilakukan oleh Ce Xiao dan Jean-Yves Buffiere pada tahun 2021 dengan dataset retakan pada material dari mesin CT (*Computerized Tomography*). Penelitian ini bertujuan untuk mensegmentasi bentuk retakan (*crack*) tiga dimensi. Proses segmentasi ini dilakukan dengan metode Convolutional Neural Network (CNN) dan dikombinasikan dengan Hessian Matrix filter. Hasil dari penelitian ini menunjukkan bahwa retakan pada material teridentifikasi lebih baik dengan menggunakan Hessian Matrix dibandingkan dengan metode segmentasi klasik (Xiao & Buffiere, 2021).

Penelitian dengan judul *A Deep Learning Approach for Segmentation of Red Blood Cell Images and Malaria Detection* yang dilakukan oleh M. Delgado-Ortet, A. Molina, S. Alférez, J. Rodellar, dan A. Merino pada tahun 2020 dengan dataset citra mikroskop hapsus darah merah dengan pewarna MGG (May-Grünwald-Giemsa). Penelitian ini bertujuan untuk mendeteksi kehadiran parasit segera pada gambar mikroskop sel darah merah untuk mempercepat pengobatan dan menurunkan angka kematian dari penyakit malaria. Proses penelitian ini terbagi menjadi tiga proses, yaitu segmentasi sel darah merah, memotong dan menutupi bagian bukan sel darah merah, dan mengklasifikasi sel darah merah dalam kategori sehat atau terinfeksi malaria. Hasil dari penelitian ini adalah mendapatkan model dengan akurasi global sebesar 93.72% dalam segmentasi sel darah merah dan spesifitas sebesar 87.04% (Delgado-Ortet et al., 2020).

Penelitian dengan judul *Automated recognition of white blood cells using deep learning* yang dilakukan oleh Amin Khouani, Mostafa El Habib Daho, Sidi Ahmed Mahmoudi, Mohammed Amine Chikh, dan Brahim Benzineb pada tahun 2020 dengan dataset sebanyak 145 sel yang terlabel dari 87 gambar yang terdiri dari lima tipe leukosit, 49 sel normal, 24 sel distropik, dan 72 sel lain. Penelitian ini bertujuan untuk melakukan klasifikasi, deteksi, dan segmentasi antara sitoplasma dan nukleus pada sel darah putih. Metode yang digunakan adalah melakukan persiapan data, *pre-processing* dengan normalisasi warna dan augmentasi data, melatih model, dan *post-processing* dengan melakukan *bounding-box* dan *masking* pada label. Beberapa eksperimen dilakukan, yaitu melakukan percobaan dengan arsitektur yang berbeda (Resnet50, Resnet101, VGG16, VGG19, dan InceptionV3). Hasil yang didapatkan adalah pendekatan yang diajukan oleh tim peneliti berhasil melakukan klasifikasi, deteksi, dan segmentasi dari gambar yang sudah diberikan, terutama pada bagian segmentasi. Hasil yang didapatkan adalah peningkatan F1-Score dari 79.66% menjadi 95.73% (Khouani et al., 2020).

Penelitian dengan judul *Automated Semantic Segmentation of Red Blood Cells for Sickle Cell Disease* yang dilakukan oleh Mo Zhang, Xiang Li, Mengjia Xu, dan Quanzheng Li pada tahun 2020 dengan dataset gambar sel darah merah dari pasien *Sickle Cell Disease* (SCD) yang tersedia secara publik. Gambar yang dipakai adalah 314 gambar mikroskop dari lima pasien SCD yang berbeda dan sekitar 3000 sel terlibat dalam penelitian. Penelitian ini bertujuan untuk melakukan klasifikasi dan segmentasi pada bagian sel darah merah untuk diagnosis SCD. Metode yang diajukan adalah memperbaiki arsitektur U-Net dengan menambahkan *deformable convolution* (dU-Net) untuk mengatasi permasalahan pada antar sel. Model dengan arsitektur tersebut akan dilatih dengan dataset yang sudah dijelaskan dan diuji performa model dengan berbagai metrik dan perbandingan pada model *unsupervised* (region growing, Ilastik) dan model *supervised* (U-Net, PSPNet, DeeplabV3+). Metrik yang digunakan untuk tingkatan pixel adalah Dice Coefficient, Jaccard Index, dan Hausdorff Distance. Metrik yang digunakan untuk tingkatan sel adalah Error I (keberhasilan model pada memisahkan sel yang bersentuhan), Error II (artifak yang terekognisi sebagai sel), dan Error III (jumlah hasil segmentasi yang tidak mencakup seluruh sel). Hasil yang didapat dari penelitian ini adalah dU-Net berhasil meningkatkan performa dalam klasifikasi dan

segmentasi sel darah merah untuk diagnosis SCD dan lebih tahan dalam ukuran, tekstur, dan bentuk sel (Zhang et al., 2020).

Penelitian dengan judul *Blood Cell Images Segmentation using Deep Learning Semantic Segmentation* yang dilakukan oleh Tran et al. pada tahun 2018 dengan dataset gambar berasal dari ALL-IDB1 basis data. Gambar yang dipakai sebanyak 42 gambar. Gambar-gambar tersebut dilakukan augmentasi dan menghasilkan gambar menjadi sebanyak 145 gambar dan 145 label. Penelitian ini bertujuan untuk melakukan segmentasi sel darah putih dan sel darah merah. Metode yang diajukan adalah menggunakan arsitektur SegNet yang dibentuk dari VGG16. Metrik yang digunakan adalah akurasi, IoU, dan BF Score. Hasil yang didapatkan adalah arsitektur SegNet yang diajukan dapat mengklasifikasi piksel sebagai sel darah putih, sel darah merah, dan lingkungan dengan baik berdasarkan metrik akurasi bernilai 0.94929, 0.91112, dan 0.87319 secara berurutan (Tran et al., 2018).

Penelitian dengan judul *Alveolar Bone and Mandibular Canal Segmentation on Cone Beam Computed Tomography Images Using U-Net* yang dilakukan oleh Widiasri et al. pada tahun 2023 dengan menggunakan gambar gigi 2D CBCT dari Rumah Sakit Universitas Airlangga. Gambar yang digunakan sebanyak 563 gambar berisi kanal mandibular dan premolar 1 hingga molar 3. Penelitian ini dilakukan untuk memberi bantuan kepada dokter dalam perencanaan implan gigi dengan parameter segmentasi *Alveolar Bone* (AB) dan *Mandibular Canal* (MC). Metode yang diajukan adalah model segmentasi dengan arsitektur U-Net dengan dilakukan beberapa tahap (segmentasi multikelas dan segmentasi biner). Metrik yang digunakan sebagai acuan performa model adalah mean IoU dengan dilakukan K-Fold. Hasil yang didapatkan adalah model U-Net berhasil mendapatkan skor mean IoU sebesar 0.85 dalam segmentasi multikelas, sedangkan model dengan segmentasi biner pada AB dan MC mendapatkan skor mean IoU sebesar 0.98 dan 0.81 (Widiasri et al., 2023).

2.2 Dasar Teori

Bab ini membahas teori-teori yang menjadi dasar dan digunakan dalam pembuatan Tugas Akhir ini. Pembuatan bab ini bertujuan untuk memberikan penjelasan mengenai teori yang dipakai dalam pembuatan sistem.

2.2.1 Golongan Darah

Setiap darah pada manusia dapat dikategorikan berdasarkan pada ada atau tidak dari antigen dan antibodi pada sel darah merah. Hingga saat ini, terdapat 44 sistem pengelompokan golongan darah berdasarkan ISBT (International Society of Blood Transfusion). Sistem golongan darah yang paling umum dipakai adalah sistem darah ABO dan Rh untuk kecocokan transfusi darah (Mathon et al., 1993).

Sistem golongan darah ABO adalah pengelompokan golongan darah berdasarkan ada atau tidaknya antigen atau antibodi pada eritrosit (sel darah merah). Antibodi yang dimaksud adalah anti-A dan anti-B antibodi yang diproduksi sebagai bentuk reaksi dari kontak langsung dengan antigen pada sel darah merah. Darah digolongkan sebagai A apabila mempunyai antigen A pada sel darah merah, begitu pula dengan B, AB, dan O. Pada umumnya, sistem golongan darah Rh dipakai setelah sistem golongan darah ABO. Sebagai contoh, golongan darah A- adalah sel darah merah memiliki antigen A dan tidak memiliki antigen Rh(D). Pembentukan antibodi Rh terjadi apabila terpapar dengan positif Rh (Encyclopædia Britannica, n.d.).

2.2.2 Uji Silang Serasi

Uji silang serasi (*Crossmatch testing*) adalah tes yang dilakukan sebelum pemberian produk darah kepada pasien. Tes ini dilakukan untuk menentukan kecocokan antara darah

resipien dengan darah pendonor, lebih tepatnya kecocokan antara antibodi resipien dengan sel darah merah pendonor. Ketidakcocokan pemberian darah kepada resipien dapat mengakibatkan kegumpalan darah, anemia hemolitik, bahkan kematian. Terdapat dua variasi dalam *crossmatch testing*, yaitu *major crossmatch testing* dan *minor crossmatch testing*. Dalam *major crossmatch testing*, sel darah donor akan dimasukkan dalam serum resipien, sedangkan *minor crossmatch testing* melakukan sebaliknya, yaitu sel darah resipien dimasukkan ke dalam serum darah donor. *Crossmatch testing* dilakukan menggunakan kartu gel yang berisi gel dan larutan darah yang sudah sesuai dengan tipe tes yang akan dicoba. Hasil dari *crossmatch* bisa dikategorikan ke dalam beberapa golongan pola reaksi, yaitu negatif, positif satu, positif dua, positif tiga, positif empat, hemolisis, dan D.P.



Gambar 2.1 Petunjuk Kategori Penggumpalan Darah

Karakteristik dari tipe pola reaksi golongan negatif (-) adalah semua darah terkumpul pada dasar tabung kartu gel. Karakteristik dari tipe pola reaksi golongan positif satu (+1) adalah terdapat beberapa darah yang naik dan tidak menjadi satu gumpalan darah. Karakteristik dari tipe pola reaksi golongan positif dua (+2) adalah beberapa gumpalan darah berada pada tengah hingga permukaan di bawah gel, namun beberapa darah masih ada yang berada pada di bagian tengah hingga dasar gel. Karakteristik dari tipe pola reaksi golongan positif tiga (+3) adalah gumpalan darah sudah berada pada bagian tengah atas dari gel dan sudah sedikit darah berada pada bagian tengah bawah gel. Karakteristik dari tipe pola reaksi golongan positif empat (+4) adalah darah berada pada di atas permukaan gel. Rangkuman karakteristik pola reaksi dapat dilihat pada Tabel 2.1.

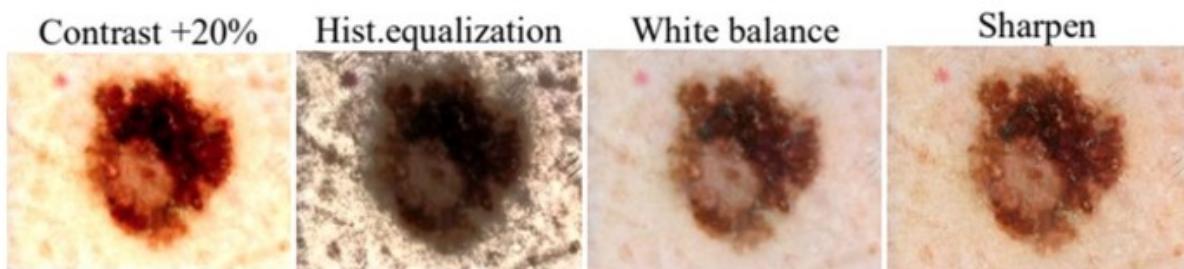
Tabel 2.1 Karakteristik Penggolongan Pola Reaksi Darah pada Uji Silang Serasi

Negatif (-)	Semua darah terkumpul pada dasar tabung kartu gel
Positif satu (+1)	Darah tersebar pada bagian tengah bawah gel, darah tidak berada pada

Tipe Penggumpalan	Karakteristik
	bagian tengah atas permukaan gel
Positif dua (+2)	Darah tersebar pada bagian tengah atas gel dan masih ada darah pada bagian tengah bawah gel
Positif tiga (+3)	Darah tersebar pada bagian tengah atas dan tidak ada atau sedikit darah berada pada bagian tengah bawah gel
Positif empat (+4)	Darah sudah berada pada diatas permukaan gel

2.2.3 Augmentasi Citra

Augmentasi citra adalah teknik yang dilakukan pada citra yang digunakan sebagai sumber latihan dari model untuk memperbanyak keragaman data latih dengan menerapkan transformasi pada data citra yang tersedia. Augmentasi citra dilakukan ketika pengumpulan citra yang akan digunakan masih sedikit dan belum memenuhi kriteria. Beberapa teknik augmentasi citra adalah *flipping* (secara horizontal maupun vertikal), *color space*, *cropping*, rotasi, translasi, penambahan *noise*, dan transformasi *color space*. Augmentasi citra diharapkan dapat mempermudah model dalam menghadapi citra yang diambil dengan sumber latih yang minim (Shorten & Khoshgoftaar, 2019). Contoh augmentasi citra dapat dilihat pada Gambar 2.2.



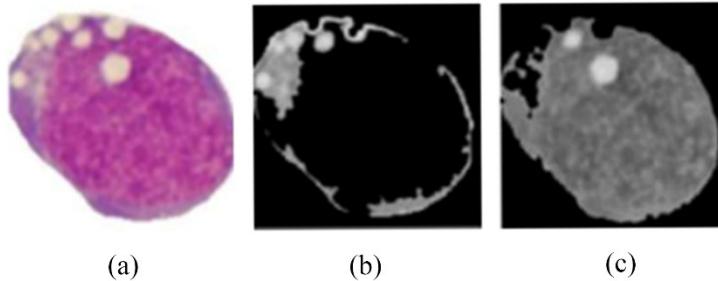
Gambar 2.2 Contoh Augmentasi Citra Warna Klasifikasi Melanoma (Mikolajczyk & Grochowski, 2018).

2.2.4 Segmentasi Citra

Segmentasi citra merupakan masalah mendasar dalam penglihatan komputer sejak awal perkembangan bidang ini. Hal ini melibatkan pemisahan citra atau *frame* video menjadi segmen dan objek yang berbeda. Segmentasi citra memiliki peran penting dalam berbagai sistem pemahaman visual dan digunakan dalam banyak aplikasi, termasuk analisis citra medis seperti ekstraksi batas tumor dan pengukuran volume jaringan, kendaraan otonom, pemantauan video, dan realitas tambahan.

Segmentasi citra dapat dilakukan dalam beberapa bentuk, yaitu segmentasi semantik, segmentasi instansi, atau keduanya. Segmentasi semantik melibatkan pelabelan piksel dengan kategori objek seperti manusia, mobil, pohon, dan langit. Ini adalah masalah yang lebih kompleks daripada klasifikasi citra keseluruhan yang hanya memprediksi satu label untuk seluruh citra. Segmentasi instansi memperluas segmentasi semantik dengan mendeteksi dan membatasi setiap objek individu dalam citra, seperti orang-orang secara terpisah. Meskipun telah ada banyak algoritma segmentasi citra dalam literatur, namun dalam beberapa tahun terakhir, model *deep learning* (DL) telah menghasilkan kemajuan yang signifikan,

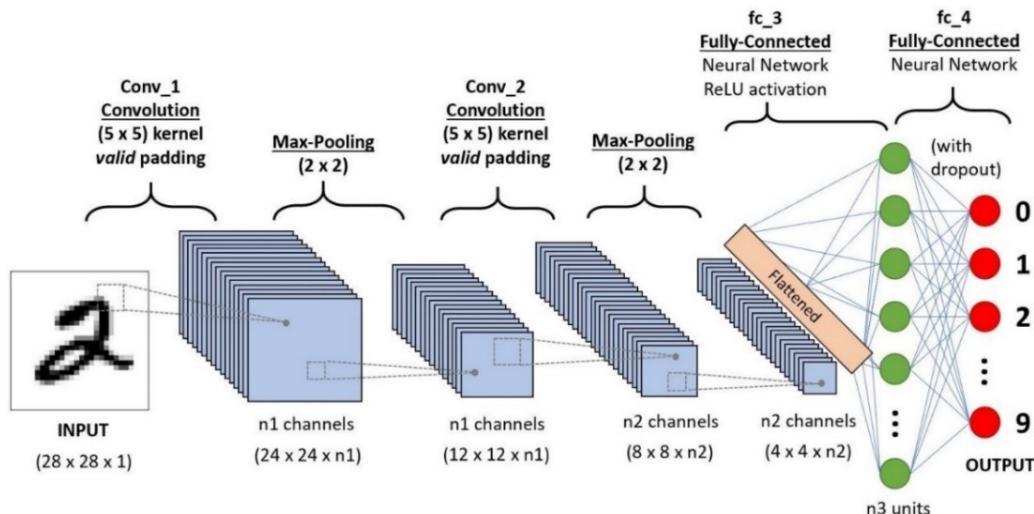
memberikan hasil yang luar biasa dalam segmentasi citra dan mencapai tingkat akurasi tertinggi dalam berbagai *benchmark*. Hal ini telah mengubah paradigma dalam bidang segmentasi citra (Minaee et al., 2021).



Gambar 2.3 Hasil Segmentasi Sel Leukimia Sel Orisinal (a), Segmentasi Sitoplasma (b), Segmentasi Area dalam (c) (Khashman & Al-Zgoul, 2010)

2.2.5 CNN (Convolutional Neural Network)

Arsitektur CNN lebih terfokus dalam mengatasi masukan yang terdiri dari beberapa citra. Salah satu perbedaan CNN dan ANN adalah neuron yang mendirikan arsitektur CNN terdiri dari tiga dimensi, yaitu dimensi citra masukan (tinggi dan lebar citra) dan kedalaman dari *activation volume*. Contohnya adalah citra yang dimasukkan terdiri dari citra sebesar $32 \times 32 \times 3$ yang nantinya akan menghasilkan lapisan terakhir dengan dimensi $1 \times 1 \times n$ (banyaknya kelas) (O'Shea & Nash, 2015).



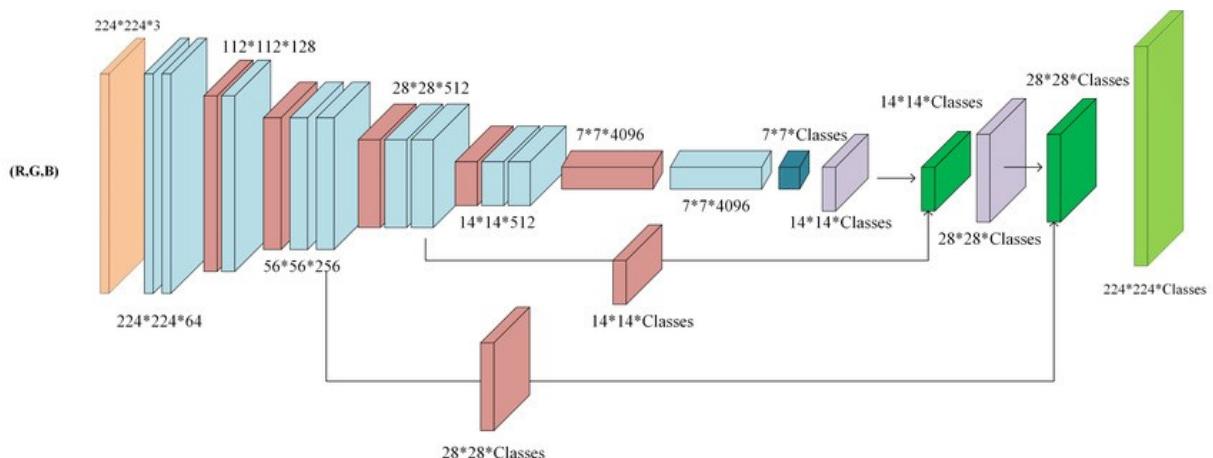
Gambar 2.4 Ilustrasi Komponen pada CNN

Arsitektur CNN terdiri dari tiga macam lapisan, yaitu *convolutional layer*, *pooling layer*, dan *fully-connected layer*. *Convolutional layer* sendiri dipakai untuk menentukan hasil keluaran dari neuron menggunakan hasil kalkulasi dari neuron yang terhubung dari bagian kecil dari citra masukan dan *weight* dari neuron itu sendiri. *Pooling layer* akan melakukan *downsampling* dimensionalitas dari masukan dan mengurangi parameter. *Fully-connected layer* melakukan fungsi sama halnya pada ANN untuk mendapatkan hasil kelas dari aktivasi (O'Shea & Nash, 2015).

2.2.6 Arsitektur Klasifikasi CNN

1. FCN (Fully Convolutional Network)

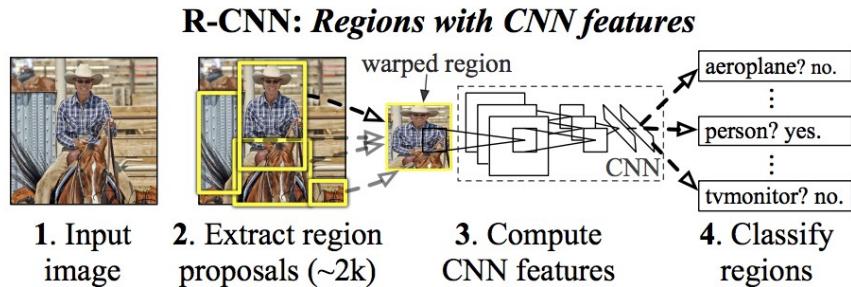
Setiap citra masukan pada jaringan konvolusional ini terdiri dari tiga dimensi dengan ukuran dari $h \times w \times d$ (tinggi, lebar, dan color channel dari citra). Setiap hasil dari tiap lapisan hanya berlaku secara lokal dan relatif terhadap lapisan tersebut saja. Dalam FCN, terdapat tahapan convolution yang sama dilakukan pada arsitektur CNN. Perbedaan dari CNN dan FCN adalah tidak adanya *fully-connected layer* atau *dense layer* yang digantikan dengan proses deconvolution, yaitu melakukan upsampling hasil dari convolution dengan filter yang sudah ditentukan. Hasil dari *deconvolution* ini akan digunakan bersamaan dengan hasil *convolution* dengan dimensi yang sama untuk menggabungkan kedua informasi yang dapat digunakan untuk melakukan *upsampling* ke dimensi masukan awal dan mendapatkan hasil segmentasi. Proses tersebut dinamakan *Skip Connections*. Ilustrasi arsitektur FCN dapat dilihat pada Gambar 2.5 (J. Long et al., 2015).



Gambar 2.5 Contoh Visualisasi dari Arsitektur FCN (J. Long et al., 2015)

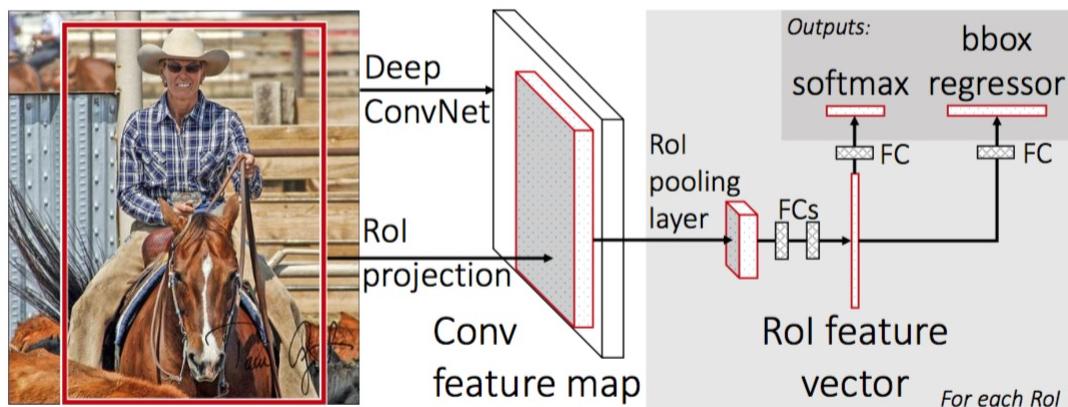
2. R-CNN (Regional Convolutional Neural Network)

Regional Convolutional Neural Network adalah model yang diajukan oleh Ross Girchick. R-CNN melakukan seleksi bagian tertentu dari sebuah citra dengan fitur yang penting dan menandai bagian tersebut dengan *bounding boxes* beserta kategori dari bagian citra tersebut. Setelah itu, CNN akan digunakan untuk mengklasifikasi hasil dari penandaan *bounding boxes* dan memperbaiki *bounding boxes*. Komponen dari R-CNN terdiri dari *Selective Search*, CNN, dan *Support Vector Machine* (SVM). *Selective Search* digunakan untuk mendapatkan *bounding boxes* awal dengan besar dan lebar yang berbeda – beda. Setelah itu, CNN digunakan untuk mengolah area citra dan mendapatkan fitur – fitur dari citra tersebut. Setelah itu, SVM digunakan untuk mengolah fitur – fitur yang didapatkan dari CNN untuk memprediksi atau mengklasifikasi label dari area citra tersebut. Kelemahan dari R-CNN adalah *Selective Search Algorithm* yang kurang fleksibel, performa yang lambat, serta pelatihan model membutuhkan waktu yang cukup lama. Ilustrasi arsitektur dari R-CNN dapat dilihat pada Gambar 2.6 (A.K. Shetty et al., 2021).



3. Fast R-CNN

Dalam arsitektur Fast R-CNN, keseluruhan citra menjadi masukan dan area citra dengan label objeknya. Beberapa lapisan konvolusional dan *max pooling* digunakan untuk memproses citra dan menghasilkan peta konvolusi fitur. RoI *pooling* digunakan untuk mengambil vektor fitur dengan panjang tetap dari peta fitur untuk setiap objek. Semua vektor fitur digunakan sebagai input dari lapisan klasifikasi, yaitu *Softmax Probability* untuk hasil prediksi. Fast R-CNN dapat mengkonsumsi waktu komputasi lebih sedikit karena keseluruhan citra adalah masukan CNN untuk mendapatkan fitur, menggunakan Softmax sebagai klasifikasi, dan terdapat lapisan RoI Pooling untuk mempercepat dan mempermudah dalam pelatihan model. Ilustrasi Fast R-CNN dapat dilihat pada Gambar 2.7 (A.K. Shetty et al.,2021).

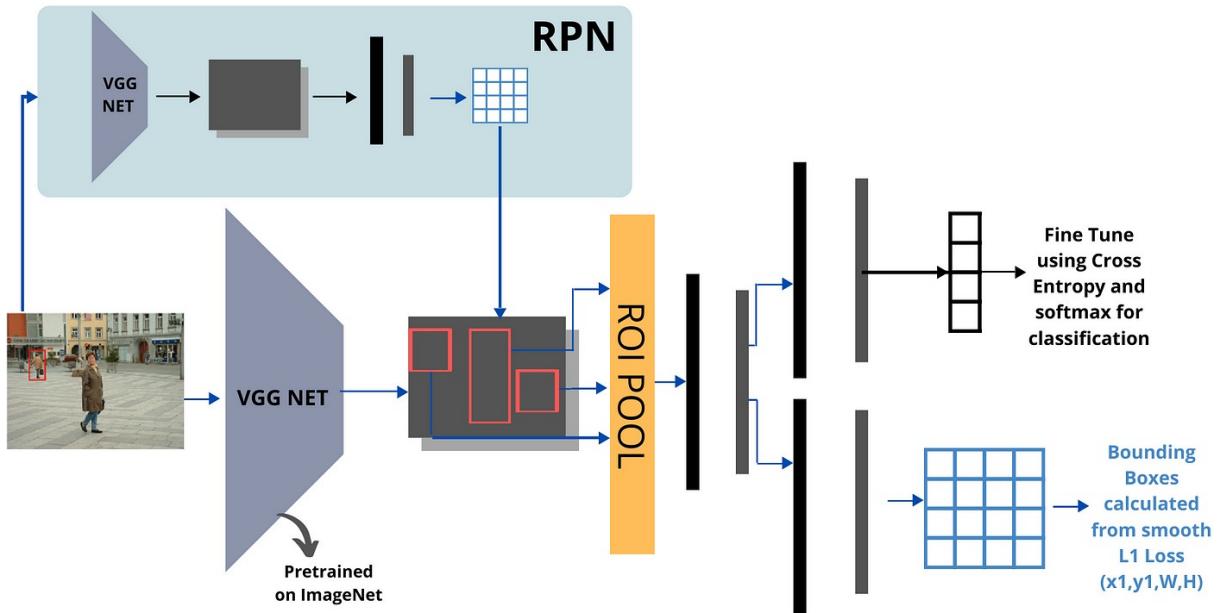


Gambar 2.7 Arsitektur dari Fast R-CNN (A.K. Shetty et al.,2021)

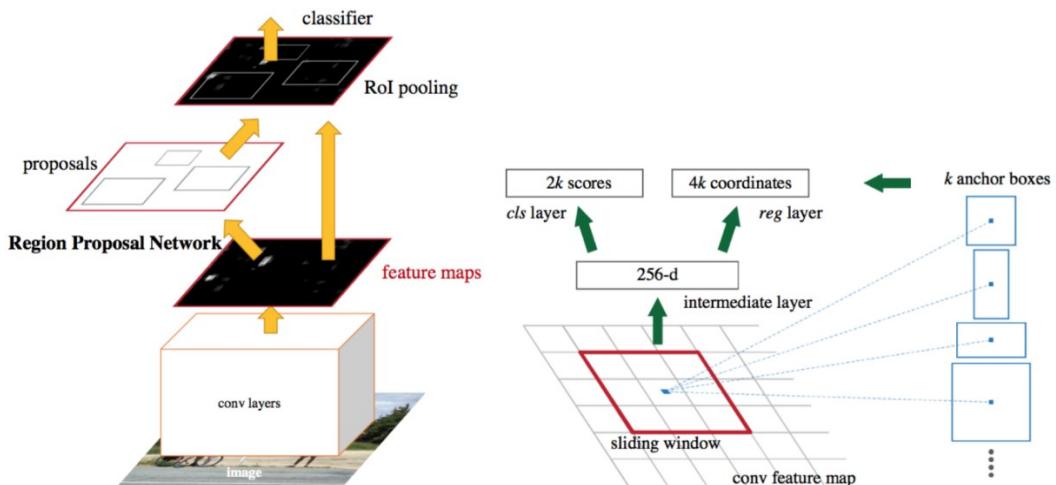
4. Faster R-CNN

Salah satu kelemahan dalam arsitektur Fast R-CNN adalah menggunakan algoritma *Selective Search* yang memperlambat performa. Dalam Faster R-CNN, algoritma *Selective Search* tidak dipakai dan digantikan oleh *Region Proposal Network* (RPN). RPN akan memasukkan citra (dengan ukuran yang tidak ditentukan) sebagai masukan dan akan mengeluarkan kotak dengan nilai keobjekan dari area tersebut. RPN menggunakan arsitektur FCN untuk mendapatkan map fitur dari citra. Untuk mendapatkan proposal area objek, akan dibentuk *neural network* kecil dengan ukuran dimensi sesuai dengan hasil map fitur sebelumnya. Map fitur akan dipetakan dan *anchor* akan ditentukan untuk *sliding window*. *Sliding window* akan dihubungkan dengan lapisan *2k scores* dan *4k coordinates*. Lapisan *2k scores* adalah lapisan yang menentukan karea adalah objek atau tidak, sedangkan *4k coordinates* adalah koordinat dari *k* area. Ilustrasi arsitektur Faster R-CNN dapat dilihat pada Gambar 2.8 dan ilustrasi RPN dapat dilihat pada Gambar 2.9 (A.K.

Shetty et al.,2021) (S. Ren et al., 2015).



Gambar 2.8 Arsitektur dari Faster R-CNN (A.K. Shetty et al.,2021)



Gambar 2.9 Ilustrasi dari RPN (A.K. Shetty et al.,2021)

5. Mask R-CNN

Mask R-CNN digunakan untuk melakukan segmentasi dalam level piksel. Mask R-CNN adalah modifikasi dari arsitektur Faster R-CNN dengan tambahan tahapan *masking*. Selain itu, Mask R-CNN menggunakan lapisan RoI Alignment yang menggantikan RoI Pooling. Lapisan RoI Alignment memungkinkan untuk memakai tambahan FCN untuk klasifikasi hingga level piksel. Dapat disimpulkan bahwa terdapat dua jaringan konvolusi, yaitu FCN untuk masking dengan konvolusi hingga level piksel dan R-CNN untuk mendapatkan *bounding box* dan *classifier* untuk kategori dari objek tersebut. Ilustrasi arsitektur Mask R-CNN dapat dilihat pada Gambar 2.10 (A.K. Shetty et al.,2021).

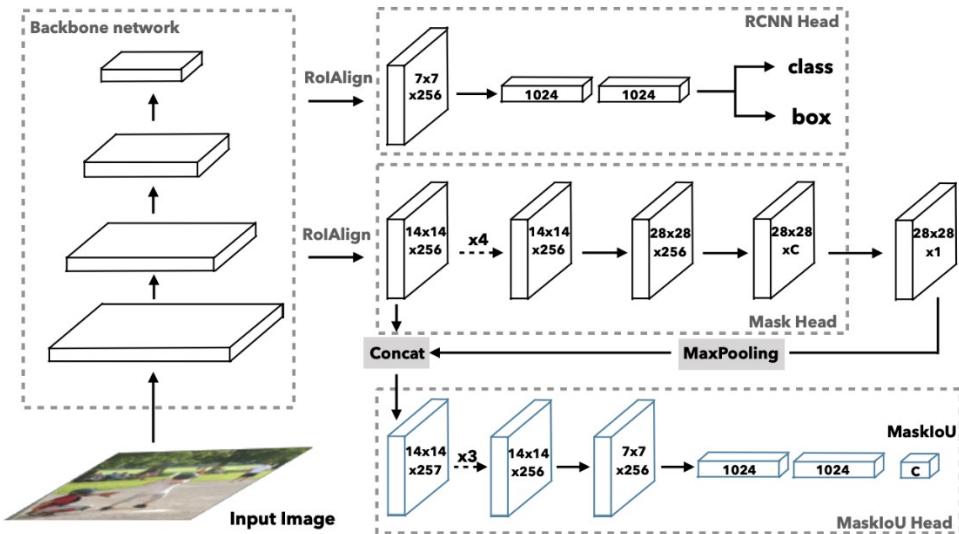


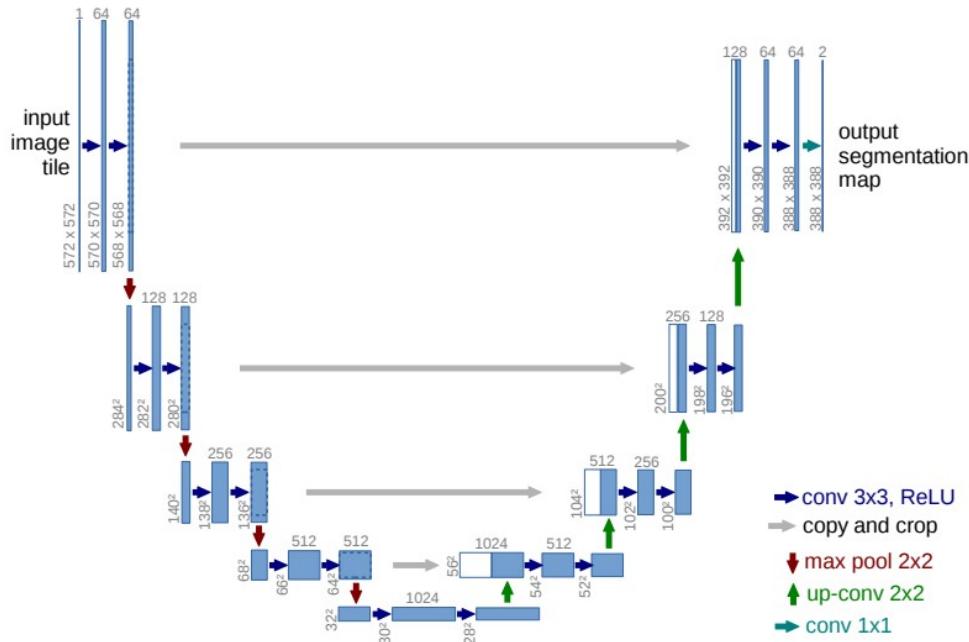
Figure 3. Network architecture of Mask Scoring R-CNN. The input image is fed into a backbone network to generate RoIs via RPN and RoI features via RoIPooling. The RCNN head and Mask head are standard components of Mask R-CNN. For predicting MaskIoU, we use the predicted mask and ROI feature as input. The MaskIoU head has 4 convolution layers (all have kernel=3 and the final one uses stride=2 for downsampling) and 3 fully connected layers (the final one outputs C classes MaskIoU.)

Gambar 2.10 Arsitektur dari Mask R-CNN (A.K. Shetty et al.,2021)

2.2.7 Arsitektur Segmentasi CNN pada Uji Silang Serasi

Model Uji Silang Serasi akan menggunakan beberapa arsitektur segmenasi CNN. Berikut adalah arsitektur yang digunakan :

1. U-Net



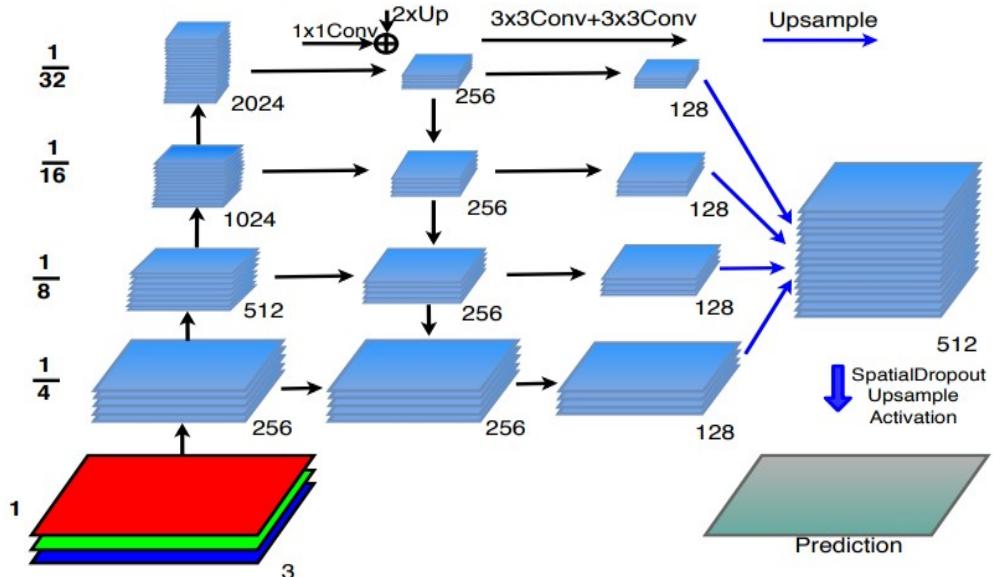
Gambar 2.11 Ilustrasi Arsitektur dari U-Net (Ronneberger et al., 2015).

U-Net adalah arsitektur yang diajukan untuk mengatasi masalah permasalahan klasifikasi setiap piksel ke dalam label yang sudah ditentukan. Hal ini sering sering dijumpai pada ranah medis. Arsitektur U-Net dibentuk untuk mengatasi masalah tersebut. U-Net terdiri dari dua 3×3 convolution layer yang masing-masing diikuti dengan Rectified

Linear Unit (ReLU) dan 2×2 *max pooling layer* dengan *stride* 2 untuk *downsampling*. Setiap tahap *downsampling* akan menggandakan *feature channel*. Selanjutnya diikuti dengan tahap *upsampling* yang terdiri dari 2×2 *up-convolution layer* dan dua 3×3 *convolution layer* dan masing-masing diikuti dengan ReLU. Pada lapisan terakhir terdapat 1×1 *convolution layer* yang digunakan untuk memetakan setiap *feature vector* ke dalam label yang sudah ditentukan. Arsitektur ini memiliki total 23 *convolutional layer*. Ilustrasi arsitektur U-Net dapat dilihat pada Gambar 2.11 (Ronneberger et al., 2015).

2. Feature Pyramid Network (FPN)

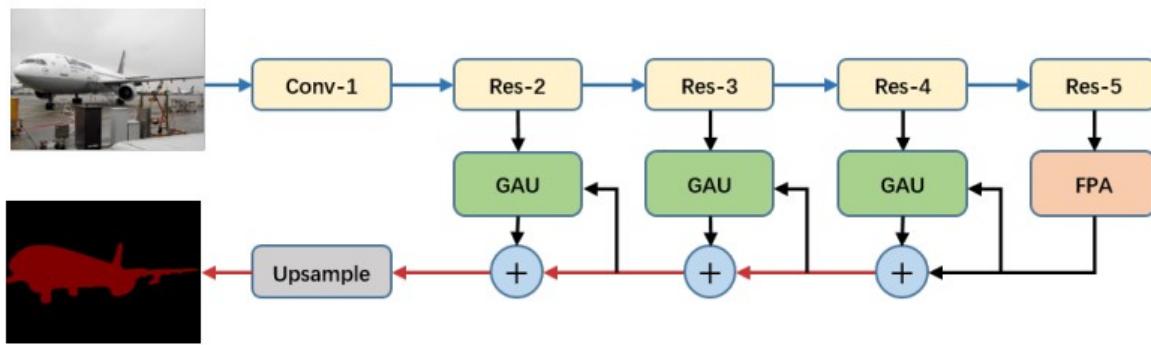
FPN menggunakan hirarki piramida dari *convolutional network* untuk membuat piramida fitur. FPN terdiri dari dua jalur, yaitu *bottom-up* dan *top-down*. Jalur *bottom-up* adalah *convolutional network* seperti biasa. Puncak pada jalur *bottom-up* adalah 1×1 *convolution filter* untuk mengurangi kedalaman *channel*, kemudian masuk ke dalam dua 3×3 *convolution layer* untuk mendapat peta fitur. Seiring berjalan melalui jalur *top-down*, *upsampling* dilakukan dari *layer* sebelumnya. Setelah setiap *layer* dilalui, semua hasil konvolusi akan dilakukan *upsampling* dan dilakukan 3×3 *convolution filters*, *batch normalization*, dan aktivasi ReLU. Ilustrasi arsitektur FPN dapat dilihat pada Gambar 2.12 (Seferbekov et al., 2018).



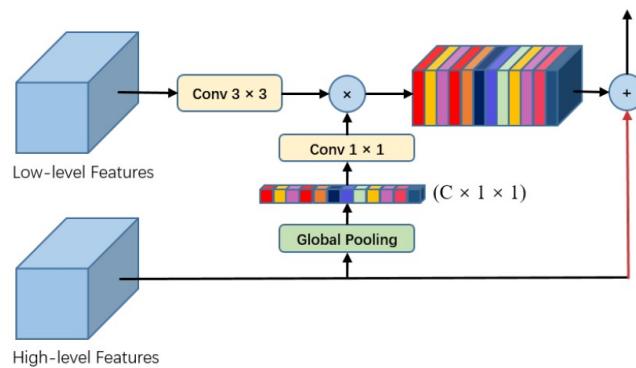
Gambar 2.12 Ilustrasi Arsitektur dari FPN (Seferbekov et al., 2018).

3. Pyramid Attention Network (PAN)

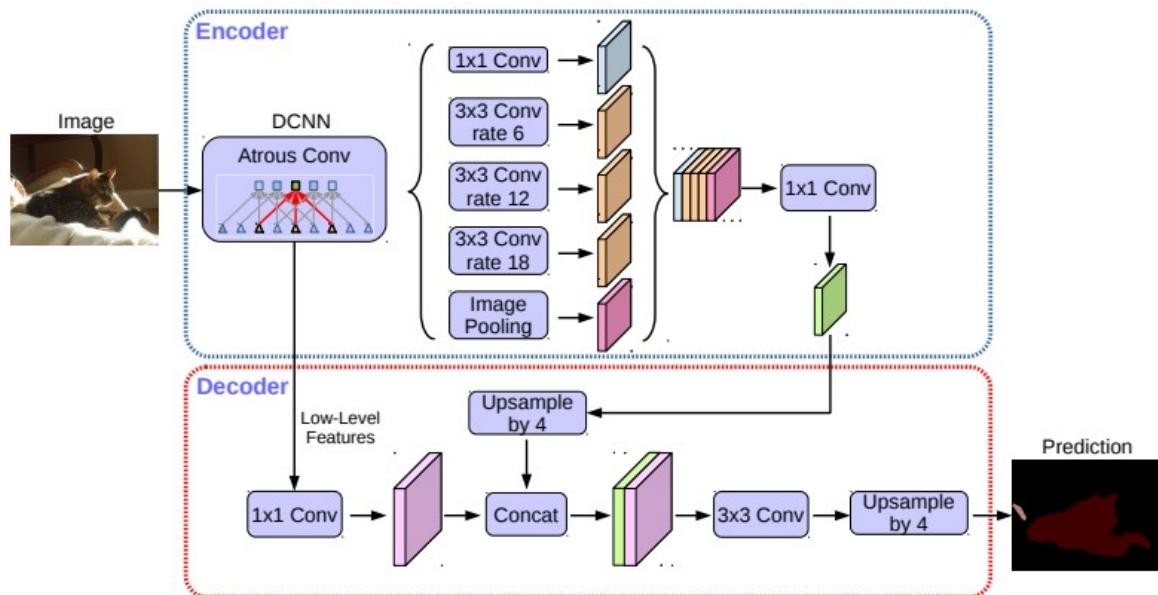
PAN adalah *neural network* yang menggunakan mekanisme atensi dan piramida spasial untuk mengekstrak fitur secara presisi dalam tingkatan piksel. Terdapat dua komponen penting, yaitu *Feature Pyramid Attention* (FPA) dan *Global Attention Upsample* (GAU). FPA adalah inovasi dari *Spatial Pyramid Pooling* dengan menggunakan pendekatan dari FPN, yaitu menggunakan tiga macam tingkatan dalam menggabungkan fitur. Hasil dari prosedur tersebut digabung dengan fitur original dan *global pooled* fitur. Modul GAU melakukan *global average pooling* untuk menyediakan konteks global sebagai panduan dalam fitur *low-level* untuk mereduksi kanal, kemudian fitur *high-level* digabung dengan fitur *low-level* dan dilakukan *upsampling*. Ilustrasi arsitektur PAN dapat dilihat pada Gambar 2.13, sedangkan ilustrasi komponen GAU yang dipakai pada Gambar 2.14 (Li et al., 2018).



Gambar 2.13 Ilustrasi dari *Feature Pyramid Attention* (Li et al., 2018).



Gambar 2.14 Ilustrasi GAU pada Arsitektur PAN (Li et al., 2018)



Gambar 2.15 Ilustrasi dari DeepLabV3+ (Chen et al., 2018)

4. DeepLabV3+

DeepLabV3+ adalah inovasi untuk meningkatkan performa dari DeepLabV3. Inovasi yang dilakukan adalah perubahan struktur menjadi struktur *encoder-decoder*. Pada bagian *encoder*, *atrous convolution* digunakan untuk mengontrol resolusi dari fitur dan mengatur filter *field of view* untuk menangkap informasi. Bagian *decoder* terdapat penggabungan dengan fitur *low-level* dengan hasil dari bagian *encoder* dan dimasukkan ke 3×3 *convolutional layer* dan *upsampling*. *Decoder* ditambahkan ke dalam DeepLabV3+ untuk meningkatkan kemampuan model dalam segmentasi. Dalam *decoder*, komponen yang digunakan adalah hasil *atrous convolution* dengan *low level feature* dan dikombinasikan dengan hasil *atrous convolution* yang sama, tetapi dilakukan *upsampling* sebesar empat. Setelah itu, model akan dilakukan konvolusi 3×3 dan dilakukan *upsampling* dengan ukuran empat untuk mendapatkan hasil prediksi model. Ilustrasi arsitektur DeepLabV3+ dapat dilihat pada Gambar 2.15 (Chen et al., 2018).

2.2.8 Fungsi Kerugian untuk Uji Silang Serasi

Fungsi Kerugian adalah fungsi yang memetakan sebuah tugas ke dalam angka riil sebagai representasi dari “biaya” secara intuitif. Setiap tugas pembelajaran mesin adalah untuk meminimalisasi nilai dari “biaya”. Fungsi kerugian dalam pembelajaran mesin umumnya representasi dari nilai asli dan nilai prediksi dari model pembelajaran mesin. Fungsi kerugian ini juga representasi dari performa model selama pelatihan. Fungsi kerugian ini menjadi patokan dalam memperbarui parameter dalam model *deep learning* (DL) berdasarkan nilai dari fungsi kerugian. Pada awal pelatihan, nilai fungsi kerugian bernilai sangat besar. Seiringnya pelatihan berjalan, nilai fungsi kerugian ini akan turun dengan metode *gradient descent* dalam proses *backpropagation*. Pada akhirnya, nilai prediksi model mendekati nilai asli dan nilai fungsi kerugian ini mencapai batasan yang sudah ditentukan atau tidak berkurang (Xu et al., 2023).

Terdapat dua fungsi kerugian yang digunakan dalam pembuatan model DL pada penelitian ini, yaitu

1. Jaccard Loss

Jaccard Loss (J) adalah fungsi kerugian yang berdasar pada indeks Jaccard. Indeks Jaccard adalah pengukuran kemiripan antara dua set. Kalkulasi dari indeks Jaccard mengacu pada Persamaan 2.1.

$$J(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \quad (2.1)$$

dengan X dan Y representasi dari nilai prediksi model dan nilai asli secara berurutan. $|X \cap Y|$ merepresentasikan anggota pada set X dan Y , $|X|$ merepresentasikan jumlah anggota pada set X , dan $|Y|$ merepresentasikan jumlah anggota pada set Y . Representasi prediksi model semakin mendekati nilai asli jika nilai pada indeks Jaccard semakin mendekati 1. Kalkulasi fungsi kerugian menggunakan indeks Jaccard mengacu pada Persamaan 2.2.

$$L_{Jaccard} = 1 - J = 1 - \frac{\sum_{i=1}^N y_i \hat{p}_i + 1}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{p}_i - \sum_{i=1}^N y_i \hat{p}_i + 1} \quad (2.2)$$

dengan $y_i \in \{0, 1\}$ adalah label kelas pada piksel i dan $\hat{p}_i \in [0, 1]$ adalah prediksi dengan nilai probabilitas bahwa piksel i hingga jumlah piksel N akan terkласifikasi menjadi kelas positif.

2. Dice Loss

Sama seperti Jaccard index, koefisien Dice digunakan untuk menghitung kemiripan dari dua set. Koefisien Dice didefinisikan pada Persamaan 2.3.

$$D(X, Y) = 2 \vee X \cap Y \vee \frac{|X| + |Y| - 2|X \cup Y|}{|X| + |Y|} \quad (2.3)$$

Fungsi kerugian dalam menggunakan koefisien Dice (D) dapat dilihat pada Persamaan 2.4.

$$L_{Dice} = 1 - D = 1 - \frac{2 \sum_{i=1}^N y_i \hat{p}_i + 1}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{p}_i + 1} \quad (2.4)$$

2.2.9 Evaluasi Performa Model Uji Silang Serasi

Evaluasi Performa model segmentasi dengan teknik DL perlu dilakukan untuk menentukan seberapa baik dalam menjalankan tugas yang diberikan, dalam Tugas Akhir ini adalah melakukan segmentasi darah. Penentu dari performa model segmentasi sama dengan model klasifikasi. Terdapat empat kategori klasifikasi biner nilai prediksi berdasarkan nilai asli, yaitu :

1. **True Positive (TP)** : TP adalah kelas pada nilai prediksi yang berhasil memprediksi label positif pada nilai asli sebagai label positif.
2. **True Negative (TN)** : TN adalah kelas pada nilai prediksi yang berhasil memprediksi label negatif pada nilai asli sebagai label negative.
3. **False Positive (FP)** : FP adalah kelas pada nilai prediksi yang salah memprediksi label negatif pada nilai asli sebagai label positif. Skenario ini diketahui sebagai *Type 1 Error*.
4. **False Negative (FN)** : FN adalah kelas pada nilai prediksi yang salah memprediksi label positif pada nilai asli sebagai label negatif. Skenario ini diketahui sebagai *Type 2 Error*.

Hasil dari pengkategorian beberapa metrik yang digunakan ini dapat digunakan dalam metrik yang akan dipakai. Metrik yang dipakai pada Tugas Akhir ini adalah *IoU-score*, *F1-score*, akurasi, presisi, *recall*, *specificity*, *negative predictive value* (NPV), dan waktu inferensi (Vujovic, 2021).

F1-score dan *Intersection over Union* atau *IoU-score* adalah metrik yang sering digunakan terutama pada model pembelajaran mesin visi komputer. *F1-score* diukur dari sensitivitas atau *recall* dan presisi dari prediksi. Presisi digunakan untuk mengatasi *false positive* terutama pada dataset yang *imbalance*. *F1-score* atau indeks Sørensen-Dice (DSC)

didefinisikan sebagai rata-rata harmonik antara sensitivitas dan presisi. Persamaan *F1-Score* dapat dilihat pada Persamaan 2.5. *IoU-score* juga dikenal sebagai indeks Jaccard. Definisi persamaan *IoU-score* dapat dilihat pada Persamaan 2.6 (Müller et al., 2022).

$$DSC = \frac{2TP}{2TP+FP+FN} \quad (2.5)$$

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

Sensitivitas dan spesifisitas sudah sangat umum digunakan dan dijadikan standar terutama pada bidang medis sebagai tolak ukur. Sensitivitas atau *true positive rate* (TPR) atau *recall* adalah metrik yang berfokus pada kemampuan model dalam memprediksi TP, sedangkan spesifisitas (*specificity*) adalah metrik yang berfokus pada kemampuan model dalam memprediksi TN. Spesifisitas dalam segmentasi citra medis diasosiasikan sebagai kemampuan model untuk mendeteksi kelas *background* pada citra. Persamaan sensitivitas dapat terlihat pada Persamaan 2.7. Persamaan spesifisitas dapat terlihat pada Persamaan 2.8 (Müller et al., 2022).

$$Sensitivity = \frac{TP}{TP+FN} \quad (2.7)$$

$$Specificity = \frac{TN}{TN+FP} \quad (2.8)$$

Akurasi atau diketahui sebagai indeks Rand atau akurasi piksel adalah metrik evaluasi paling umum dipakai dalam statistik. Akurasi didefinisikan sebagai jumlah prediksi yang benar, terdiri dari TP dan TN, dibandingkan dengan seluruh total prediksi. Namun, akurasi sangat tidak disarankan dipakai dalam menghadapi data *imbalance*, terutama pada segmentasi citra medis. Hal ini terjadi karena TN yang berjumlah banyak sehingga terjadi ketimpangan dan nilai dari akurasi cenderung tinggi. Nilai akurasi sering bernilai 90% atau mendekati 100% dan sering terjadi *misleading*. Persamaan akurasi terlihat pada Persamaan 2.9 (Müller et al., 2022).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.9)$$

[Halaman ini sengaja dikosongkan]

BAB III METODOLOGI

3.1 Deskripsi Umum Sistem

Penelitian ini akan membuat model dengan kapabilitas melakukan segmentasi bagian darah pada setiap citra tabung kartu gel hasil uji silang serasi pada pelayanan di Unit Transfusi Darah di RSUD Dr. Soetomo. Model yang diajukan pada penelitian ini adalah model *semantic segmentation* dengan teknik *deep learning* dan metode *transfer learning* untuk mempercepat waktu pelatihan model dan mempertimbangkan *dataset* yang sudah diambil cukup terbatas. Model *deep learning semantic segmentation* sendiri terdiri dari *encoder* dan *decoder* yang sudah ditentukan. *Encoder* yang akan dipakai adalah ResNet34, ResNeXt50, dan EfficientNetB5, sedangkan *decoder* yang akan dipakai adalah FPN, PAN, DeepLabV3+. Penjelasan mengenai *encoder* akan dibahas pada Subbab 3.4, sedangkan penjelasan mengenai *decoder* ada pada Subbab 2.2.7.

Performa dari model sendiri perlu dilihat setelah melakukan pelatihan model. Metrik yang akan dipakai adalah skor IoU, skor F1, akurasi, presisi, *recall*, spesifisitas, NPV, dan waktu inferensi. Metrik yang akan menjadi prioritas penentu adalah skor IoU, skor F1, dan *recall*. Hal ini diperlukan karena metrik terutama skor IoU merupakan penilaian prediksi model terhadap gabungan prediksi model dan *ground truth* label segmentasi, skor F1 sebagai validasi dan mencegah bias, dan *recall* merupakan metrik yang sangat penting untuk mengatasi masalah medis, terutama pada segmentasi citra medis yang lebih membutuhkan kemampuan model dalam mendekteksi darah tanpa meninggalkan *true positive* yang lain.

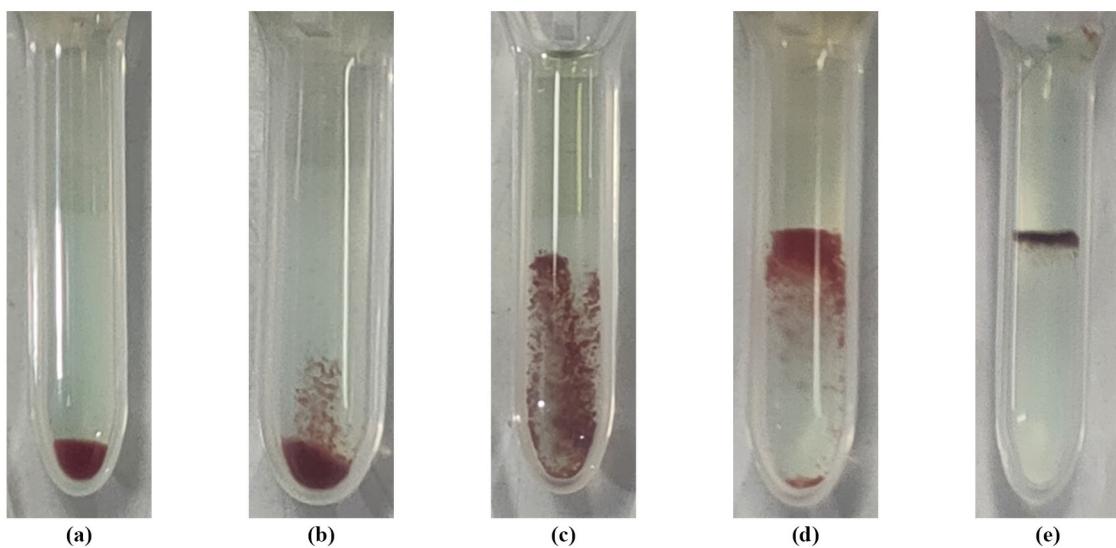
3.2 Dataset yang Digunakan

Dataset yang digunakan untuk penelitian ini adalah hasil pengambilan citra pada Unit Transfusi Darah (UTD) RSUD Dr. Soetomo, Surabaya. *Dataset* kartu gel uji silang serasi yang diambil adalah hasil pengajuan setiap pasien yang ingin melakukan transfusi darah pada RSUD Dr. Soetomo. Mengambil citra pada RSUD Dr. Soetomo harus mengajukan perizinan kode etik kepada komite etik penelitian RSUD Dr. Soetomo. Pengajuan sudah dilakukan dengan nomor 2439/121/4/IX/2023 dan sudah mendapatkan persetujuan. Terdapat lima kategori yang diambil pada darah tersebut. Kategori tersebut adalah negatif, positif satu, positif dua, positif tiga, dan positif empat. Citra yang diambil tidak memiliki kriteria pasien tertentu. Hasil yang didapatkan adalah total sebanyak 131 citra mentah kartu gel uji silang serasi. Salah satu contoh gambar yang diambil dapat dilihat pada Gambar 3.1.



Gambar 3.1 Kartu Gel untuk Uji Silang Serasi

Proses selanjutnya adalah melakukan pemotongan atau *cropping* pada setiap tabung yang ada pada citra uji silang serasi. Hal ini harus dilakukan karena pada sistem yang diajukan pada Tugas Akhir ini tidak mendukung *object detection* dan harus dilakukan *cropping* secara manual. Setiap hasil *cropping* akan dikelompokkan pada kategori sesuai dengan *labelling* yang dilakukan oleh ahli. Proses ini menghasilkan citra sebanyak 719 citra tabung yang sudah sesuai dan dapat dimasukkan ke dalam dataset pengujian. Detail dari 719 citra tersebut adalah 368 citra kategori negatif, 148 citra kategori positif satu, 130 citra positif dua, 32 citra positif tiga, dan 41 citra positif empat. Contoh hasil gambar yang didapat dengan kelas pola reaksinya dapat dilihat pada Gambar 3.2.



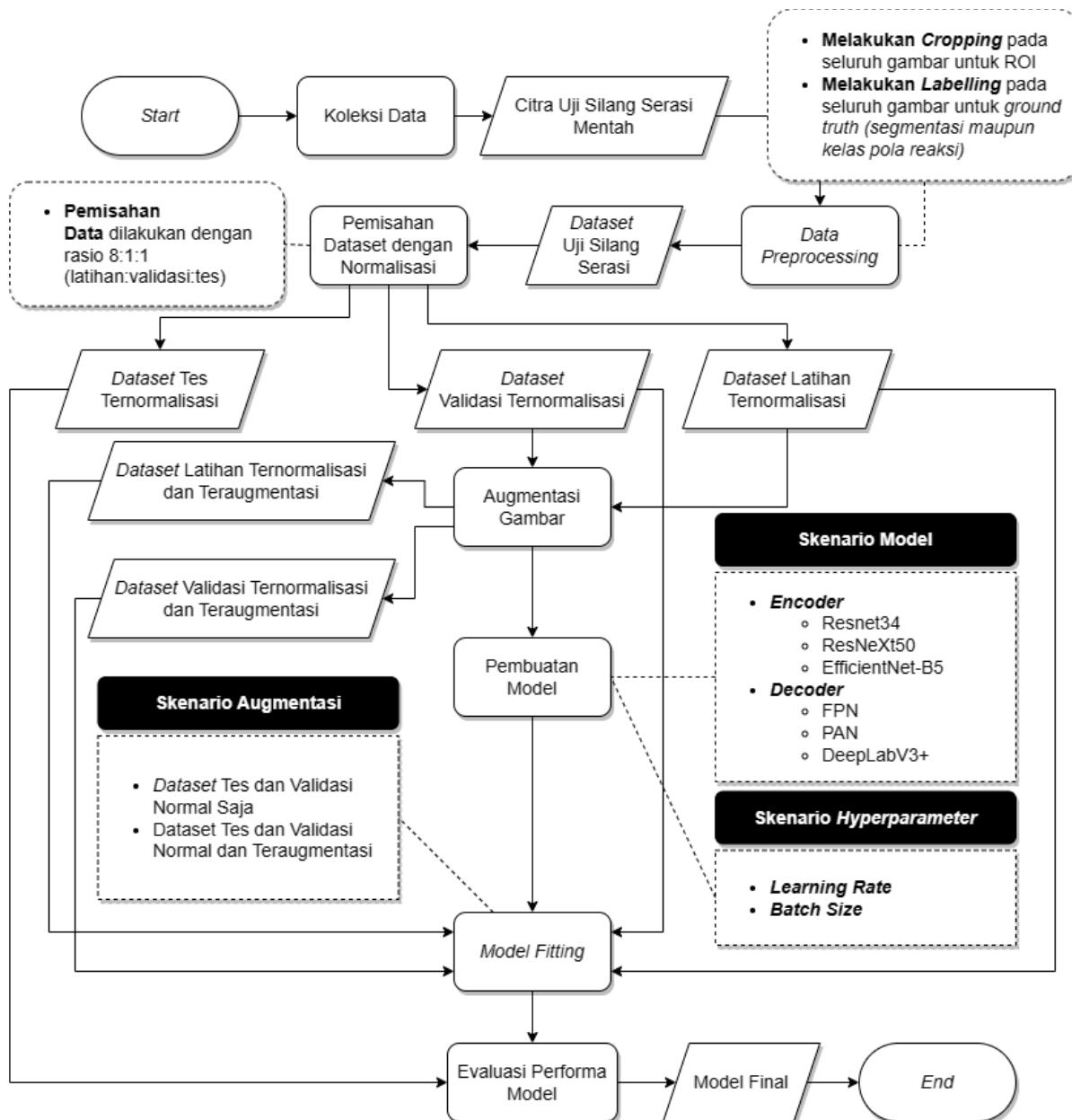
Gambar 3.2 Contoh Pemotongan Gambar untuk Tabung dengan Kelas Negatif (a), Positif Satu (b), Positif dua (c), Positif Tiga (d), dan Positif Empat (d).

3.3 Peralatan pendukung

Tugas Akhir ini dilakukan pada sistem lokal dan sistem awan (*cloud system*) dengan spesifikasi sistem. Tabel 3.1 berisi spesifikasi sistem lokal yang digunakan dengan bantuan sistem awan yang disediakan oleh Google melalui Google Drive dan Google Colab.

No	Aspek	Spesifikasi
Tabel 3.1 Spesifikasi Peralatan Pendukung		
2	<i>Processor</i>	AMD Ryzen 5600H
3	<i>Memory (RAM)</i>	16 GB
4	<i>Internal Memory</i>	512 GB
5	<i>Graphics Processing Unit (GPU)</i>	NVIDIA RTX 3050 4GB
6	Bahasa Pemrograman	Python 3.10
7	<i>Software</i>	Google Colaboratory, Microsoft Edge, Adobe Photoshop 2021
8	<i>Python Library</i>	PyTorch, Numpy, Matplotlib, Pandas, Label Studio, imgaug

3.4 Perancangan Sistem



Gambar 3.3 Diagram Alur Proses Penelitian

Proses penelitian ini terdiri dari beberapa tahap, yaitu koleksi data, *data preprocessing*, pembagian *dataset*, augmentasi data, *model building*, *model fitting*, evaluasi performa, dan mendapatkan model terbaik berdasarkan evaluasi performa. Gambar 3.3 adalah alur pengerjaan dari penelitian ini.

3

3.1

3.2

3.3

3.4

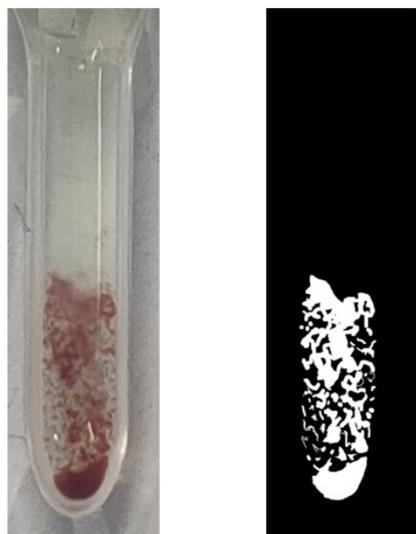
3.4.1 Koleksi data

Koleksi data dilakukan untuk mengumpulkan citra hasil uji silang serasi yang dibutuhkan. Pengumpulan data dilakukan pada Unit Transfusi Darah RSUD Dr. Soetomo,

Surabaya, Jawa Timur. Pengumpulan citra menggunakan *smartphone* dan dipastikan bahwa citra yang ada sudah fokus dan jelas dengan *background* polos untuk memudahkan dalam proses anotasi. Pengumpulan citra hasil uji silang serasi dilakukan selama Juni 2023 hingga Agustus 2023. Hasil yang didapatkan adalah 131 citra kartu gel uji silang serasi. Setiap kartu gel uji silang serasi memiliki delapan tabung individu yang dapat digunakan, maupun tidak digunakan. Setiap citra yang sudah diambil harus dilakukan *cropping* untuk mendapatkan bagian tabung saja (*Region of Interest*).

3.4.2 Data Preprocessing

Preprocessing Data dilakukan pada setiap citra kartu gel hasil uji silang serasi untuk mendapatkan ROI (*Region of Interest*), yaitu setiap tabung yang telah digunakan dari dasar hingga ujung atas tabung. Hasil *cropping* citra kartu gel hasil uji silang serasi dapat dilihat pada Gambar 3.4. Semua citra diurutkan dengan angka 1 hingga banyaknya jumlah tabung yang sudah terambil, yaitu 719. Semua citra tabung dilakukan *labelling* kategori pola reaksi oleh ahli dalam bidang ini dan disusun rapi ke dalam file JSON dengan nama *file* citra dan label dari kategori pola reaksi.



Gambar 3.4 Contoh Gambar Tabung dengan *Ground Truth* Segmentasi Darah

Setelah melakukan *cropping*, 719 citra akan dilakukan anotasi pada setiap piksel termasuk darah menggunakan Label Studio. Label Studio adalah *platform* untuk melakukan *labelling* pada tugas visi komputer seperti klasifikasi citra, deteksi objek, dan *semantic segmentation* (tugas utama pada Tugas Akhir ini). Hasil dari anotasi ini adalah *ground truth* dari segmentasi darah pada citra tabung hasil uji silang serasi. Setelah dilakukan anotasi, label segmentasi darah tersebut diekspor dan disimpan sesuai dengan pasangan citra uji silang serasi yang asli. Semua citra uji silang serasi, label segmentasi citra uji silang serasi, dan label kategori pola reaksi disimpan di dalam satu tempat dan tersusun untuk memudahkan penelitian ini. Contoh citra uji silang serasi beserta hasil label segmentasi darah dapat dilihat pada Gambar 3.4.

3.4.3 Pemisahan Dataset dengan Normalisasi

Crossmatch dataset yang sudah dibentuk, yaitu citra asli tabung uji silang serasi dan label segmentasi darah tabung uji silang serasi, akan dilakukan pemisahan secara rata berdasarkan label kategori pola reaksi dari citra tersebut. Hal ini dilakukan supaya model DL ini dapat mendeteksi darah pada pola reaksi berbagai macam, yaitu ketika darah terkumpul

hanya pada bagian bawah tabung saja, gumpalan darah kecil mulai melayang pada gel, ataupun sudah berkumpul pada bagian atas. *Crossmatch dataset* akan dibagi menjadi tiga bagian, yaitu latihan, validasi, dan tes dengan rasio 8:1:1 atau 80% latihan, 10% validasi, dan 10% tes. Hal ini umum dilakukan terutama pada pelatihan model pembelajaran mesin menggunakan teknik DL. *Dataset* latihan akan menjadi bagian utama untuk model berlatih, validasi digunakan untuk *fine-tuning* pada model, dan tes digunakan untuk sebagai penentu dalam evaluasi performa model. Setelah melakukan pembagian sesuai rasio, citra-citra tersebut akan dimasukkan dan dilakukan *resize* dengan dimensi 160x544 piksel untuk konsistensi pada *input layer*.

3.4.4 Augmentasi Citra

Augmentasi citra perlu dilakukan untuk mengatasi *overfitting* dan dapat membuat model pembelajaran mesin lebih *robust* terhadap masukan yang tidak bisa dipastikan. Augmentasi dilakukan pada *dataset* latihan dan validasi dengan tiga macam augmentasi, yaitu kontras, kecerahan, dan *noise*. Untuk *dataset* latihan dan validasi, augmentasi dilakukan dengan mengaplikasikan semua jenis augmentasi secara bersamaan dan menghasilkan dua citra teraugmentasi tiap citra pada *dataset* latihan dan validasi, sedangkan pada *dataset* tes tidak dilakukan augmentasi untuk menentukan hasil performa model dengan data riil. Citra yang sudah teraugmentasi ini akan menjadi penentu *robustness* pada model yang sudah dibuat. Hasil dari augmentasi yang sudah dilakukan pada penilitian akan dijelaskan pada Subbab 3.5.3.

3.4.5 Pembuatan Model

Penelitian ini menggunakan metode *pretrained weights* pada semua *encoder* yang digunakan dalam skenario model. Keputusan menggunakan *pretrained weights* adalah keterbatasan *dataset* yang ada dan waktu untuk melakukan pelatihan pada model segmentasi. Skenario model terdiri dari penggunaan berbagai *encoder*, *decoder* dan penggunaan *learning rate* yang berbeda pada *optimizer* Adam. *Encoder* yang digunakan pada penelitian ini adalah ResNet34, ResNext50_32x4d, dan EfficientNet-B5. *Decoder* yang digunakan adalah FPN, PAN, dan DeepLabV3+. Selanjutnya adalah skenario ketiga, yaitu perbandingan pelatihan augmentasi dan tanpa augmentasi. Model yang digunakan adalah model terbaik dari hasil skenario pertama dan kedua. Semua kombinasi dari skenario ini memiliki total lima belas model.

Setiap model yang digunakan dalam skenario pengujian penelitian ini berdasarkan CNN sehingga terdapat beberapa komponen dasar CNN yang sama digunakan dalam empat belas model. Komponen-komponen umum CNN seperti *convolutional layer*, *max pooling layer*, dan lain-lain dipakai pada seluruh model, namun beberapa modul yang ada hanya pada *encoder* atau *decoder* sendiri. Penjelasan komponen-komponen CNN yang digunakan pada penelitian ini dijelaskan pada Tabel 3.2.

Model yang dipakai pada penelitian ini berdasarkan implementasi dari penelitian-penelitian lain, terutama pada kasus *semantic segmentation* pada citra medis. Beberapa alasan tersebut adalah model-model yang dipilih memiliki kompleksitas yang mendalam tanpa mengorbankan komponen penting seperti waktu pendekripsi. Selain itu, *pretrained weights* yang digunakan berasal pada dataset besar, yaitu ImageNet pada semua *encoder* yang dipakai. Keuntungan memakai *pretrained weights* adalah skenario yang sudah dilatih cukup banyak dan tidak memerlukan waktu lama pada pelatihan model yang akan digunakan pada tugas ini. Berikut adalah penjelasan masing-masing *state of the art* model yang digunakan pada penelitian ini :

1. **ResNet34** : ResNet34 memiliki 34 *layer* yang digunakan untuk melakukan ekstraksi

fitur dengan teknik *residual blocks* yang digunakan untuk menyimpan informasi pola spasial yang ada. Penelitian berjudul *OCT layer segmentation using U-NET semantic segmentation and RESNET34 encoder-decoder* menjadi contoh ResNet34 sebagai *encoder* (Yojana & Thillai Rani, 2023).

2. **ResNeXt50** : ResNeXt50 adalah perkembangan *residual network* (ResNet) dengan penambahan kardinalitas untuk meningkatkan kemampuan model tanpa harus menambah *layer*. ResNeXt50 digunakan sebagai *encoder* pada penelitian berjudul *Multi-modal semantic image segmentation* dan berhasil mendapatkan hasil cukup baik dibandingkan dengan ResNet50 (Pemasiri et al., 2021) (Hitawala, 2018).
3. **EfficientNet-B5** : EfficientNet adalah pengembangan yang dilakukan oleh tim riset dari Google untuk mencapai akurasi dengan efisiensi yang tinggi. EfficientNet merupakan pengembangan dari MobileNetV2. Hal ini dapat dicapai dengan metode *scaling* pada *network* secara *compounding*. Faktor *scaling* disini ditunjukkan pada “B” dengan 0 paling kecil dan 7 paling besar. Penelitian dengan judul *EfficientNet family U-Net models for deep learning semantic segmentation of kidney tumors on CT images* menggunakan EfficientNet dari B0 hingga B7 dan pada penelitian ini menggunakan B5 karena keterbatasan *resource* (Abdelrahman & Viriri, 2023) (Tan &

Tabel 3.2 Deskripsi *Layer* dalam Arsitektur CNN Segmentasi

Le, 2020).

Layer	Deskripsi
<i>Input</i>	Lapisan awal pada model CNN sebagai tempat awal data masuk dan akan diproses pada seluruh <i>layer</i>
<i>Convolutional 2D</i>	Lapisan konvolusi dua dimensi yang bertugas menjalankan konvolusi dengan <i>sliding window</i> (<i>kernel</i> atau <i>filter</i>) dan menghasilkan data dengan pola spasial dari operasi konvolusi
<i>Batch Normalization</i>	Lapisan yang bertugas untuk normalisasi dari masukan data pada setiap lapisan model. Lapisan ini membantu model untuk stabilisasi dan akselerasi pada pelatihan
<i>Max Pooling</i>	Digunakan untuk mengurangi dimensionalitas spasial dengan menyimpan nilai informasi paling signifikan dan mengabaikan informasi kurang penting. Teknik ini digunakan untuk mencegah <i>overfitting</i> .
<i>ReLU</i>	Fungsi matematis non linear yang digunakan untuk mempelajari pola kompleks
<i>Group Normalization</i>	Teknik yang digunakan pada <i>neural networks</i> untuk normalisasi <i>activation</i>
<i>Merge Block</i>	Komponen pada jaringan yang digunakan untuk mengkombinasikan informasi dari berbagai cabang. Komponen ini ditemukan pada ResNet, U-Net, dan lain-lain.
<i>Dropout</i>	Lapisan dengan menggunakan teknik <i>regularization</i> untuk mencegah <i>overfitting</i> dengan cara menonaktifkan beberapa unit secara acak sesuai dengan probabilitas yang sudah ditentukan
<i>Upsampling</i>	Teknik yang digunakan untuk menambahkan dimensi spasial dari masukan

3.4.6 Model Fitting

Setelah melakukan pemilihan arsitektur model dengan *encoder* dan *decoder* yang sudah ditentukan pada skenario uji coba, tahap selanjutnya adalah *model compiling* dengan

memilih *optimizer*, *loss function*, dan *metrics*. Setelah melakukan kompilasi hal-hal tersebut, *fitting* data pada model dapat dilakukan. Dalam proses ini, *hyperparameter tuning* juga dilakukan untuk mendapatkan hasil terbaik. Skenario yang ditentukan pada *hyperparameter tuning* ini adalah *learning rate* dan *batch size* pada *dataset*. Citra akan masuk dalam arsitektur *fully convolutional network* sesuai dengan skenario yang sudah disebutkan dan akan menghasilkan klasifikasi setiap pixel dari *upsampling* ekstraksi fitur yang dilakukan. *Optimizer* pada model digunakan untuk mengganti *weight* pada model berdasarkan hasil skor dari *loss function* yang digunakan. Pembaharuan *weight* ini dapat dilakukan secara signifikan maupun tidak sesuai dengan nilai *learning rate* yang sudah ditentukan. Proses pembaharuan *weight* ini dinamakan *backward propagation*. Setiap model yang digunakan akan melakukan sebanyak lima puluh *epoch*. Penjelasan dari *hyperparameter* yang digunakan pada penelitian ini ada pada Tabel 3.3 (Liu et al., 2018).

Tabel 3.3 Deskripsi <i>Hyperparameter</i> yang Digunakan (Bagian 1)	
<i>Loss Function</i>	<p>[Dice Loss, Jaccard Loss]</p> <ul style="list-style-type: none"> • Dice Loss adalah <i>loss function</i> yang menggunakan Dice Coefficient atau indeks Sørensen dengan cara menghitung kemiripan dari dua set. Persamaan dapat dilihat pada Persamaan 2.4. • Jaccard Loss adalah <i>loss function</i> yang menggunakan indeks Jaccard. Indeks Jaccard juga menghitung kemiripan dan diversitas dari dua set. Persamaan dapat dilihat pada Persamaan 2.2.
<i>Optimizer</i>	<p>[Adam]</p> <ul style="list-style-type: none"> • Adam (Adaptive Moment Estimation) adalah kombinasi algoritma dari Root Mean Square Propagation Algorithm (RMSProp) dan Stochastic Gradient Descent with Momentum (SGDM). Adam menggunakan <i>moving average</i> secara eksponensial untuk estimasi momen setiap <i>batch</i> dalam setiap iterasi. Formula dari Adam dapat dilihat pada Persamaan 3.1 dan 3.2 dimana m dan v adalah <i>moving average</i>, g adalah gradien pada <i>batch</i>, t adalah jumlah iterasi, dan β_1 dan β_2 adalah <i>hyperparameter</i> dari algoritma (SEN & OZKURT, 2020). $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.1)$ $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3.2)$
<i>Learning Rate</i>	<p>[0.001, 0.0001 (<i>Default</i>)]</p> <ul style="list-style-type: none"> • Learning Rate adalah nilai yang menentukan seberapa besar pembaharuan <i>weight</i> pada model. Nilai ini sangat berpengaruh pada konvergensi dari model sehingga nilai tidak bisa terlalu besar karena akan menyebabkan <i>divergence</i> dan tidak bisa terlalu kecil karena akan membutuhkan waktu yang lama untuk mencapai konvergensi.

Hyperparameter	Jenis dan Deskripsi
<i>Batch Size</i>	[8, 16, 32] <ul style="list-style-type: none"> • Batch Size adalah jumlah citra yang digunakan secara bersamaan dalam satu <i>batch</i> pada pelatihan model untuk mendapatkan stabilitas pada model dan efisiensi memori.

Tabel 3.3 Deskripsi *Hyperparameter* yang Digunakan (Bagian 2)

Hyperparameter	Jenis dan Deskripsi
<i>Activation Function</i>	[ReLU] <ul style="list-style-type: none"> • ReLU adalah fungsi matematis non linear yang digunakan untuk mempelajari pola kompleks.
<i>Epoch</i>	[50] <ul style="list-style-type: none"> • Epoch adalah iterasi atau perulangan yang dilakukan model untuk melakukan pembelajaran dari <i>dataset</i> citra yang sudah ada. Penelitian ini menggunakan lima puluh <i>epoch</i>. <i>Epoch</i> sendiri menentukan seberapa jauh model akan dilakukan pembelajaran, namun terlalu banyak juga kurang efektif.
<i>Dropout</i>	[0.2 (<i>Default</i>)] <ul style="list-style-type: none"> • Dropout adalah teknik yang digunakan pada <i>deep learning</i> dengan cara menonaktifkan beberapa unit pada lapisan selama pelatihan untuk mencegah <i>overfitting</i>. Nilai yang dipakai pada penelitian ini adalah 0.2, dengan arti sebanyak 20% bagian dari lapisan dinonaktifkan.

3

3.1

3.2

3.3

3.4

3.4.1

3.4.2

3.4.3

3.4.4

3.4.5

3.4.6

3.4.7 Evaluasi Performa

Setelah proses pelatihan model dijalankan, evaluasi performa setiap model yang sudah ditentukan perlu dilakukan untuk memilih model terbaik. Metrik atau pengukur performa dari model tersebut pada penelitian ini adalah skor IoU, skor F1, akurasi, presisi, *recall*, spesifitas, dan NPV. Pengujian ini dilakukan pada *dataset* tes yang sudah dibagi sesuai dengan proporsi label. Setelah melakukan evaluasi terhadap model terbaik pada *dataset* normal, pelatihan ulang dengan *encoder* dan *decoder* terbaik dilakukan dengan *dataset* normal dan *dataset* teraugmentasi dengan augmentasi *brightness*, kontras, dan *noise*. Hasil

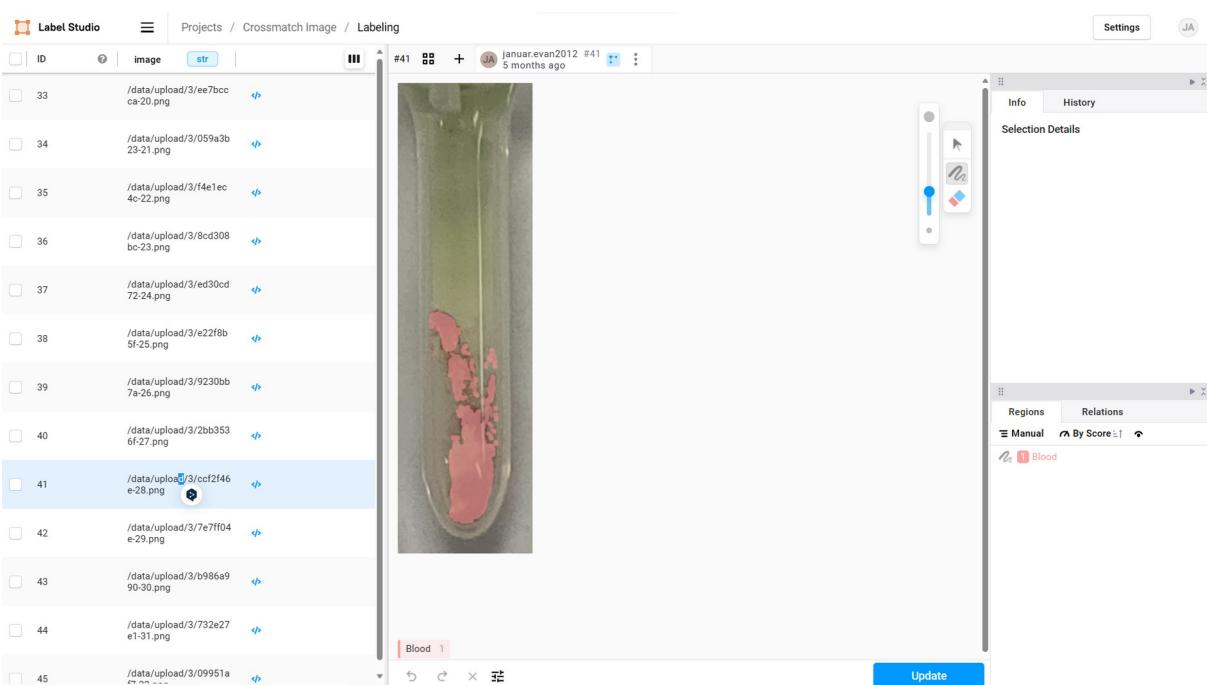
pelatihan dengan *dataset* teraugmentasi akan dibandingkan dengan model tanpa pelatihan dengan *dataset* augmentasi. Pengujian evaluasi dilakukan secara individu (masing-masing citra) jika memungkinkan dan dalam bentuk *batch* jika tidak memungkinkan karena terdapat lapisan *batch normalization*.

3.5 Implementasi

Subbab ini akan menjelaskan implementasi dari setiap tahapan yang dilakukan dalam penelitian ini. Setiap kode yang tertera pada sub bab ini merupakan *pseudocode*. Hasil dari implementasi juga akan dijelaskan pada bagian ini.

3.5.1 Implementasi Data Preprocessing

Proses pertama dalam *Data Preprocessing* adalah melakukan *cropping* untuk mendapatkan ROI, yaitu tabung pada kartu gel hasil uji silang serasi. Proses ini menggunakan aplikasi Adobe Photoshop 2021. Proses awal adalah memasukkan citra awal kartu gel hasil uji silang serasi dan dilakukan *cropping* dengan ukuran citra 155x541 piksel dan disimpan dalam satu folder dengan nama file sesuai dengan urutan pembuatan. Terdapat beberapa tabung yang ada pada citra kartu gel tidak digunakan karena tabung tersebut belum dipakai atau darah yang sudah ada dalam tabung tersebut sudah kering sehingga terdapat beberapa tabung yang ada di kartu gel uji silang serasi tidak digunakan.



Gambar 3.5 *User Interface* dari Label Studio

Setelah semua citra terkumpul dalam satu folder, proses selanjutnya adalah anotasi. Proses ini menggunakan aplikasi Label Studio. Proses awal pada tahap ini adalah mengunggah semua citra yang digunakan ke dalam *project* yang sudah dibuat. Setelah semua citra sudah terunggah, proses selanjutnya adalah anotasi segmentasi darah. Terdapat tombol yang dapat digunakan untuk memilih *brush* untuk anotasi dan *eraser* untuk menghapus anotasi. Selain tombol tersebut, terdapat *slider* untuk mengatur besar dari *brush* atau *eraser*.

Anotasi dilakukan dengan cara mengarahkan kursor dan klik kiri. Terdapat panel *regions* yang dapat membantu dalam anotasi *multilabel*. Setelah memastikan semua darah terlabel dengan benar, tombol *Update* perlu ditekan untuk memperbarui hasil segmentasi darah terbaru. Tampilan aplikasi Label Studio dapat terlihat pada Gambar 3.5.

Setelah melakukan anotasi pada seluruh citra, langkah selanjutnya adalah melakukan *export* ke dalam format JSON. File tersebut memiliki informasi lokasi dan nama citra sesuai dengan urutan karena terjadi pengacakan citra setelah pengunggahan ke Label Studio. Implementasi proses pemindahan citra berdasarkan informasi pada file JSON ke dalam folder label tertera pada Kode Semu 3.1.

```

1 import json
2 import os
3 import shutil
4 png_dir ← '/label/'
5 img_dir ← '/label-studio/'
6 task = json_load('export.json')
7 for task in tasks do
8     copy_image(png_dir + task['data']['image_key'], img_dir + task['id'] +
'.jpg')
9 end for

```

Kode Semu 3.1 Pemindahan Citra sesuai JSON dari Label Studio

Setelah melakukan anotasi segmentasi pada darah, proses selanjutnya adalah melakukan *labelling* pola reaksi pada setiap citra tabung hasil uji silang serasi. Proses ini melibatkan ahli bidang darah dengan menanyakan kategori pola reaksi pada setiap citra tabung hasil uji silang serasi. Pada awalnya, semua hasil kategori masuk ke dalam file dengan format TXT. Dalam file tersebut, setiap baris adalah pengkategorian citra sesuai dengan nama file citra. Proses selanjutnya adalah membuat file JSON yang berisi nama file dan kategori pola reaksi. Implementasi proses tersebut tertera pada Kode Semu 3.2.

```

1 import json
2
3 file ← 'label.txt'
4 all ← []
5 counter ← 1
6 for i in open(file, 'r') do
7     i = replace(i, '\n', '')
8     all.append({counter + '.jpg' : i})
9     counter += 1
10.end for
11.with open('label.json', 'w') as outfile
12    json.dump(all, outfile)

```

Kode Semu 3.2 Pembuatan JSON Citra dengan Kelas Pola Reaksi

3.5.2 Implementasi Pemisahan *Dataset* dengan Normalisasi

Implementasi proses ini dilakukan secara manual dengan cara menghitung total dari setiap kategori pola reaksi dan memiliki *counter* selama iterasi berjalan. Pembagian ini dilakukan dengan cara menyimpan namafile ke dalam sebuah *list* dan akan dilakukan *data loading* secara terpisah. *Data loading* pada penelitian ini menggunakan *library* opencv ke dalam bentuk numpy. Implementasi *dataset splitting* dapat dilihat pada Kode Semu 3.3.

```

1 import json
2
3 file ← '/label.json'
4 label_json ← load_json(file)
5 counter ← [0,0,0,0,0]

```

```

6. temp_counter ← [0,0,0,0,0]
7. train_namefile ← []
8. val_namefile ← []
9. test_namefile ← []
10.
11. for i in label_json do
12.     counter[i['label']] += 1
13. end for
14.
15. for i in label_json do
16.     temp_counter[i['label']] += 1
17.     if counter[i['label']] * 0.8 > temp_counter[i['label']] then
18.         append i['name'] to train_namefile
19.     else if counter[i['label']] * 0.9 < temp_counter[i['label']] then
20.         append i['name'] to test_namefile
21.     else
22.         append i['name'] to val_namefile
23.     end if
24. end for

```

Kode Semu 3.3 Pembagian Latihan, Validasi, dan Tes sesuai Rasio

Setelah semua citra sudah terpisah ke dalam bagian latihan, validasi, dan tes dengan rasio 8:1:1, semua data dimasukkan ke dalam DatasetLoader untuk memudahkan pelatihan dalam bentuk *batch*. Pengubahan ukuran dari citra juga dilakukan. Penjelasan mengenai implementasi *Dataset Splitting* dan *Loading*. Berikut adalah implementasi *data loading* dan memasukkan data ke dalam DatasetLoader (Kode Semu 3.4).

```

1. x_train ← []
2. y_train ← []
3. x_val ← []
4. y_val ← []
5. x_test ← []
6. y_test ← []
7.
8. for i in train_namefile do
9.     append loaded image of i to x_train
10.    append loaded label of i to y_train
11. end for
12.
13. for i in val_namefile do
14.     append loaded image of i to x_val
15.     append loaded image of i to y_val
16. end for
17.
18. for i in test_namefile do
19.     append loaded image of i to x_test
20.     append loaded image of i to y_test
21. end for
22.
23.
24. train_dataset ← create dataset from x_train with label of y_train
25. val_dataset ← create dataset from x_val with label of y_val
26. test_dataset ← create dataset from x_test with label of y_test
27. train_dataloader ← create data loader from train dataset
28. val_dataloader ← create data loader from val dataset
29. test_dataloader ← create data loader from test dataset

```

Kode Semu 3.4 Loading Citra sesuai Pembagian

3.5.3 Implementasi Augmentasi Citra

Proses implementasi *Image Augmentation* dilakukan setelah pemisahan *dataset*. Proses ini dibantu dengan library imgaug. Pada proses ini, citra pada *dataset* latihan dan validasi dilakukan tiga jenis augmentasi, yaitu *brightness*, *contrast*, dan *noise*. Terdapat masing-masing satu citra untuk setiap jenis augmentasi dan tiga citra untuk pengaplikasian tiga jenis augmentasi tersebut secara bersamaan sehingga total terdapat lima citra augmentasi dari citra asli. Hasil yang didapatkan adalah 2840 citra untuk latihan dan 385 citra untuk validasi. Baris enam hingga lima belas adalah implementasi augmentasi kontras. Untuk implementasi augmentasi yang lain juga mirip dengan baris enam hingga lima belas, namun hanya diubah menjadi jenis augmentasi yang sudah ditentukan.

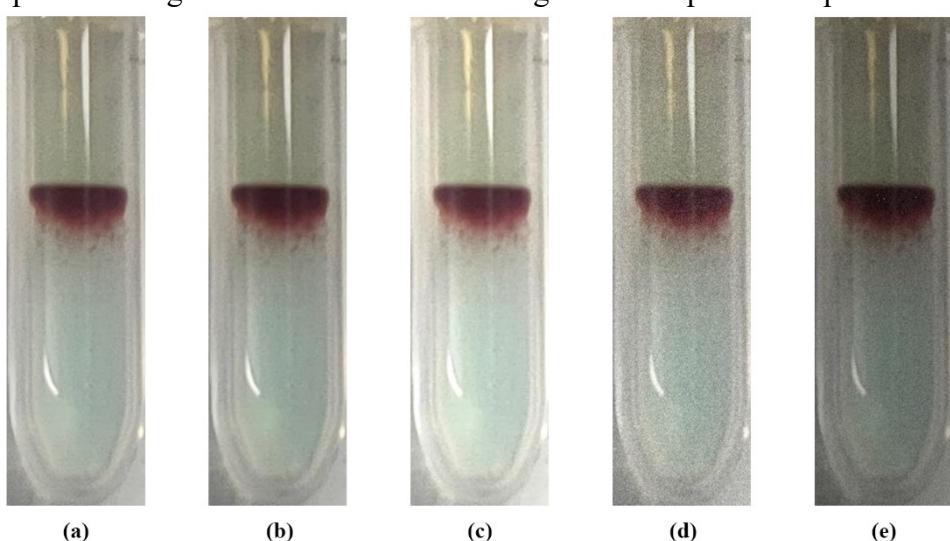
```

1. all_augmented ← []
2. counter ← 1
3. for i in range(len(x_test)) do
4.     class ← label(i + '.jpg')
5.     temp_image ← x_test[i]
6.     for j = 1 to 1 do
7.         contrast_value ← random(0.5, 1.5)
8.         apply_aug ← set apply_aug value to apply_contrast with value of
    contrast_value
9.         aug_img ← apply_aug to temp_image
10.        aug_label ← y_test[i]
11.        save aug_img to img_path
12.        save aug_label to label_path
13.        append {counter+'.jpg', contrast_value} to all_augmented
14.        counter += 1
15.    end for

```

Kode Semu 3.5 Implementasi Augmentasi Gambar

Pengaplikasian augmentasi citra dilakukan dengan sesuai dengan urutan *brightness* terlebih dahulu, lalu *contrast*, lalu *noise*. Hal ini dilakukan untuk mencapai penggambaran augmentasi yang realistik. Nilai pengaplikasian augmentasi *brightness* bernilai rentang 0.5 hingga 1.15. Nilai pengaplikasian augmentasi *contrast* bernilai rentang 0.5 hingga 1.5. Nilai pengaplikasian augmentasi *noise* bernilai rentang 0.02 hingga 0.06. *Pseudocode* untuk implementasi *image augmentation* dapat dilihat pada Kode Semu 3.5. Setiap nilai pengaplikasian juga disimpan dalam bentuk file JSON untuk kemudahan akses dalam melihat data pengaplikasian augmentasi citra. Hasil citra augmentasi dapat dilihat pada Gambar 3.6.



Gambar 3.6 Hasil Augmentasi dari Gambar Normal (a) dengan Kontras (b), Kecerahan (c), *Noise* (d), dan Kombinasi Ketiga Jenis Augmentasi (e)

**3.5.4
I**

implementasi Pembuatan Model

Proses ini melakukan pemilihan *encoder* dan *decoder* yang sudah ditentukan. Proses ini menggunakan bantuan library segmentation_models.pytorch. Dengan kombinasi tiga *encoder*, tiga *decoder*, dan *hyperparameter tuning*, model yang sudah didapat dan dilakukan pelatihan berjumlah empat belas model. Metode pelatihan model juga menggunakan *transfer learning* untuk melatih model dengan cepat dan keterbatasan data latih yang ada. Tahapan awal pada proses ini adalah deklarasi model. Deklarasi setiap *encoder* dan *decoder* relatif sama dengan perbedaan deklarasi kelas yang berbeda dan nilai *encoder* yang ingin dilatih. Setelah deklarasi *encoder* dan *decoder* pada model, deklarasi *optimizer*, *learning rate*, *epoch*, *optimizer*, *loss function* dan *batch size* pada DataLoader.

```
1. model ← FPN with ResNet34
2. optimizer ← set to Adam with Learning Rate of 0.0001
3. criterion1 ← set to DiceLoss
4. criterion2 ← set to JaccardLoss
5. num_epochs ← 50
6. if cuda is available then
7.     device = set to 'cuda'
8. else
9.     device = set to 'cpu'
10. end if
```

Kode Semu 3.6 Pendefinisian Model

Pada penelitian ini, implementasi model tidak perlu deklarasi ukuran masukan dan hanya dipastikan bahwa ukuran lebar dan tinggi berkelipatan 32. Selain itu, deklarasi *function loss* dipastikan menggunakan mode biner karena penelitian ini menggunakan segmentasi biner darah. Penggunaan GPU juga harus dideklarasikan untuk mempercepat pelatihan model segmentasi. Implementasi *model building* dapat dilihat pada Kode Semu 3.6.

3.5.5 Implementasi *Model Fitting*

Setelah melakukan *model building*, proses selanjutnya adalah *model fitting* untuk melakukan pelatihan model pada data latih yang sudah ditentukan. Pada tahap sebelumnya, nilai dari banyaknya *epoch* sudah ditentukan. Pada penelitian ini, nilai dua *loss function* yang digunakan bernilai sama, sehingga masing-masing *loss function* akan dibagi 0.5. Setiap metrik yang sudah ditentukan akan dijadikan satu dan akan dilakukan rata-rata setiap akhir iterasi. Hasil dari nilai semua metrik dan *loss function* akan disimpan. Perbedaan pada bagian validasi dan latihan terletak pada proses *backpropagation*. *Pseudocode* implementasi *model fitting* dapat dilihat pada Kode Semu 3.7.

```
1. for j = p to num_epochs do
2.     train_metrics ← []
3.     for batch in train_data_loader do
4.         input ← batch['input']
5.         mask ← batch['mask']
6.         set optimizer to zero gradient
7.         output ← model(input)
8.         loss1 ← criterion1(output,mask)
9.         loss2 ← criterion2(output,mask)
10.        train_loss = (0.5 * loss1) + (0.5 * loss2)
11.        calculate metrics
12.        append metric_result to train_metrics
13.        compute gradient from loss
14.        backpropagation using optimizer
15.    end for
```

```

16.    set model to eval
17.    val_metrics ← []
18.    disable gradient_computation
19.    for batch in val_data_loader do
20.        input ← batch['input']
21.        mask ← batch['mask']
22.        output ← model(input)
23.        loss1 ← criterion1(output, mask)
24.        loss2 ← criterion2(output, mask)
25.        val_loss = (0.5 * loss1) + (0.5 * loss2)
26.        calculate metrics
27.        append metric_result to val_metrics
28.    end for
29.    Save avg(train_metrics), avg(val_metrics), avg(train_loss), and
      avg(val_loss)
30.end for

```

Kode Semu 3.7 Iterasi pada Pelatihan Model

3.5.6 Implementasi Evaluasi Performa Model

Terdapat dua proses pada tahapan evaluasi performa ini. Proses pertama adalah melakukan evaluasi selama proses pelatihan. Proses ini terletak pada iterasi pelatihan yang tertera pada Kode Semu 3.7. Proses ini mengambil rata-rata dari hasil penghitungan setiap *batch* dalam satu iterasi. Proses ini menggunakan library Torch untuk menyimpan setiap metrik dan nilai *loss function*. Setiap metrik dan *loss function* disimpan secara terstruktur dengan nama model beserta keterangan *hyperparameter* yang dicoba. Implementasi evaluasi performa pada tahap validasi hampir sama dengan Kode Semu 3.7 dengan penggantian angka dari *train* menjadi *validation*. Kode Semu 3.8 merupakan penjabaran dari baris 11 dan baris 27 pada Kode Semu 3.7 dengan isi penghitungan metrik yang sudah ditentukan pada awal. Proses pengambilan metrik yang digunakan

```

1. tp, fp, fn, tn = get_stats(output, mask)
2. append iou_score(tp,fp,fn,tn) to train_iou_score
3. append f1_score(tp,fp,fn,tn) to train_f1_score
4. append accuracy(tp,fp,fn,tn) to train_acc
5. append precision(tp,fp,fn,tn) to train_precision
6. append recall(tp,fp,fn,tn) to train_recall
7. append specificity(tp,fp,fn,tn) to train_specificity
8. append negative_predictive_value(tp,fp,fn,tn) to train_npv

```

Kode Semu 3.8 Kalkulasi Tiap Metrik

Proses kedua adalah melakukan evaluasi berdasarkan *dataset* pengujian normal. Pada proses ini, model dicoba untuk menggunakan evaluasi pada individu citra. Semua hasil metrik dengan tambahan *inference time* dijadikan satu dalam satu pandas DataFrame. Beberapa model tidak bisa melakukan evaluasi hanya menggunakan satu citra sehingga harus menggunakan proses dalam tiap *batch*. Hal ini berakibat pada ketidakakuratan pada hasil pada *inference time* karena hasil dari *inference time* pada setiap *batch* adalah hasil rata-rata dari banyaknya citra pada *batch* tersebut.

```

1. augmented_data_loader = create data loader from test_data_loader
2. for batch in test_data_loader do
3.     image = batch['image']
4.     mask = batch['label']
5.     with torch.no_grad() do
6.         start_time = calculate current time
7.         output = model(image)
8.         end_time = calculate current time

```

```

9.         inference_time = end_time - start_time
10.    end
11.    for j = 0 to len(image)-1 do
12.        loss1 = criterion1(output[j], mask[j])
13.        loss2 = criterion2(output[j], mask[j])
14.        tp, fp, fn, tn = metrics.get_stats(output, mask)
15.        iou_score = iou_score(tp,fp,fn,tn)
16.        f1_score = f1_score(tp,fp,fn,tn)
17.        accuracy = accuracy(tp,fp,fn,tn)
18.        precision = precision(tp,fp,fn,tn)
19.        recall = recall(tp,fp,fn,tn)
20.        specificity = specificity(tp,fp,fn,tn)
21.        npv = negative_predictive_value(tp,fp,fn,tn)
22.        save iou_score, f1_score, accuracy, precision, recall specificity
     npv, inference_time
23.    endfor
24. Endfor

```

Kode Semu 3.9 Pengujian Model dengan *Dataset Tes*

BAB IV HASIL DAN PEMBAHASAN

Bab ini akan membahas hasil dari skenario model yang ditentukan dan skenario uji coba untuk mendapatkan model dengan parameter yang terbaik berdasarkan uji coba yang dilakukan.

4.1 Skenario Uji Coba dan Analisis

Skenario uji coba yang dilakukan pada penelitian ini dilakukan dengan tahapan-tahapan sebagai berikut:

1. Perbandingan metode *encoder*
2. Perbandingan metode *decoder*
3. Perbandingan pelatihan dengan data augmentasi dan tanpa augmentasi

Seluruh model yang digunakan dipastikan menggunakan *weight* pada pelatihan menggunakan dataset ImageNet, menggunakan optimizer Adam, dan dilakukan pelatihan dan validasi menggunakan *dataset* normal tanpa augmentasi. Rangkuman model-model yang sudah dicoba tertera pada Tabel 4.1.

Tabel 4.1 Rangkuman *Encoder*, *Decoder*, dan *Hyperparameter* yang Digunakan

No	(Decoder)	Encoder	Learning Rate	Batch Size
1	FPN	ResNet34	0.0001	8
2			0.0001	32
3			0.001	32
4		ResNeXt50	0.0001	8
5			0.0001	16
6			0.001	16
7		EfficientNet-B5	0.0001	8
8			0.001	8
9	DeepLabV3+	ResNet34	0.001	32
10		ResNeXt50	0.001	16
11		EfficientNet-B5	0.001	8
12	PAN	ResNet34	0.001	32
13		ResNeXt50	0.001	16
14		EfficientNet-B5	0.001	8

4

4.1

4.1.1 Skenario Perbandingan *Encoder*

Skenario perbandingan *encoder* ini dilakukan untuk menentukan *decoder* dengan *hyperparameter* yang sama pada model segmentasi darah. Skenario ini diuji dengan *dataset* latih dan *dataset* validasi normal tanpa augmentasi yang digunakan. Berikut adalah tahapan dalam pelaksanaan skenario ini.

1. Penggunaan setiap *encoder* dengan garis dasar pada *hyperparameter* yang sama, yaitu optimizer Adam dengan *learning rate* 0.0001 dan *batch size* sebesar delapan.
2. Analisis data setelah melakukan iterasi *epoch* sebanyak lima puluh.

Untuk skenario ini, semua *decoder* dan *hyperparameter* yang digunakan disamakan untuk

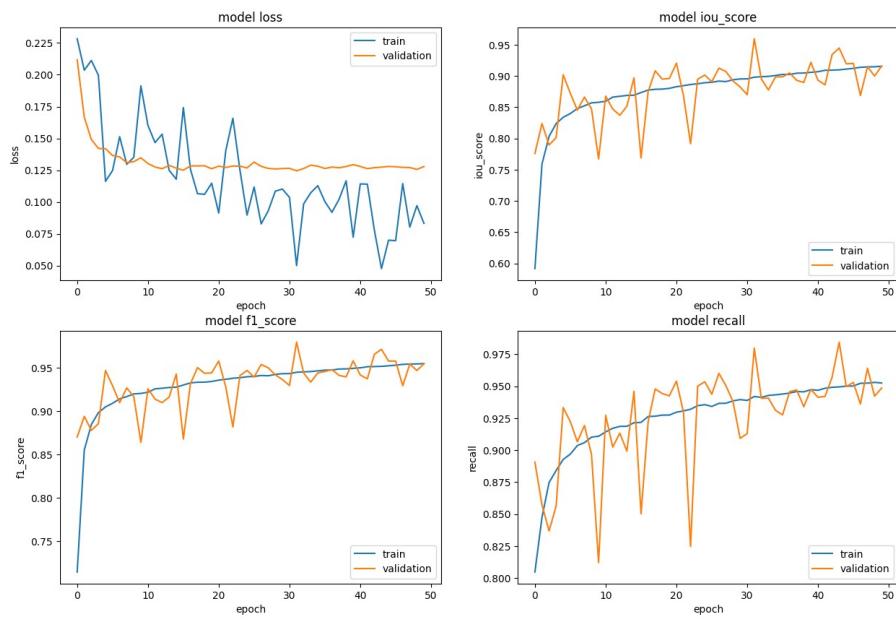
mengetahui performa dari setiap *encoder* yang sudah ditentukan. Hasil pertama uji coba yang dilakukan adalah menggunakan *decoder* FPN dengan *hyperparameter optimizer* Adam dengan *learning rate* 0.0001.

Hasil yang didapatkan pada skenario ini tertera pada Tabel 4.2. Hasil model dengan *decoder* dan *hyperparameter* mendapatkan hasil pengujian terbaik dengan *encoder* ResNet34. Hasil ini dapat dilihat berdasarkan metrik skor IoU yang didapatkan pada model ResNet34. Selain itu, hasil skor F1, *recall*, dan akurasi mendapatkan hasil terbaik dibandingkan *encoder* ResNeXt50 dan EfficientNet-B5. Namun, hasil model terbaik pada skenario ini tidak mendapatkan hasil presisi yang terbaik, justru terendah meskipun nilai tidak berbeda secara signifikan. Hasil presisi yang lebih rendah ini masih dapat diterima. Hal ini merupakan hal yang cukup umum dilakukan pada citra medis. Citra medis lebih mengutamakan hasil dengan nilai *recall* yang lebih baik sehingga model dengan ResNet34 diambil sebagai model terbaik.

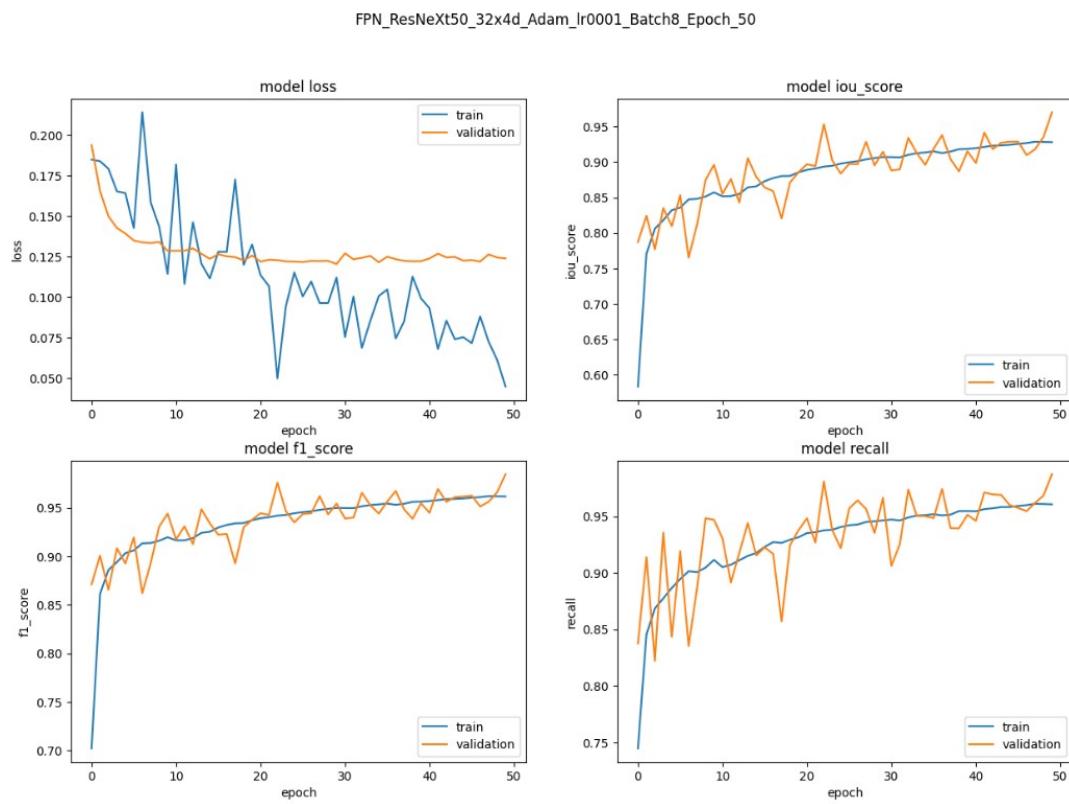
Tabel 4.2 Hasil Uji Coba Model pada Skenario Perbandingan <i>Encoder</i>						
Batch Size = 8 Optimizer = Adam LR = 0.0001	ResNet34	0.8507	0.9203	0.9184	0.9166	0.9791
	ResNeXt50	0.8427	0.9302	0.9000	0.9118	0.9778
	EfficientNet-B5	0.8407	0.9275	0.9003	0.9105	0.9774

Selain hasil metrik yang sudah dikumpulkan pada fase uji coba, hasil pelatihan dan validasi juga diliat pada setiap model yang digunakan. Hasil pelatihan dan validasi yang digunakan pada model ini dapat dilihat pada Gambar 4.1 (ResNet34), Gambar 4.2 (ResNeXt50), dan Gambar 4.3 (EfficientNet-B5). Hasil yang didapatkan adalah model dengan kestabilan paling tinggi terjadi pada model ResNeXt50. Hasil ini dapat dilihat pada dekatnya hasil validasi dan pelatihan pada metrik skor IoU, skor F1, dan *recall*. Hasil pada model ResNeXt50 mendapatkan nilai rata-rata *loss function* lebih rendah dibandingkan pada kedua model, yaitu sekitar 0.05, diikuti model ResNet34 dengan nilai sekitar 0.075 dan EfficientNet-B5 dengan nilai sekitar 0.09. Hasil *loss function* ini dapat menandakan bahwa model mengalami *overfitting* berlebihan. Ketiga model mengalami *overfitting* dengan model EfficientNet-B5 mengalami *overfitting* paling kecil dengan perbedaan nilai validasi dan pelatihan cukup rendah.

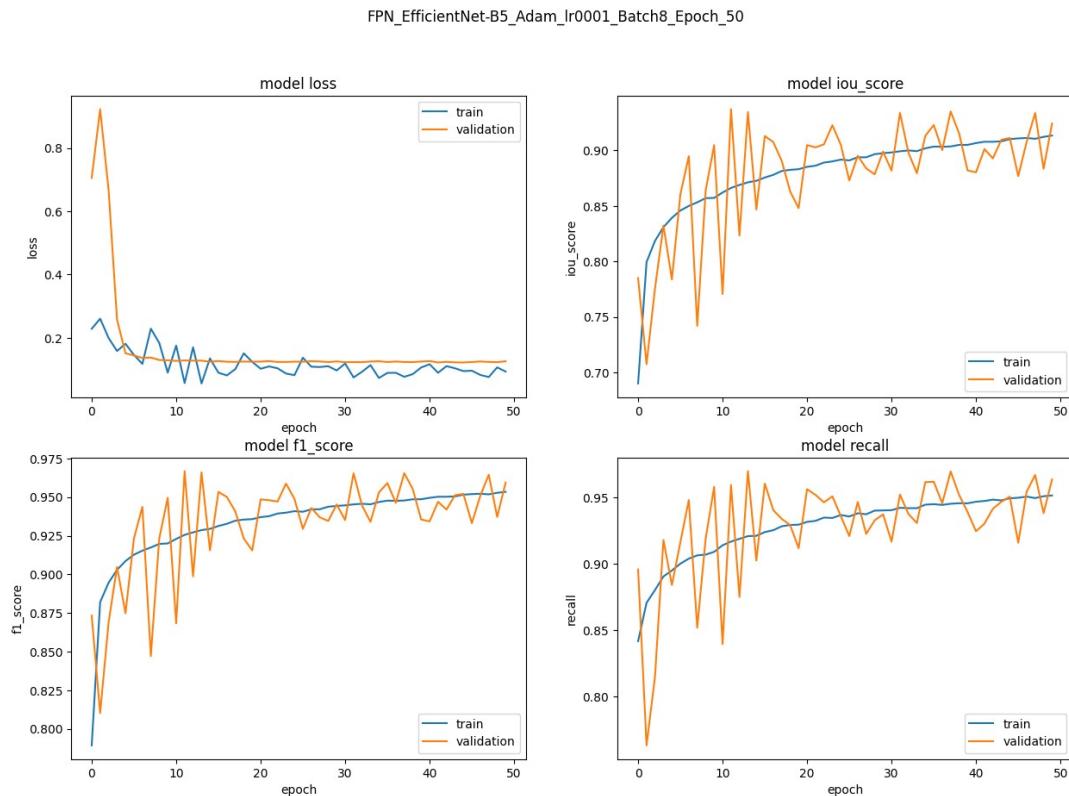
FPN_ResNet34_Adam_lr0001_Batch8_Epoch_50



Gambar 4.1 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr0001-Batch8

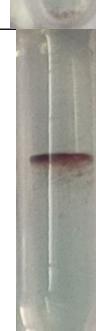


Gambar 4.2 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr0001-Batch8



Gambar 4.3 Grafik Pelatihan dan Validasi Model FPN-EfficientNet-B5-Adam-lr0001-Batch8

Tabel 4.3 Hasil Model Terbaik pada Skenario Perbandingan *Encoder*

Nama	Kelas	Citra	Ground Truth	Output
511.jpg	0			
609.jpg	1			
658.jpg	2			
691.jpg	3			
718.jpg	4			

Tabel 4.4 Metrik dari Citra dengan Skor IoU Terendah pada Skenario Perbandingan *Encoder*

511.jpg	0	0.7641	0.7645	0.9992	0.8662	0.9950
609.jpg	1	0.6193	0.9246	0.6523	0.7649	0.9353
658.jpg	2	0.7134	0.8241	0.8415	0.8327	0.9392
691.jpg	3	0.8391	0.9329	0.8928	0.9124	0.9716
718.jpg	4	0.6738	0.8403	0.7727	0.8051	0.9827

Setelah itu, tahap selanjutnya adalah melakukan pemeriksaan pada hasil prediksi model terbaik pada skenario ini. Pengecekan dilakukan pada citra dengan nilai skor IoU terendah pada masing-masing representasi kelas yang ada. Hasil yang didapatkan setelah pengecekan ini adalah setiap citra pada kelas yang memiliki nilai skor IoU terendah memiliki nilai dibawah rata-rata, namun hal tersebut tidak terjadi pada metrik lain seperti *recall*. Terdapat dua jenis kesalahan pada setiap citra dengan skor IoU terendah, yaitu model mengalami masalah pada presisi dengan nilai *recall* yang tinggi dan bermasalah pada *recall* dengan nilai presisi yang tinggi. Masalah pertama menyebabkan hasil prediksi model pada citra tidak mengambil seluruh bagian darah yang sudah terlabel positif (TP) sehingga model tidak menghasilkan citra segmentasi dengan detil yang cukup (FN). Masalah kedua adalah model mengalami *overshoot* dan mengambil bagian pada citra yang tidak termasuk darah (FP). Masalah kedua adalah masalah yang dapat ditoleransi karena model dapat mengambil sebagian besar bagian citra bernilai asli darah (TP) dengan detil yang tinggi sehingga tidak mengubah informasi dan dapat merepresentasikan kelas pola reaksi citra tersebut. Hasil dapat dilihat di Tabel 4.3 dan Tabel 4.4.

4.1.2 Skenario Perbandingan *Hyperparameter*

Skenario perbandingan *hyperparameter* ini dilakukan pada *decoder* yang sama, yaitu FPN. Skenario ini merupakan tahapan lanjutan dari skenario pertama untuk mendapatkan model dengan hasil yang terbaik setelah melakukan beberapa perubahan kecil pada *hyperparameter*. Perubahan yang dilakukan pada scenario ini adalah perubahan *hyperparameter* pada *batch size* semampu *resource* yang diberikan pada Google Colab. Selain itu, perubahan *hyperparameter* pada *learning rate* (LR) dari *optimizer* Adam yang digunakan.

Untuk hasil yang didapatkan akan dilakukan terpisah untuk setiap *encoder*. Berikut adalah hasil dari pengubahan *hyperparameter* :

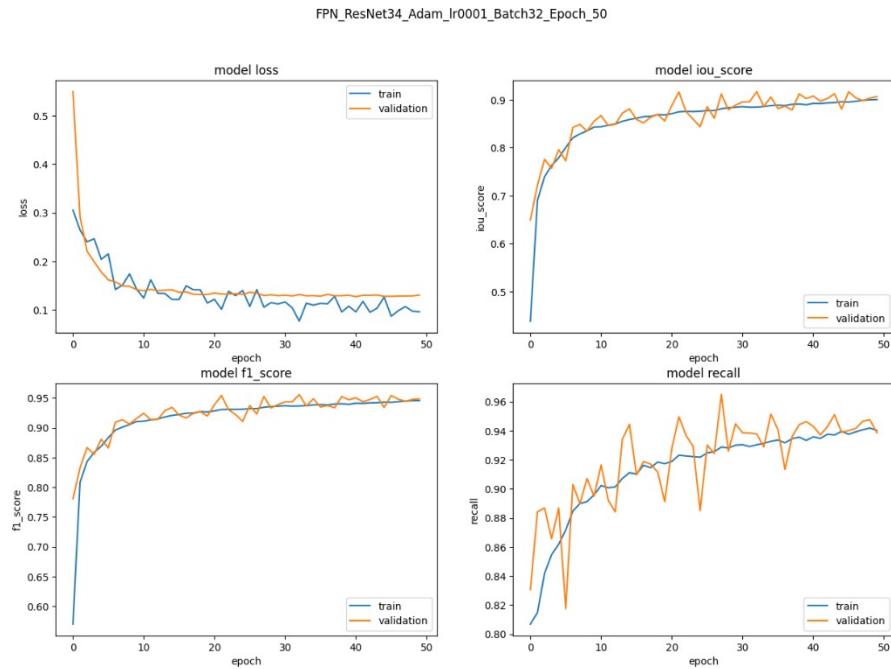
1. ResNet34

Setelah melakukan beberapa pengujian dengan mengganti *batch size* menjadi 32 saja dan kombinasi *batch size* 32 dan *learning rate* menjadi 0.001 adalah model dengan *batch size* 8 dan *learning rate* 0.0001 adalah model dengan hasil terbaik pada skor IoU, skor F1, Akurasi, dan *recall*. Untuk presisi, model terbaik kembali mendapatkan hasil presisi terendah. Model dengan presisi rendah masih bisa diterima asalkan nilai *recall* yang didapatkan tinggi. *Recall* menjadi prioritas karena hasil prediksi model mampu mengambil label positif (darah) lebih baik sehingga representasi pola reaksi dari hasil prediksi model masih berada pada kelas yang sama. Hasil uji coba model ini dapat dilihat pada Tabel 4.5.

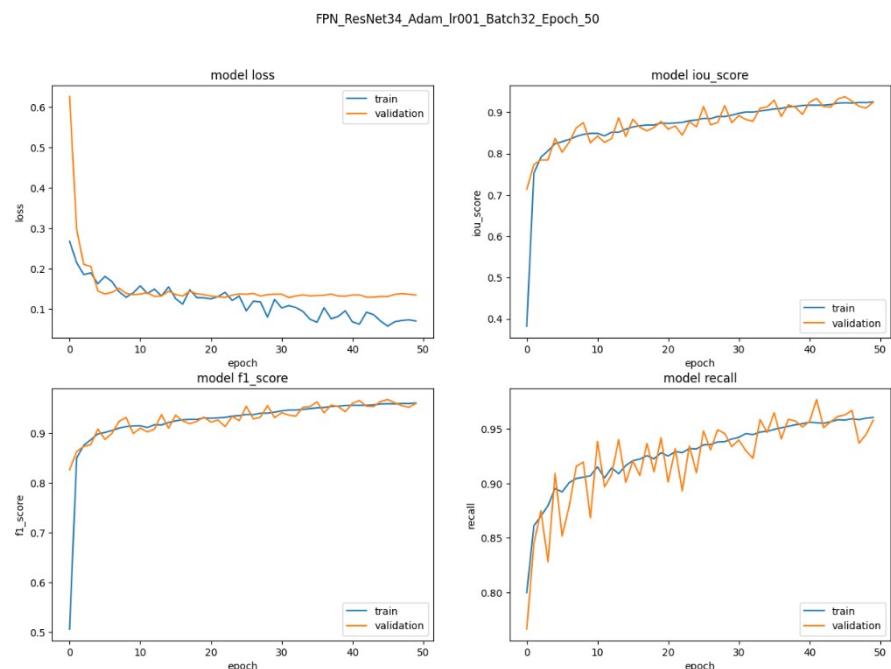
Tabel 4.5 Hasil Uji Coba Model ResNet34 pada Skenario Perbandingan *Hyperparameter*

<i>Batch Size = 8</i> LR = 0.0001	0.8507	0.9203	0.9184	0.9166	0.9791
<i>Batch Size = 32</i> LR = 0.0001	0.8169	0.9318	0.8714	0.8955	0.9742
<i>Batch Size = 32</i> LR = 0.001	0.8247	0.9332	0.8784	0.8989	0.9739

Selain itu, hasil pelatihan dan validasi yang didapatkan dibandingkan untuk melihat performa model selama pelatihan dan validasi terhadap perubahan *hyperparameter* yang digunakan. Hasil pelatihan dan validasi dari ketiga model ini dapat dilihat pada Gambar 4.1 (*batch size* 8 dan LR 0.0001), Gambar 4.4 (*batch size* 32 dan LR 0.0001), dan Gambar 4.5 (*batch size* 32 dan LR 0.001). Model dengan *batch size* 32 dan LR 0.001 mendapatkan hasil paling stabil dibandingkan kedua model lainnya. Hasil ini dapat dilihat pada grafik validasi dan pelatihan yang dihasilkan mengalami fluktuasi tidak signifikan. Hasil model dengan kestabilan tertinggi akan digunakan pada skenario ketiga.



Gambar 4.4 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr0001-Batch32



Gambar 4.5 Grafik Pelatihan dan Validasi Model FPN-ResNet34-Adam-lr001-Batch32

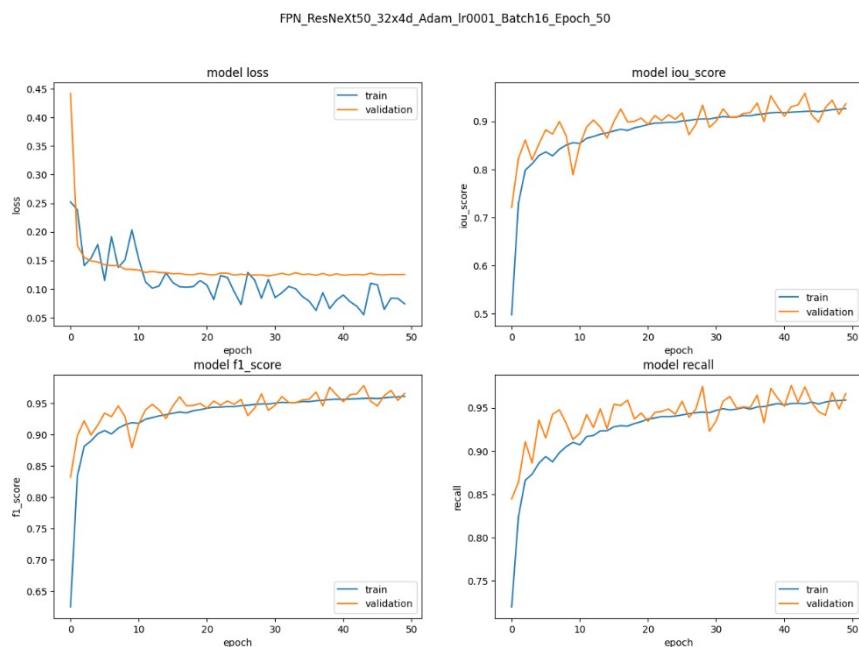
2. ResNeXt50

Beberapa pengujian dilakukan pada *encoder* ResNeXt50, yaitu hanya mengganti pada *batch size* menjadi 16 dan mengganti *batch size* 16 beserta *learning rate* menjadi 0.001. Hasil yang didapatkan dapat dilihat pada Tabel 4.6. Hasil yang didapatkan pada setiap metrik adalah model dengan *batch size* 16 dan *learning rate* 0.001 mendapatkan hasil yang terbaik dengan nilai skor IoU, skor F1, *recall*, dan akurasi terbaik. Hal serupa terjadi pada model sebelumnya, yaitu model terbaik yang diujicobakan mendapatkan hasil presisi lebih rendah dibanding model lain. Hasil model dengan *recall* terbaik tetap diprioritaskan karena kemampuan mendeteksi TP dengan baik direpresentasikan oleh nilai metrik *recall*.

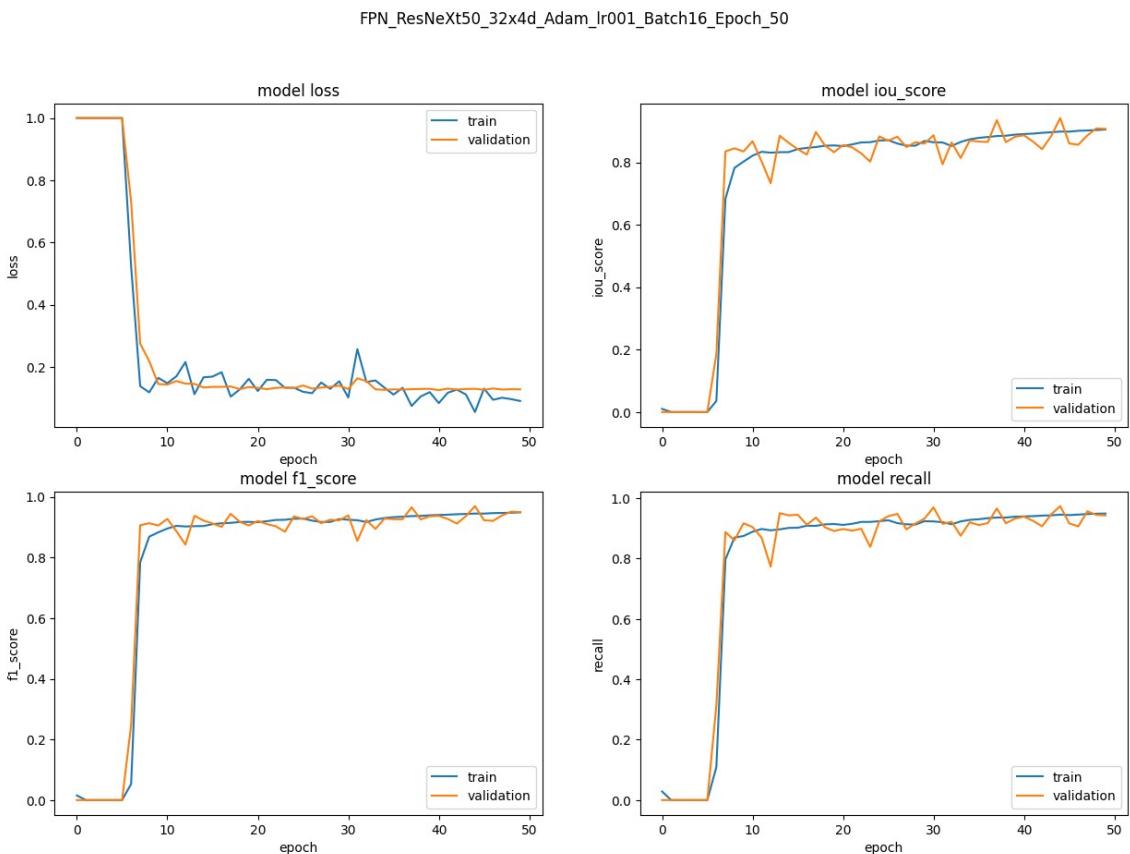
Tabel 4.6 Hasil Uji Coba Model ResNeXt50 pada Skenario Perbandingan *Hyperparameter*

<i>Batch Size</i> = 8 LR = 0.0001	0.8427	0.9302	0.9000	0.9118	0.9778
<i>Batch Size</i> = 16 LR = 0.0001	0.8269	0.9353	0.8771	0.9021	0.9756
<i>Batch Size</i> = 16 LR = 0.001	0.8444	0.9261	0.9045	0.9122	0.9778

Selain perbandingan hasil setiap metrik pada pengujian, tahap selanjutnya adalah menentukan model yang akan dipakai untuk skenario selanjutnya, yaitu perbandingan *decoder*. Model yang akan diambil adalah model dengan stabilitas terbaik untuk mencegah adanya *overfitting* yang terjadi pada model. Hasil pelatihan ketiga model tersebut dapat dilihat pada Gambar 4.2 (*batch size* 8 dan LR 0.0001), Gambar 4.6 (*batch size* 16 dan LR 0.0001), dan Gambar 4.7 (*batch size* 16 dan LR 0.001). Hasil yang didapatkan adalah model dengan *batch size* 16 dan LR 0.001 mendapatkan hasil lebih stabil. Namun, saat pelatihan, model mengalami stagnasi pembelajaran hingga *epoch* keenam, tetapi hasil yang didapatkan setelah *epoch* ketujuh terjadi perubahan yang signifikan dan menjadi lebih stabil dibandingkan ketiga model lainnya. Selain itu, model dengan *learning rate* 0.001 mendapatkan hasil konvergensi lebih cepat dibandingkan model dengan *learning rate* 0.0001. Hasil ini juga terjadi pada model ResNet34 dengan model mencapai konvergensi lebih cepat dari grafik yang dihasilkan.



Gambar 4.6 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr0001-Batch16



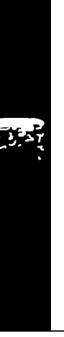
Gambar 4.7 Grafik Pelatihan dan Validasi Model FPN-ResNeXt50-Adam-lr001-Batch16

Tahap selanjutnya adalah melakukan pemeriksaan pada citra dengan skor IoU terendah pada tiap kelas pola reaksi pada model terbaik dari FPN ResNeXt50, yaitu model dengan *batch size* 16 dan *learning rate* 0.001. Masalah yang terjadi pada model ini juga serupa pada model terbaik FPN ResNet34 (*batch size* 8 dan *learning rate* 0.0001), yaitu permasalahan pada presisi dan *recall*. Contoh masalah model pada presisi adalah hasil prediksi model pada citra 519.jpg yang menunjukkan bahwa model mampu mendeteksi label positif (TP) dengan baik sehingga FN yang dihasilkan oleh prediksi model cukup rendah namun menghasilkan presisi yang cukup rendah. Masalah kedua, yaitu model tidak bisa memprediksi nilai asli darah dengan baik (TP) sehingga jumlah FN pada model cukup banyak dan tinggi. Masalah ini dapat dilihat pada hasil prediksi model pada citra 609.jpg dengan nilai *recall* relatif rendah. Hasil prediksi model pada citra 609.jpg gagal memprediksi bagian darah pada sisi kiri dan kanan yang berada di atas gumpalan darah yang ada pada bagian dasar tabung (gagal melihat bagian darah yang cukup kecil). Hasil ini dapat dilihat pada Tabel 4.7 dan Tabel 4.8.

Tabel 4.7 Metrik dari Citra dengan Skor IoU Terendah Model ResNeXt50 Skenario Perbandingan *Hyperparameter*

609.jpg	1	0.6165	0.8988	0.6624	0.7627	0.9336
658.jpg	2	0.6599	0.8096	0.7811	0.7951	0.9276
691.jpg	3	0.8233	0.9596	0.8528	0.9036	0.9697
718.jpg	4	0.6272	0.8314	0.7187	0.7709	0.9803

Tabel 4.8 Hasil Model ResNeXt50 Terbaik pada Skenario Perbandingan *Hyperparameter*

Nama	Kelas	Citra	Ground Truth	Output
519.jpg	0			
609.jpg	1			
658.jpg	2			
691.jpg	3			
718.jpg	4			

3. EfficientNet-B5

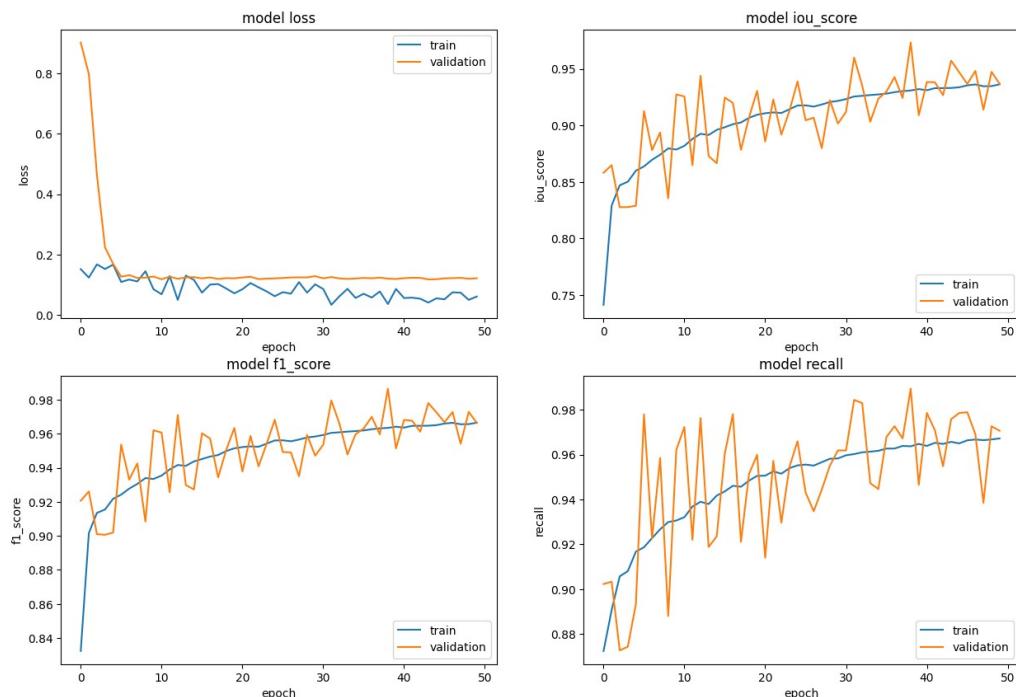
Pengujian yang dilakukan pada perbandingan *hyperparameter* untuk FPN dengan *encoder* EfficientNet-B5 hanya perubahan pada *learning rate* karena keterbatasan *resource* Video Random Access Memory (VRAM) yang tersedia pada Google Colab. Hasil yang didapatkan untuk pengujian pengubahan *learning rate* ini mendapatkan hasil terbaik pada model dengan *learning rate* lebih besar, yaitu 0.001. Hasil ini dapat dilihat pada metrik skor IoU, skor F1, akurasi, presisi, dan *recall*. Model ini tidak mendapatkan permasalahan dalam presisi yang terjadi pada FPN dengan *encoder* ResNet34 dan ResNeXt50 sebelumnya.

Tabel 4.9 Hasil Uji Coba Model EfficientNet-B5 pada Skenario Perbandingan *Hyperparameter*

<i>Batch Size</i> = 8 LR 0.0001	0.8407	0.9275	0.9003	0.9105	0.9774
<i>Batch Size</i> = 8 LR 0.001	0.8499	0.9339	0.9039	0.9159	0.9791

Tahap selanjutnya adalah melihat hasil pelatihan dan validasi dari setiap model. Untuk grafik hasil pelatihan dan validasi model dapat dilihat pada Gambar 4.3 (*batch size* 8 dan LR 0.0001) dan Gambar 4.8 (*batch size* 8 dan LR 0.001). Hasil model dengan stabilitas terbaik akan digunakan untuk percobaan pada skenario ketiga (perbandingan metode *decoder*). Hasil yang didapatkan adalah model dengan *learning rate* 0.001 memiliki stabilitas terbaik daripada model dengan *learning rate* 0.0001. Hasil ini dapat dilihat pada metrik skor IoU dan skor F1 yang menunjukkan fluktuasi validasi terhadap pelatihan memiliki perbedaan lebih rendah. Meskipun model mengalami perbedaan nilai rata-rata *loss function* antara validasi dan latihan lebih jauh, namun metrik lain menjadi prioritas. Selain itu, model dengan *learning rate* pada 0.001 mengalami konvergensi model lebih cepat daripada model dengan *learning rate* 0.0001.

FPN_EfficientNet-B5_Adam_Lr001_Batch8_Epoch_50



Gambar 4.8 Grafik Pelatihan dan Validasi Model FPN-EfficientNet-B5-Adam-lr001-Batch8

Tabel 4.10 Hasil Model EfficientNet-B5 Terbaik pada Skenario Perbandingan *Hyperparameter*

Nama	Kelas	Citra	Ground Truth	Output
532.jpg	0			
593.jpg	1			
658.jpg	2			
691.jpg	3			
718.jpg	4			

Tahap selanjutnya adalah melakukan pemeriksaan pada citra dengan skor IoU terendah pada tiap kelas pola reaksi. Masalah yang terjadi untuk citra dengan skor IoU terendah pada tiap kelas pada model ini hanya terjadi dengan nilai *recall* yang cukup rendah. Hal ini terjadi pada citra dengan kelas positif satu. Hasil prediksi model tidak dapat melakukan segmentasi darah pada bagian gumpalan darah yang kecil. Hal ini dapat menghilangkan informasi pola reaksi darah dan tidak representasi dari kelas pola reaksi tersebut. Secara keseluruhan, model masih memprediksi bagian darah dengan baik berdasarkan nilai metrik *recall* pada citra dengan skor IoU terendah pada beberapa kelas dimulai dari 0.8. Hasil pada model ini menjadi dasar untuk pencarian *decoder* pada skenario dua dengan *hyperparameter* yang sama karena hasil pada model ini merupakan hasil terbaik pada *encoder* EfficientNet-B5 dengan *decoder* FPN. Hasil dapat dilihat pada Tabel 4.10 dan Tabel 4.11.

Tabel 4.11 Metrik dari Citra dengan Skor IoU Terendah Model EfficientNet-B5 Skenario Perbandingan *Hyperparameter*

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
532.jpg	0	0.7688	0.9689	0.7883	0.8693	0.9786
593.jpg	1	0.6532	0.9708	0.6663	0.7902	0.9679
658.jpg	2	0.7084	0.8437	0.8153	0.8293	0.9396
691.jpg	3	0.8231	0.9695	0.845	0.9029	0.9699
718.jpg	4	0.6442	0.7486	0.8221	0.7836	0.9791

4.1.3 Skenario Perbandingan Metode *Decoder*

Skenario perbandingan *decoder* bertujuan untuk mencari *decoder* terbaik dalam tugas segmentasi darah dengan berbagai *encoder* beserta *hyperparameter* terbaik hasil dari skenario pertama dan kedua. *Dataset* yang digunakan adalah *dataset* latihan dan validasi tanpa augmentasi. Berikut adalah langkah-langkah yang dilakukan pada skenario ini :

1. Pembuatan model dilakukan pada setiap *encoder* yang digunakan dengan pasangan *hyperparameter* yang sama dan *decoder* yang sudah ditentukan, yaitu PAN dan DeepLabV3+.
2. Analisis hasil model setelah melakukan pelatihan sebanyak lima puluh *epoch*.
3. Menentukan model dengan *decoder* terbaik berdasarkan metrik prioritas.

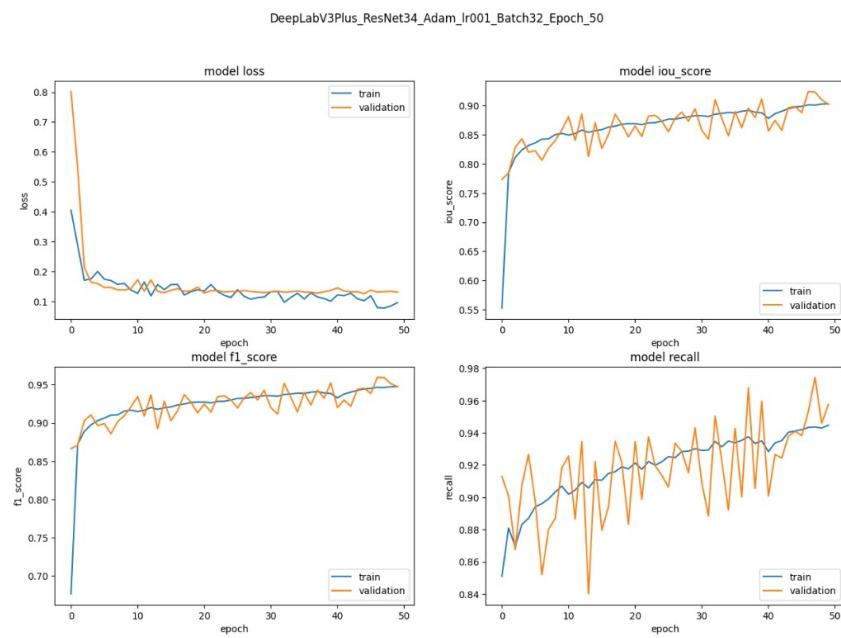
Untuk hasil model-model yang sudah dicoba akan dipisah pada setiap *encoder*. Berikut adalah hasil dari skenario ini :

1. ResNet34

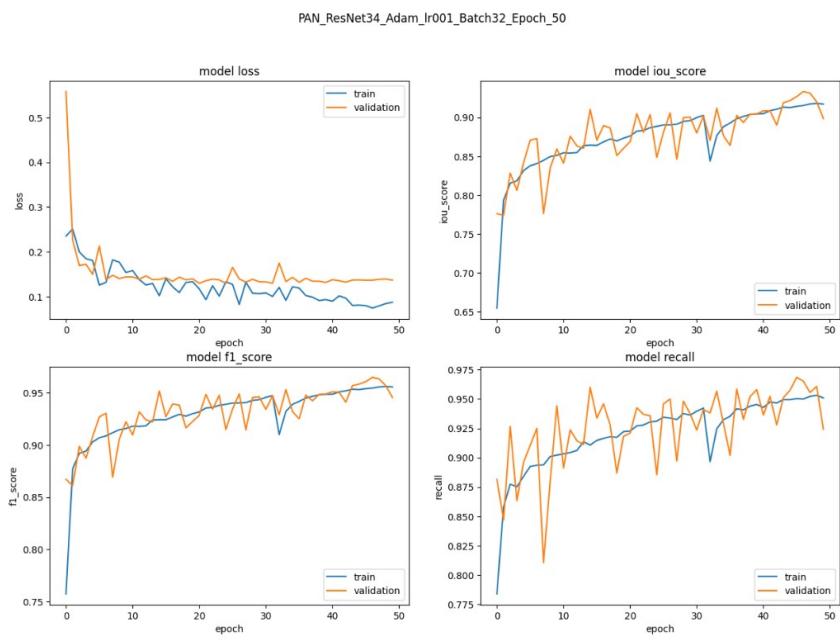
Pada bagian *encoder* ResNet34, *hyperparameter* yang digunakan pada ketiga *decoder* sama, yaitu *batch size* 32 dengan *optimizer* Adam dengan *learning rate* 0.001. Hasil ini dapat dilihat pada Tabel 4.12. Hasil yang didapatkan pada perbandingan dengan *encoder* ResNet34 ini adalah model dengan *decoder* DeepLabV3+ mendapatkan hasil yang terbaik, sedangkan model dengan *decoder* FPN mendapatkan hasil terburuk berdasarkan metrik skor IoU. Untuk model *decoder* DeepLabV3+ mendapatkan nilai *recall* paling tinggi dan memiliki perbedaan sekitar 0.02 pada kedua model lainnya. Metrik *recall* ini merepresentasikan kemampuan model dalam mendeteksi TP dengan meminimalisasi pendekripsi FN.

Decoder	Skor IoU	Presisi	Recall	Skor F1	Akurasi
Tabel 4.12 Hasil Uji Coba Model ResNet34 pada Skenario Perbandingan <i>Decoder</i>					
DeepLabV3+	0.8559	0.9425	0.9023	0.9191	0.9803
PAN	0.8485	0.9573	0.8798	0.9149	0.9795

Tahap selanjutnya adalah melihat kestabilan model dalam pelatihan dan validasi. Hasil grafik yang didapatkan selama pelatihan dan validasi dapat dilihat pada Gambar 4.5 (FPN), Gambar 4.9 (DeepLabV3+), dan Gambar 4.10 (PAN). Hasil yang didapatkan pada ketiga model ini adalah model memiliki kemiripan dalam hal kestabilan terutama pada metrik skor IoU dan skor F1. Hasil yang didapatkan mengalami fluktuasi, namun perbedaan yang didapatkan tidak signifikan. Hasil akhir yang didapatkan pada grafik *loss function* untuk model DeepLabV3+ tidak jauh berbeda pada pelatihan dan validasi. Model PAN mengalami beberapa masalah pada saat pertengahan pelatihan sehingga nilai skor IoU, skor F1, dan *recall* mengalami penurunan cukup signifikan namun dapat kembali ke posisi sebelum turun setelah beberapa *epoch*. Untuk hasil *loss function*, model PAN memiliki perbedaan antara pelatihan dan validasi lebih tinggi jika dibandingkan pada model DeepLabV3+. Hasil model FPN masih menjadi model dengan nilai kestabilan paling baik dibandingkan pada ketiga model lainnya.

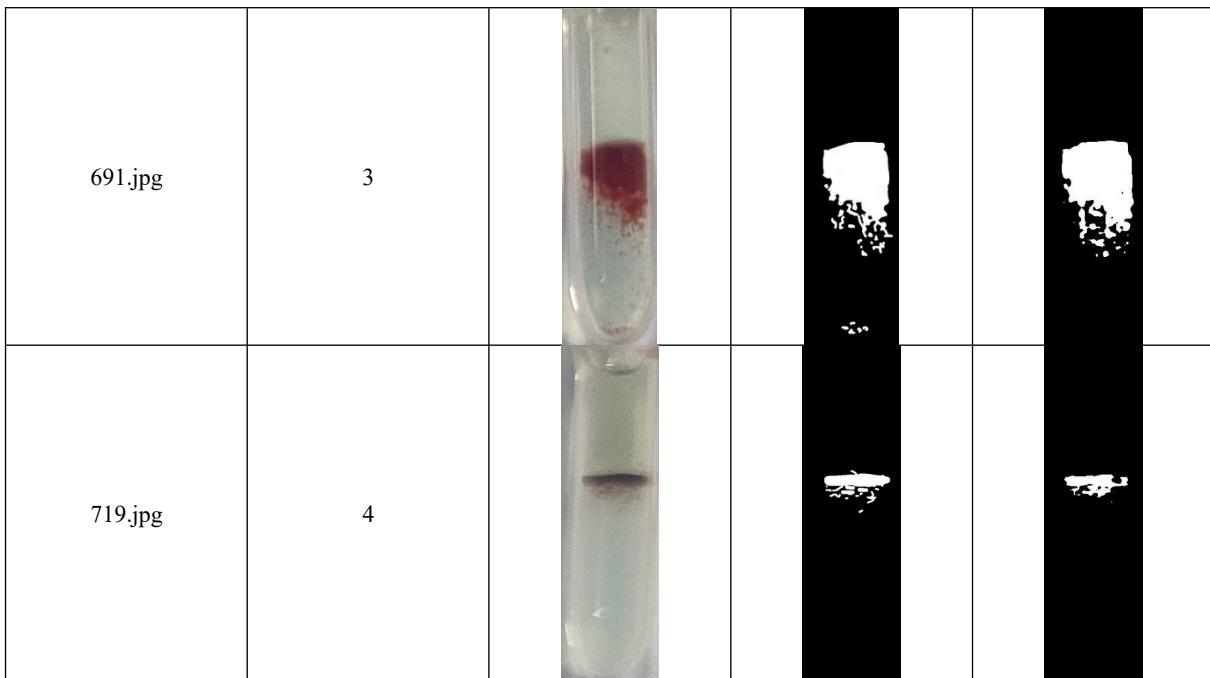


Gambar 4.9 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-ResNet34-Adam-lr001-Batch32



Tabel 4.13 Hasil Model ResNet34 Terbaik pada Skenario Perbandingan *Decoder*

Nama	Kelas	Citra	Ground Truth	Output
511.jpg	0			
609.jpg	1			
663.jpg	2			



Setelah itu, pada penelitian ini melihat performa dari model terbaik (DeepLabV3+) pada *encoder* ResNet34 dengan *batch size* 32 dan *learning rate* 0.001 dengan *dataset* pengujian dengan cara melakukan pemeriksaan citra-citra dengan nilai skor IoU terendah pada masing-masing representasi kelas citra tersebut. Hasil yang didapatkan pada model ini masih bermasalah pada kelas positif satu dan nilai *recall* elative rendah pada citra 609.jpg (0.6791) dan pada kelas positif empat pada citra 719.jpg dengan nilai *recall* sebesar 0.5731. Hasil *recall* pada citra 714.jpg cukup rendah. Secara keseluruhan, model dapat merepresentasikan kelas pola reaksi citra yang ada. Hasil dapat dilihat pada Tabel 4.13 dan Tabel 4.14.

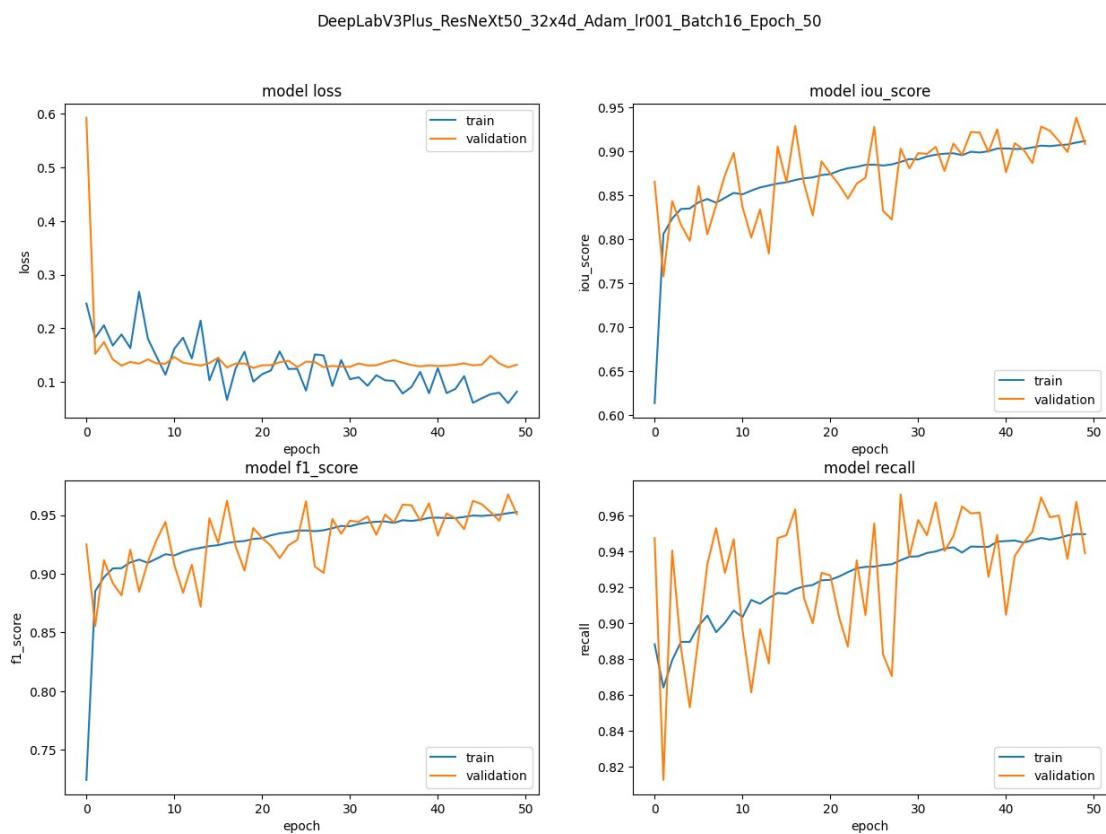
Tabel 4.14 Metrik dari Citra dengan Skor IoU Terendah Model ResNet34 Skenario Perbandingan <i>Decoder</i>						
211.jpg	0	0.1501	0.1002	0.9151	0.8510	0.9948
609.jpg	1	0.6476	0.9332	0.6791	0.7861	0.9405
663.jpg	2	0.7207	0.9204	0.7686	0.8377	0.9446
691.jpg	3	0.8297	0.9497	0.8678	0.9069	0.9705
719.jpg	4	0.5629	0.9695	0.5731	0.7204	0.9836

2. ResNeXt50

Pada *encoder* ResNeXt, hasil *hyperparameter* terbaik yang dipakai untuk skenario kedua adalah *batch size* 32 dan *optimizer* Adam dengan *learning rate* 0.001. Hasil pada *encoder* ini dapat dilihat pada Tabel 4.15. Hasil yang didapatkan untuk *encoder* ini adalah model dengan *decoder* PAN adalah model terbaik yang didapatkan. Hasil ini berdasarkan metrik utama dari penelitian ini, yaitu pada metrik skor IoU, skor F1, dan *recall*. Selain itu, metrik akurasi pada model PAN mendapatkan hasil yang terbaik juga. Namun, model PAN masih mendapatkan nilai presisi di urutan kedua, dibawah DeepLabV3+. Hasil ini masih bisa diterima karena nilai *recall* pada model mendapatkan hasil paling tinggi.

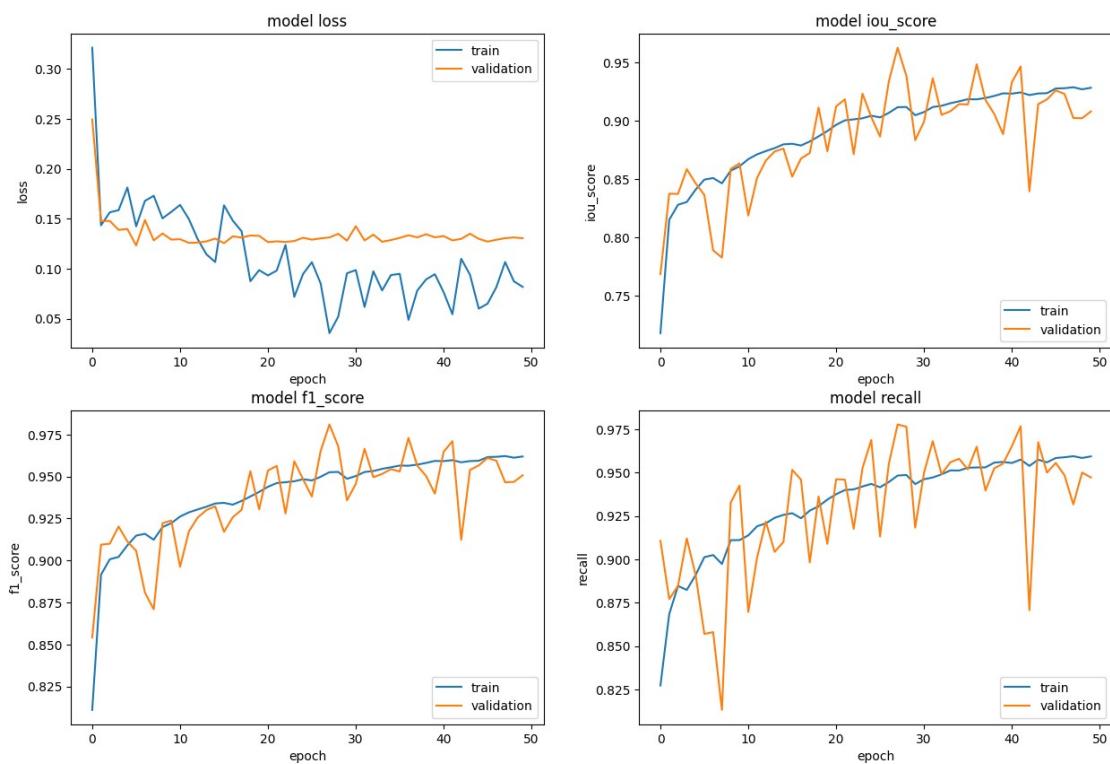
Tabel 4.15 Hasil Uji Coba Model ResNeXt50 pada Skenario Perbandingan <i>Decoder</i>				
FPN	0.8444	0.9261	0.9045	0.9122
DeepLabV3+	0.8594	0.9399	0.9074	0.9212
PAN	0.8636	0.9371	0.9152	0.9238
				0.9811

Tahap selanjutnya adalah melihat kestabilan model dalam pelatihan dan validasi. Hasil grafik yang didapatkan selama pelatihan dan validasi dapat dilihat pada Gambar 4.7 (FPN), Gambar 4.11 (DeepLabV3+), dan Gambar 4.12 (PAN). Hasil yang didapatkan pada ketiga model ini adalah pada model DeepLabV3+, model mengalami hasil awal pelatihan rendah dan langsung meroket pada *epoch* selanjutnya, sedangkan model PAN mengalami hasil awal yang tidak terlalu jauh. Selain itu, model DeepLabV3+ mengalami hasil pelatihan model dan validasi lebih stabil daripada model PAN. Hasil ini dapat terlihat dari fluktuasi model PAN dengan perubahan cukup signifikan masih terjadi pada *epoch* sekitar empat puluh, sedangkan model DeepLabV3+ terjadi fluktuasi relatif kecil. Meskipun hasil kedua model ini lebih bagus, model FPN masih mendapatkan hasil terbaik dalam kestabilan. Hasil model FPN pada awal *epoch* hingga *epoch* kelima masih berada pada skor IoU bernilai 0, namun mulai meroket pada *epoch* kedelapan dan berlanjut hingga akhir pelatihan dengan lebih stabil. Secara keseluruhan, model dengan *decoder* selain FPN mampu meningkatkan performa model pada pengujian. Hal ini juga terjadi pada *encoder* ResNet34.



Gambar 4.11 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-ResNeXt50-Adam-lr001-Batch16

PAN_ResNeXt50_32x4d_Adam_lr001_Batch16_Epoch_50



Tabel 4.16 Hasil Model ResNeXt50 Terbaik pada Skenario Perbandingan *Decoder*

Nama	Kelas	Citra	Ground Truth	Output
532.jpg	0			
609.jpg	1			

658.jpg	2			
691.jpg	3			
718.jpg	4			

Tahap selanjutnya adalah melakukan pemeriksaan pada citra dengan skor IoU terendah pada tiap kelas pola reaksi. Hasil yang didapatkan pada model ini adalah model memiliki nilai *recall* terutama pada kelas positif satu lebih tinggi dibandingkan pada model-model lain dengan nilai berada diatas 0.7. Hasil ini menggambarkan bahwa model mampu melakukan tugas segmentasi dengan detil lebih tinggi, namun model kurang mampu melakukan segmentasi citra pada golongan positif empat. Hasil ini dapat dilihat pada citra 718.jpg dengan hasil prediksi model kurang memiliki detil pada bagian gumpalan darah ukuran kecil Untuk permasalahan presisi, model sudah melakukan tugas segmentasi darah dengan presisi baik. Hasil dapat dilihat pada Tabel 4.16 dan Tabel 4.17.

Tabel 4.17 Metrik dari Citra dengan Skor IoU Terendah Model ResNeXt50 Skenario Perbandingan Decoder						
532.jpg	0	0.7497	0.9895	0.7557	0.8569	0.9712
609.jpg	1	0.66	0.9103	0.7059	0.7952	0.9414
658.jpg	2	0.6945	0.7998	0.8406	0.8197	0.9335
691.jpg	3	0.8325	0.9429	0.8767	0.9086	0.9708
718.jpg	4	0.6403	0.8985	0.6903	0.7807	0.9821

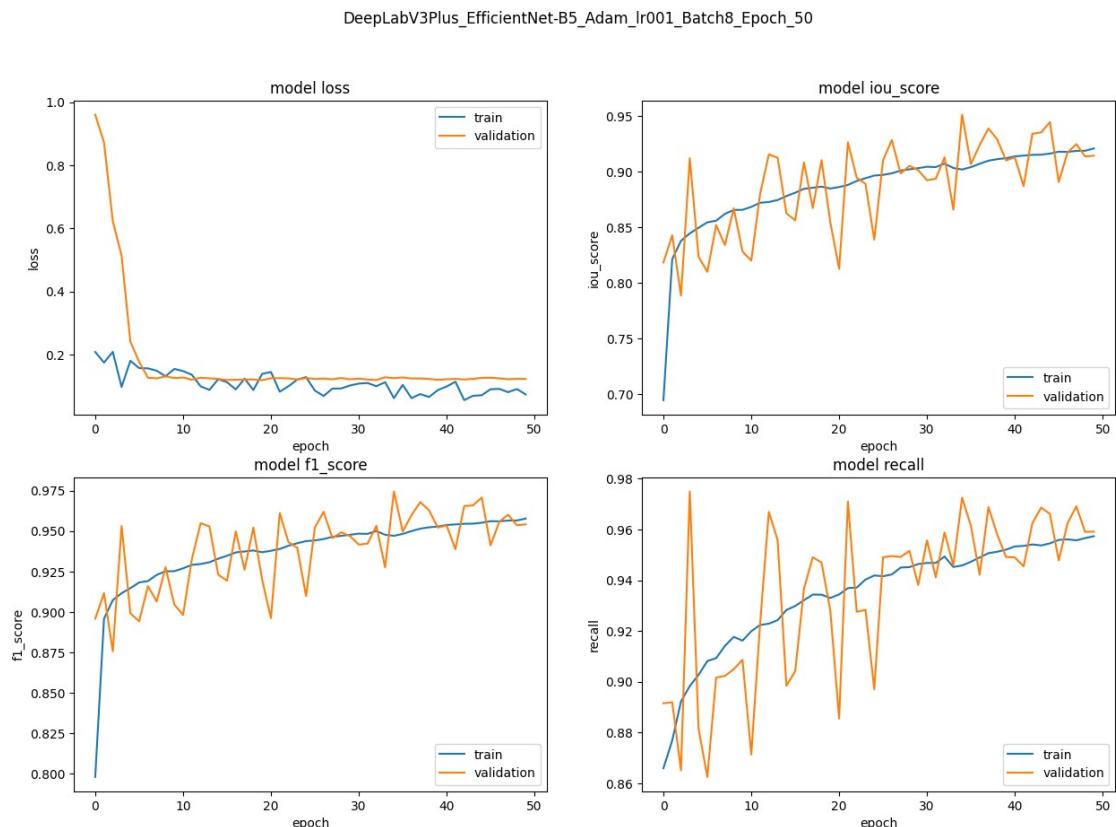
3. EfficientNet-B5

Untuk *encoder* EfficientNet-B5, *hyperparameter* yang digunakan adalah *batch size* 8 dan *optimizer* Adam dengan *learning rate* 0.001. Hasil dari setiap model dapat dilihat pada Tabel 4.18. Hasil yang didapatkan pada *encoder* EfficientNet-B5 adalah model dengan *decoder* PAN mendapatkan hasil terbaik diantara ketiga *encoder* lainnya. Hasil ini

ditunjukkan pada metrik utama dari penelitian ini, yaitu skor IoU, skor F1, dan *recall*. Selain itu, model juga mendapatkan hasil akurasi lebih baik daripada model lainnya. Untuk *encoder* ini juga terjadi hal yang serupa pada *encoder-encoder* sebelumnya, yaitu model terbaik mendapatkan presisi pada urutan kedua. Model dengan *recall* yang lebih tinggi akan menjadi prioritas meskipun hasil skor F1 yang didapatkan lebih rendah. Oleh karena itu, model yang diambil menjadi model terbaik adalah model dengan *decoder* PAN pada *encoder* EfficientNet-B5.

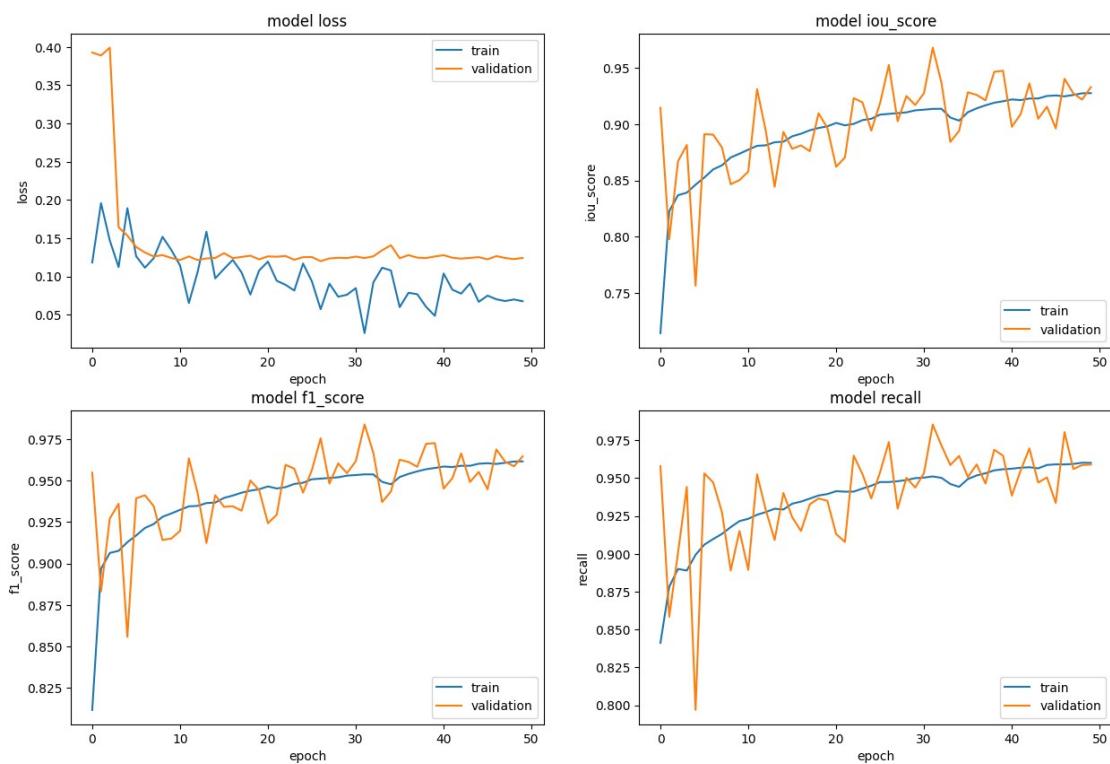
Tabel 4.18 Hasil Uji Coba Model EfficientNet-B5 pada Skenario Perbandingan <i>Decoder</i>					
FPN	0.8499	0.9339	0.9039	0.9159	0.9791
DeepLabV3+	0.8594	0.9365	0.9112	0.9217	0.9804
PAN	0.8609	0.9343	0.9145	0.9225	0.9809

Tahap selanjutnya adalah melihat kestabilan model dalam pelatihan dan validasi. Hasil grafik yang didapatkan selama pelatihan dan validasi dapat dilihat pada Gambar 4.8 (FPN), Gambar 4.13 (DeepLabV3+), dan Gambar 4.14 (PAN). Hasil yang didapatkan pada ketiga model ini adalah pada model DeepLabV3+ mendapatkan kestabilan lebih tinggi jika dibandingkan pada *decoder* PAN. Hasil ini dapat ditunjukkan pada grafik metrik skor IoU dan skor F1. Selain itu, model mendapatkan hasil *loss function* pelatihan dan validasi tidak berbeda jauh jika dibandingkan dengan *encoder* PAN. Namun, dari ketiga model ini, FPN masih memegang kestabilan model lebih tinggi. Fluktuasi yang dialami pada model FPN paling kecil dibandingkan pada semua model. Fluktuasi yang dialami semua model berkurang seiring pelatihan berjalan. Hal ini dapat terlihat pada grafik *recall* dari ketiga model mengalami fluktuasi lebih kecil seiringnya pelatihan.



Gambar 4.13 Grafik Pelatihan dan Validasi Model DeepLabV3Plus-EfficientNet-B5-Adam-lr001-Batch8

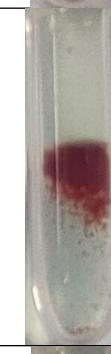
PAN_EfficientNet-B5_Adam_lr001_Batch8_Epoch_50



Gambar 4.14 Grafik Pelatihan dan Validasi Model PAN-EfficientNet-B5-Adam-lr001-Batch8

Tabel 4.19 Hasil Model EfficientNet-B5 Terbaik pada Skenario Perbandingan *Decoder*

Nama	Kelas	Citra	Ground Truth	Output
532.jpg	0			
610.jpg	1			

658.jpg	2						
691.jpg	3						
718.jpg	4						

Tahap selanjutnya adalah melakukan pemeriksaan pada citra dengan skor IoU terendah pada tiap kelas pola reaksi. Hasil *recall* pada model ini termasuk baik jika dibandingkan pada model-model sebelumnya. Hal ini ditunjukkan pada nilai *recall* pada kelas positif satu dengan nilai *recall* diatas 0.7, sedangkan beberapa model masih berada di bawah 0.7. Hasil *recall* rendah dapat dilihat pada citra 610.jpg pada bagian gumpalan darah ukuran kecil yang tersebar di atas gumpalan besar darah pada dasar tabung. Secara keseluruhan, model masih dapat merepresentasikan kelas pola reaksi yang ada. Hasil ini dapat dilihat pada Tabel 4.19 dan Tabel 4.20.

Tabel 4.20 Metrik dari Citra dengan Skor IoU Terendah Model EfficientNet-B5 Skenario Perbandingan <i>Decoder</i>						
532.jpg	0	0.773	0.9718	0.7908	0.8719	0.9791
610.jpg	1	0.6828	0.9172	0.7277	0.8115	0.9538
658.jpg	2	0.7129	0.8266	0.8382	0.8324	0.9393
691.jpg	3	0.8377	0.9293	0.8946	0.9117	0.9713
718.jpg	4	0.6444	0.8127	0.7568	0.7837	0.9808

4.1.4 Skenario Perbandingan Tanpa Augmentasi dan Augmentasi pada Latihan

Skenario perbandingan augmentasi dan tanpa augmentasi dilakukan untuk melihat performa model setelah dilakukan pelatihan terhadap *dataset* latihan dan *validasi* yang teraugmentasi. Setelah pelatihan dilakukan, performa model dilihat pada *dataset* tes dan dilakukan perbandingan pada model yang sama tanpa dilakukan augmentasi. Augmentasi ini juga dilakukan untuk meningkatkan performa model pada *dataset* tes. Berikut adalah langkah-

langkah pada skenario ini :

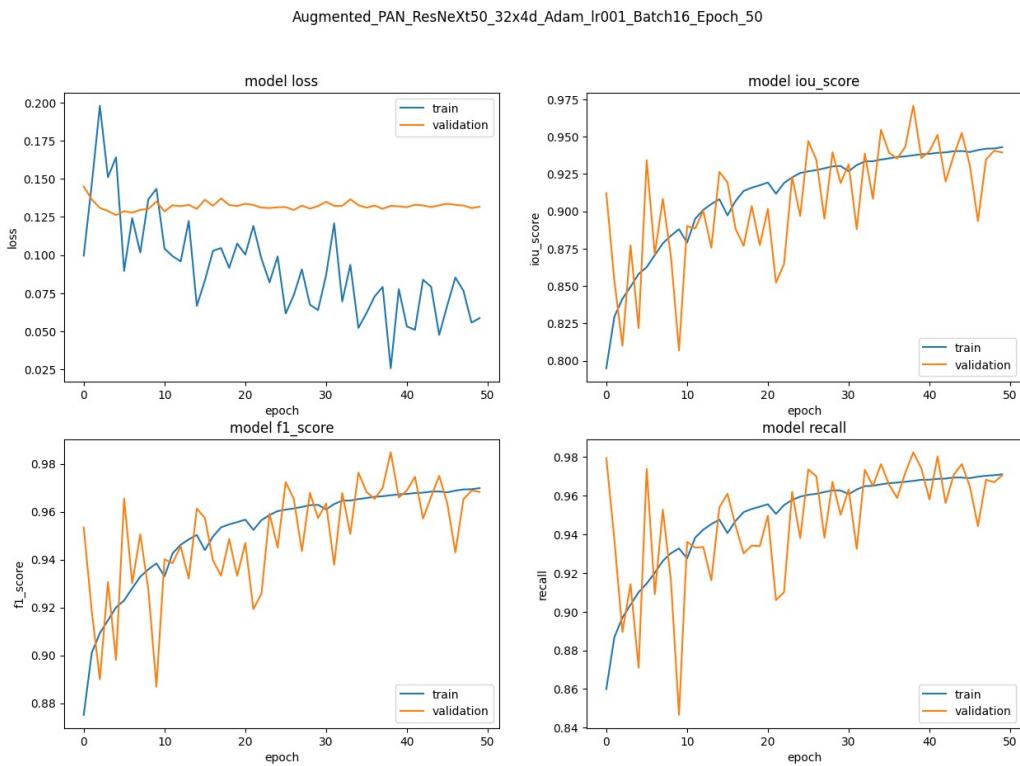
1. Pemilihan hasil model terbaik berdasarkan skenario satu dan dua.
2. Melakukan pelatihan sesuai dengan *decoder*, *encoder*, dan *hyperparameter* terbaik pada model terbaik dengan *dataset* latihan dan *validasi* sudah teraugmentasi.
3. Melakukan uji performa pada model teraugmentasi dengan *dataset* tes tidak teraugmentasi.

Model terbaik berdasarkan hasil skenario satu dan dua adalah PAN_ResNeXt50_32x4d dengan *hyperparameter batch size* 16 dan *optimizer* Adam dengan *learning rate* 0.001. Model ini ditentukan berdasarkan performa model tersebut pada *dataset* tes merupakan hasil terbaik dalam segi skor IoU dan skor F1. Selain itu, hasil metrik *recall* untuk *dataset* tes pada model ini merupakan hasil kedua terbaik dari model-model yang sudah dicoba. *Dataset* latihan dan validasi yang teraugmentasi juga memiliki citra original sebelum citra tersebut teraugmentasi.

Setelah melakukan pelatihan dengan *dataset* pelatihan dan validasi dengan *dataset* normal dan teraugmentasi, hasil yang didapatkan pada pengujian dapat dilihat pada Tabel 4.21. Hasil yang didapatkan adalah model dengan augmentasi adalah model dengan *dataset* teraugmentasi mendapatkan hasil pengujian lebih rendah dibandingkan model dengan pelatihan dan validasi tanpa augmentasi. Hasil ini dapat dilihat dari setiap metrik pengujian yang sudah ditentukan. Kejadian pada model terbaik dengan nilai presisi yang lebih rendah tidak terjadi pada skenario ini. Hal ini menandakan bahwa model mengalami *overfitting* cukup tinggi dan model gagal melakukan generalisasi dalam pembelajaran untuk meningkatkan hasil pengujian.

Tabel 4.21 Hasil Uji Coba Model pada Skenario Perbandingan Augmentasi dan Tanpa Augmentasi					
Non-augmentasi	0.8636	0.9371	0.9152	0.9238	0.9811
Augmentasi	0.8413	0.9195	0.9073	0.9106	0.9765

Hasil yang didapatkan pada pelatihan dan validasi model augmentasi dapat dilihat pada Gambar 4.12 dan Gambar 4.15 pada model tanpa augmentasi. Hasil yang didapatkan pada model augmentasi selama pelatihan dan validasi adalah stabilitas model lebih rendah dibandingkan model tanpa augmentasi. Hasil ini didapatkan pada grafik skor IoU dan skor F1. Fluktuasi yang terjadi pada grafik tersebut menunjukkan ketidakstabilan model selama pelatihan dan validasi. Hasil awal model augmentasi mendapatkan hasil cukup tinggi dibandingkan model-model lainnya. Selain itu, model mengalami penurunan performa pada bagian akhir iterasi *epoch*. Hal ini dapat dilihat pada grafik skor IoU yang mengalami penurunan pada validasi. Hasil akhir pada pelatihan dan validasi model augmentasi mendapatkan hasil lebih tinggi dibandingkan model tanpa augmentasi.



Gambar 4.15 Grafik Pelatihan dan Validasi Model Augmentasi

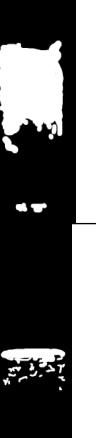
Tahap selanjutnya adalah melakukan pemeriksaan pada citra dengan skor IoU terendah pada tiap kelas pola reaksi. Hasil yang didapatkan pada model ini adalah model mayoritas mengalami masalah pada nilai *recall*. Hasil *recall* yang didapat pada model mendapatkan nilai cukup tinggi dibandingkan model-model lainnya. Masalah lain yang didapatkan adalah presisi model yang terjadi pada citra 660.jpg. Hasil ini dapat dilihat pada Tabel 4.23 dan Tabel 4.24. Secara umum, model dapat mempertahankan representasi kelas pola reaksi sebelumnya, namun hasil yang didapatkan lebih buruk daripada model tanpa

Tabel 4.22 Metrik dari Citra dengan Skor IoU Terendah Model Augmentasi
augmentasi.

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
532.jpg	0	0.7555	0.9709	0.7731	0.8608	0.9774
593.jpg	1	0.6555	0.9459	0.6810	0.7919	0.9676
660.jpg	2	0.6631	0.7201	0.8935	0.7975	0.9338
687.jpg	3	0.8072	0.956	0.838	0.8933	0.9623
718.jpg	4	0.6551	0.8871	0.7147	0.7916	0.9827

Tabel 4.23 Hasil Model Augmentasi

Nama	Kelas	Citra	Ground Truth	Output
------	-------	-------	--------------	--------

532.jpg	0			
593.jpg	1			
660.jpg	2			
687.jpg	3			
718.jpg	4			

4.1.5 Hasil Performa Model Terbaik pada Tiap Kelas

Subbab ini membahas performa model terbaik berdasarkan hasil prediksi model pada

metrik yang sudah ditentukan pada pengujian. Bagian ini lebih fokus pada performa model di setiap kelas yang ada. Hasil yang didapatkan dapat dilihat pada Tabel 4.23. Hasil yang didapatkan adalah performa model terbaik didapatkan pada kelas 0 (negatif) berdasarkan kelima metrik yang digunakan. Performa model yang didapatkan terburuk terjadi pada kelas 4 atau positif 4 jika melihat pada sisi metrik skor IoU. Selain itu, kelas 1 dan 2 atau positif satu dan positif dua mendapatkan hasil metrik skor IoU cukup rendah. Namun, jika melihat dari metrik *recall*, performa kelas positif satu merupakan kelas dengan performa model terburuk dengan rata-rata *recall* 0.8355. Hasil terendah pada setiap kelas pada model ini sudah dibahas pada Tabel 4.16 dan Tabel 4.17. Hasil pada kelas positif satu mayoritas bernilai dibawah 0.8 untuk metrik *recall*. Meskipun model mendapatkan hasil nilai *recall* yang cukup rendah, model masih mampu mendapatkan detil yang bagus terutama pada kelas positif satu pada bagian gumpalan darah berukuran kecil yang berada di atas gumpalan darah lebih besar. Performa model pada kelas positif empat lebih baik daripada kelas positif satu. Hasil yang didapatkan pada positif empat cenderung melakukan *overshoot* pada prediksi sehingga hasil yang didapatkan mengalami masalah presisi dan sering mengambil bagian bukan darah dijadikan darah (FP). Masalah pada FP ini masih bisa ditoleransi karena tujuan model segmentasi mendapatkan prediksi bagian darah sebanyak-banyaknya. Performa model pada kelas dua juga bermasalah pada presisi. Hasil yang didapatkan mendapatkan FP yang lebih banyak. Performa model pada kelas positif tiga mendapatkan hasil terbaik dengan nilai *recall* dan *presisi* sangat tinggi dibandingkan pada kelas-kelas lain.

Kelas	Citra I ₁₁	Presisi	Recall	Citra E ₁	Akurasi
Tabel 4.24 Metrik dari Citra dengan Skor IoU Terendah Model Augmentasi					
1	0.7853	0.9270	0.8355	0.8762	0.9661
2	0.7914	0.8671	0.9014	0.8825	0.9508
3	0.9347	0.9749	0.9562	0.9652	0.9894
4	0.7697	0.8511	0.8951	0.8676	0.9881

4.2 Pembahasan/Diskusi

Subbab ini akan melakukan pembahasan berdasarkan hasil yang sudah didapat pada Subbab 4.1 dengan pembagian pembahasan pada setiap skenario yang sudah dilakukan.

4.2

4.2.1 Pembahasan Skenario Perbandingan Metode *Encoder*

Hasil yang didapatkan pada skenario ini adalah *encoder* dengan *hyperparameter* terbaik yang dapat digunakan untuk skenario kedua. Tujuan dari skenario ini adalah mencari *encoder* yang dapat mengolah citra dan mendapatkan informasi atau pola dari citra-citra yang digunakan pada latihan dan validasi. Untuk keterangan model, *decoder* yang digunakan adalah FPN dengan *hyperparameter batch size* 8 dan *optimizer* Adam dengan *learning rate* 0.0001. Hasil yang didapatkan pada skenario ini mendapatkan hasil terbaik pada model ResNet34. Hasil ini merupakan hasil dengan nilai *recall* terbaik dari semua model yang sudah dilatih. Hasil ini menandakan bahwa model FPN dengan ResNet34 dapat melakukan pelatihan dengan *batch size* relatif kecil karena arsitektur yang digunakan tidak menggunakan *batch normalization*. Tentunya model mendapatkan kestabilan cukup rendah karena model masih menggunakan *batch size* cukup rendah. Hasil ini tidak menutup kemungkinan model lain dengan perubahan *hyperparameter* dan *decoder* dapat meningkatkan performa model saat pengujian. Hal ini dapat disimpulkan bahwa *encoder* ResNet34 lebih cocok digunakan pada

FPN dengan *batch size* rendah meskipun kedalaman, jumlah parameter, dan komponen CNN yang digunakan pada ResNeXt50 dan EfficientNet-B5 lebih banyak.

4.2.2 Pembahasan Skenario Perbandingan Metode *Hyperparameter*

Tujuan dilakukannya skenario ini adalah mendapatkan konfigurasi model dengan kestabilan yang cukup tinggi pada *encoder* yang sudah ditentukan untuk melakukan perbandingan skenario ketiga, yaitu *decoder*. Selain itu, skenario ini dilakukan untuk meningkatkan performa model pada saat pengujian. *Hyperparameter* yang diganti adalah model *batch size* dan *learning rate* sesuai dengan kemampuan Google Colab yang digunakan.

Untuk hasil pada model FPN dengan *encoder* ResNet34, hasil yang didapatkan adalah model mendapatkan hasil terbaik pengujian pada *batch size* 8 dengan *learning rate* 0.0001. Hasil ini memang direpresentasikan pada metrik skor IoU, skor F1, dan *recall*. Namun, hasil yang didapatkan pada model dengan *batch size* 32 dengan *learning rate* 0.001 lebih stabil meskipun tidak adanya *batch normalization* dari komponen model ini. Hasil yang didapatkan ini menjadi patokan untuk digunakan pada skenario ketiga. Kesimpulan yang didapat pada eksperimen ini adalah model mengalami *overfitting* karena hasil yang didapatkan pada metrik skor IoU untuk pelatihan dan validasi tidak jauh berbeda antara model dengan *batch size* 8 dan *learning rate* 0.0001 dan model dengan *batch size* 32 dan *learning rate* 0.001 dan lebih memilih untuk mengambil model kedua karena *decoder* PAN dan DeepLabV3+ memiliki komponen *batch normalization layer* untuk memanfaatkan *batch size* tinggi.

Untuk hasil pada model FPN dengan *encoder* ResNeXt50, hasil yang didapatkan adalah model mendapatkan hasil terbaik pada *hyperparameter batch size* 8 dan *optimizer Adam* dengan *learning rate* 0.0001 untuk latihan dan validasi, sedangkan hasil yang didapatkan untuk pengujian tes lebih tinggi pada model dengan *hyperparameter batch size* 32 dan *optimizer Adam* dengan *learning rate* 0.001. Hasil pada model dengan *batch size* 8 dan *optimizer Adam* dengan *learning rate* 0.0001 mengalami *overfitting* dan kurang stabil, sedangkan model *batch size* 32 dan *optimizer Adam* dengan *learning rate* 0.001 memiliki kestabilan lebih tinggi selama pelatihan dan mengalami *overfitting* lebih rendah. Hasil yang didapatkan ini menggambarkan pentingnya *batch size* tinggi dan dapat berpengaruh pada kestabilan model dan mencegah model mengalami *overfitting* berlebih sama seperti percobaan pada ResNet34.

Untuk hasil model FPN dengan *encoder* EfficientNet-B5, hasil yang didapatkan adalah model dengan *learning rate* lebih tinggi menghasilkan model lebih stabil dan lebih baik pada latihan, validasi, dan tes. Hasil ini menggambarkan bahwa model membutuhkan *learning rate* lebih tinggi untuk melakukan pembelajaran dan menuju konvergensi lebih cepat dan menghasilkan model lebih baik. Untuk perbandingan *encoder* dengan *hyperparameter* yang sama, model dengan *encoder* ResNet34 lebih baik dalam latihan, validasi, dan tes. Untuk urutan kedua, model dengan *encoder* ResNeXt50 dan untuk urutan ketiga diikuti oleh model dengan *encoder* EfficientNet-B5. Hal ini menunjukkan bahwa kedalaman model dan kompleksitas model tidak dapat menentukan keberhasilan dalam menjalankan tugas. Untuk tugas ini, pada model FPN, justru model dengan kedalaman lebih rendah mampu menjalankan tugas lebih baik daripada model EfficientNet-B5 dengan kedalaman model dan parameter lebih banyak. Namun, hal ini tidak dapat disimpulkan secara umum dan *hyperparameter* yang berbeda. Penggunaan *batch size* berbeda tidak dapat dilakukan karena limitasi *resource* pada Google Colab.

Kesimpulan yang dapat diambil pada skenario ini adalah model membutuhkan *learning rate* pada *optimizer Adam* lebih tinggi dari nilai *default* dan diiringi *batch size* yang tinggi. Hasil ini ditunjukkan pada model FPN dengan *encoder* ResNet34, model mengalami

kestabilan lebih tinggi dengan *hyperparameter batch size* 32 dan Adam dengan *learning rate* 0.001, untuk *encoder* ResNeXt50, model dengan *hyperparameter batch size* 16 dan Adam dengan *learning rate* 0.001 mengalami kestabilan model lebih tinggi, dan model EfficientNet-B5 mengalami kestabilan lebih tinggi dengan perubahan *learning rate* saja.

4.2.3 Pembahasan Skenario Perbandingan Metode *Decoder*

Perbandingan skenario ini dilakukan untuk pencarian pendekripsi atau bagian melakukan segmentasi hasil dari informasi yang sudah diolah dari citra-citra latihan dan validasi pada *encoder* yang sudah ditentukan. Untuk *decoder* FPN sudah dilakukan pada skenario satu. Hasil model yang didapatkan pada *decoder* DeepLabV3+ lebih baik jika dibandingkan pada model dengan *decoder* FPN untuk bagian tes. Hal ini terjadi pada skor IoU, skor F1, presisi, dan *recall*. Hal ini menunjukkan bahwa *decoder* DeepLabV3+ memiliki performa lebih baik jika dibandingkan dengan *decoder* FPN. Untuk perbandingan *encoder* pada *decoder* DeepLabV3+, hasil yang didapatkan adalah *encoder* ResNet34 melakukan tugas paling buruk dibandingkan model-model lain. Hasil terbaik didapatkan pada *encoder* EfficientNet-B5 dengan nilai skor F1 lebih tinggi namun skor IoU tidak jauh berbeda. Selain itu, *encoder* EfficientNet-B5 mendapatkan nilai *recall* lebih tinggi. Secara efisiensi, *encoder* ResNeXt50_32x4d lebih tinggi dengan adanya *bottleneck layer* untuk melakukan kompresi representasi citra yang diolah dan mengurangi parameter yang harus diubah dalam model. Hal ini terjadi pada *inference time* model ResNeXt50_32x4d lebih rendah cukup banyak, sekitar 0.4 detik pada bagian tes. Untuk *decoder* DeepLabV3+ sangat baik untuk peningkatan hasil tes jika dibandingkan FPN meskipun dari segi arsitektur, FPN menggabungkan beberapa lapis dari hasil konvolusi model dengan dimensi terkecil dengan *feature map* tinggi dan model dengan dimensi besar dengan *feature map* rendah dan dilakukan *upsampling* untuk menggabungkan menjadi satu dengan dimensi yang sama. Dari segi arsitektur, DeepLabV3+ hanya menggabungkan *low level feature* biasa dengan *low level feature* yang sudah dilakukan *upsampling*. Hal ini juga tidak menjadi jaminan performa model dalam tugas didasarkan pada kerumitan model.

Untuk hasil pada model dengan *decoder* PAN, untuk model dengan *encoder* ResNet34 mengalami *overfitting* pada validasi dan tes. Hasil dari *encoder* ini justru mengorbankan metrik *recall* dengan nilai sebesar 0.8798 yang berbeda jauh jika dibandingkan pada model dengan *encoder* ResNext50_32x4d dan EfficientNet-B5. Untuk model dengan *encoder* ResNeXt50_32x4d lebih baik daripada *encoder* EfficientNet-B5 dengan perbedaan tidak terlalu jauh pada tes yaitu sekitar 0.003 pada metrik skor IoU. Pada *encoder* EfficientNet-B5, hasil yang didapatkan adalah model menghasilkan presisi lebih baik, sedangkan *encoder* ResNeXt50_32x4d mengorbankan hasil presisi untuk *recall* lebih baik. Dalam perbandingan *encoder* untuk *decoder* PAN ini, model ResNeXt50_32x4d mampu menjalankan tugas segmentasi lebih baik jika dibandingkan pada *encoder* lain, terutama pada EfficientNet-B5 karena kedalaman *encoder* tersebut lebih dalam, tetapi menghasilkan hasil yang lebih buruk. Untuk perbandingan *decoder* dengan *encoder* yang sama, PAN mampu menjalankan tugas lebih baik dibandingkan *decoder* FPN dan DeepLabV3+ pada *encoder* ResNeXt50_32x4d dan EfficientNet-B5. Hasil ini didapatkan karena arsitektur PAN memiliki FPA (*Feature Pyramid Attention*) dan GAU (*Global Attention Upsampling*). FPA sendiri digunakan untuk mengekstraksi fitur secara presisi dan mengadopsi mekanisme *attention*. Mekanisme *attention* sendiri bertugas untuk melakukan seleksi dan fokus pada hal terpenting dari fitur. Setelah FPA dilakukan, GAU melakukan *upsampling*. GAU digunakan bersamaan dengan FPA dan digunakan untuk memberikan konteks secara global berdasarkan fitur yang sudah ada dilakukan pada FPA. Konteks global sendiri digunakan dengan cara

menggabungkan fitur *high level* yang dimasukkan pada konvolusi 1x1 dengan *batch normalization* dan ReLU dan akan dimultiplikasi dengan fitur *low level* (Li et al., 2018).

Kesimpulan pada skenario ini adalah *decoder* PAN lebih baik jika dibandingkan dengan *decoder* FPN dan DeepLabV3+ karena kelebihan arsitektur PAN memiliki FPA dan GAU untuk retensi informasi yang didapatkan. Meskipun EfficientNet-B5 lebih dalam, model PAN lebih baik melakukan tugas dengan *encoder* ResNeXt50. Rangkuman hasil dari skenario satu sampai tiga dapat dilihat pada Tabel 4.25.

Tabel 4.25 Hasil Model Terbaik dari Skenario Satu hingga Tiga					
	0.8559	0.9425	0.9023	0.9191	0.9803
ResNet34	0.8636	0.9371	0.9152	0.9238	0.9811
EfficientNet-B5	0.8609	0.9343	0.9145	0.9225	0.9809

4.2.4 Pembahasan Skenario Perbandingan Augmentasi dan Tanpa Augmentasi

Skenario ini dilakukan untuk meningkatkan performa model dalam menjalankan tugas segmentasi darah pada bagian tes. Hasil yang didapatkan pada model terbaik untuk skenario satu dan dua pada pelatihan augmentasi adalah model tidak dapat melakukan tugas segmentasi darah lebih baik dibandingkan pada model tanpa augmentasi. Hasil yang didapatkan turun sekitar 0.02 untuk skor IoU dan terjadi *overfitting* karena hasil tes memiliki perbedaan nilai skor IoU sekitar 0.1 dengan latihan dan sekitar 0.09 untuk validasi. Selain metrik skor IoU, model ini juga menghasilkan Hasil ini menunjukkan bahwa dataset *latihan* dan *validasi* sudah benar, namun terjadi *overfitting* karena jumlah perbandingan *augmentasi* dan data normal adalah 5:1 sehingga model tidak mampu belajar secara general citra yang dimasukkan ke pada *dataset* latihan.

Salah satu dugaan berdasarkan hasil ini adalah *decoder* yang digunakan adalah PAN dan di dalam PAN terdapat FPA dan GAU. FPA dan GAU mengambil informasi cukup banyak pada *dataset* latihan yang teraugmentasi dan gagal melakukan generalisasi yang terjadi pada *dataset* tes karena FPA dan GAU melakukan fokus pada fitur paling relevan yang diambil. Hasil ini terbukti bahwa model gagal melakukan segmentasi pada *dataset* tes dan cenderung mengarah pada citra teraugmentasi. Selain arsitektur PAN, masalah lainnya adalah kurangnya akurasi dan presisi pada *ground truth* segmentasi darah yang dilakukan. Hal ini terjadi karena *human error* dan seringnya terjadi kesalahan pada bagian gumpalan darah dengan ukuran kecil atau *smearing* terjadi pada darah sehingga darah terdilusi dan berwarna tidak pekat. Kekonsistennan *labelling* pada satu citra dengan citra inilah menjadi dugaan model gagal melakukan generalisasi. Pada model tanpa augmentasi, model lebih baik dalam generalisasi sedangkan pada augmentasi, model sudah mengambil informasi cenderung pada *daaset* latihan dan validasi.

4.2.5 Pembahasan Performa Model Terbaik pada Tiap Kelas

Performa model terbaik yang didapatkan pada tiap kelas berbeda-beda. Performa kelas terburuk didapatkan pada golongan positif empat jika dilihat pada metrik skor IoU. Hasil yang didapatkan adalah pada kelas positif empat, model mendapatkan nilai *recall* cukup tinggi, namun model kurang presisi dalam melakukan prediksi sehingga mendapatkan FP cukup banyak. Hal ini tidak terlalu menjadi masalah karena nilai *recall* yang didapatkan masih cukup tinggi. Hal ini berbeda jika melihat pada kelas positif satu. Kelas positif satu memiliki rata-rata *recall* paling rendah dari semua kelas. Hal ini berpotensi menghasilkan prediksi model yang tidak sesuai dengan representasi kelas dari masukan citra yang ada. Salah

satu alasan model kurang detil dalam hal pembelajaran ini adalah CNN mendapatkan informasi yang tidak sepenuhnya sehingga beberapa detil kecil yang ada pada gambar tidak didapatkan secara baik dari hasil *feature map* yang sudah dilakukan oleh *encoder* ResNeXt50 itu sendiri. Memperbanyak *feature map* dengan memperdalam *encoder* itu sendiri tidak menjamin model dapat mengambil informasi secara baik karena hasil dari *encoder* EfficientNet-B5 mendapatkan hasil lebih rendah dibandingkan *encoder* ResNeXt50. *Decoder* PAN sendiri sudah lebih bagus daripada *decoder* DeepLabV3+, namun detil-detil darah yang diambil juga masih kurang detil sehingga berpotensi menghilangkan bagian termasuk darah.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui tahapan-tahapan dalam pengumpulan citra untuk pembuatan dataset, perancangan model, pembuatan model, dan uji coba, Kesimpulan yang didapat sebagai berikut:

1. Pengumpulan citra dilakukan menggunakan *smartphone* yang ada dengan memastikan citra terlihat jelas detil darah yang ada pada setiap tabung yang sudah digunakan dalam kartu gel uji silang serasi. Setelah itu, dilakukan *cropping* untuk mendapatkan RoI yaitu setiap tabung yang digunakan sehingga mendapatkan hingga delapan tabung setiap kartu gel. Setelah melakukan *cropping*, citra disusun secara rapi dengan cara diurutkan berdasarkan angka dan terstruktur dalam *file system* untuk memudahkan akses pengambilan citra jika dibutuhkan. Setelah citra tabung tersusun, setiap citra dilakukan anotasi pada bagian darah sebagai *ground truth*. Namun, anotasi yang dilakukan kurang akurat dan konsisten untuk bagian darah yang kecil dan *smearing* darah. Selanjutnya adalah melakukan *labelling* kelas pola reaksi dengan melakukan *crosscheck* dengan ahli pada bidang ini. Setelah semua data darah terkumpul, data tersebut akan dipisahkan menjadi latihan, validasi, dan tes dengan rasio 8:1:1 pada setiap kelas pola reaksi. Setiap citra akan dilakukan *loading* dan *resize* menjadi 160x544.
2. Proses pembuatan model ini dilakukan dengan cara melakukan beberapa uji coba dengan berbagai *encoder*, *decoder*, dan *hyperparameter tuning* untuk mendapatkan model dengan hasil terbaik berdasarkan hasil pengujian model pada *dataset* tes. Setiap model dilatih menggunakan *dataset* latihan dan validasi yang sama. Hasil setiap model yang sudah dilatih akan dievaluasi dengan metrik yang sudah ditentukan, yaitu skor IoU, skor F1, akurasi, presisi, *recall*, spesifitas, dan NPV. Metrik yang akan dijadikan prioritas dan penentu model terbaik skor IoU, skor F1, dan *recall*. Selanjutnya adalah memilih model terbaik berdasarkan metrik yang sudah ditentukan dan melakukan latihan dengan *dataset* latihan dan validasi yang teraugmentasi beserta normal.
3. Hasil yang didapatkan pada setiap model setelah melakukan uji coba pada *dataset* tes tidak berbeda jauh, namun terdapat beberapa model mengalami *overfit* berlebih sehingga hasil model tidak terepresentasi dengan baik pada *dataset* tes. Hasil terbaik yang didapatkan adalah model PAN-ResNeXt50-32x4d-Adam-lr001-Batch16. Hasil model ini mendapatkan hasil terbaik pada skor IoU, skor F1, dan *recall* terbaik, tetapi mengorbankan presisi. Setelah model dilakukan pelatihan ulang dengan *dataset* latihan dan validasi teraugmentasi, hasil yang didapatkan adalah model mengalami *overfit* pada *dataset* tes. Hasil ini diduga pada arsitektur PAN yang berfokus pada fitur paling relevan dari *dataset* latihan dan validasi sehingga tidak mampu melakukan generalisasi pada *dataset* tes sehingga model yang akan digunakan tetap PAN-ResNeXt50-32x4d-Adam-lr001-Batch16 tanpa augmentasi.
4. Setelah melakukan pengecekan hasil prediksi segmentasi darah pada model terbaik, masih terdapat beberapa detil darah yang hilang sehingga representasi kelas pola reaksi citra berubah, namun hasil ini terjadi pada hasil uji silang serasi kelas positif satu menjadi kelas negatif. Setelah melakukan konsultasi pada ahli bidang ini, hasil ini bermasalah namun tidak fatal sehingga perlu dilakukan ulang pelatihan model.

5.2 Saran

Dalam penelitian selanjutnya, hasil model dapat ditingkatkan pada detil yang lebih tinggi dengan cara menambahkan citra latihan yang lebih banyak sehingga model mampu melakukan generalisasi lebih baik karena berdasarkan hasil penelitian yang sudah dilakukan, augmentasi *dataset* latihan dan validasi tidak mampu mengatasi masalah *overfitting*. Selain itu, perlu adanya perbaikan anotasi segmentasi darah yang sudah dilakukan dengan menentukan standar bagian darah yang termasuk pada bagian darah sehingga anotasi darah yang dilakukan konsisten dan model dapat melakukan pembelajaran dengan baik.

DAFTAR PUSTAKA

- Abdelrahman, A., & Viriri, S. (2023). EfficientNet family U-Net models for deep learning semantic segmentation of kidney tumors on CT images. *Frontiers in Computer Science*, 5. <https://doi.org/10.3389/fcomp.2023.1235622>
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481-2495.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018, August 22). *Encoder-decoder with atrous separable convolution for Semantic Image segmentation*. arXiv.org. <https://arxiv.org/abs/1802.02611>
- Crossmatch testing*. Cornell University College of Veterinary Medicine. (2019, March 25). Retrieved July 4, 2023, from <https://www.vet.cornell.edu/animal-health-diagnostic-center/testing/protocols/immunology/crossmatch>
- Delgado-Ortet, M., Molina, A., Alférez, S., Rodellar, J., & Merino, A. (2020). *A Deep Learning Approach for Segmentation of Red Blood Cell Images and Malaria Detection*. *Entropy*, 22(6), 657. <https://doi.org/10.3390/e22060657>
- Encyclopædia Britannica, inc. (n.d.-c). *ABO Blood Group System*. Retrieved July 4, 2023, from <https://www.britannica.com/science/ABO-blood-group-system>
- Glossary: Crossmatch - Blood Bank guy glossary*. Blood Bank Guy. (2019, July 26). Retrieved July 4, 2023, from <https://www.bbguy.org/education/glossary/glc23/>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Hitawala, S. (2018, May 9). *Evaluating resnext model architecture for Image Classification*. arXiv.org. <https://arxiv.org/abs/1805.08700>
- Khashman, A., & Al-Zgoul, E. (2010). Image segmentation of blood cells in leukemia patients. *Recent advances in computer engineering and applications*, 104-109.
- Khouani, A., El Habib Daho, M., Mahmoudi, S., Chikh, M., & Benzineb, B. (2020). Automated recognition of white blood cells using Deep Learning. *Biomedical Engineering Letters*, 10(3), 359–367. <https://doi.org/10.1007/s13534-020-00168-3>
- Li, H., Xiong, P., An, J., & Wang, L. (2018). Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180.
- Liu, Y., Liu, S., Wang, Y., Lombardi, F., & Han, J. (2018). A stochastic computational multi-layer perceptron with backward propagation. *IEEE Transactions on Computers*, 67(9), 1273–1286. <https://doi.org/10.1109/tc.2018.2817237>
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
- Mathon, A., Hopkins, J., McLaughlin, C. W., Johnson, S., Warner, M. Q., LaHart, D., & Wright, J. D. (1993). *Human Biology and Health*. Prentice Hall. Britannica, T. Editors of Encyclopaedia (2020, May 22). ABO blood group system.
- Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*. <https://doi.org/10.1109/iiphdw.2018.8388338>
- Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

- <https://doi.org/10.1109/tpami.2021.3059968>
- Müller, D., Soto-Rey, I., & Kramer, F. (2022). Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15(1). <https://doi.org/10.1186/s13104-022-06096-y>
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Oksuz, I., Clough, J. R., Ruijsink, B., Anton, E. P., Bustin, A., Cruz, G., ... & Schnabel, J. A. (2020). Deep learning-based detection and correction of cardiac MR motion artefacts during reconstruction for high-quality segmentation. *IEEE Transactions on Medical Imaging*, 39(12), 4001-4010.
- Pemasiri, A., Nguyen, K., Sridharan, S., & Fookes, C. (2021). Multi-modal Semantic Image Segmentation. *Computer Vision and Image Understanding*, 202, 103085. <https://doi.org/10.1016/j.cviu.2020.103085>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* 18 (pp. 234-241). Springer International Publishing.
- Salazar-Gonzalez, A., Kaba, D., Li, Y., & Liu, X. (2014). Segmentation of the blood vessels and optic disk in retinal images. *IEEE journal of biomedical and health informatics*, 18(6), 1874-1886.
- Seferbekov, S., Iglovikov, V., Buslaev, A., & Shvets, A. (2018). Feature pyramid network for multi-class land segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. <https://doi.org/10.1109/cvprw.2018.00051>
- SEN, S. Y., & OZKURT, N. (2020). Convolutional neural network hyperparameter tuning with adam optimizer for ECG Classification. *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. <https://doi.org/10.1109/asyu50717.2020.9259896>
- Shetty, A. K., Saha, I., Sanghvi, R. M., Save, S. A., & Patel, Y. J. (2021, April). A review: Object detection models. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-8). IEEE.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- Tran, T., Kwon, O.-H., Kwon, K.-R., Lee, S.-H., & Kang, K.-W. (2018). Blood cell images segmentation using Deep Learning Semantic segmentation. *2018 IEEE International Conference on Electronics and Communication Engineering (ICECE)*. <https://doi.org/10.1109/icecome.2018.8644754>
- Tuzkaya, G., Sennaroglu, B., Kalender, Z. T., & Mutlu, M. (2019). Hospital Service Quality Evaluation with IVIF-promethee and a case study. *Socio-Economic Planning Sciences*, 68, 100705.
- Vujovic, Ž. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, 12(6). <https://doi.org/10.14569/ijacsa.2021.0120670>
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., & Xiao, B. (2021). Deep high-resolution representation learning for visual

- recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3349–3364. <https://doi.org/10.1109/tpami.2020.2983686>
- Widiasri, M., Suciati, N., Faticahah, C., Astuti, E. R., Putra, R. H., & Arifin, A. Z. (2023). Alveolar bone and mandibular canal segmentation on cone beam computed tomography images using U-Net. *2023 8th International Conference on Instrumentation, Control, and Automation (ICA)*. <https://doi.org/10.1109/ica58538.2023.10273108>
- Xiao, C., & Buffiere, J. Y. (2021). Neural network segmentation methods for fatigue crack images obtained with X-ray tomography. *Engineering Fracture Mechanics*, 252, 107823.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017, April 11). *Aggregated residual transformations for deep neural networks*. arXiv.org. <https://doi.org/10.48550/arXiv.1611.05431>
- Xu, H., He, H., Zhang, Y., Ma, L., & Li, J. (2023). A comparative study of loss functions for road segmentation in remotely sensed road datasets. *International Journal of Applied Earth Observation and Geoinformation*, 116, 103159. <https://doi.org/10.1016/j.jag.2022.103159>
- Yojana, K., & Thillai Rani, L. (2023). OCT layer segmentation using U-NET semantic segmentation and RESNET34 encoder-decoder. *Measurement: Sensors*, 29, 100817. <https://doi.org/10.1016/j.measen.2023.100817>
- Yu, C., Hu, Z., Li, R., Xia, X., Zhao, Y., Fan, X., & Bai, Y. (2022). Segmentation and density statistics of mariculture cages from remote sensing images using mask R-CNN. *Information Processing in Agriculture*, 9(3), 417-430.
- Zhang, M., Li, X., Xu, M., & Li, Q. (2020). Automated semantic segmentation of red blood cells for sickle cell disease. *IEEE Journal of Biomedical and Health Informatics*, 24(11), 3095–3102. <https://doi.org/10.1109/jbhi.2020.3000484>

[Halaman ini sengaja dikosongkan]

LAMPIRAN-LAMPIRAN

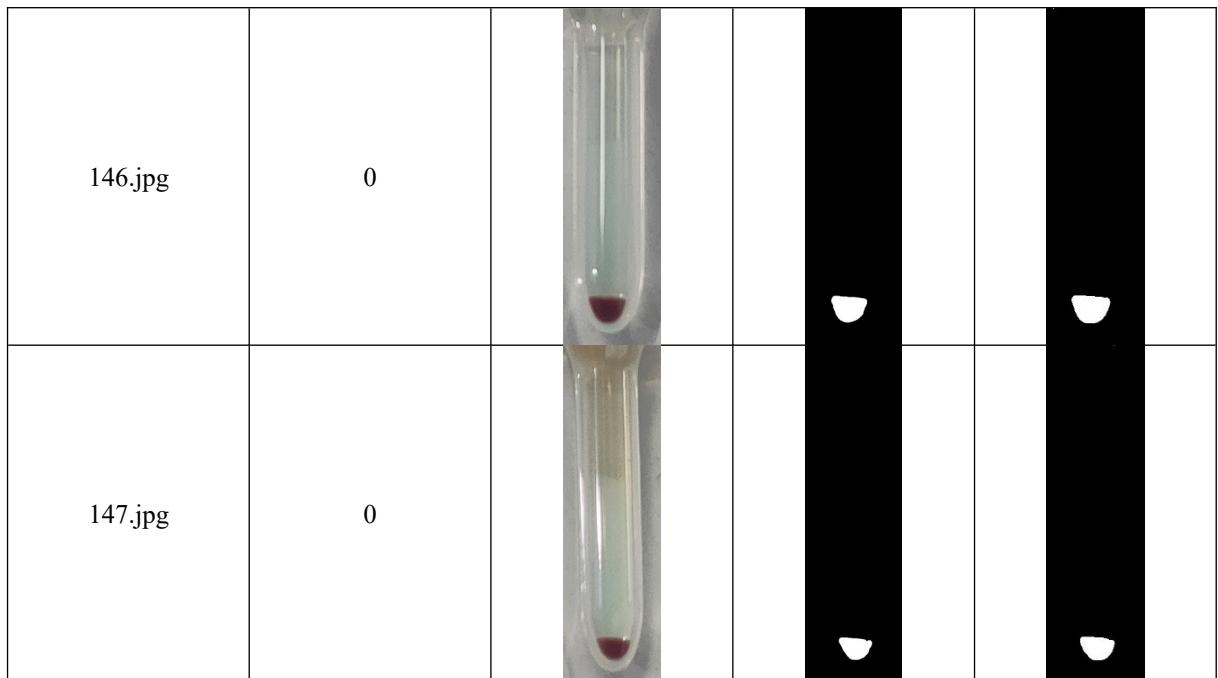
L1. Hasil Model Terbaik pada Skenario *Encoder*

Model terbaik adalah FPN ResNet34 *batch size* 8 Adam *learning rate* 0.0001

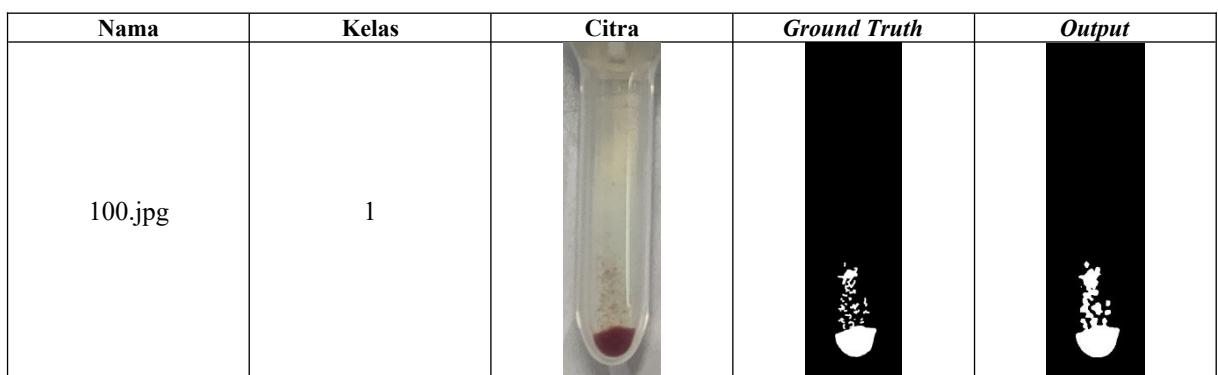
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
131.jpg	0	0.9506	0.9587	0.9912	0.9746	0.9987
132.jpg	0	0.9759	0.9805	0.9952	0.9878	0.9992
133.jpg	0	0.9403	0.9434	0.9965	0.9693	0.9987
134.jpg	0	0.9464	0.9503	0.9957	0.9725	0.9988
135.jpg	0	0.9381	0.9418	0.9958	0.9681	0.9987
136.jpg	0	0.9540	0.9552	0.9987	0.9765	0.9986
137.jpg	0	0.9803	0.9861	0.9940	0.9901	0.9992
145.jpg	0	0.9305	0.9341	0.9958	0.9640	0.9985
146.jpg	0	0.9511	0.9541	0.9966	0.9749	0.9987
147.jpg	0	0.9582	0.9705	0.9869	0.9786	0.9992

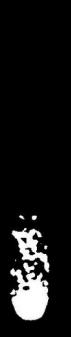
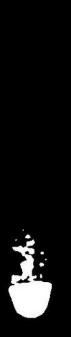
Nama	Kelas	Citra	Ground Truth	Output
131.jpg	0			
132.jpg	0			
133.jpg	0			

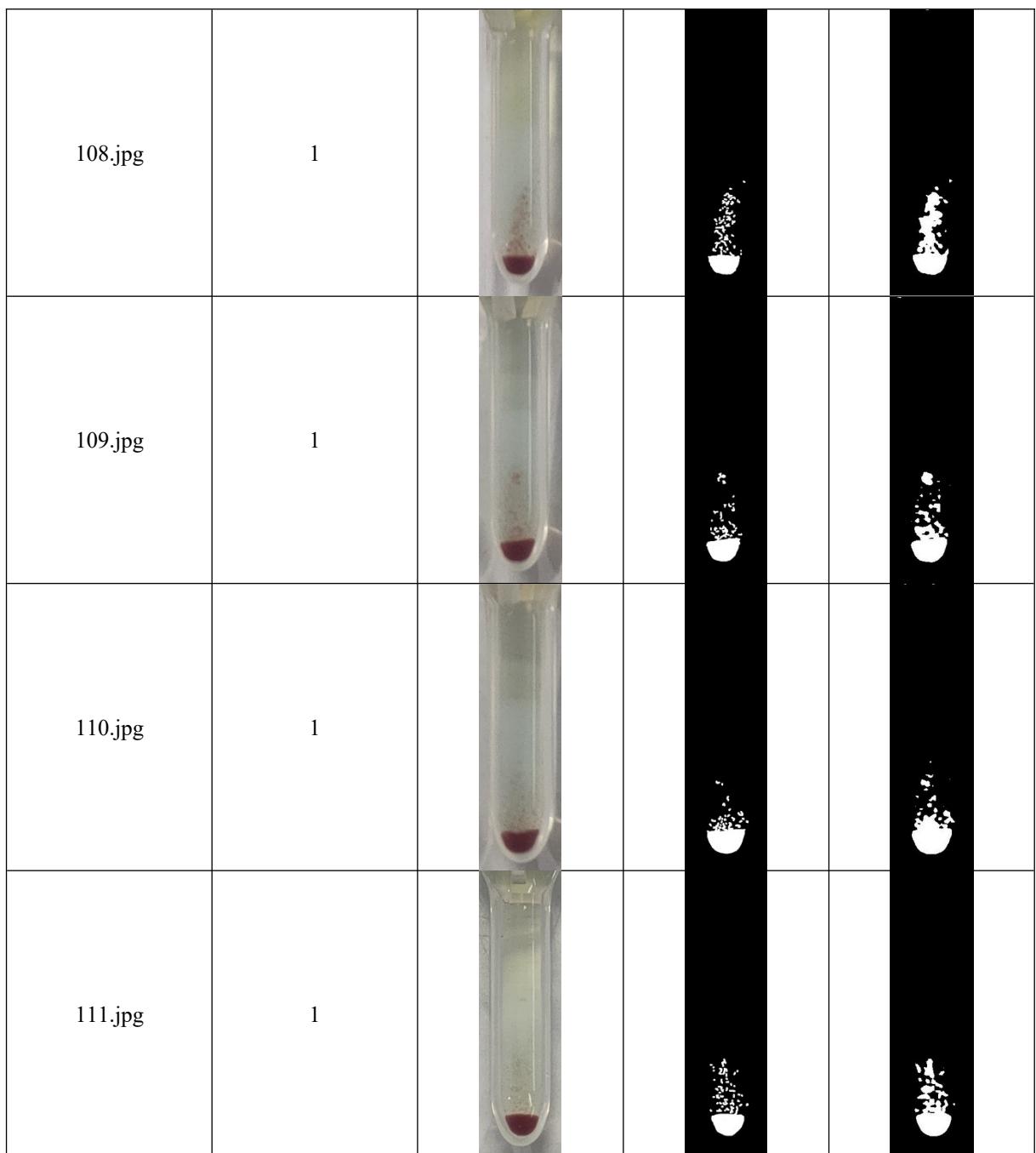
134.jpg	0			
135.jpg	0			
136.jpg	0			
137.jpg	0			
145.jpg	0			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
100.jpg	1	0.8675	0.9191	0.9391	0.9291	0.9921
102.jpg	1	0.9094	0.9235	0.9834	0.9525	0.9895
104.jpg	1	0.9230	0.9422	0.9784	0.9599	0.9907
105.jpg	1	0.7008	0.8074	0.8415	0.8241	0.9791
113.jpg	1	0.8854	0.9185	0.9608	0.9392	0.9866
107.jpg	1	0.8375	0.8899	0.9341	0.9115	0.9908
108.jpg	1	0.8043	0.8343	0.9571	0.8915	0.9873
109.jpg	1	0.8247	0.8620	0.9500	0.9039	0.9903
110.jpg	1	0.8454	0.8832	0.9518	0.9162	0.9911
111.jpg	1	0.8209	0.9033	0.8999	0.9016	0.9893



102.jpg	1			
104.jpg	1			
105.jpg	1			
113.jpg	1			
107.jpg	1			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
101.jpg	2	0.8807	0.8951	0.9820	0.9365	0.9851
103.jpg	2	0.9022	0.9280	0.9701	0.9486	0.9883
112.jpg	2	0.8434	0.9006	0.9299	0.9150	0.9771
115.jpg	2	0.8665	0.9072	0.9508	0.9285	0.9819
116.jpg	2	0.8760	0.9325	0.9353	0.9339	0.9837
117.jpg	2	0.9018	0.9442	0.9525	0.9483	0.9880
118.jpg	2	0.8493	0.8915	0.9471	0.9185	0.9828
15.jpg	2	0.8456	0.8972	0.9363	0.9163	0.9746
16.jpg	2	0.8551	0.9139	0.9299	0.9218	0.9777
18.jpg	2	0.8864	0.9163	0.9645	0.9397	0.9827

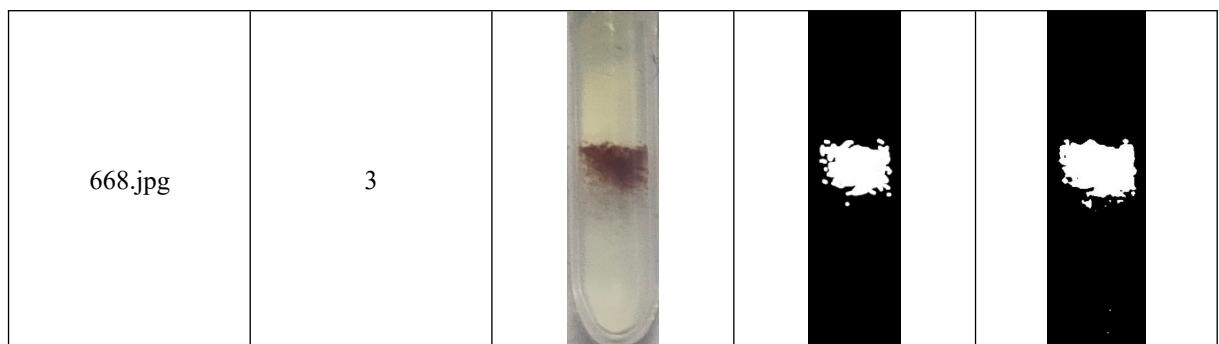
Nama	Kelas	Citra	Ground Truth	Output
101.jpg	2			
103.jpg	2			
112.jpg	2			
115.jpg	2			
116.jpg	2			

117.jpg	2			
118.jpg	2			
15.jpg	2			
16.jpg	2			
18.jpg	2			

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
13.jpg	3	0.8973	0.9301	0.9621	0.9458	0.9926
14.jpg	3	0.8417	0.9347	0.8942	0.9140	0.9801
17.jpg	3	0.7967	0.8306	0.9513	0.8868	0.9748
634.jpg	3	0.8996	0.9428	0.9515	0.9471	0.9811
642.jpg	3	0.8432	0.9095	0.9204	0.9149	0.9601
664.jpg	3	0.8622	0.8766	0.9812	0.9260	0.9894
665.jpg	3	0.8981	0.9185	0.9758	0.9463	0.9934
666.jpg	3	0.8710	0.8858	0.9811	0.9310	0.9907
667.jpg	3	0.8847	0.9089	0.9707	0.9388	0.9898
668.jpg	3	0.9072	0.9437	0.9591	0.9513	0.9897

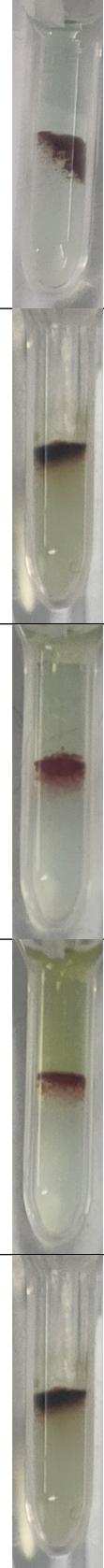
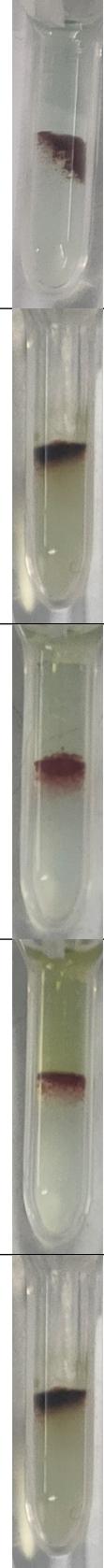
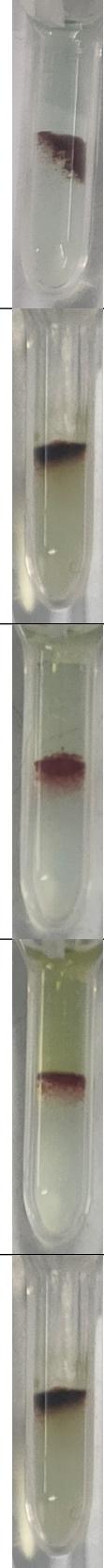
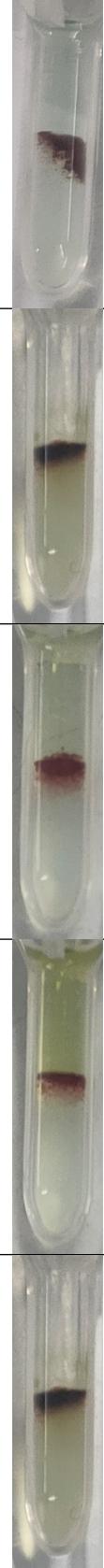
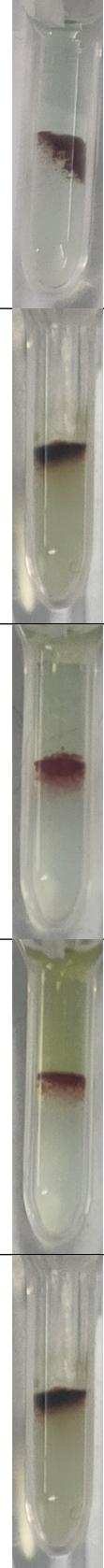
Nama	Kelas	Citra	Ground Truth	Output
13.jpg	3			
14.jpg	3			
17.jpg	3			
634.jpg	3			

642.jpg	3			
664.jpg	3			
665.jpg	3			
666.jpg	3			
667.jpg	3			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
1.jpg	4	0.9324	0.9517	0.9788	0.9650	0.9987
11.jpg	4	0.8678	0.9050	0.9547	0.9292	0.9889
12.jpg	4	0.8827	0.9355	0.9399	0.9377	0.9919
2.jpg	4	0.9150	0.9337	0.9785	0.9556	0.9959
3.jpg	4	0.9189	0.9389	0.9773	0.9577	0.9962
4.jpg	4	0.9215	0.9303	0.9898	0.9591	0.9952
5.jpg	4	0.8583	0.8803	0.9717	0.9237	0.993
6.jpg	4	0.8290	0.8502	0.9706	0.9065	0.9916
675.jpg	4	0.8258	0.9282	0.8821	0.9046	0.9816
689.jpg	4	0.9137	0.9369	0.9736	0.9549	0.9945

Nama	Kelas	Citra	Ground Truth	Output
1.jpg	4			
11.jpg	4			

12.jpg	4			
2.jpg	4			
3.jpg	4			
4.jpg	4			
5.jpg	4			

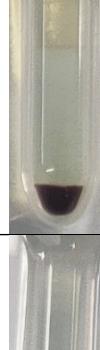
6.jpg	4					
675.jpg	4					
689.jpg	4					

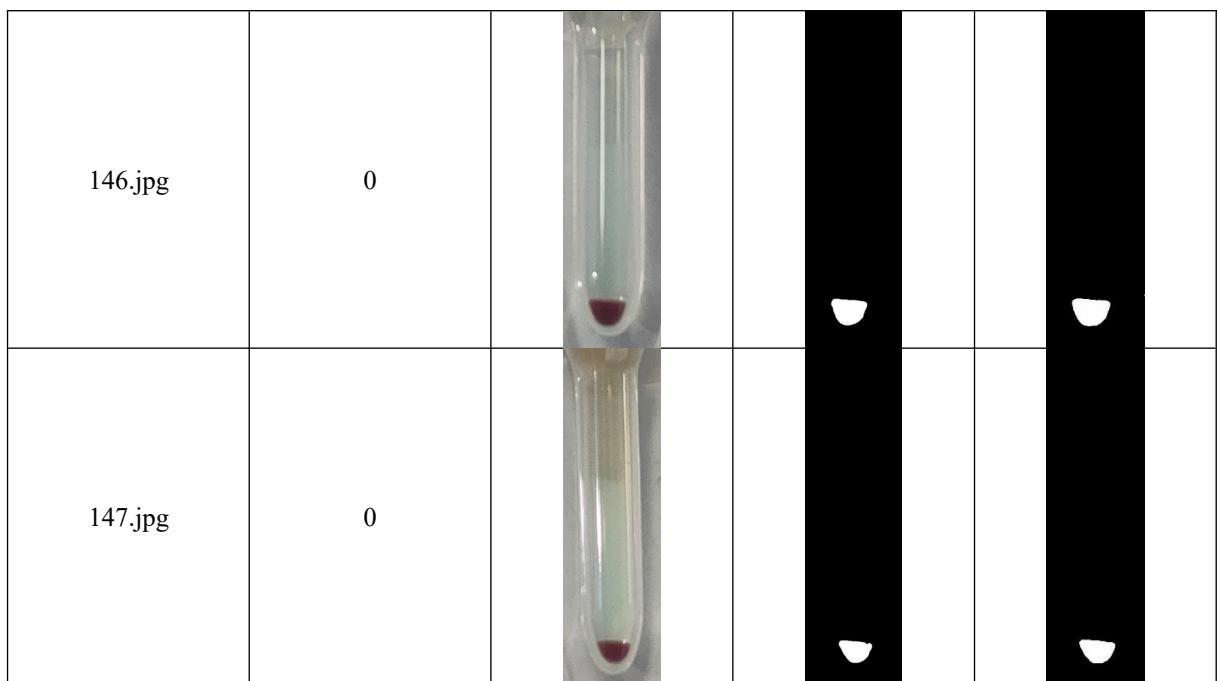
L2. Hasil Model Terbaik pada Skenario *Hyperparameter*

Model terbaik adalah FPN EfficientNet-B5 *batch size* 8 *learning rate* 0.001.

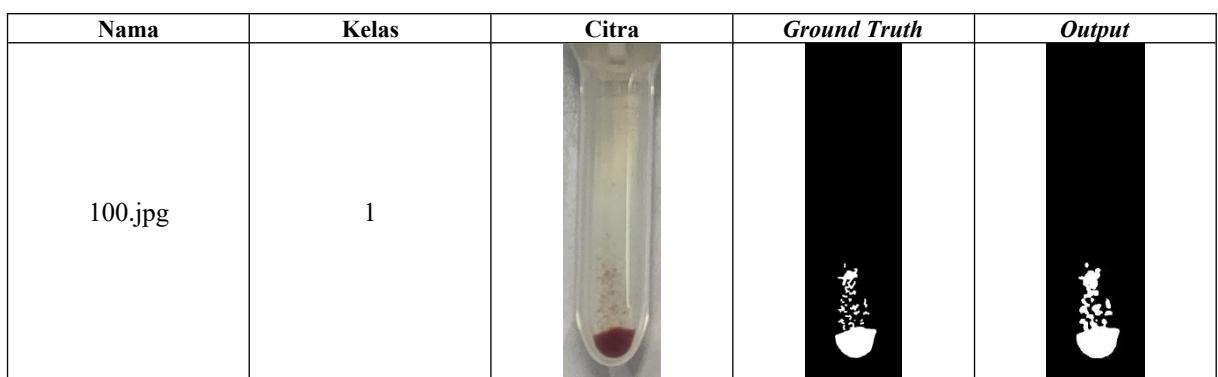
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
131.jpg	0	0.9757	0.9901	0.9852	0.9877	0.9994
132.jpg	0	0.9807	0.9955	0.9850	0.9902	0.9993
133.jpg	0	0.9791	0.9911	0.9877	0.9894	0.9995
134.jpg	0	0.9742	0.9994	0.9747	0.9869	0.9994
135.jpg	0	0.9692	0.9794	0.9893	0.9843	0.9993
136.jpg	0	0.9897	0.9938	0.9958	0.9948	0.9997
137.jpg	0	0.9787	0.9878	0.9906	0.9892	0.9992
145.jpg	0	0.9405	0.9501	0.9893	0.9693	0.9987
146.jpg	0	0.9663	0.9745	0.9913	0.9828	0.9991
147.jpg	0	0.9732	0.9767	0.9962	0.9864	0.9994

Nama	Kelas	Citra	Ground Truth	Output
131.jpg	0			
132.jpg	0			
133.jpg	0			

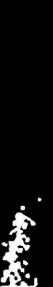
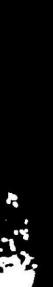
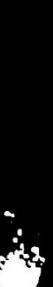
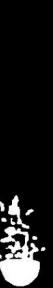
134.jpg	0			
135.jpg	0			
136.jpg	0			
137.jpg	0			
145.jpg	0			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
100.jpg	1	0.9094	0.9452	0.9599	0.9525	0.9947
102.jpg	1	0.9347	0.9556	0.9770	0.9662	0.9927
104.jpg	1	0.9546	0.9772	0.9763	0.9767	0.9947
105.jpg	1	0.7287	0.8673	0.8201	0.8430	0.9822
113.jpg	1	0.9233	0.9567	0.9635	0.9601	0.9914
107.jpg	1	0.9071	0.9354	0.9676	0.9513	0.9950
108.jpg	1	0.8626	0.9262	0.9262	0.9262	0.9920
109.jpg	1	0.9036	0.9349	0.9642	0.9494	0.9951
110.jpg	1	0.9132	0.9454	0.9639	0.9546	0.9953
111.jpg	1	0.8687	0.9448	0.9151	0.9297	0.9924



102.jpg	1			
104.jpg	1			
105.jpg	1			
113.jpg	1			
107.jpg	1			

108.jpg	1			
109.jpg	1			
110.jpg	1			
111.jpg	1			

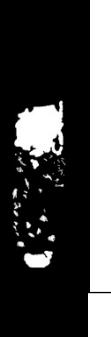
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
101.jpg	2	0.9189	0.9427	0.9733	0.9577	0.9903
103.jpg	2	0.9400	0.9681	0.9701	0.9691	0.9931
112.jpg	2	0.8894	0.9399	0.9429	0.9414	0.9845
115.jpg	2	0.9172	0.9633	0.9504	0.9568	0.9894
116.jpg	2	0.9124	0.9496	0.9588	0.9542	0.9887
117.jpg	2	0.9284	0.9614	0.9643	0.9628	0.9914
118.jpg	2	0.8933	0.9380	0.9492	0.9436	0.9884
15.jpg	2	0.8895	0.9340	0.9491	0.9415	0.9824
16.jpg	2	0.8982	0.9479	0.9447	0.9463	0.9848
18.jpg	2	0.9237	0.9589	0.9617	0.9603	0.9889

Nama	Kelas	Citra	Ground Truth	Output
101.jpg	2			
103.jpg	2			
112.jpg	2			
115.jpg	2			
116.jpg	2			

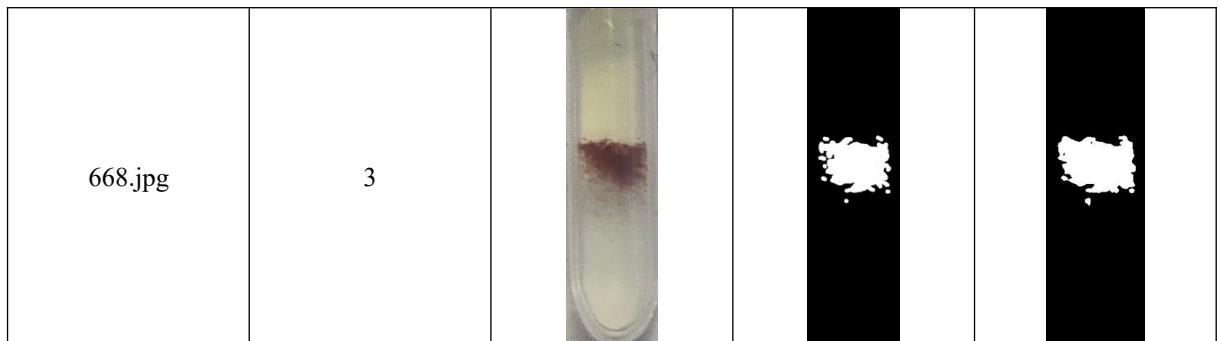
117.jpg	2			
118.jpg	2			
15.jpg	2			
16.jpg	2			
18.jpg	2			

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
------	-------	----------	---------	--------	---------	---------

13.jpg	3	0.9447	0.9644	0.9789	0.9716	0.9961
14.jpg	3	0.9011	0.9470	0.9489	0.9480	0.9877
17.jpg	3	0.8899	0.9364	0.9472	0.9417	0.9878
634.jpg	3	0.9115	0.9670	0.9407	0.9537	0.9838
642.jpg	3	0.7265	0.9185	0.7765	0.8416	0.9319
664.jpg	3	0.8995	0.9268	0.9683	0.9471	0.9927
665.jpg	3	0.9218	0.9529	0.9658	0.9593	0.9951
666.jpg	3	0.9116	0.9194	0.9907	0.9538	0.9939
667.jpg	3	0.9198	0.9648	0.9516	0.9582	0.9933
668.jpg	3	0.9434	0.9743	0.9675	0.9709	0.9939

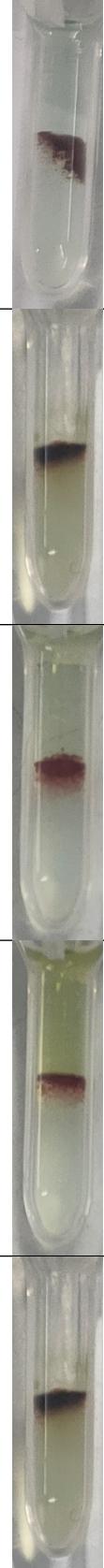
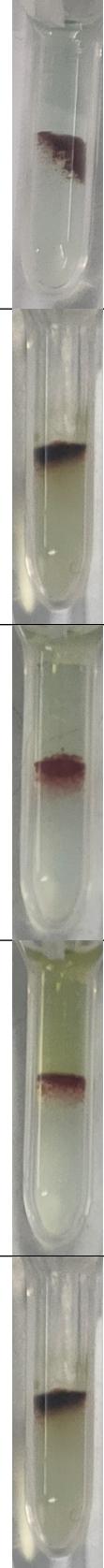
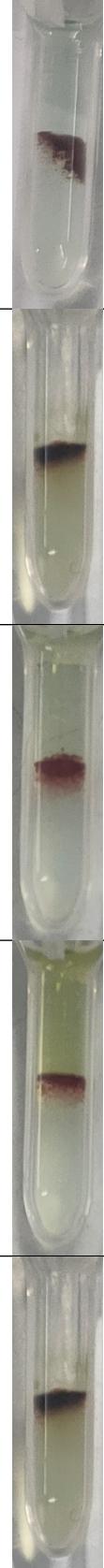
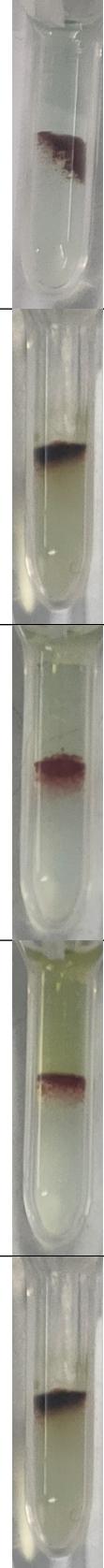
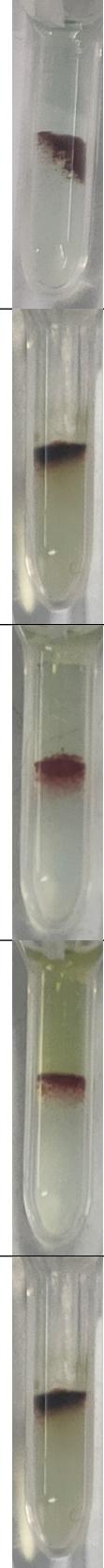
Nama	Kelas	Citra	Ground Truth	Output
13.jpg	3			
14.jpg	3			
17.jpg	3			
634.jpg	3			

642.jpg	3			
664.jpg	3			
665.jpg	3			
666.jpg	3			
667.jpg	3			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
1.jpg	4	0.9454	0.9569	0.9874	0.9719	0.9990
11.jpg	4	0.9210	0.9552	0.9626	0.9589	0.9937
12.jpg	4	0.9188	0.9544	0.961	0.9576	0.9944
2.jpg	4	0.9528	0.9774	0.9743	0.9758	0.9978
3.jpg	4	0.9585	0.9766	0.9810	0.9788	0.9981
4.jpg	4	0.9669	0.9768	0.9896	0.9831	0.9980
5.jpg	4	0.9160	0.9365	0.9766	0.9561	0.9962
6.jpg	4	0.8835	0.9135	0.9641	0.9381	0.9946
675.jpg	4	0.8930	0.9478	0.9392	0.9435	0.9889
689.jpg	4	0.8692	0.9061	0.9552	0.9300	0.9914

Nama	Kelas	Citra	Ground Truth	Output
1.jpg	4			
11.jpg	4			

12.jpg	4			
2.jpg	4			
3.jpg	4			
4.jpg	4			
5.jpg	4			

6.jpg	4				
675.jpg	4				
689.jpg	4				

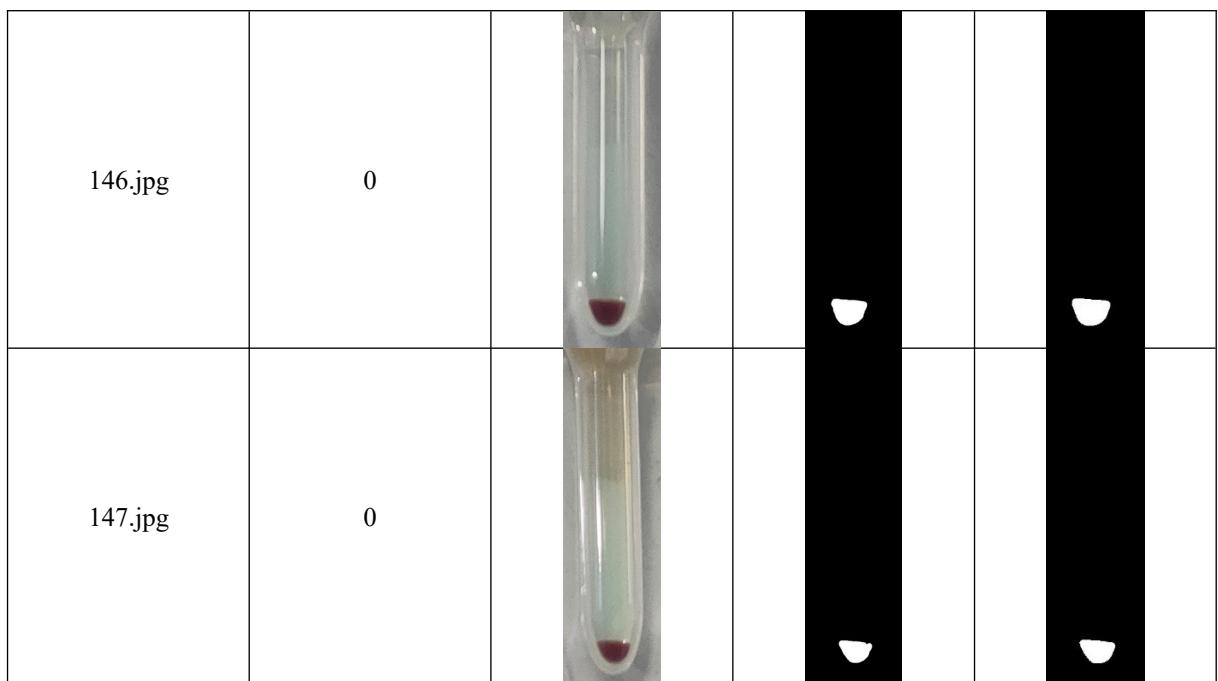
L3. Hasil Model Terbaik pada Skenario Decoder

Model terbaik adalah PAN ResNeXt50 *batch size* 16 Adam *learning rate* 0.001.

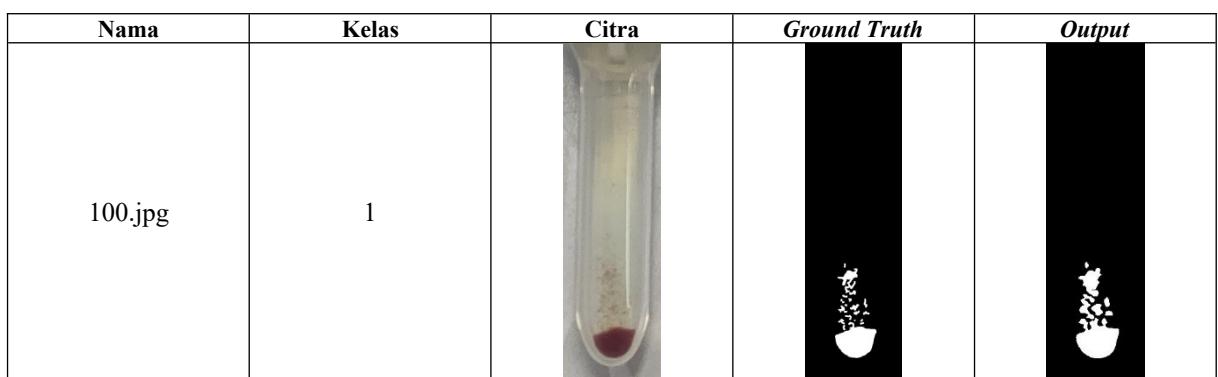
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
131.jpg	0	0.9714	0.9848	0.9862	0.9855	0.9993
132.jpg	0	0.9832	0.9944	0.9886	0.9915	0.9994
133.jpg	0	0.9774	0.9871	0.9900	0.9886	0.9995
134.jpg	0	0.9766	0.9876	0.9887	0.9881	0.9994
135.jpg	0	0.9742	0.9892	0.9846	0.9869	0.9994
136.jpg	0	0.9829	0.9889	0.9938	0.9913	0.9995
137.jpg	0	0.9855	0.9907	0.9947	0.9927	0.9994
145.jpg	0	0.9698	0.9840	0.9852	0.9846	0.9994
146.jpg	0	0.9834	0.9848	0.9985	0.9916	0.9995
147.jpg	0	0.9749	0.9809	0.9937	0.9873	0.9995

Nama	Kelas	Citra	Ground Truth	Output
131.jpg	0			
132.jpg	0			
133.jpg	0			

134.jpg	0			
135.jpg	0			
136.jpg	0			
137.jpg	0			
145.jpg	0			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
100.jpg	1	0.8963	0.9315	0.9595	0.9453	0.9939
102.jpg	1	0.9622	0.9727	0.9888	0.9807	0.9958
104.jpg	1	0.9532	0.9655	0.9867	0.9760	0.9944
105.jpg	1	0.7090	0.8320	0.8274	0.8297	0.9802
113.jpg	1	0.9152	0.9473	0.9642	0.9557	0.9904
107.jpg	1	0.9009	0.9350	0.9610	0.9478	0.9946
108.jpg	1	0.8527	0.8801	0.9648	0.9205	0.9909
109.jpg	1	0.8951	0.9167	0.9744	0.9446	0.9945
110.jpg	1	0.9135	0.9657	0.9441	0.9548	0.9954
111.jpg	1	0.8785	0.9304	0.9402	0.9353	0.9929



102.jpg	1			
104.jpg	1			
105.jpg	1			
113.jpg	1			
107.jpg	1			

108.jpg	1			
109.jpg	1			
110.jpg	1			
111.jpg	1			

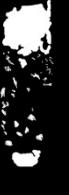
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
101.jpg	2	0.9489	0.9576	0.9905	0.9738	0.9940
103.jpg	2	0.9402	0.9566	0.9821	0.96919	0.9931
112.jpg	2	0.8993	0.9296	0.9649	0.9470	0.9857
115.jpg	2	0.9065	0.9284	0.9746	0.9509	0.9875
116.jpg	2	0.9086	0.9496	0.9546	0.9521	0.9882
117.jpg	2	0.9278	0.9706	0.9546	0.9625	0.9914
118.jpg	2	0.8818	0.9310	0.9434	0.9372	0.9870
15.jpg	2	0.8923	0.9308	0.9556	0.9431	0.9828
16.jpg	2	0.9041	0.9345	0.9653	0.9496	0.9855
18.jpg	2	0.9334	0.9459	0.9860	0.9655	0.9901

Nama	Kelas	Citra	Ground Truth	Output
101.jpg	2			
103.jpg	2			
112.jpg	2			
115.jpg	2			
116.jpg	2			

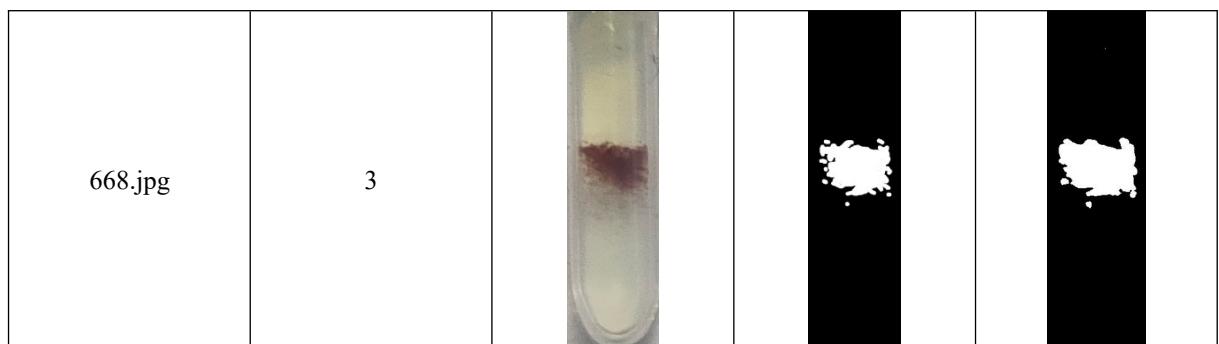
117.jpg	2			
118.jpg	2			
15.jpg	2			
16.jpg	2			
18.jpg	2			

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
------	-------	----------	---------	--------	---------	---------

13.jpg	3	0.9423	0.9777	0.9630	0.9703	0.9960
14.jpg	3	0.8787	0.9468	0.9243	0.9354	0.9849
17.jpg	3	0.8819	0.9340	0.9405	0.9372	0.9869
634.jpg	3	0.9347	0.9564	0.9762	0.9662	0.9879
642.jpg	3	0.8504	0.8690	0.9753	0.9191	0.9600
664.jpg	3	0.9034	0.9221	0.9780	0.9492	0.9929
665.jpg	3	0.9186	0.9447	0.9708	0.9576	0.9948
666.jpg	3	0.9204	0.9376	0.9804	0.9585	0.9946
667.jpg	3	0.9443	0.9747	0.9680	0.9713	0.9954
668.jpg	3	0.9459	0.9612	0.9834	0.9722	0.9941

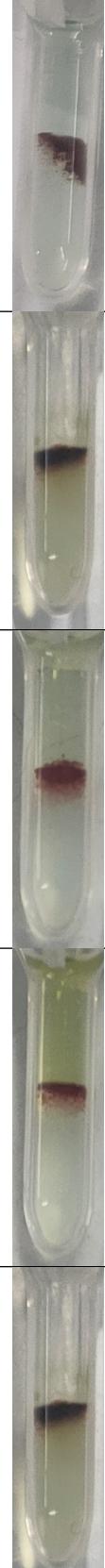
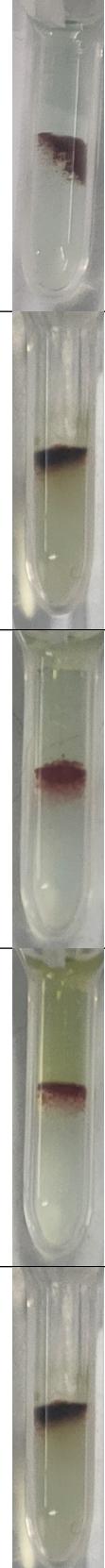
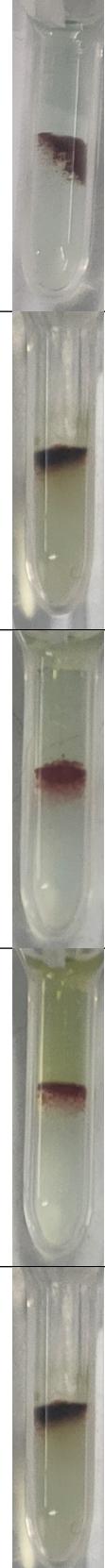
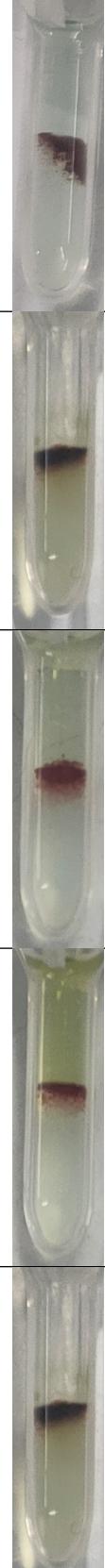
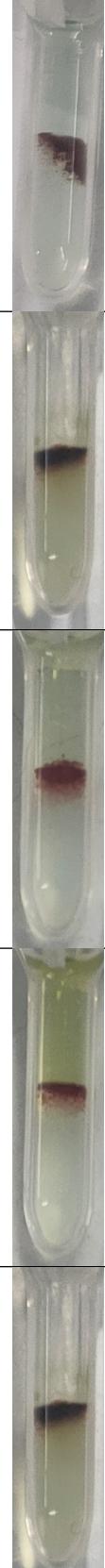
Nama	Kelas	Citra	Ground Truth	Output
13.jpg	3			
14.jpg	3			
17.jpg	3			
634.jpg	3			

642.jpg	3			
664.jpg	3			
665.jpg	3			
666.jpg	3			
667.jpg	3			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
1.jpg	4	0.9497	0.9729	0.9754	0.9742	0.9991
11.jpg	4	0.9265	0.9582	0.9654	0.9618	0.9941
12.jpg	4	0.9253	0.9532	0.9693	0.9612	0.9949
2.jpg	4	0.9206	0.9771	0.9409	0.9587	0.9964
3.jpg	4	0.9738	0.9925	0.9810	0.9867	0.9988
4.jpg	4	0.9698	0.9805	0.9888	0.9846	0.9982
5.jpg	4	0.9379	0.9531	0.9832	0.9679	0.9972
6.jpg	4	0.8831	0.9239	0.9523	0.9379	0.9947
675.jpg	4	0.8858	0.9090	0.9719	0.9394	0.9876
689.jpg	4	0.8977	0.9353	0.9571	0.9461	0.9935

Nama	Kelas	Citra	Ground Truth	Output
1.jpg	4			
11.jpg	4			

12.jpg	4			
2.jpg	4			
3.jpg	4			
4.jpg	4			
5.jpg	4			

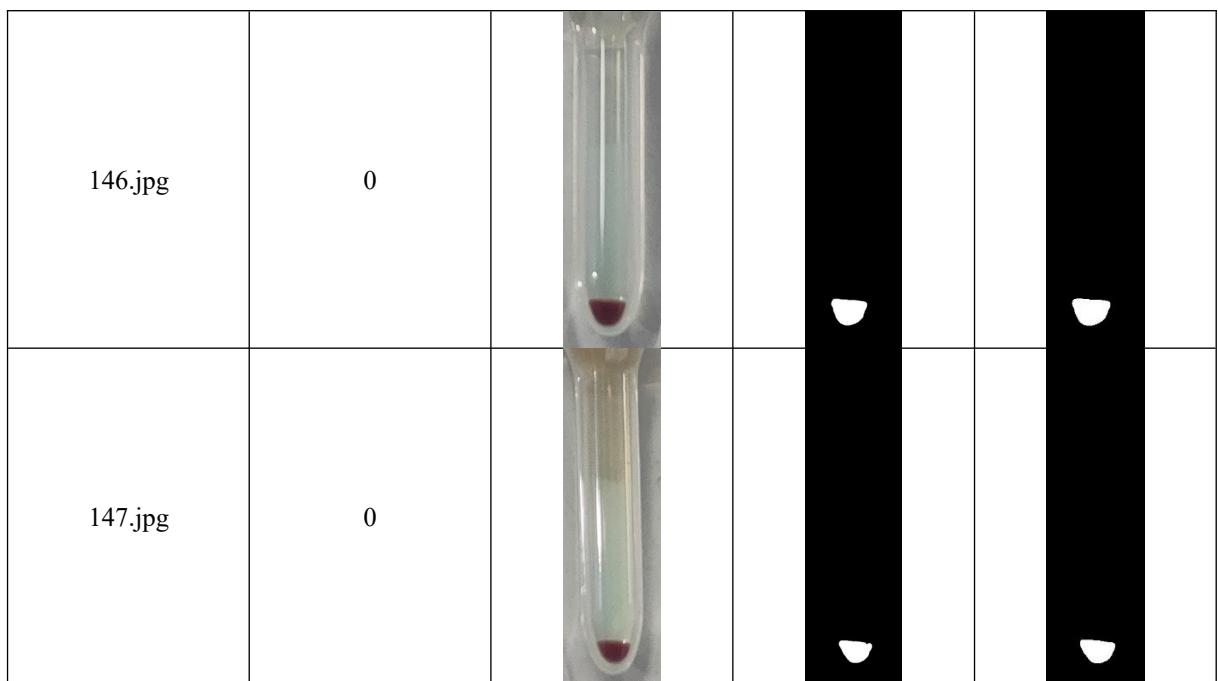
6.jpg	4					
675.jpg	4					
689.jpg	4					

L4. Hasil Model dengan Augmentasi

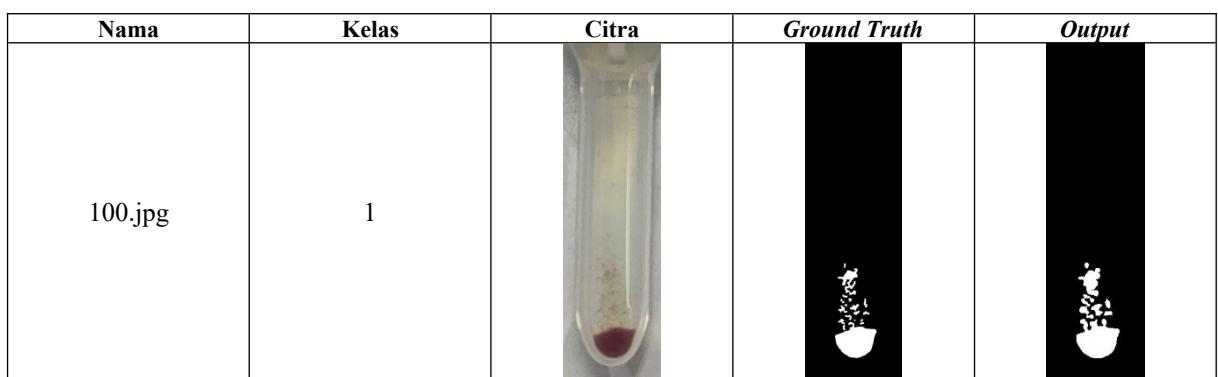
Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
131.jpg	0	0.9844	0.9926	0.9916	0.9921	0.9996
132.jpg	0	0.9886	0.9981	0.9904	0.9943	0.9996
133.jpg	0	0.9883	0.9953	0.9929	0.9941	0.9997
134.jpg	0	0.9893	0.9951	0.9940	0.9946	0.9997
135.jpg	0	0.9882	0.9917	0.996	0.9940	0.9997
136.jpg	0	0.9910	0.9926	0.9983	0.9954	0.9997
137.jpg	0	0.9910	0.9956	0.9953	0.9954	0.9996
145.jpg	0	0.9894	0.9952	0.9941	0.9946	0.9997
146.jpg	0	0.9851	0.9985	0.9865	0.9925	0.9996
147.jpg	0	0.9870	0.9913	0.9956	0.9934	0.9997

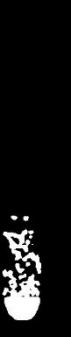
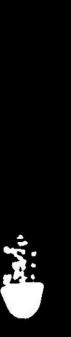
Nama	Kelas	Citra	Ground Truth	Output
131.jpg	0			
132.jpg	0			
133.jpg	0			

134.jpg	0			
135.jpg	0			
136.jpg	0			
137.jpg	0			
145.jpg	0			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
100.jpg	1	0.9194	0.9614	0.9547	0.9580	0.9954
102.jpg	1	0.9731	0.9864	0.9862	0.9863	0.9971
104.jpg	1	0.9657	0.9833	0.9818	0.9825	0.9960
105.jpg	1	0.7155	0.8427	0.8258	0.8342	0.9809
113.jpg	1	0.9303	0.9617	0.9660	0.9639	0.9922
107.jpg	1	0.9188	0.9582	0.9571	0.9577	0.9957
108.jpg	1	0.8781	0.9165	0.9544	0.9351	0.9928
109.jpg	1	0.9075	0.9393	0.9640	0.9515	0.9953
110.jpg	1	0.9334	0.9587	0.9725	0.9655	0.9964
111.jpg	1	0.9008	0.9409	0.9548	0.9478	0.9942



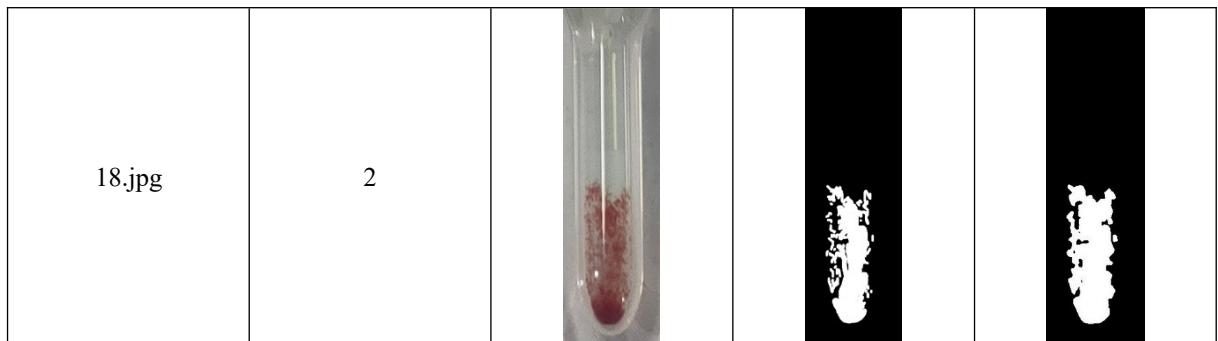
102.jpg	1			
104.jpg	1			
105.jpg	1			
113.jpg	1			
107.jpg	1			

108.jpg	1			
109.jpg	1			
110.jpg	1			
111.jpg	1			

Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
101.jpg	2	0.9633	0.9705	0.9922	0.9813	0.9957
103.jpg	2	0.9602	0.9822	0.9772	0.9797	0.9955
112.jpg	2	0.9121	0.9379	0.9707	0.9540	0.9876
115.jpg	2	0.9277	0.9602	0.9648	0.9625	0.9907
116.jpg	2	0.9204	0.9562	0.9609	0.9585	0.9898
117.jpg	2	0.9424	0.9710	0.9696	0.9703	0.9931
118.jpg	2	0.9063	0.9392	0.9627	0.9508	0.9898
15.jpg	2	0.9239	0.9522	0.9687	0.9604	0.9881
16.jpg	2	0.9175	0.9524	0.9616	0.9569	0.9877
18.jpg	2	0.9451	0.9708	0.9728	0.9718	0.9921

Nama	Kelas	Citra	Ground Truth	Output
101.jpg	2			
103.jpg	2			
112.jpg	2			
115.jpg	2			

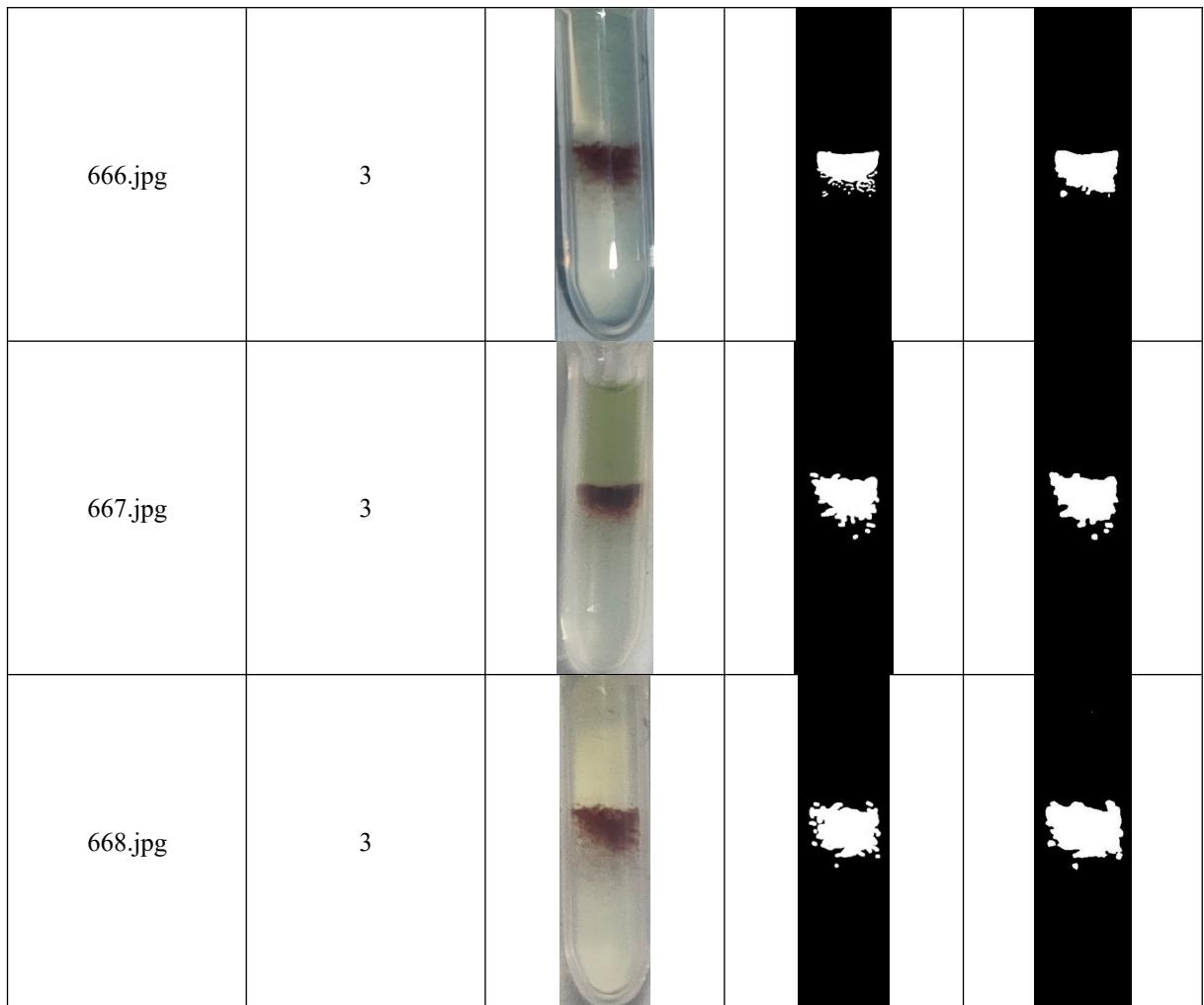
116.jpg	2			
117.jpg	2			
118.jpg	2			
15.jpg	2			
16.jpg	2			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
13.jpg	3	0.9531	0.9741	0.9778	0.9760	0.9968
14.jpg	3	0.9096	0.9387	0.9670	0.9526	0.9886
17.jpg	3	0.9153	0.9546	0.9569	0.9558	0.9908
634.jpg	3	0.9571	0.9777	0.9784	0.9781	0.9922
642.jpg	3	0.9545	0.9817	0.9718	0.9767	0.9892
664.jpg	3	0.9156	0.9402	0.9722	0.9559	0.9939
665.jpg	3	0.9340	0.9528	0.9793	0.9659	0.9958
666.jpg	3	0.9294	0.9468	0.9806	0.9634	0.9952
667.jpg	3	0.9648	0.9808	0.9834	0.9821	0.9971
668.jpg	3	0.9635	0.9727	0.9902	0.9814	0.9961

Nama	Kelas	Citra	Ground Truth	Output
13.jpg	3			
14.jpg	3			

17.jpg	3			
634.jpg	3			
642.jpg	3			
664.jpg	3			
665.jpg	3			



Nama	Kelas	Skor IoU	Presisi	Recall	Skor F1	Akurasi
1.jpg	4	0.9810	0.9849	0.9960	0.9904	0.9996
11.jpg	4	0.9413	0.9652	0.9744	0.9698	0.9953
12.jpg	4	0.9384	0.9559	0.9808	0.9682	0.9958
2.jpg	4	0.9299	0.9755	0.9520	0.9637	0.9968
3.jpg	4	0.9815	0.9866	0.9947	0.9906	0.9991
4.jpg	4	0.9787	0.9912	0.9872	0.9892	0.9987
5.jpg	4	0.9474	0.9569	0.9895	0.9729	0.9977
6.jpg	4	0.9070	0.9377	0.9652	0.9512	0.9958
675.jpg	4	0.9097	0.9449	0.9606	0.9527	0.9905
689.jpg	4	0.9666	0.9821	0.9839	0.9830	0.9979

Nama	Kelas	Citra	Ground Truth	Output
------	-------	-------	--------------	--------

1.jpg	4			
11.jpg	4			
12.jpg	4			
2.jpg	4			
3.jpg	4			

4.jpg	4			
5.jpg	4			
6.jpg	4			
675.jpg	4			
689.jpg	4			

BIODATA PENULIS



Penulis dilahirkan di Jayapura, 06 Januari 2003, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Gracia Surabaya, SD Gracia Surabaya, SMPK Petra 3 Surabaya dan SMAN 5 Surabaya. Setelah lulus dari SMAN tahun 2020, Penulis mengikuti UBTK dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2020 dan terdaftar dengan NRP 5025201210.

Di Departemen Teknik Informatika, Penulis sempat aktif di beberapa kegiatan kepanitiaan yang diselenggarakan oleh Departemen, Himpunan Mahasiswa Teknik *Computer* (HMTC). Selain kepanitiaan, Penulis juga mengikuti beberapa perlombaan *Competitive Programming* secara nasional yang dilakukan oleh ICPC. Selain itu, Penulis juga sempat mengikuti beberapa proyek yang dilakukan oleh dosen, menjalani program Studi Independen dengan mitra Zenius, dan menjalani Magang melalui program Merdeka Belajar Kampus Merdeka (MBKM) yang diselenggarakan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) selama empat bulan secara luring di GoTo Company (Gojek), Jakarta, DKI Jakarta.