

**DETEKSI DAN SEGMENTASI INSTANCE SEL DARAH PUTIH
BERBASIS MASK R-CNN PADA PASIEN ACUTE LYMPHOBLASTIC
LEUKEMIA TIPE L1**

SKRIPSI

untuk memenuhi sebagian persyaratan
untuk memperoleh derajat Sarjana S-1
Program Studi Teknik Fisika



Diajukan oleh
Mahatma Ageng Wisesa
17/411372/TK/45857

Kepada
**DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
YOGYAKARTA**
2021

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Mahatma Ageng Wisesa
NIM : 17/411372/TK/45857
Tahun terdaftar : 2017
Program Studi : Teknik Fisika
Fakultas : Teknik

menyatakan bahwa dokumen ilmiah skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 28 Desember 2021



Mahatma Ageng Wisesa

NIM. 17/411372/TK/45857

HALAMAN PENGESAHAN

SKRIPSI

DETEKSI DAN SEGMENTASI INSTANCE SEL DARAH PUTIH BERBASIS MASK R-CNN PADA PASIEN ACUTE LYMPHOBLASTIC LEUKEMIA TIPE L1

Nama Mahasiswa : Mahatma Ageng Wisesa

Nomor Mahasiswa : 17/411372/TK/45857

Pembimbing Utama : Ir. Nopriadi, S.T., M.Sc., Ph.D.

Pembimbing Pendamping : Ir. Nazrul Effendy, S.T., M.T., Ph.D., IPM.

Skripsi ini telah dipertahankan di depan Tim Penguji
pada tanggal 28 Desember 2021

Ketua Sidang : Ir. Nopriadi, S.T., M.Sc., Ph.D.

Penguji Utama : Dr. Faridah, S.T., M.Sc.

Anggota Penguji : Dr. Nur Abdillah Siddiq, S.T.

Skripsi ini telah diterima dan dinyatakan memenuhi
syarat kelulusan pada tanggal 28 Desember 2021

Ketua Departemen Teknik Nuklir dan Teknik Fisika
Fakultas Teknik UGM

Dr. Ir. Alexander Agung, S.T., M.Sc.
NIP. 19720916 199803 1002

We have created everything by measure (QS 54:49)

God does not play dice with the universe

We are all machine learning

We live in a simulation

It's not about you, it's not even about us, it's about legacy

It's about what we choose to leave behind

KATA PENGANTAR

Segala puji dan syukur kepada Allah *subhanahu wa ta'ala*, Tuhan semesta alam yang telah menciptakan segala sesuatu menurut ukuran. Tugas akhir dengan judul “**Deteksi dan Segmentasi Instance Sel Darah Putih berbasis Mask R-CNN pada Pasien Acute Lymphoblastic Leukemia tipe L1**” merupakan hasil karya seorang yang memiliki keingintahuan bagaimana alam semesta bekerja. Penulis berharap dengan disusunnya skripsi ini, penulis dapat ikut berkontribusi dalam perkembangan ilmu pengetahuan kecerdasan buatan. Selesainya penulisan Tugas Akhir ini, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu serta memberi dukungan kepada penulis dalam proses penggerjaan tugas akhir ini, khususnya:

1. Ibu dan ayah saya yang selalu memberi dukungan terbaik dalam segala hal hingga saat ini.
2. Bapak Ir. Nopriadi, S.T., M.Sc., Ph.D. dan Bapak Ir. Nazrul Effendy, S.T., M.T., Ph.D., IPM. selaku dosen pembimbing Tugas Akhir yang telah memberikan masukan, bimbingan, dan nasihat lainnya.
3. Sahabat-sahabat semasa kuliah Nico, Evy, dan Hasna yang telah berbagi tawa dan cerita bersama.
4. Teman-teman yang telah 3,5 tahun kuliah bersama, berbagi ilmu bersama, kemudian berpisah sendiri-sendiri.
5. Orang-orang yang secara langsung ataupun tidak langsung memberikan kontribusi pada skripsi ini, baik berupa membantu dalam penggerjaan teknis maupun memotivasi penulis untuk segera menyelesaikan.

Semoga hasil penelitian ini dapat bermanfaat bagi semua pihak dan dapat diimplementasikan pada bidang instrumentasi biomedis.

Yogyakarta, 16 November 2021



Mahatma Ageng Wisesa

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN BEBAS PLAGIASI.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMBANG DAN SINGKATAN	xviii
INTISARI.....	xxi
ABSTRACT.....	xxii
BAB I PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Perumusan Masalah	3
I.2.1. Batasan Masalah.....	3
I.3. Tujuan Penelitian	4
I.4. Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA.....	5
II.1. Klasifikasi WBC pada Kasus <i>Acute Lymphoblastic Leukemia</i>	5
II.2. Pendekripsi WBC pada Kasus <i>Acute Lymphoblastic Leukemia</i>	11
II.3. Perhitungan Sel Darah Otomatis dengan basis Pendekripsi, Segmentasi <i>Instance</i> , dan Pengolahan Citra	28
BAB III DASAR TEORI	43
III.1. Sel Darah	43
III.1.1. Hemopoiesis.....	44
III.1.2. Morfologi WBC.....	45
III.1.3. <i>Acute Lymphoblastic Leukemia</i> Tipe L1	50
III.2. <i>Deep Learning</i>	51
III.2.1. <i>Convolutional Neural Networks</i>	53
III.2.2. <i>Transfer Learning</i>	57
III.2.3. Distribusi Data	58

III.2.4. Augmentasi Data.....	60
III.3. <i>Computer Vision</i>	60
III.3.1. Klasifikasi dan Lokalisasi Objek (<i>Object Detection</i>)	61
III.3.2. Segmentasi Citra	62
III.3.3. Pengolahan Citra.....	64
III.4. <i>Mask Regional-Convolutional Neural Network</i>	68
III.4.1. <i>Backbone Network Architecture</i>	69
III.4.2. <i>Region Proposal Networks</i> (RPN).....	72
III.4.3. <i>Anchor Boxes</i>	73
III.4.4. <i>Intersection over Union</i> (IoU)	74
III.4.5. <i>Non-maximum Suppression</i> (NMS)	74
III.4.6. <i>Loss Function</i>	75
III.5. <i>Hyperparameter</i>	78
III.5.1. <i>Epoch</i> , <i>Batch Size</i> , dan <i>Step</i>	78
III.5.2. <i>Learning Rate</i> dan <i>Learning Momentum</i>	78
III.6. Evaluasi Performa	81
III.6.1. Akurasi.....	82
III.6.2. Presisi.....	82
III.6.3. <i>Recall</i>	83
III.6.4. <i>F1-Score</i>	83
III.6.5. <i>Average Precision</i> (AP) dan <i>mean Average Precision</i> (mAP)	83
III.6.6. <i>Mean Absolute Error</i> (MAE).....	85
III.6.7. <i>Overfitting</i> dan <i>Underfitting</i>	85
BAB IV PELAKSANAAN PENELITIAN	88
IV.1. Alat dan Bahan Penelitian	88
IV.1.1. Alat Penelitian	88
IV.1.2. Bahan Penelitian	89
IV.2. Tata Laksana Penelitian	90
IV.2.1. Studi Literatur.....	91
IV.2.2. Eksplorasi Data.....	91
IV.2.3. Persiapan Data	92
IV.2.4. Pelatihan Algoritma.....	102

IV.2.5. Pengujian Algoritma Deteksi.....	103
IV.2.6. Perhitungan Prediksi Objek Terdeteksi	104
IV.3. Analisis Hasil Penelitian	105
IV.4. Penulisan Laporan Penelitian	105
BAB V HASIL DAN PEMBAHASAN.....	106
V.1. Pelatihan Algoritma.....	106
V.1.1. Penentuan Nilai <i>Learning Rate</i>	106
V.1.2. Penentuan Nilai <i>Steps per Epoch</i>	108
V.1.3. Penentuan Nilai <i>Validation Steps</i>	111
V.1.4. Penentuan <i>Detection Minimum Confidence</i>	113
V.1.5. Evaluasi Performa <i>Hyperparameter</i> Terpilih	116
V.1.6. Evaluasi Citra Hasil Prediksi	119
V.2. Pengujian Algoritma Deteksi.....	123
V.2.1. Penentuan <i>Detection NMS Threshold</i> dan <i>RPN NMS Threshold</i>	124
V.2.2. Evaluasi Performa Alogritma Deteksi	125
V.3. Evaluasi Perhitungan Otomatis Sel Limfoblas dan Non-Limfoblas	131
V.4. Perbandingan Hasil Deteksi WBC Metode <i>Mask R-CNN</i> dengan Penelitian-Penelitian Sebelumnya pada Pasien ALL Tipe L1	132
BAB VI KESIMPULAN DAN SARAN	135
VI.1. Kesimpulan.....	135
VI.2. Saran.....	135
DAFTAR PUSTAKA	136
LAMPIRAN	142
LAMPIRAN A	143
LAMPIRAN B	150
LAMPIRAN C	157
LAMPIRAN D	158

DAFTAR TABEL

Tabel 2.1. Perbandingan kinerja dalam <i>dataset ALL-IDB1</i> dengan 70% data pelatihan dan 30% pengujian (rata-rata <i>10-Fold cross validation</i>).....	7
Tabel 2.2. Perbandingan kinerja dalam <i>dataset ALL-IDB2</i> dengan 70% data pelatihan dan 30% pengujian (rata-rata <i>10-Fold cross validation</i>).....	7
Tabel 2.3. Perbandingan jumlah fitur dan kinerja untuk kedua <i>dataset</i>	10
Tabel 2.4. Hasil pada <i>test-set</i> selama 24 <i>training runs</i> menggunakan model terbaik dari setiap <i>run</i> yang diukur dengan <i>F1-score</i>	11
Tabel 2.5. Perbandingan penelitian lain yang diusulkan peneliti untuk <i>dataset ALL-IDB1</i>	14
Tabel 2.6. Perbandingan penelitian lain dengan metode yang diusulkan peneliti untuk <i>dataset C-NMC 2019</i>	15
Tabel 2.7. Distribusi data limfoblas dan non-limfoblas.	19
Tabel 2.8. Distribusi data limfoblas dan non-limfoblas pasca <i>oversampling</i> dan <i>undersampling</i>	19
Tabel 2.9. Performa model terbaik peneliti dengan <i>epoch</i> 400.....	23
Tabel 2.10. Evaluasi MAE dengan perhitungan manual.	24
Tabel 2.11. Distribusi data pasca <i>oversampling</i> dan <i>undersampling</i>	25
Tabel 2.12. <i>Confusion matrix</i> ini hanya menargetkan sel yang terinfeksi dalam 60 citra.....	30
Tabel 2.13. Hasil Algoritma Perhitungan Sel.....	31
Tabel 2.14. Perbandingan setiap tahap dengan <i>Faster R-CNN</i>	32
Tabel 2.15. Daftar fitur yang digunakan dalam XGBoost <i>classifier</i>	38
Tabel 2.16. Perbandingan kinerja klasifikasi pada RBC individu menggunakan <i>Mask R-CNN</i> , XGBoost dengan delineasi manual, dan XGBoost dengan hasil segmentasi <i>Mask R-CNN</i>	38
Tabel 2.17. Hasil perhitungan metrik	42
Tabel 2.18. Rangkuman tinjauan pustaka.....	39
Tabel 3.1. Fitur struktural granulosit dan agranulosit.	46
Tabel 3.2. Morfologi WBC dari sel punca myeloid.	47
Tabel 3.3. Morfologi WBC dari sel punca limfoid.	49
Tabel 3.4. Ringkasan karakteristik ALL L1.....	50
Tabel 3.5. Perbedaan klasifikasi, lokalisasi, dan deteksi objek.....	61
Tabel 4.1. Jenis WBC dan label pada citra.....	91
Tabel 5.1. Hasil pelatihan dengan variasi nilai <i>steps per epoch</i>	108
Tabel 5.2. Hasil pelatihan dengan variasi nilai <i>validation steps</i>	112
Tabel 5.3. Performa hasil latih pada citra mentah.	113
Tabel 5.4. Performa hasil latih pada citra disegmentasi.	114
Tabel 5.5. Nilai <i>loss</i> akhir pada pelatihan.	114
Tabel 5.6. Rangkuman <i>hyperparameter</i> terpilih.	116
Tabel 5.7. Hasil deteksi pada data validasi.....	116

Tabel 5.8. Hasil performa pada data validasi.....	116
Tabel 5.9. Hasil deteksi pada data uji.....	117
Tabel 5.10. Rangkuman performa pada data uji.....	118
Tabel 5.11. Hasil deteksi pada data validasi.....	127
Tabel 5.12. Hasil performa pada data validasi.....	128
Tabel 5.13. Hasil deteksi pada data uji.....	128
Tabel 5.14. Hasil performa pada data uji.....	129
Tabel 5.15. Rangkuman performa model terbaik pada data validasi.....	130
Tabel 5.16. Rangkuman performa model terbaik pada data uji.....	130
Tabel 5.17. Waktu pelatihan yang dibutuhkan untuk kedua model terbaik.....	131
Tabel 5.18. Evaluasi MAE pada model terbaik.....	132
Tabel 5.19. Perbandingan hasil penelitian menggunakan metode <i>Mask R-CNN</i> dengan penelitian yang telah dilakukan oleh Tim Riset Studio Sensor Visual DTNTF UGM.....	134

DAFTAR GAMBAR

Gambar 2.1. Arsitektur deteksi ALL berbasis CNN yang diusulkan peneliti [22].	6
Gambar 2.2. Alur kerja dari pendekatan yang diusulkan peneliti [21].....	9
Gambar 2.3. Kinerja pendekatan VGGNet dan SESSA <i>hybrid</i> yang diusulkan pada <i>dataset</i> ALL-IDB [21]. (a) kinerja rata-rata selama 10 <i>runs</i> , (b) akurasi untuk 10 <i>runs</i> terbaik dan terburuk.	9
Gambar 2.4. Contoh gambar yang diambil dari <i>training-set</i> [23]. (a) Sel ALL, (b) Sel Normal, (c) ALL sel dengan bagian sel terputus karena segmentasi yang tidak sempurna, (d) Sel normal dengan latar belakang yang berlebihan karena segmentasi yang tidak sempurna.....	11
Gambar 2.5. Alur kerja <i>screening</i> ALL peneliti [24].....	12
Gambar 2.6. Citra sel limfoblas dan augmentasi untuk <i>dataset</i> ALL-IDB1 [24].	12
Gambar 2.7. Citra sel limfoblas dan augmentasi untuk C-NMC 2019 [24].....	12
Gambar 2.8. Prediksi di YOLOv4 untuk <i>dataset</i> ALL-IDB1 [24].....	13
Gambar 2.9. Prediksi pada YOLOv4 untuk <i>dataset</i> C-NMC 2019 [24].	14
Gambar 2.10. Ilustrasi <i>enhancement</i> citra ALL [25].....	16
Gambar 2.11. Proses segmentasi [25].	17
Gambar 2.12. Ilustrasi proses segmentasi [25].....	17
Gambar 2.13. Ilustrasi hasil segmentasi [25].	17
Gambar 2.14. Proses ekstraksi fitur [25].	18
Gambar 2.15. Normalisasi dan pengukuran akurasi [25].	18
Gambar 2.16. Distribusi awal data [26].....	19
Gambar 2.17. Alur pelatihan algoritma deteksi peneliti [26].....	20
Gambar 2.18. Alur pengujian algoritma deteksi peneliti [26].....	21
Gambar 2.19. Kurva <i>loss function</i> data latih dan data validasi ResNet50 [26]... ..	22
Gambar 2.20. Kurva <i>loss function</i> data latih dan data validasi ResNet101 [26]. ..	22
Gambar 2.21. Evaluasi performa model keseluruhan peneliti [26].....	23
Gambar 2.22. Distribusi data WBC [16].	24
Gambar 2.23. Distribusi data sel limfoblas dan non-limfoblas [16].	25
Gambar 2.24. Kurva <i>loss function</i> pada pelatihan <i>Tiny YOLOv2</i> [16].....	26
Gambar 2.25. Kurva <i>loss function</i> pada pelatihan YOLOv2 [16].....	26
Gambar 2.26. Evaluasi performa seluruh model pelatihan [16].....	27
Gambar 2.27. Diagram pohon diturunkan dari Tabel 2.11. Jumlah normosit sebenarnya dihilangkan karena bukan salah satu dari <i>dataset</i> berlabel [27].	30
Gambar 2.28. Bagan batang dari <i>true results</i> vs. persentase <i>predicted results</i> [27].	31

Gambar 2.29. Citra secara otomatis dianalisis oleh model <i>Mask R-CNN</i> [27]. (A) Citra asli diambil dengan objektif imersi $100\times$, (B) Klasifikasi <i>bounding box</i> dan probabilitas kelas untuk semua retikulosit dan iRBC yang tidak terinfeksi yang diekstraksi, (C-F) <i>Mask</i> selektif untuk tahap ekstraksi dengan urutan sebagai berikut: Retikulosit, Cincin, Trofozoit, dan Skizon yang tidak terinfeksi.	33
Gambar 2.30. Kerangka kerja segmentasi dan penghitung sel yang diusulkan. <i>Dataset</i> citra mikroskopis diberi label dan dimasukkan ke dalam model <i>deep learning</i> yang melakukan segmentasi sel, mengklasifikasikan ke dalam WBC dan RBC, dan menghitung jumlahnya masing-masing [13].....	34
Gambar 2.31. Evolusi fungsi <i>loss</i> [13].	35
Gambar 2.32. Evolusi citra apusan darah selama proses segmentasi dan klasifikasi [13].....	36
Gambar 2.33. Contoh hasil delineasi dan deteksi dengan model <i>Mask R-CNN</i> . Dua <i>slide</i> dengan hRBC dan tRBC ditunjukkan pada (a) dan (b). Garis biru adalah kontur penggambaran manual sel darah merah sebagai <i>ground truth</i> . Garis hijau dan merah menandai kontur yang diprediksi mode dari sel darah merah yang diklasifikasikan sebagai hRBC dan tRBC, masing-masing. Kasus yang salah diklasifikasikan dalam (a) dan (b) diperbesar dan ditunjukkan dalam (c) dan (d). Kontur yang diprediksi model cocok dengan kontur penggambaran manual, menunjukkan bahwa kesalahan klasifikasi bukan karena segmentasi yang gagal [35].	39
Gambar 2.34. Metodologi yang diusulkan peneliti [36].....	40
Gambar 2.35. Beberapa gambar sampel (kiri) yang digunakan dalam eksperimen dan gambar keluaran masing-masing (kanan) [36].	41
Gambar 2.36. Perbandingan perhitungan manual dan otomatis [36].	41
Gambar 3.1. Komposisi darah lengkap [36].....	43
Gambar 3.2. Penyiapan apusan darah [1].	44
Gambar 3.3. Tahap asal dan diferensiasi sel darah [1].	45
Gambar 3.4. Leukosit: jenis dan fitur struktural [1].	46
Gambar 3.5. Representasi mendalam dipelajari oleh model klasifikasi digit [47].	51
Gambar 3.6. Jaringan saraf diparameterisasi oleh nilai bobot dan bias-nya [47].	52
Gambar 3.7. <i>Loss function</i> mengukur kualitas <i>output</i> jaringan dan digunakan sebagai sinyal umpan balik untuk menyesuaikan nilai bobot [47].	53
Gambar 3.8. Citra visual membentuk <i>spatial hierarchies: edges</i> hiperlokal bergabung menjadi objek lokal seperti mata atau telinga yang digabungkan menjadi konsep tingkat tinggi seperti "cat" [47].....	54
Gambar 3.9. Operasi konvolusi pada citra dengan kernel/filter 3×3 [48].	55
Gambar 3.10. <i>Feature map</i> yang dihasilkan oleh <i>single filter</i> 3×3 [47].	55
Gambar 3.11. <i>Max pooling</i> dengan filter 2×2 dan <i>stride</i> bernilai 2 [48].....	56
Gambar 3.12. Contoh arsitektur CNN yang merupakan gabungan dari <i>Convolution Layer</i> , <i>Pooling Layer</i> , dan <i>FC Layer</i> [49].	57
Gambar 3.13. Contoh kasus <i>transfer learning</i>	58

Gambar 3.14. <i>Undersampling</i> dan <i>oversampling</i> [50].	60
Gambar 3.15. Contoh hasil segmentasi semantik (kiri) dan segmentasi <i>instance</i> (kanan) [54].....	62
Gambar 3.16. Arsitektur dasar segmentasi citra [55].	63
Gambar 3.17. Penambahan cabang pada <i>Faster R-CNN</i> untuk menghasilkan segmentasi <i>instance</i> [24].	64
Gambar 3.18. Hasil proses filter bilateral dengan berbagai nilai parameter rentang dan domain [58].	66
Gambar 3.19. Kiri: Masukkan citra <i>RagingBull</i> dengan histogram. Kanan: Gambar yang sama setelah pemerataan histogram [53].....	68
Gambar 3.20. Kerangka kerja <i>Mask R-CNN</i> untuk segmentasi <i>instance</i> [20]....	69
Gambar 3.21. Arsitektur <i>Mask R-CNN</i> [60].	69
Gambar 3.22. Error pelatihan (kiri) dan <i>error uji</i> (kanan) dengan 20-layer and 56-layer “plain” networks [19].	70
Gambar 3.23. Residual learning: a building block [19].....	71
Gambar 3.24. Plain networks dengan 18 dan 34 layers; ResNet dengan 18 dan 34 layers [19].	72
Gambar 3.25. Region Proposal Network (RPN) [34].....	73
Gambar 3.26. Intersection over union [48].	74
Gambar 3.27. Contoh hasil non-max suppression [61].	75
Gambar 3.28. Stochastic Gradient Descent 1D (satu parameter <i>learnable</i>) [47].	79
Gambar 3.29. Minimum lokal dan minimum global [47].	80
Gambar 3.30. Efek dari nilai <i>learning rate</i> [63].....	80
Gambar 3.31. Contoh hasil prediksi <i>confidence score</i> dan <i>bounding box</i> [64]... ..	82
Gambar 3.32. Kurva presisi \times recall [65].....	85
Gambar 3.33. Contoh <i>loss function</i> dengan <i>overfitting</i> [47].	86
Gambar 3.34. Contoh <i>loss function</i> tanpa <i>overfitting</i> [47].	87
Gambar 4.1. Ilustrasi akuisisi data.....	90
Gambar 4.2. Pola pergerakan preparat.	90
Gambar 4.3. Diagram alir tata laksana penelitian.	90
Gambar 4.4. Diagram alir persiapan data.	92
Gambar 4.5. Distribusi data WBC.....	94
Gambar 4.6. Distribusi data sel limfoblas dan non-limfoblas.	94
Gambar 4.7. Distribusi data latih.....	95
Gambar 4.8. Distribusi data validasi.	96
Gambar 4.9. Distribusi data latih biner (limfoblas dan non-limfoblas).	96
Gambar 4.10. Distribusi data validasi biner (limfoblas dan non-limfoblas).	97
Gambar 4.11. Distribusi data uji biner (limfoblas dan non-limfoblas).	97
Gambar 4.12. Ilustrasi proses segmentasi sel darah dari citra mentah.	98
Gambar 4.13. Proses anotasi pada citra mentah.	100
Gambar 4.14. Proses anotasi pada citra yang telah disegmentasi.	100
Gambar 4.15. Augmentasi citra.....	101
Gambar 4.16. Diagram alir proses pelatihan model <i>Mask R-CNN</i>	102

Gambar 4.17. Diagram alir pengujian algoritma deteksi.....	104
Gambar 4.18. Diagram alir perbandingan perhitungan manual dengan hasil prediksi model.....	105
Gambar 5.1. Kurva <i>loss function</i> pada 60 <i>epochs</i> dengan <i>learning rate</i> sebesar 0,0001.....	107
Gambar 5.2. Kurva <i>loss function</i> pada 60 <i>epochs</i> dengan <i>learning rate</i> sebesar 0,001.....	107
Gambar 5.3. Kurva <i>loss function</i> pada <i>steps per epoch</i> 75	109
Gambar 5.4. Kurva <i>loss function</i> pada <i>steps per epoch</i> 100.	109
Gambar 5.5. Kurva <i>loss function</i> pada <i>steps per epoch</i> 125.	110
Gambar 5.6. Kurva <i>loss function</i> pada <i>steps per epoch</i> 150.	110
Gambar 5.7. Kurva <i>loss function</i> pada <i>validation steps</i> 150.	111
Gambar 5.8. Kurva <i>loss function</i> pada <i>validation steps</i> 250.	112
Gambar 5.9. Kurva <i>loss function</i> pada citra mentah dengan nilai <i>detection minimum confidence</i> 0,70.....	115
Gambar 5.10. Kurva <i>loss function</i> pada citra disegmentasi dengan nilai <i>detection minimum confidence</i> 0,85.....	115
Gambar 5.11. Evaluasi performa data validasi dengan <i>hyperparameter</i> terbaik.	117
Gambar 5.12. Evaluasi performa data uji dengan <i>hyperparameter</i> terbaik.....	118
Gambar 5.13. Perbedaan hasil <i>confidence score</i> pada citra mentah dan citra disegmentasi. (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: non-limfoblas 88,7%, biru: limfoblas 65,2%, hijau: non-limfoblas 99,8%), (D) Citra disegmentasi setelah prediksi (biru: non-limfoblas 99,8%, merah: limfoblas 73,7%, hijau: non-limfoblas 99,8%)......	119
Gambar 5.14. Hasil prediksi citra mentah terdapat <i>false negative</i> (lingkaran kuning bagian C). (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (biru: limfoblas 81,7%, merah: limfoblas 91,3%, hijau: limfoblas 76,4%), (D) Citra disegmentasi setelah prediksi (ungu: limfoblas 99,4%, merah: limfoblas 99,5%, kuning: limfoblas 65,9%, biru muda: limfoblas 91,5%).	120
Gambar 5.15. Hasil prediksi citra mentah berupa <i>false positive</i> . (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: limfoblas 60,2%), (D) Citra disegmentasi setelah prediksi (merah: non-limfoblas 96,8%).	121
Gambar 5.16. Hasil prediksi citra mentah dan disegmentasi memberikan <i>false positive</i> karena terdapat objek menyerupai WBC. (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: non-limfoblas 95,4%, biru muda: non-limfoblas 94,2%), (D) Citra disegmentasi setelah prediksi (biru muda: non-limfoblas 100%, merah: limfoblas 64,5%).	122

Gambar 5.17. Nilai <i>threshold</i> menentukan jumlah <i>bounding box</i> yang dihasilkan untuk setiap objek. (A) RPN NMS bernilai 0,7 dan <i>detection</i> NMS bernilai 0,2 (warna sel terdeteksi tumpang-tindih), (B) RPN NMS bernilai 0,2 dan <i>detection</i> NMS bernilai 0,2 (hijau: limfoblas 98,4%, biru: limfoblas 80,4%, merah: non-limfoblas 99,2%).....	124
Gambar 5.18. Nilai <i>threshold</i> yang terlalu kecil menghilangkan <i>bounding box</i> untuk objek yang berdempetan. (A) RPN NMS bernilai 0,2 dan <i>detection</i> NMS bernilai 0,2 (biru muda: limfoblas 80,4%, merah: non-limfoblas 90,5%), (B) RPN NMS bernilai 0,25 dan <i>detection</i> NMS bernilai 0,25 (hijau: limfoblas 98,5%, ungu: limfoblas 80,4%, merah: non-limfoblas 90,5%).....	125
Gambar 5.19. Hasil prediksi citra mentah pada data uji. (A) RPN NMS bernilai 0,2 dan <i>detection</i> NMS bernilai 0,2 terdapat <i>false negative</i> (biru: limfoblas 81,7%, merah: limfoblas 91,3%, hijau: limfoblas 76,4%), (B) RPN NMS bernilai 0,25 dan <i>detection</i> NMS bernilai 0,25 terdapat <i>false positive</i> (merah: limfoblas 81,7%, biru muda: limfoblas 91,3%, ungu: non-limfoblas 80,9%, kuning: limfoblas 76,4%).	126
Gambar 5.20. Hasil prediksi citra mentah pada data uji. (A) RPN NMS bernilai 0,2 dan <i>detection</i> NMS bernilai 0,2 memberikan satu <i>bounding box</i> (<i>mask</i> berwarna biru muda) pada sel di atas (biru muda: limfoblas 79,6%, merah: limfoblas 99,2%), (B) RPN NMS bernilai 0,25 dan <i>detection</i> NMS bernilai 0,25 memberikan dua <i>bounding box</i> (<i>mask</i> berwarna biru dan merah) pada sel di atas (biru: limfoblas 61,9%, merah: limfoblas 89,2%, hijau: limfoblas 99,2%).....	127
Gambar 5.21. Perbandingan performa data validasi pada NMS sebesar 0,25 dan 0,20.....	128
Gambar 5.22. Perbandingan performa data uji pada NMS sebesar 0,25 dan 0,20.	129

DAFTAR LAMBANG DAN SINGKATAN

Lambang Romawi

<i>Lambang</i>	<i>Kuantitas</i>	<i>Satuan</i>
x	Koordinat piksel terhadap sumbu x	-
y	Koordinat piksel terhadap sumbu y	-

Lambang Yunani

<i>Lambang</i>	<i>Kuantitas</i>	<i>Satuan</i>
σ	Parameter <i>smoothing</i>	-
λ	Parameter penyeimbang	-

Subskrip

<i>Lambang</i>	<i>Deskripsi</i>
i	Indeks <i>anchor</i> pada mini-batch
cls	Klasifikasi
reg	Regresi
a	<i>Anchor box</i>
w	Lebar kotak prediksi

Superskrip

<i>Lambang</i>	<i>Deskripsi</i>
*	<i>ground truth</i>
j	Indeks koordinat

Singkatan

WBC	<i>White Blood Cell</i>
RBC	<i>Red Blood Cell</i>
PLT	<i>Platelets</i>
FAB	<i>French-American-British</i>
ALL	<i>Acute Lymphoblastic Leukemia</i>
AI	<i>Artificial Intelligence</i>
R-CNN	<i>Region-based Convolutional Neural Network</i>
ANN	<i>Artificial Neural Network</i>
FC	<i>Fully Connected</i>
FCN	<i>Fully Convolutional Network</i>
RoI	<i>Region of Interest</i>
ResNet	<i>Residual Neural Network</i>
FPN	<i>Feature Pyramid Network</i>
RPN	<i>Region Proposal Networks</i>
IoU	<i>Intersection over Union</i>
NMS	<i>Non-Maximum Suppression</i>
TP	<i>True Positive</i>
TN	<i>True Negative</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>

AP	<i>Average Precision</i>
mAP	<i>Mean Average Precision</i>
AUC	<i>Area Under the Curve</i>
MAE	<i>Mean Absolute Error</i>

**DETEKSI DAN SEGMENTASI INSTANCE SEL DARAH PUTIH
BERBASIS MASK R-CNN PADA PASIEN ACUTE LYMPHOBLASTIC
LEUKEMIA TIPE L1**

Mahatma Ageng Wisesa
17/411372/TK/45857

Diajukan kepada Departemen Teknik Nuklir dan Teknik Fisika Fakultas Teknik
Universitas Gadjah Mada pada tanggal 28 Desember 2021
untuk memenuhi sebagian persyaratan untuk memperoleh derajat
Sarjana Program Studi Teknik Fisika

INTISARI

Acute Lymphoblastic Leukemia adalah kanker paling umum untuk anak-anak dan penyebab utama kematian pada anak-anak. Sebanyak 70% pasien penderita ALL termasuk dalam kategori L1. Penyakit tersebut ditandai dengan produksi sel darah putih muda (sel limfoblas) lebih dari 20% dari total sel darah putih di sumsum tulang belakang. Respon cepat dan diagnosis dini sangat dibutuhkan untuk rehabilitasi pasien, terutama pada anak-anak. Metode tradisional untuk perhitungan sel darah bergantung pada bantuan manusia, di mana perhitungan dan klasifikasi sel dilakukan secara manual melalui mikroskop. Oleh karena itu, diperlukan alternatif untuk menggantikan metode tradisional dengan sistem otomatis berbasis *deep learning* yang telah berkembang pesat beberapa tahun terakhir agar menggantikan perhitungan dan klasifikasi sel secara manual.

Pada penelitian ini digunakan model *deep learning* bernama *Mask R-CNN* yang melakukan klasifikasi, lokalisasi, dan menghasilkan *segmentation mask* untuk mempermudah visualisasi sel terdeteksi. *Mask R-CNN* dilatih pada sel limfoblas dan non-limfoblas. Teknik pengolahan citra seperti filter bilateral, *thresholding*, dan pemerataan histogram diterapkan pada penelitian ini untuk meningkatkan performa perhitungan otomatis dan mengurangi waktu pelatihan. Penelitian ini telah berhasil mengimplementasikan *Mask R-CNN* dengan akurasi sebesar 91,59%, *F1-Score* sebesar 95,61%, dan mAP sebesar 92,75%. Hasil implementasi pada penelitian ini memiliki potensi untuk mengurangi kesalahan pemeriksaan manual dan menghemat banyak sumber daya manusia, waktu, dan biaya.

Kata kunci: *computer vision*, pendekslan objek, *deep learning*, *Mask R-CNN*, pengolahan citra, *acute lymphoblastic leukemia*

Pembimbing Utama : Ir. Nopriadi, S.T., M.Sc., Ph.D.

Pembimbing Pendamping : Ir. Nazrul Effendy, S.T., M.T., Ph.D., IPM.

**DETECTION AND INSTANCE SEGMENTATION OF WHITE BLOOD
CELLS BASED ON MASK R-CNN IN ACUTE LYMPHOBLASTIC
LEUKEMIA L1 PATIENT**

Mahatma Ageng Wisesa
17/411372/TK/45857

Submitted to the Departement of Nuclear Engineering and Engineering Physics
Faculty of Engineering Universitas Gadjah Mada on December 28, 2021
in partial fulfillment of the requirement for the Degree of
Bachelor of Engineering in Engineering Physics

ABSTRACT

Acute Lymphoblastic Leukemia is the most common cancer in children and the leading cause of death among children. As much as 70% of patients with ALL are included in the L1 category. The disease is characterized by the production of young white blood cells (lymphoblast cells) more than 20% of the total white blood cells in the bone marrow. Rapid response and early diagnosis are highly needed for the rehabilitation of patients, especially children. The traditional method of human-assisted blood cell counting, in which counting and classification is carried out manually through a microscope. Therefore, an alternative is needed to replace traditional method with deep learning-based automated systems that have developed rapidly in recent years to replace manual cell counting and classification.

In this research, a deep learning model called Mask R-CNN is used which performs classification, localization, and generates mask segmentation to facilitate easy visualization of the detected cells. The Mask R-CNN were trained on lymphoblast and non-lymphoblast cells. Image processing techniques such as bilateral filtering, thresholding, and histogram equalization were applied in this work to improve the performance of automatic counting and reduce training time. This research has successfully implemented Mask R-CNN with accuracy, F1-Score, and mAP of 91.59%, 95.61%, and 92.75%. Results in this research have the potential to reduce manual microscopic examination errors and save a lot of human resources, time, and costs.

Keywords: computer vision, object detection, deep learning, Mask R-CNN, image processing, acute lymphoblastic leukemia

Supervisor : Ir. Nopriadi, S.T., M.Sc., Ph.D..

Co-supervisor : Ir. Nazrul Effendy, S.T., M.T., Ph.D., IPM.

BAB I

PENDAHULUAN

I.1. Latar Belakang

Darah merupakan sarana pendistribusian, pengangkut O₂, CO₂, metabolit, hormon, dan zat-zat lain ke sel-sel di seluruh tubuh [1]. Darah mengandung tiga jenis sel: Sel Darah Merah (RBC), Trombosit (PLT), dan Sel Darah Putih (WBC) [2]. Sel Darah Merah (RBC) penting untuk transportasi oksigen dari jantung ke semua jaringan dan membawa karbon dioksida [3]. Sel Darah Putih (WBC) juga memiliki fungsi penting untuk sistem kekebalan tubuh karena merupakan pertahanan utama tubuh terhadap infeksi dan penyakit [4]. Leukemia adalah jenis kanker darah atau sumsum tulang yang ditandai dengan peningkatan dramatis yang tidak teratur dalam jumlah WBC yang belum matang yang disebut "sel limfoblas" [5].

Leukemia diklasifikasikan menjadi leukemia akut dengan kemampuan progresif cepat dan leukemia kronis yang berlangsung lambat [6]. Leukemia tipe akut diklasifikasikan menjadi dua kelas berdasarkan model *French-American-British* (FAB) yang merupakan model klasifikasi leukemia yang paling umum, yaitu *Acute Myeloid leukemia* (AML) dan *Acute Lymphoblastic Leukemia* (ALL) [5,7,8]. *Acute Lymphoblastic Leukemia* adalah kanker paling umum untuk anak-anak [5]. Penyakit kanker adalah penyakit kedua yang paling mematikan pada anak-anak, dan *Acute Lymphoblastic Leukemia* adalah penyebab utama kematian pada anak-anak [9].

French-American-British (FAB) mengklasifikasi ALL berdasarkan ukuran sel, sitoplasma, nukleous, vakuola, dan basofilia menjadi tiga subtipe, yaitu L1 (anak-anak), L2 (anak yang lebih tua dan orang dewasa), dan L3 (pasien dengan leukemia sekunder akibat limfoma burkitt) [10]. Sebanyak 70% pasien penderita ALL termasuk dalam kategori L1 [11]. Penyakit tersebut ditandai dengan produksi jumlah WBC muda (bernama sel limfoblas) yang berlebihan [5]. *Threshold*

(ambang batas) jumlah sel limfoblas adalah tidak lebih 20% dari total WBC di sumsum tulang belakang untuk batas normal populasi sel limfoblas [11]. Oleh karena itu, respon cepat dan diagnosis dini sangat dibutuhkan untuk rehabilitasi pasien, terutama pada anak-anak [12].

Teknik perhitungan sel darah tradisional yang digunakan di laboratorium cukup menghabiskan waktu dan tenaga [13]. Metode tradisional untuk perhitungan sel darah bergantung pada bantuan manusia, di mana perhitungan dan klasifikasi sel dilakukan secara manual melalui mikroskop [13]. Pemeriksaan mikroskopis secara manual terhadap apusan darah menghabiskan banyak waktu dan mengalami bias, seperti faktor pengalaman hematologis, kelelahan, dan faktor manusia umum lainnya yang menyebabkan hasil laporan tidak konsisten dan subjektif [13,14]. Saat ini, beberapa rumah sakit dan laboratorium telah menerapkan penganalisis secara otomatis menggunakan *analyzer* untuk mendeteksi dan menghitung elemen dan sel darah, tetapi instrumen tersebut sangat mahal dan sebagian besar laboratorium dan rumah sakit di pedesaan tidak mampu memilikinya [13]. Oleh karena itu, muncul kebutuhan untuk sistem deteksi dan perhitungan sel darah otomatis secara hemat biaya, cepat, dan efisien tanpa dipengaruhi oleh kelelahan manusia.

Artificial Intelligence (AI) telah sangat maju dalam beberapa tahun terakhir yang mendapatkan manfaat dari kemajuan teknologi *Graphics Processing Unit* (GPU) [17]. Banyak model *Convolutional Neural Network* (CNN) dengan arsitektur berbeda yang dikembangkan dan diterapkan dalam berbagai bidang [17]. *Deep Learning* (pembelajaran mendalam) dianggap sebagai *state-of-the-art* untuk klasifikasi citra [18,19]. Arsitektur CNN yang merupakan bagian dari *deep learning* dapat menghasilkan penurunan tingkat kesalahan klasifikasi sekitar 25% pada tahun 2011 menjadi 3,6% pada tahun 2015 [18,19].

Penelitian yang sudah dilakukan sebelumnya adalah hematologis didukung dengan sebuah mikroskop termodifikasi yang memiliki algoritma kecerdasan buatan [15,16,26]. Mikroskop yang termodifikasi tersebut diberi kamera pada lensa okuler untuk menggantikan mata pengamat atau hematologis. Kemudian, citra apusan darah akan diambil oleh kamera. Hasil citra tersebut akan dilakukan

pemrosesan dengan algoritma kecerdasan buatan (AI) untuk pengklasifikasian jenis WBC. Penelitian menggunakan mikroskop termodifikasi telah dilakukan oleh Tim Riset Studio Sensor Visual dengan menggunakan model YOLOv2 dan *Faster R-CNN* untuk melakukan pendekslsian sel limfoblas dan non-limfoblas [16,26]. Namun, model YOLOv2 dan *Faster R-CNN* yang dikembangkan tersebut tidak dapat melakukan klasifikasi tingkat piksel-ke-piksel (segmentasi *instance*) serta tidak digunakan pengolahan citra yang berpotensi untuk meningkatkan performa deteksi.

Pada penelitian ini digunakan model *Mask Region-based Convolutional Neural Network* (R-CNN) yang dapat melakukan deteksi dan segmentasi *instance* sel limfoblas dan non limfoblas pada apusan darah pasien ALL tipe L1. *Mask R-CNN* merupakan pengembangan dari *Faster R-CNN* dengan adanya penambahan cabang untuk melakukan segmentasi *instance*, sehingga dapat melakukan klasifikasi tingkat piksel-ke-piksel [20]. Kelebihan dari penelitian ini adalah kemampuannya dalam menghasilkan segmentasi *instance* untuk mempermudah visualisasi WBC serta penerapan teknik pengolahan citra yang berpotensi untuk meningkatkan performa lebih optimal dari penelitian telah dilakukan oleh Tim Riset Studio Sensor Visual dengan menggunakan model deteksi objek YOLOv2 [16] dan *Faster R-CNN* [26].

I.2. Perumusan Masalah

1. Bagaimana performa deteksi dan segmentasi *instance* model *Mask R-CNN* pada sel limfoblas dan non-limfoblas?
2. Bagaimana upaya peningkatan performa model *Mask R-CNN* dalam deteksi dan segmentasi *instance* sel limfoblas dan non-limfoblas?

I.2.1. Batasan Masalah

Penelitian berfokus pada deteksi dan segmentasi *instance* sel limfoblas dan non-limfoblas pasien ALL tipe L1 dengan melakukan pengembangan model *Mask R-CNN*. Citra apusan darah yang digunakan pada penelitian ini bersumber dari pasien Rumah Sakit Umum Pusat Dr. Sardjito. Citra apusan darah telah diambil

oleh Tim Riset Studio Sensor Visual menggunakan mikroskop termodifikasi, sehingga penelitian hanya berfokus pada pengembangan model deteksi dan segmentasi *instance*.

I.3. Tujuan Penelitian

1. Mengembangkan model *Mask R-CNN* untuk melakukan deteksi dan segmentasi *instance* sel limfoblas dan non-limfoblas.
2. Melakukan peningkatan akurasi, *F1-Score*, dan mAP yang dapat dicapai dengan model *Mask R-CNN* pada sel limfoblas dan non-limfoblas untuk memperoleh performa yang optimal.

I.4. Manfaat Penelitian

1. Memperoleh hasil implementasi metode deteksi dan segmentasi *instance* berbasis *Mask R-CNN* untuk sel limfoblas dan non-limfoblas pada pasien ALL tipe L1.
2. Hasil penelitian ini dapat menjadi pertimbangan implementasi metode *deep learning* dan pengolahan citra pada peneliti berikutnya dengan kasus serupa.

BAB II

TINJAUAN PUSTAKA

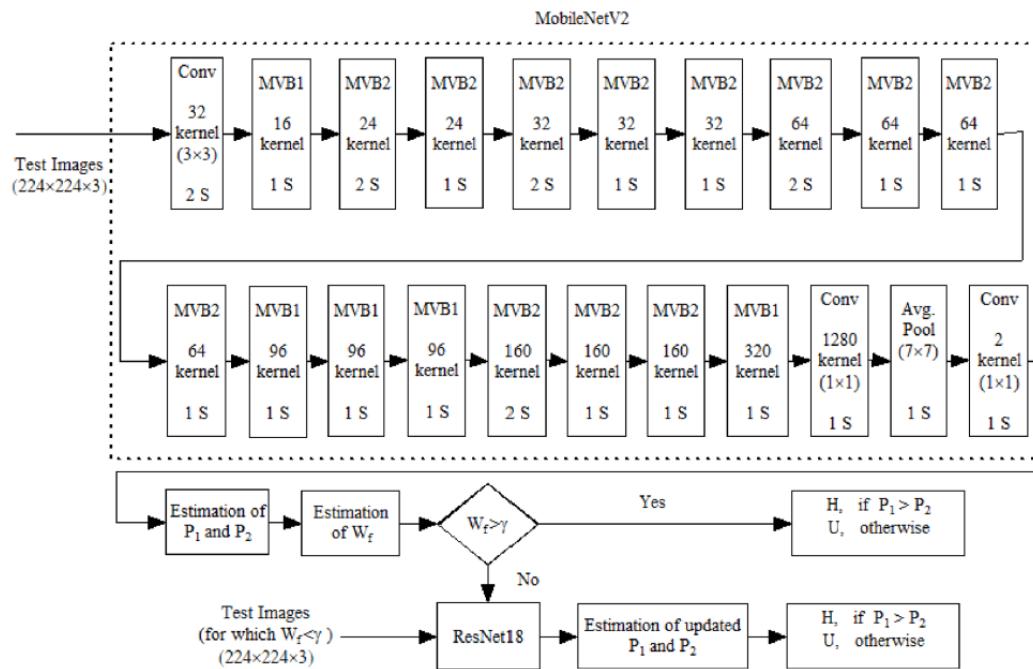
Penelitian-penelitian mengenai klasifikasi, segmentasi, dan pendektsian dari jenis sel darah manusia dengan berbagai metode *image processing* dan *deep learning* telah banyak dilakukan. Penelitian-penelitian tersebut bertujuan sebagai alternatif menganalisis kesehatan pasien yang memerlukan pengklasifikasian dan perhitungan jumlah dari jenis sel darah secara manual ataupun menggunakan *analyzer*. Adapun pendektsian objek dan segmentasi *instance* menggunakan *Mask R-CNN* telah dilakukan pada sel darah manusia yang dapat memberikan kinerja cukup baik. Pada tinjauan pustaka ini akan diuraikan berbagai penelitian yang berkaitan dengan klasifikasi, pendektsian, pengolahan citra, dan segmentasi *instance* pada sel darah manusia.

II.1. Klasifikasi WBC pada Kasus Acute Lymphoblastic Leukemia

Penelitian mengenai klasifikasi WBC pada kasus ALL berbasis *transfer learning* telah banyak dilakukan dalam beberapa tahun terakhir. Kesulitan dalam mendekksi sel leukemia dari citra terdapat pada kesamaan morfologi dan variabilitas subjek, membuat pendefinisian fitur citra yang sesuai menjadi tugas yang sangat menantang [21]. Selain itu, citra juga memiliki jenis sel lain yang dapat mengganggu proses klasifikasi WBC [21]. Pada subbab ini akan membahas penelitian terkait klasifikasi WBC kasus ALL dengan basis *transfer learning* [22].

Pada tahun 2021, Pradeep Kumar Das dan Sukadev Meher telah melakukan penelitian dengan menggunakan *probability-based weight factor* yang memiliki peran untuk menghibridisasi MobilenetV2 dan ResNet18 secara efisien dengan mempertahankan manfaat dari kedua pendekatan tersebut [22]. Hasil penelitian menunjukkan akurasi terbaik (dengan 70% pelatihan dan 30% pengujian) masing-masing nilainya adalah 99,39% dan 97,18% pada *dataset* ALL-IDB1 dan ALL-IDB2 [22]. Selain itu, penelitian tersebut juga mencapai kinerja terbaik dibandingkan dengan model *transfer learning* baru-baru ini untuk kedua *dataset*

ALL-IDB1 dan ALL-IDB2 dalam hal *sensitivity*, *specificity*, *accuracy*, *precision*, *F1-score*, dan *receiver operating characteristic* (ROC) seperti yang ditunjukkan pada Tabel 2.1. dan Tabel 2.2. [22].



Gambar 2.1. Arsitektur deteksi ALL berbasis CNN yang diusulkan peneliti [22].

Pada Gambar 2.1. di atas adalah arsitektur yang diusulkan peneliti dengan H mengartikan sehat sedangkan U mengartikan tidak sehat (*ALL affected*) [22]. P_1 adalah probabilitas sehat sedangkan P_2 adalah probabilitas *ALL affected* dan γ adalah *threshold* [22]. Dalam penelitian tersebut digunakan $\gamma = 0,043$ dan digunakan *weight factor* (W_f) yang memiliki peran penting dalam menghasilkan kinerja baik dan membuat sistem menjadi efisien secara komputasi [22].

Tabel 2.1. Perbandingan kinerja dalam *dataset ALL-IDB1* dengan 70% data pelatihan dan 30% pengujian (rata-rata 10-Fold cross validation).

Metode	Akurasi (%)	Presisi (%)	Sensitivity (%)	Specificity (%)	F1-Score	AUC
Inception V3	90,91	91,41	89,63	92,13	0,9022	0,909
ResNet 18	95,15	93,79	96,59	94,12	0,9517	0,954
DenseNet 201	94,55	91,34	98,64	91,01	0,9485	0,950
VGG16	85,76	88,38	80,35	89,90	0,8417	0,851
MobileNet V2	96,06	95,67	96,63	95,81	0,9615	0,962
GoogLe Net	95,15	94,49	95,66	94,67	0,9507	0,952
AlexNet	93,64	89,55	98,64	89,35	0,9387	0,940
Diajukan peneliti	99,39	99,33	99,55	99,47	0,9944	0,995

Sumber: [22]

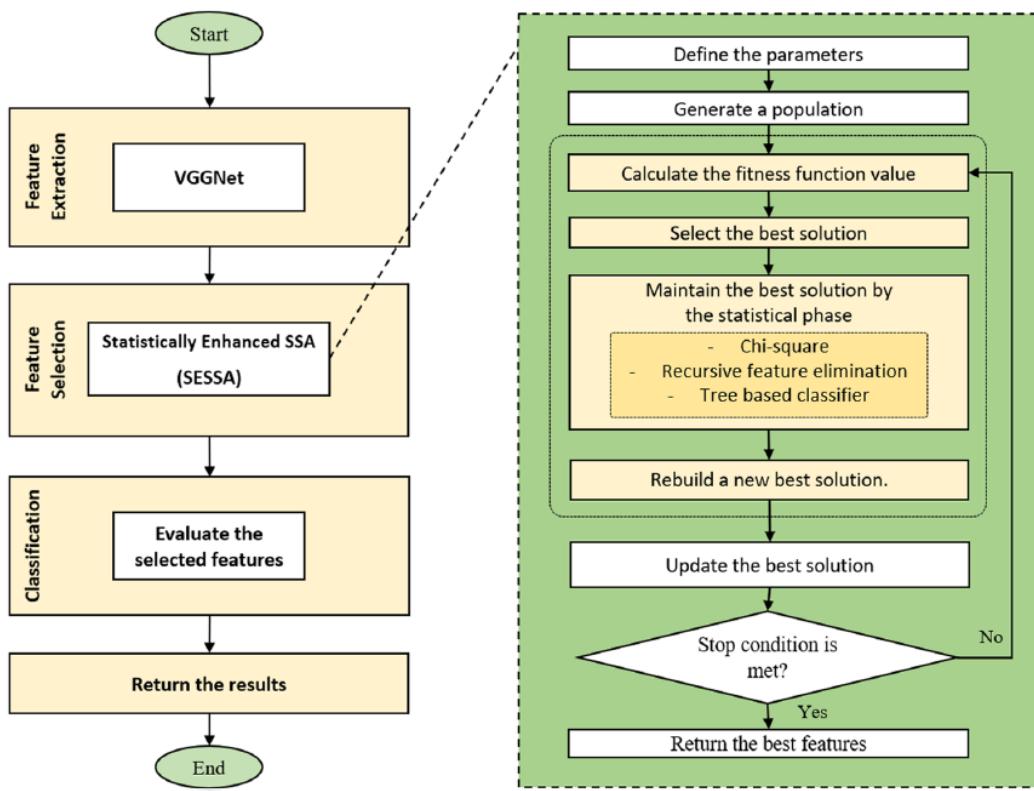
Tabel 2.2. Perbandingan kinerja dalam *dataset ALL-IDB2* dengan 70% data pelatihan dan 30% pengujian (rata-rata 10-Fold cross validation).

Metode	Akurasi (%)	Presisi (%)	Sensitivity (%)	Specificity (%)	F1-Score	AUC
Inception V3	92,82	95,66	91,28	94,35	0,9342	0,928
ResNet 18	93,34	93,82	93,34	93,34	0,9358	0,933
DenseNet 201	94,10	92,60	96,41	91,79	0,9447	0,941
VGG16	82,57	84,60	86,15	78,97	0,8537	0,826

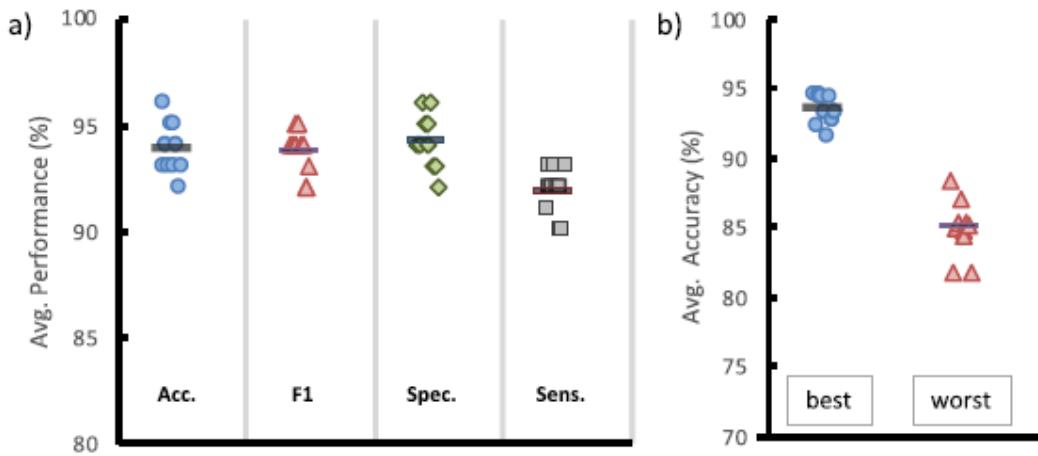
MobileNet V2	92,82	95,48	91,80	93,84	0,9360	0,928
GoogLe Net	91,54	88,26	96,41	86,67	0,9216	0,919
AlexNet	91,54	89,77	94,87	88,21	0,9225	0,915
Diajukan peneliti	97,18	98,52	95,90	98,46	0,9719	0,972

Sumber: [22]

Ahmed T. Sahlol, Philip Kollmannsberger, dan Ahmed A. Ewees pada tahun 2020 melakukan penelitian menggunakan pendekatan *hybrid* yang dioptimalkan untuk klasifikasi leukemia WBC yang efisien menggunakan *dataset* ALL-IDB1, ALL-IDB2, dan C-NMC [21]. Penelitian tersebut mengekstrak fitur dari gambar WBC menggunakan VGGNet, fitur yang diekstraksi tersebut kemudian diseleksi menggunakan algoritma *Salp Swarm Algorithm* (SESSA) yang dioptimalkan secara statistik [21]. Penggunaan seleksi fitur berdasarkan SESSA dapat mengurangi waktu pelatihan (lebih sedikit fitur yang digunakan), meningkatkan akurasi (lebih sedikit data *misleading* sehingga membuat model lebih efisien), dan mengurangi *overfitting* [21]. Ekstraksi fitur dan seleksi fitur dievaluasi menggunakan enam algoritma pengklasifikasi (*Linear SVM*, KNN, *Decision Trees*, Naive Bayes, Adaboost, dan *Multi-Layer Perceptron*) [21]. Alur kerja dan hasil kinerja model yang diusulkan penelitian tersebut ditampilkan pada Gambar 2.2., Gambar 2.3., dan Tabel 2.3.



Gambar 2.2. Alur kerja dari pendekatan yang diusulkan peneliti [21].



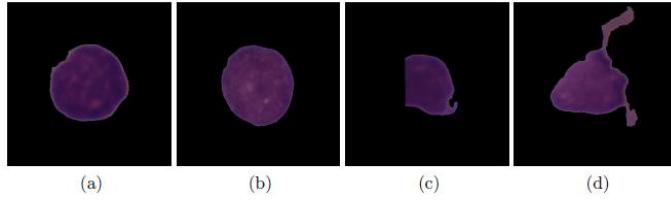
Gambar 2.3. Kinerja pendekatan VGGNet dan SESSA *hybrid* yang diusulkan pada dataset ALL-IDB [21]. (a) kinerja rata-rata selama 10 *runs*, (b) akurasi untuk 10 *runs* terbaik dan terburuk.

Tabel 2.3. Perbandingan jumlah fitur dan kinerja untuk kedua *dataset*.

<i>Dataset 1</i>	Jumlah Fitur	Persentase	Akurasi (%)	<i>Specificity</i> (%)	<i>Sensitivity</i> (%)
VGG 19	25088	100%	94,23	100	88,00
Diajukan peneliti	1087	4,00%	96,11	95,00	93,00
<i>Dataset 2</i>					
VGG 19	25088	100%	80,90	80,90	80,90
Diajukan peneliti	1115	4,40%	83,30	67,30	91,10

Sumber: [21]

Pada tahun 2020, Jonas Prellberg dan Oliver Kramer melakukan penelitian klasifikasi sel sehat dan ganas ALL menggunakan *dataset* C-NMC yang terdiri lebih dari 10.000 citra WBC [23]. Model dasar yang digunakan pada penelitian tersebut adalah ResNeXt50 dengan menggunakan augmentasi data untuk meningkatkan pelatihan model [23]. Augmentasi data tersebut berupa pembalikan horizontal dan vertikal dengan probabilitas masing-masing 50%, rotasi dengan interval sudut $[0, 360)$ derajat yang dipilih secara seragam dan acak, dan translasi acak hingga 20% dari setiap panjang sisi dalam arah horizontal dan vertikal [23]. *Dataset* yang digunakan sudah tersegmentasi seperti ditunjukkan pada Gambar 2.4. Hasil kinerja model yang digunakan peneliti tersebut ditampilkan pada Tabel 2.4.



Gambar 2.4. Contoh gambar yang diambil dari *training-set* [23]. (a) Sel ALL, (b) Sel Normal, (c) ALL sel dengan bagian sel terputus karena segmentasi yang tidak sempurna, (d) Sel normal dengan latar belakang yang berlebihan karena segmentasi yang tidak sempurna.

Tabel 2.4. Hasil pada *test-set* selama 24 *training runs* menggunakan model terbaik dari setiap *run* yang diukur dengan *F1-score*.

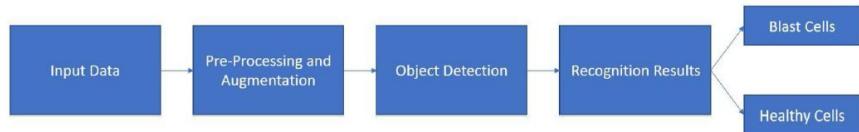
	Min.	Mean ± Std.	Max.
Akurasi	86,40	87,96 ± 0,90	89,88
Sensitivity	88,43	92,01 ± 1,44	94,50
Specificity	75,31	80,36 ± 2,39	84,72
<i>F1 score</i>	86,28	87,89 ± 0,90	89,81
Presisi	86,27	87,91 ± 0,90	89,81
Recall	86,40	87,96 ± 0,90	89,88

Sumber: [23]

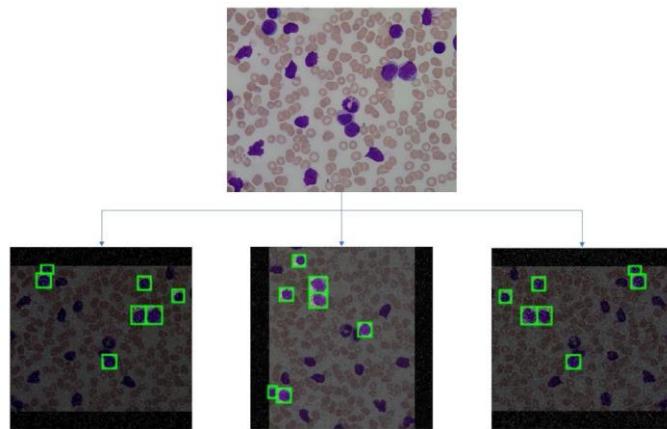
II.2. Pendeksi WBC pada Kasus Acute Lymphoblastic Leukemia

Rohan Khandekar, Prakhya Shastry, Smruthi Jaishankar, Oliver Faust, dan Niranjana Sampathila melakukan penelitian pada tahun 2021 mengenai deteksi sel limfoblas otomatis untuk diagnosis ALL dengan implementasi *You Only Look Once* versi 4 (YOLOv4) sebagai deteksi dan klasifikasi [24]. Klasifikasi tersebut ditetapkan sebagai klasifikasi biner, di mana setiap sel diberi label sebagai sel limfoblas (ALL) atau sel sehat (HEM) [24]. Algoritma deteksi objek dilatih dan diuji dengan citra dari *dataset* ALL-IDB1 dan *dataset* C-NMC 2019 [24]. *Mean Average Precision* (mAP) adalah 96,06% untuk *dataset* ALL-IDB1 dan 98,7% untuk *dataset* C-NMC 2019 [24]. Alur kerja penelitian dan contoh hasil dari *image*

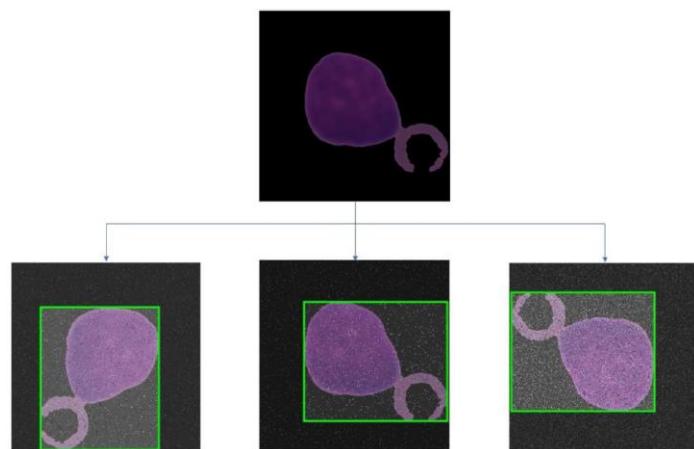
pre-processing dan *augmentation* untuk dataset ALL-IDB1 dan C-NMC 2019 ditampilkan pada Gambar 2.5., Gambar 2.6., dan Gambar 2.7.



Gambar 2.5. Alur kerja *screening* ALL peneliti [24].



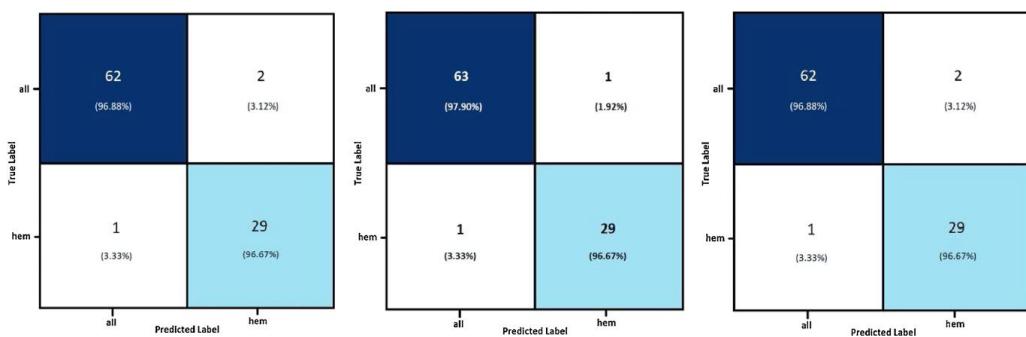
Gambar 2.6. Citra sel limfoblas dan augmentasi untuk *dataset* ALL-IDB1 [24].



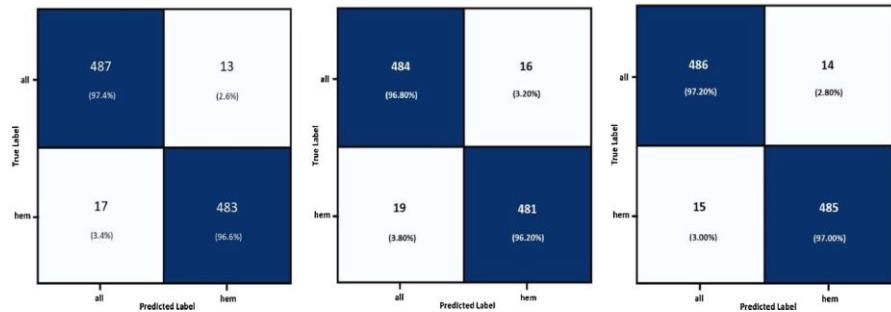
Gambar 2.7. Citra sel limfoblas dan augmentasi untuk C-NMC 2019 [24].

Augmentasi citra tersebut digunakan untuk meningkatkan properti *dataset* dan membuat algoritma lebih *robust* [24]. Pada penelitian ini terdapat dua langkah *image pre-processing*, yaitu *Auto-oriented* dan *Resizing* [24]. *Auto-oriented* dilakukan dengan mempertahankan *bounding boxes* terhadap *object of interest* setelah operasi seperti rotasi dan pembalikan [24]. *Resizing* dilakukan karena algoritma YOLO mengharuskan gambar dalam kelipatan 32×32 [24]. Selama proses *resizing*, citra di-*padding* untuk mempertahankan *aspect ratio* asli dari semua citra sehingga tetap mengubah ukuran ke ukuran yang dibutuhkan [24].

Augmentasi citra dilakukan seperti pembalikan horizontal dan vertikal, rotasi 90° , rentang kecerahan $\pm 25\%$, dan *noise* hingga 5% piksel dilakukan secara acak pada citra *dataset* ALL-IDB1 [24]. *Dataset* dipartisi menjadi pelatihan, validasi, dan pengujian dengan pembagian 70:20:10, augmentasi dan *pre-processing* dilakukan menggunakan Roboflow [24]. Selanjutnya, *dataset* C-NMC 2019, *auto-orient* dan *resizing* sama dengan *dataset* ALL-IDB1 [24]. Namun, augmentasi C-NMC dilakukan dengan empat langkah: 1) Pembalikan horizontal dan vertikal; 2) Penyesuaian kecerahan acak $\pm 40\%$; 3) Rotasi acak searah jarum jam dan berlawanan arah jarum jam; dan 4) *Gaussian blur* acak antara 0–3% piksel dan *noise salt* dan *pepper* untuk 5% piksel [24]. Pada pengujian dan inferensi, citra mentah dimasukkan langsung ke dalam algoritma tersebut [24]. Hasil kinerja penelitian tersebut ditampilkan pada Gambar 2.8., Gambar 2.9., Tabel 2.5., dan Tabel 2.6.



Gambar 2.8. Prediksi di YOLOv4 untuk *dataset* ALL-IDB1 [24].



Gambar 2.9. Prediksi pada YOLOv4 untuk *dataset* C-NMC 2019 [24].

Tabel 2.5. Perbandingan penelitian lain yang diusulkan peneliti untuk *dataset* ALL-IDB1.

Peneliti	Metode (<i>Dataset</i>)	Fitur yang menonjol	Ukuran Kinerja
Ruberto et al. (2020) [66]	Teknik Pemrosesan Citra (SMCIDB, IUSM-IDB, ALL-IDB)	Deteksi sel tunggal dari aglomerasi tanpa <i>computational workload</i>	Akurasi 94,10% untuk deteksi leukemia.
Jyoti et al. (2015) [67]	SVM <i>classifier</i> (ALL-IDB)	Penggunaan matriks <i>cooccurrence</i> tingkat abu-abu dan fitur berbasis bentuk	Akurasi fitur bentuk tekstur gabungan: 89,80%.
Nizar et al. (2019) [68]	<i>Convolutional Neural Network</i> (ALL-IDB, ASH <i>Image Bank</i>)	<i>5-fold cross validation</i>	Akurasi pada leukemia versus sehat: 88,25%.
Metode yang diajukan peneliti	<i>Object Detection</i> (ALL-IDB1)	Peningkatan akurasi, serta	Keseluruhan mAP (<i>Mean Average</i>)

		deteksi posisi <i>object of Interest</i>	<i>Precision)</i> sebesar 95,57%, <i>recall</i> sebesar 92%, dan F1- <i>score</i> sebesar 0,92.
--	--	---	--

Sumber: [24]

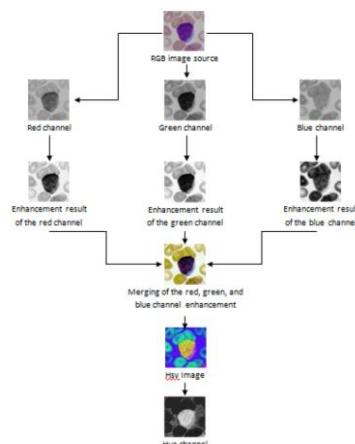
Tabel 2.6. Perbandingan penelitian lain dengan metode yang diusulkan peneliti untuk *dataset* C-NMC 2019.

Peneliti	Metode (<i>Dataset</i>)	Fitur yang menonjol	Ukuran Kinerja
Jonas et al. (2019) [69]	ResNeXT <i>convolutional neural network</i> (C-NMC 2019)	Penggunaan modul <i>Squeezeand-Excitation</i>	<i>Weighted F1-score</i> sebesar 88,91 %.
Yifan et al. (2019) [62]	<i>Convolutional Neural Network</i> (C-NMC 2019)	Penggunaan metode <i>ensemble</i> bernama <i>stacking</i> untuk meningkatkan kinerja	<i>Weighted F1-score</i> pada <i>test-set</i> sebesar 85,52 %.
Christian et al. (2019) [70]	<i>Convolutional Neural Network</i> (C-NMC 2019)	Penggunaan <i>regional proposal subnetwork</i>	<i>Weighted F1-score</i> pada <i>test-set</i> sebesar 82,84 %.
Metode yang diajukan peneliti	<i>Object Detection</i> (C-NMC 2019)	Peningkatan akurasi	<i>Weighted F1-score</i> pada <i>test-set</i> sebesar 92% dengan mAP

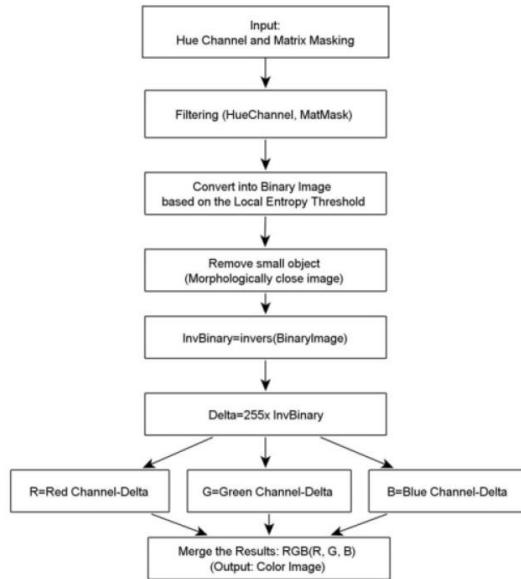
		sebesar 98,57% dan <i>recall</i> sebesar 96%.
--	--	---

Sumber: [24]

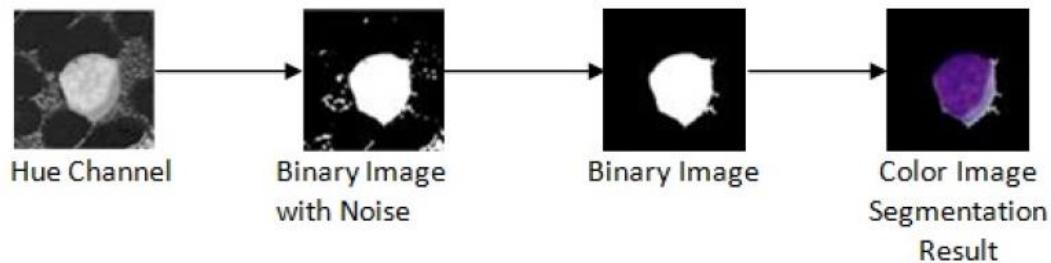
Pada tahun 2019 Arif Muntasaa dan Muhammad Yusuf melakukan penelitian yang bertujuan untuk mengembangkan model untuk mendeteksi ALL menggunakan *dataset* ALL-IDB [25]. Metode dikembangkan peneliti terdiri dari empat tahap utama sebagai berikut: *enhancement*, *segmentation*, *feature extraction*, dan *accuracy measurement* [25]. Penelitian ini mengusulkan tujuh karakteristik sebagai fitur objek, yaitu *Energy* (EN), *Entropy* (EP), *Shanon Entropy* (H(X)), dan *Log Energy*, metode pengukurnya yaitu metode Euclidean Distance, Manhattan, Canberra, dan Chebyshev [25]. Metode penelitian tersebut telah menghasilkan akurasi maksimum masing-masing sebesar 81,54% untuk Euclidean Distance, 81,54% untuk Manhattan, 76,92% untuk Canberra, dan 82,31% untuk Chebyshev [25]. Ilustrasi dari *enhancement* citra, proses segmentasi, dan hasil segmentasi ditampilkan pada Gambar 2.10., Gambar 2.11., Gambar 2.12., dan Gambar 2.13. Namun, proses pengukuran akurasi ditampilkan pada Gambar 2.14. dan Gambar 2.15.



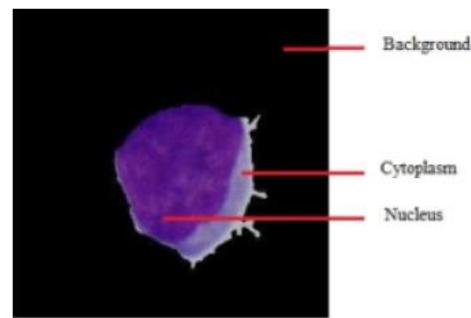
Gambar 2.10. Ilustrasi *enhancement* citra ALL [25].



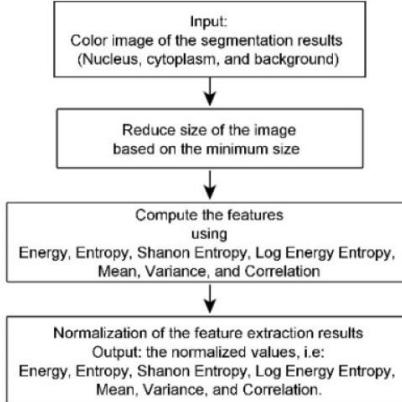
Gambar 2.11. Proses segmentasi [25].



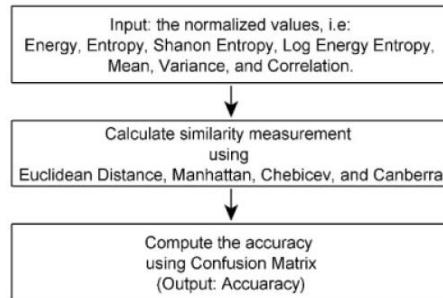
Gambar 2.12. Ilustrasi proses segmentasi [25].



Gambar 2.13. Ilustrasi hasil segmentasi [25].

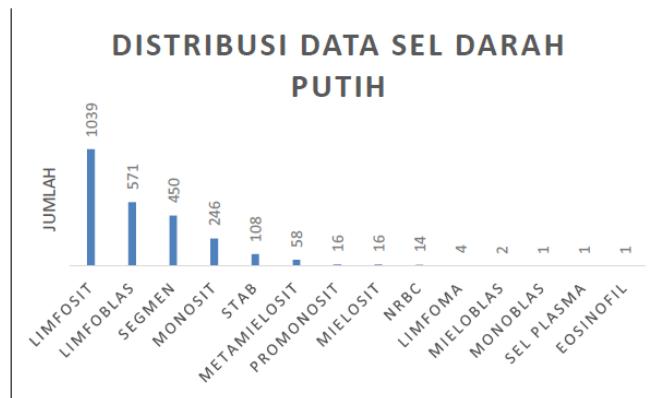


Gambar 2.14. Proses ekstraksi fitur [25].



Gambar 2.15. Normalisasi dan pengukuran akurasi [25].

Pada tahun 2020, Muhamad Inska Fajri telah melakukan penelitian menggunakan *Faster R-CNN* untuk mendekteksi sel limfoblas dan non-limfoblas menggunakan citra sel darah dari pasien pengidap ALL tipe L1 yang terdiri dari 5 jenis sampel darah [26]. Citra sel darah tersebut divalidasi dengan label oleh hematologis dari RSUP Dr. Sardjito Yogyakarta [26]. Citra WBC yang telah diberi label oleh hematologis terdiri dari 2.527 data dari lima preparat apusan darah [26]. Fokus utama pada penelitian tersebut adalah pendektsian sel limfoblas karena kebutuhan dari hematologis untuk mengenali dan menghitung sel kanker (limfoblas) dalam darah sehingga jenis WBC selain sel limfoblas dimasukkan dalam label non-limfoblas [26]. Distribusi awal data ditampilkan pada Gambar 2.16 dan Tabel 2.7. Namun, distribusi data pasca proses *oversampling* dan *undersampling* ditampilkan pada Tabel 2.8.



Gambar 2.16. Distribusi awal data [26].

Tabel 2.7. Distribusi data limfoblas dan non-limfoblas.

Jenis Label	Jumlah Citra
Limfoblas	571
Non-limfoblas	1.956
Jumlah total	2.527

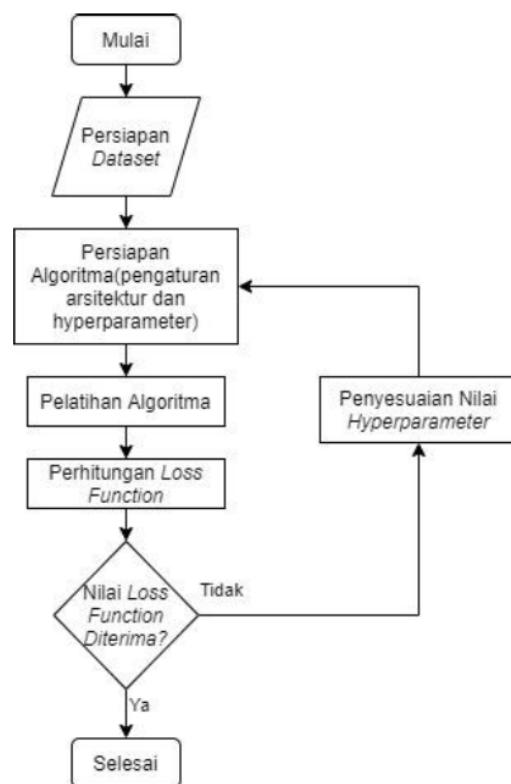
Sumber: [26]

Tabel 2.8. Distribusi data limfoblas dan non-limfoblas pasca *oversampling* dan *undersampling*.

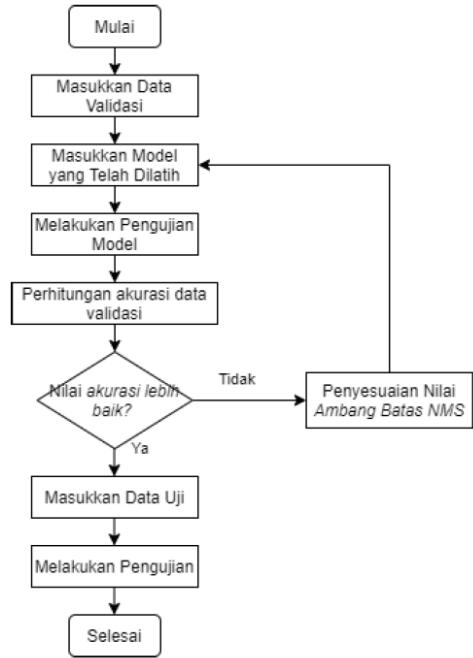
Jenis Label	Jumlah Citra
Limfoblas	633
Non-limfoblas	628
Jumlah total	1.261

Sumber: [26]

Pada penelitian tersebut *dataset* dibagi menjadi 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian [26]. Proses anotasi dilakukan satu persatu pada tiap objek WBC meliputi pada data latih, validasi, dan pengujian menggunakan LabelImg [26]. Pelatihan dilakukan menggunakan arsitektur *backbone* berupa ResNet50 dan ResNet101 dengan alur kerja ditampilkan pada Gambar 2.17. *Hyperparameter* yang diatur pada penelitian tersebut berupa: *batch size*, *learning rate*, dan *step* [26]. Pengujian model dilakukan dengan alur kerja pada Gambar 2.18. untuk memaksimalkan akurasi dengan cara penyesuaian *threshold non-max suppression* (NMS) [26]. Proses perhitungan otomatis dibandingkan dengan perhitungan secara manual dan dilakukan perhitungan *Mean Absolute Error* (MAE) sebagai pengujian kinerja model [26].

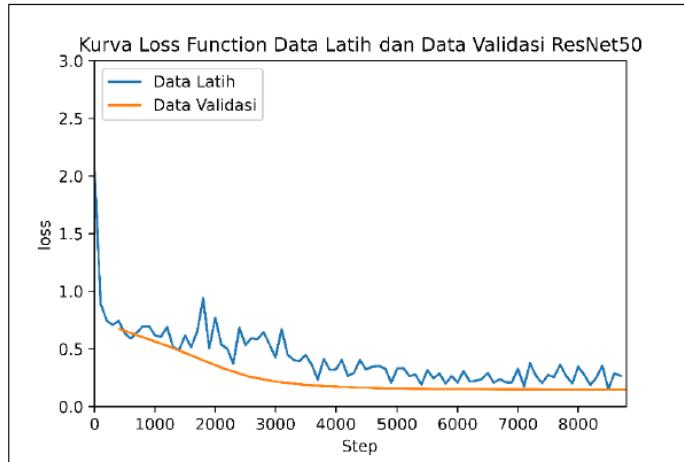


Gambar 2.17. Alur pelatihan algoritma deteksi peneliti [26].

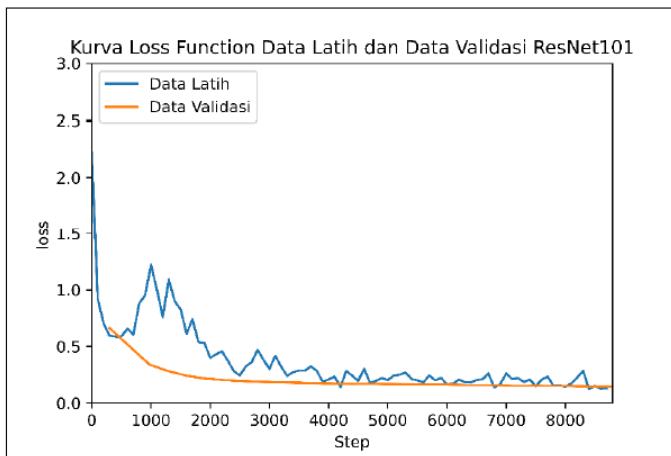


Gambar 2.18. Alur pengujian algoritma deteksi peneliti [26].

Pelatihan algoritma pada kedua arsitektur dilakukan dengan ketentuan *hyperparameter* seperti berikut: 1) *Learning rate* bernilai 10^{-5} ; 2) *Optimizer* digunakan *stochastic gradient decent with momentum* dengan besar momentum bernilai 0,9; 3) *Batch size* bernilai 8; 4) Jumlah *epoch* bernilai 80; 5) Fungsi aktivasi yang digunakan adalah ReLU pada setiap lapisan konvolusi; dan 6) L2 *regularization* diterapkan pada setiap lapisan konvolusi [26]. Pada penelitian tersebut jumlah *epoch* adalah 80 yang menunjukkan grafik *loss function* konvergen dengan membutuhkan waktu 5 jam untuk satu kali pelatihan dan dihentikan pada *step* ke-8.800 [26]. Dari penelitian tersebut arsitektur ResNet101 memberikan hasil yang lebih baik dibandingkan ResNet50 yang disebabkan faktor jumlah lapisan jaringan [26]. Kurva *loss function* ditampilkan pada Gambar 2.19. dan Gambar 2.20. yang menunjukkan tren konvergen dan mengartikan model tidak mengalami *overfitting* [26].

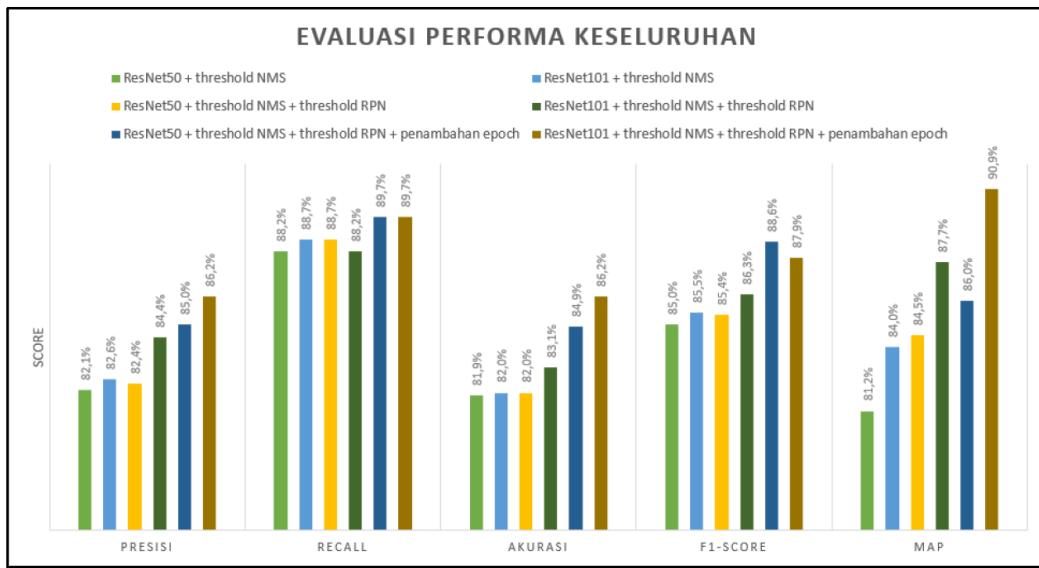


Gambar 2.19. Kurva *loss function* data latih dan data validasi ResNet50 [26].



Gambar 2.20. Kurva *loss function* data latih dan data validasi ResNet101 [26].

Penentuan *threshold* dari *non-max suppression* (NMS) dan *Region Proposal Networks* (RPN) beserta menaikkan nilai *epoch* menjadi 400 dilakukan untuk meningkatkan kinerja model *Faster R-CNN* [26]. Evaluasi performa model secara keseluruhan pada penelitian tersebut ditampilkan pada Gambar 2.21. [26]. Performa terbaik yang diusulkan peneliti ditampilkan pada Tabel 2.9. dan untuk perbandingan dengan perhitungan manual ditampilkan pada Tabel 2.10.



Gambar 2.21. Evaluasi performa model keseluruhan peneliti [26].

Tabel 2.9. Performa model terbaik peneliti dengan *epoch* 400.

Metode	Presisi	Recall	Akurasi	F1-Score	mAP
ResNet50 dengan <i>threshold</i> NMS 0,1 dan <i>threshold</i> RPN 0,55.	85,0%	89,7%	84,9%	88,6%	86,0%
ResNet101 dengan <i>threshold</i> NMS 0,15 dan <i>threshold</i> RPN 0,50.	86,2%	89,7%	86,2%	87,9%	90,9%

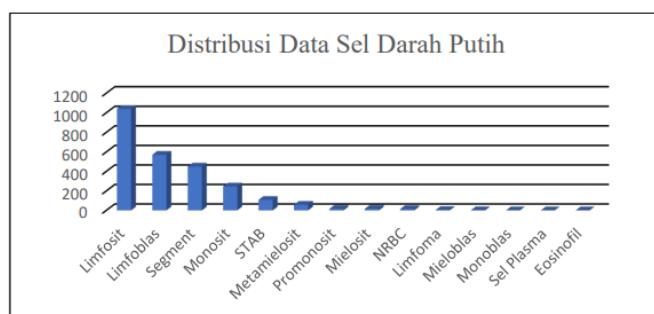
Sumber: [26]

Tabel 2.10. Evaluasi MAE dengan perhitungan manual.

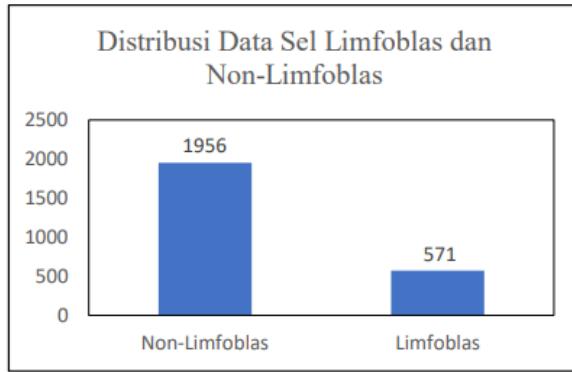
Model	Kelas	Hitung Manual	Hitung Otomatis	MAE
ResNet50 dengan <i>threshold</i> NMS 0,1 dan <i>threshold</i> RPN 0,55.	Limfoblas	97	101	0,144
	Non-limfoblas	98	105	0,173
ResNet101 dengan <i>threshold</i> NMS 0,15 dan <i>threshold</i> RPN 0,50.	Limfoblas	97	105	0,144
	Non-limfoblas	98	98	0,143

Sumber: [26]

Pada tahun 2020, Arnanda Prabaswara melakukan penelitian serupa dengan Muhamad Inska Fajri. Namun, penelitian Arnanda menggunakan deteksi objek *Tiny YOLOv2* dengan 9 lapisan konvolusi dan *YOLOv2* dengan 24 lapisan konvolusi, kemudian hasil dari kedua arsitektur tersebut dibandingkan dan dianalisis [16]. Pelatihan algoritma deteksi dan pengujian algoritma memiliki alur yang serupa dengan Muhamad Inska Fajri. Proses pengambilan citra sel darah juga serupa yang dilakukan oleh Muhamad Inska Fajri. Fokus utama pada penelitian adalah pendekripsi sel limfoblas dan non-limfoblas, sehingga jenis WBC selain sel limfoblas dimasukkan dalam label non-limfoblas (ditampilkan pada Gambar 2.22. dan Gambar 2.23.) [16].



Gambar 2.22. Distribusi data WBC [16].



Gambar 2.23. Distribusi data sel limfoblas dan non-limfoblas [16].

Distribusi awal data yang ditampilkan di atas dilakukan *oversampling* dan *undersampling*. Teknik augmentasi yang dilakukan hanya melakukan pencerminan pada sumbu *y*. Metode *oversampling* hanya dilakukan pada data latih, sedangkan pada data validasi dan data uji digunakan citra aslinya namun tetap diseimbangkan dengan metode *undersampling* [16]. Distribusi data yang telah dilakukan proses *oversampling* dan *undersampling* ditampilkan pada Tabel 2.11. Distribusi data dibagi menjadi 70% data latih, 15% data validasi, dan 15% data uji [16].

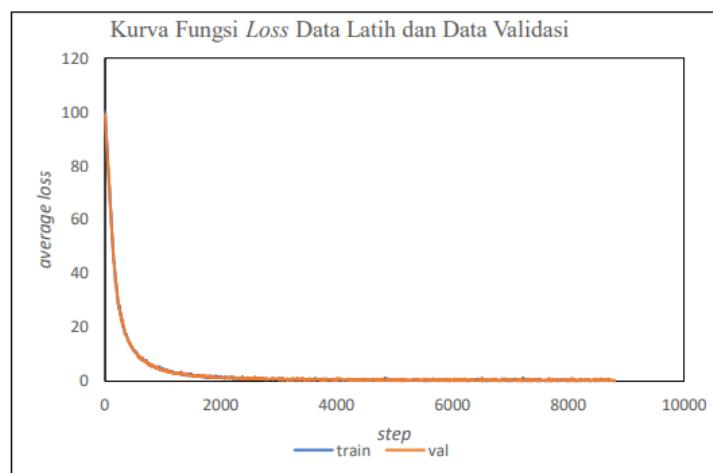
Tabel 2.11. Distribusi data pasca *oversampling* dan *undersampling*.

Jenis sel darah putih	Jumlah citra
Non-limfoblas	629
Limfoblas	632

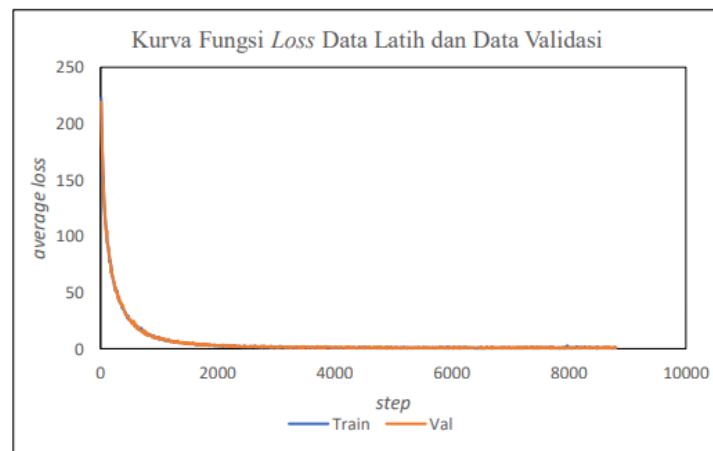
Sumber: [16]

Pelatihan dilakukan dengan menggunakan bobot *pre-train* Tiny YOLOv2 dan YOLOv2 yang dilatih pada *dataset* PASCAL VOC dan COCO, sehingga banyaknya lapisan konvolusi disesuaikan dengan arsitektur yang digunakan untuk melatih kedua *dataset* tersebut [16]. *Hyperparameter* kedua arsitektur diatur sesuai dengan ketentuan berikut: 1) *Learning rate* bernilai 10^{-5} ; 2) *Optimizer* digunakan

Adam; 3) Fungsi aktivasi digunakan *Leaky ReLU* pada setiap lapisan konvolusi; 4) *Batch normalization* digunakan juga pada setiap lapisan konvolusi; 5) Jumlah *epoch* bernilai 100; dan 6) *Batch size* berukuran 8 [16]. Kedua pelatihan berjalan selama 12 jam dan berhenti pada *step* ke 8800. Nilai *average loss* pada pelatihan jaringan *Tiny YOLOv2* berakhir pada 0,204, sedangkan pada jaringan *YOLOv2* berhenti pada 0,934 [16].

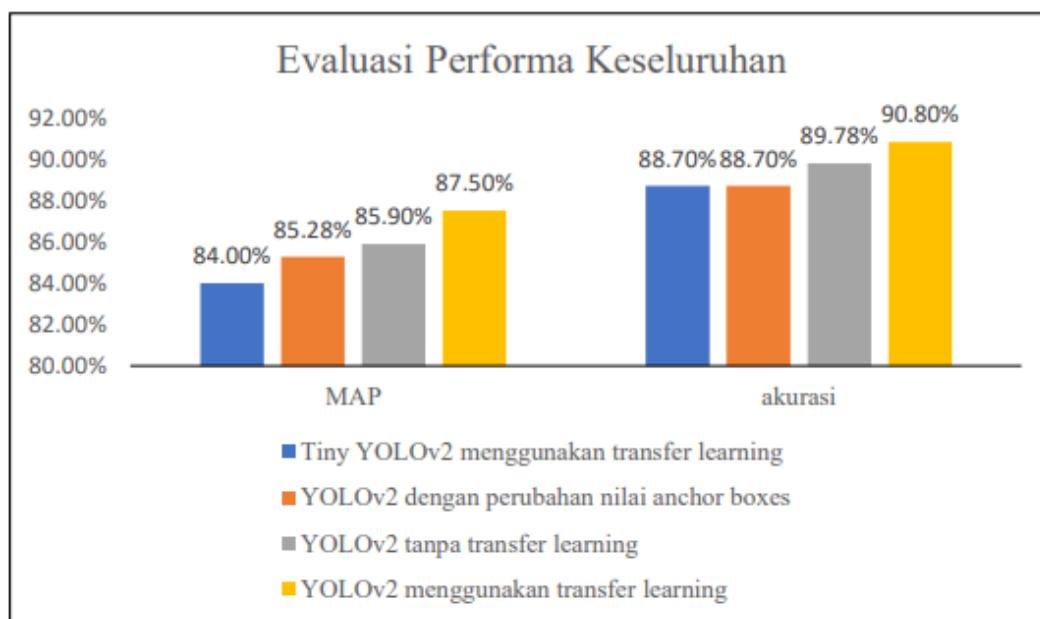


Gambar 2.24. Kurva *loss function* pada pelatihan *Tiny YOLOv2* [16].



Gambar 2.25. Kurva *loss function* pada pelatihan *YOLOv2* [16].

Kurva *loss function* ditampilkan pada Gambar 2.24. dan Gambar 2.25. yang menunjukkan tren konvergen dan mengartikan model tidak mengalami *overfitting*. Pemilihan nilai ambang batas dilakukan pada data validasi dengan pemilihan nilai ambang batas dilakukan dengan rentang antara 0,1 hingga 0,6, dengan jarak pemilihan ambang batas tiap 0,05. Nilai ambang batas yang menghasilkan mAP tertinggi ada pada nilai ambang batas 0,1. Pada penelitian ini digunakan *dataset* sebanyak 705 citra data latih, 154 citra data validasi, dan 154 citra data uji. Pada seluruh data citra tersebut terdapat objek sel limfoblas sebanyak 632, sel non-limfoblas sebanyak 629. Secara keseluruhan didapatkan performa pelatihan pada penelitian ini yang ditunjukkan pada Gambar 2.26. Model arsitektur YOLOv2 dengan *transfer learning* dan tanpa melakukan perubahan parameter *anchor boxes* memiliki performa terbaik pada pelatihan penelitian ini. Adapun arsitektur *Tiny YOLOv2* yang dilatih dengan *transfer learning* memiliki performa paling rendah di antara arsitektur pelatihan lainnya.



Gambar 2.26. Evaluasi performa seluruh model pelatihan [16].

II.3. Perhitungan Sel Darah Otomatis dengan basis Pendektsian, Segmentasi Instance, dan Pengolahan Citra

De Rong Loh, Wen Xin Yong, Jullian Yapeter, Karupppasamy Subburaj, dan Rajesh Chandramohanadas pada tahun 2021 telah melakukan penelitian mengenai perhitungan sel darah otomatis dan cepat untuk skrining infeksi malaria dengan menggunakan pendektsian objek dan segmentasi *instance* berbasis *Mask R-CNN* [27]. Penelitian tersebut menggunakan *dataset* berupa RBC yang tidak terinfeksi dan yang terinfeksi oleh plasmodium falciparum [27]. *Dataset* tersebut terdiri dari 297 citra, di antaranya 173 untuk pelatihan, 64 untuk validasi, dan 60 untuk pengujian dalam rasio perkiraan 3:1:1 [27]. Kumpulan citra tersebut dianotaskan secara manual menggunakan VGG *Image Annotator* (VIA 2.0.4) sebelum diekspor sebagai *file json* untuk fase pelatihan [27,28]. Dari lima kelas yang teridentifikasi, hanya empat kelas dalam pelabelan, yaitu retikulosit yang tidak terinfeksi, cincin, trofozoit, dan skizon [27].

Teknik *image processing* dari cv2 dan *scikit-image* digunakan dalam algoritma perhitungan sel tersebut [27,29,30]. Pada citra, kontras ditingkatkan dengan menerapkan pemerataan histogram pada ruang warna terang citra [27,31]. Operasi morfologi, seperti *opening* dan *erosion* dilakukan untuk menghilangkan *noise* citra [27,32]. Selain mengoptimalkan jumlah operasi morfologi yang digunakan, *non-maximum suppression thresholds* dan *min confidence levels* digunakan untuk lebih selektif dalam mengidentifikasi *regions* dan *bounding boxes* untuk sel [27]. Algoritma Watershed digunakan untuk mensegmentasi setiap sel, dan pada akhirnya jumlah total sel dapat dihitung [27,33].

Kinerja model dievaluasi dengan membandingkan prediksinya dengan jumlah sel spesifik secara independen oleh dua peneliti yang berpengalaman lebih dari 5 tahun dalam menghitung RBC yang terinfeksi plasmodium [27]. Secara khusus, *confusion matrix* dan *mean Average Precision* (mAP) digunakan pada penelitian tersebut karena merupakan standar metrik untuk deteksi objek [27]. Parameter untuk *confusion matrix* pada penelitian tersebut didefinisikan dalam persamaan berikut:

$$Accuracy \text{ in detecting cells} = \frac{Cluster A + Cluster B}{Cluster A + Cluster B + Cluster C} \quad (2.1)$$

$$\begin{aligned} & Accuracy \text{ in labelling infected cells} \\ &= \frac{Cluster A}{Cluster A + Cluster B + Cluster C} \end{aligned} \quad (2.2)$$

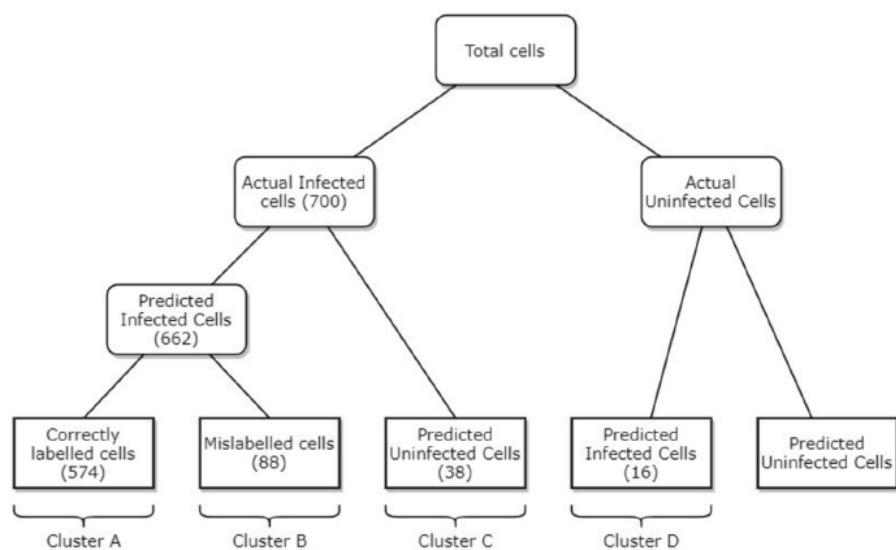
$$Error \text{ on labelling infected cells} = \frac{Cluster C + Cluster D}{Total \text{ Manual Cell Count}} \quad (2.3)$$

mAP keseluruhan dihitung dengan mengambil rata-rata skor *Average Precision* (AP) interpolasi 11 titik yang dihitung dari setiap kelas berdasarkan nilai presisi dan *recall* dengan nilai ambang batas *Intersection over Union* (IoU) sebesar 0,5 [27]. Model yang digunakan pada penelitian mencapai akurasi 94,57% dalam mendekripsi sel yang terinfeksi dengan *error* 0,55%, dan akurasi 82% dalam pelabelan bahwa sel benar-benar terinfeksi [27]. Nilai akurasi yang tinggi juga dikuatkan oleh persentase *true positives* yang tinggi di mana sebagian besar sel yang terinfeksi diprediksi dengan benar sebagai positif yang sebenarnya ditujukkan [27]. *Confusion matrix* pada Tabel 2.12. menyediakan data untuk menghitung hasil pada Gambar 2.27. dan Gambar 2.28.

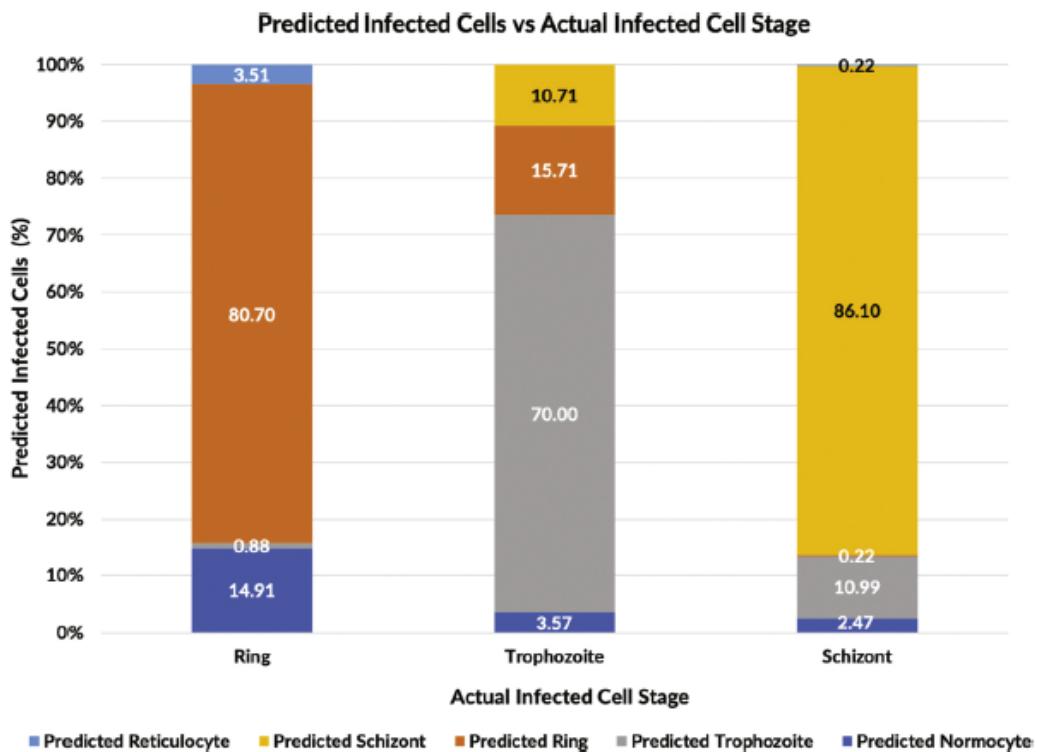
Tabel 2.12. Confusion matrix ini hanya menargetkan sel yang terinfeksi dalam 60 citra.

		Predicted cell types				
		normosit	cincin	trofozoit	skizon	retikulosit tidak terinfeksi
Actual cell types (verified through microscopy)	normosit	0	15	0	1	6
	cincin	17	92	1	0	4
	trofozoit	5	22	98	15	0
	skizon	11	1	49	384	1
	retikulosit tidak terinfeksi	0	0	0	0	11

Sumber: [27]



Gambar 2.27. Diagram pohon diturunkan dari Tabel 2.11. Jumlah normosit sebenarnya dihilangkan karena bukan salah satu dari dataset berlabel [27].



Gambar 2.28. Bagan batang dari *true results* vs. persentase *predicted results* [27].

Tabel 2.13. Hasil Algoritma Perhitungan Sel.

Metode	Jumlah Sel Total	Perbedaan (%)
Perhitungan Manual	6908	10.44
Perhitungan Sel	6187	

Sumber: [27]

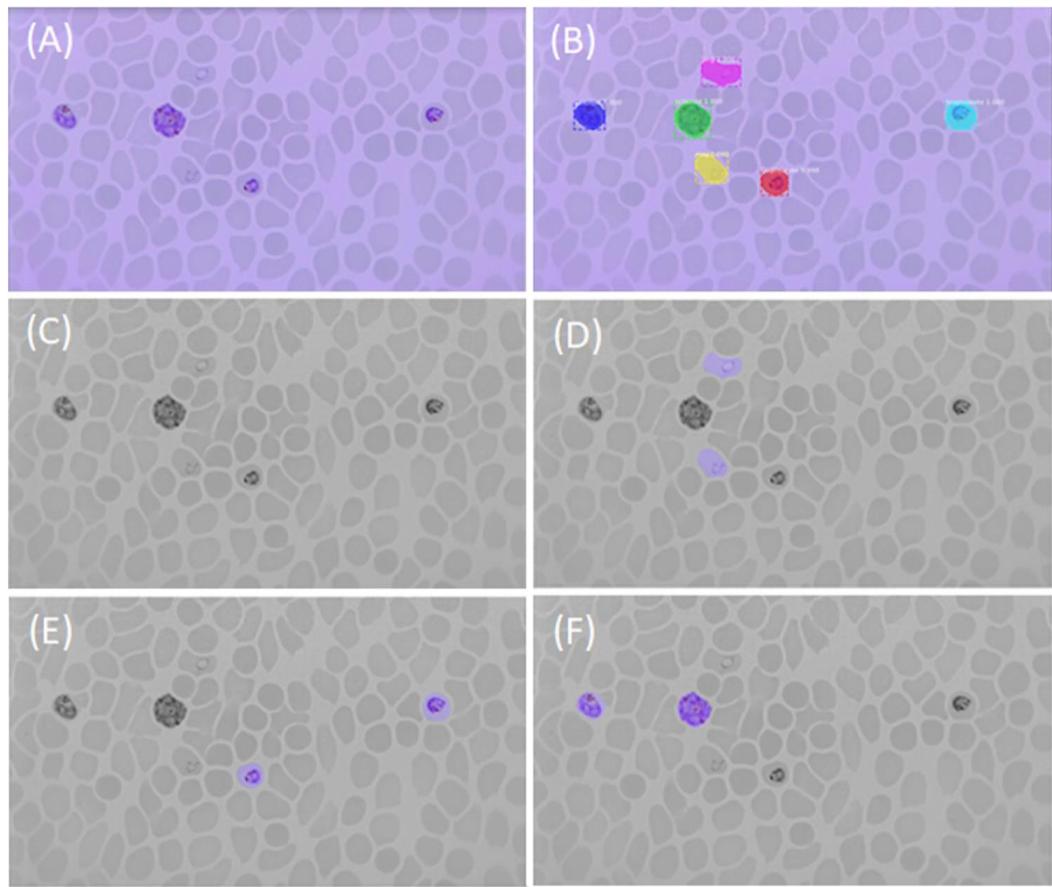
mAP keseluruhan dievaluasi menjadi 0,731 dengan mengambil rata-rata 11 titik interpolasi AP dari semua 4 kelas berlabel: cincin (0,649), trofozoit (0,462), schizont (0,814) dan retikulosit (1,000) [27]. Hampir semua proposal *Region of Interest* (RoI) dihasilkan dengan nilai IoU tinggi dengan RoI *ground-truth* seperti yang terlihat dari kotak ketat di sekitar batas sel pada Gambar 2.29. [27]. Peneliti-peneliti tersebut juga menganggap nilai mAP yang diperoleh cukup memuaskan

karena skor tidak akan turun drastis bahkan pada nilai ambang batas IoU yang lebih tinggi [27]. Secara khusus, model penelitian tersebut menunjukkan peningkatan yang signifikan dari model *Faster R-CNN*, meskipun beroperasi pada nilai *minimum confidence threshold* yang lebih tinggi yaitu 0,8 dibandingkan dengan 0,65 pada model *One Stage Faster R-CNN*, model tersebut mampu memprediksi cincin dan sel schizont dengan *error* yang lebih kecil [27].

Tabel 2.14. Perbandingan setiap tahap dengan *Faster R-CNN*.

Kelas	Error Margin (%)	
	<i>One Stage Faster R-CNN</i>	<i>Mask R-CNN</i>
cincin	157.95	5.15
trofozoit	18.36	27.41
skizon	39.28	11.72

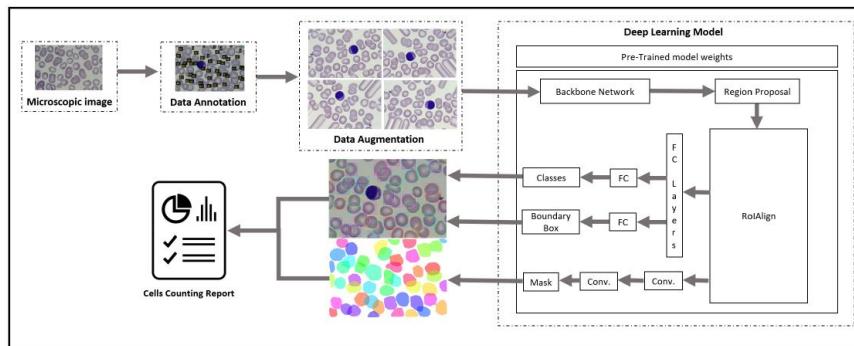
Sumber: [27]



Gambar 2.29. Citra secara otomatis dianalisis oleh model *Mask R-CNN* [27]. (A) Citra asli diambil dengan objektif imersi 100 \times , (B) Klasifikasi *bounding box* dan probabilitas kelas untuk semua retikulosit dan iRBC yang tidak terinfeksi yang diekstraksi, (C-F) *Mask* selektif untuk tahap ekstraksi dengan urutan sebagai berikut: Retikulosit, Cincin, Trofozoit, dan Skizon yang tidak terinfeksi.

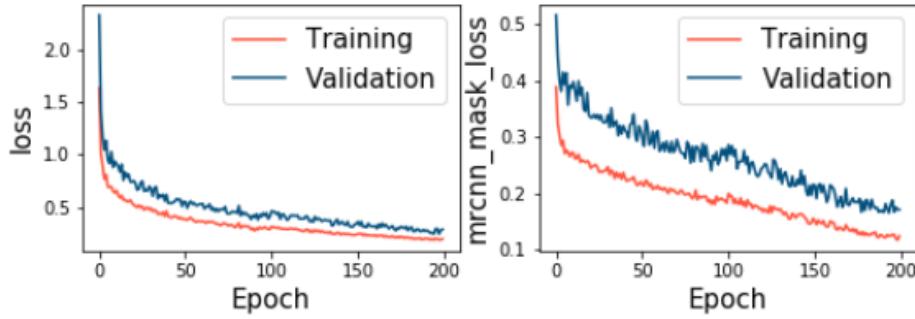
Pada tahun 2019, Najmeddine Dhibe, Hakim Ghazzai, Hichem Besbes, dan Yehia Massoud telah melakukan penelitian serupa, yakni perhitungan sel darah otomatis menggunakan CNN, segmentasi *instance*, dan *transfer learning*, berbasis *Mask R-CNN* [13]. RBC dan WBC diidentifikasi, diklasifikasikan, dan dihitung dari gambar apusan darah mikroskopis [13]. Hasil yang diperoleh menunjukkan tingkat deteksi yang tinggi dari sel darah yang berbeda [13]. Selain itu, tidak seperti teknik *state-of-the-art* lainnya, metode yang diusulkan peneliti-peneliti tersebut memiliki kemampuan untuk mengidentifikasi sel yang tumpang tindih dan memudar [13].

Tujuan dari penelitian tersebut adalah untuk menerapkan teknik segmentasi citra untuk melokalisasi, memprediksi batas, mengklasifikasikan, dan menghitung RBCs dan WBCs pada citra apusan darah [13]. Dalam penelitian tersebut digunakan *Mask R-CNN* tidak hanya karena efisiensinya dalam deteksi dan klasifikasi objek, tetapi juga untuk kinerja dalam segmentasi *instance* [13]. Sebagai hasilnya, metode tersebut dapat digunakan untuk mendeteksi sel-sel yang berbeda pada apusan darah dan membedakan antara sel yang tumpang tindih dan objek yang termasuk dalam kelas yang sama dan dapat dilakukan perhitungan [13]. Penelitian tersebut mengusulkan penggunaan ResNet-101 sebagai *backbone* dari model *feature pyramid network* (FPN) [13].



Gambar 2.30. Kerangka kerja segmentasi dan penghitung sel yang diusulkan.
Dataset citra mikroskopis diberi label dan dimasukkan ke dalam model *deep learning* yang melakukan segmentasi sel, mengklasifikasikan ke dalam WBC dan RBC, dan menghitung jumlahnya masing-masing [13].

Pada penelitian tersebut digunakan *dataset* citra apusan darah mikroskopis yang mencakup 150 citra dengan lebih dari 24.000 sampel sel [13]. Proses anotasi dan pelabelan pada citra apusan darah dilakukan menggunakan VGG *Image Annotator* secara manual [13,28]. Penelitian tersebut menggunakan teknik augmentasi pada data latih untuk meningkatkan efisiensi model *deep learning* dan mengurangi masalah *overfitting* [13]. Rotasi acak, *zooming*, pergeseran, dan transformasi membalik juga diterapkan untuk memperluas dan memvariasikan data [13].



Gambar 2.31. Evolusi fungsi *loss* [13].

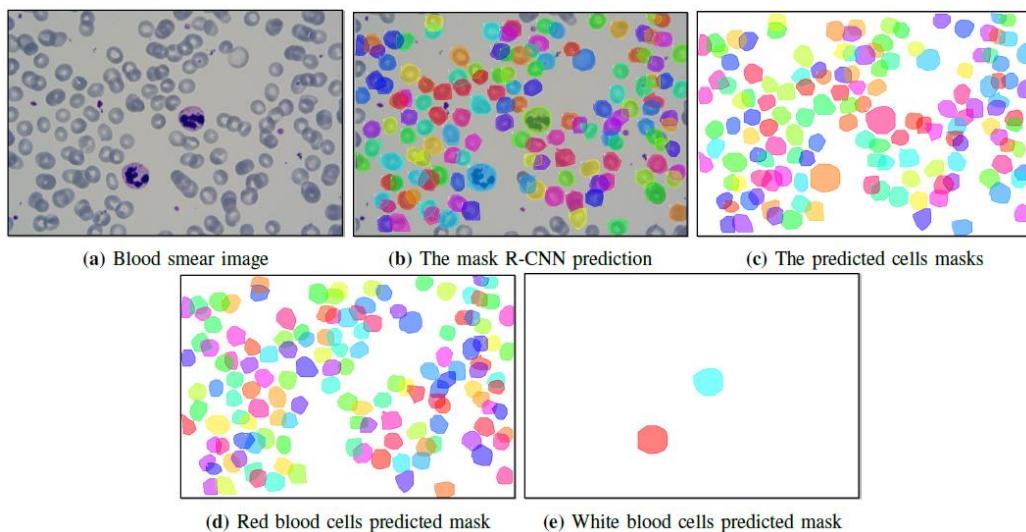
Pada penelitian tersebut untuk mengatasi tumpang tindih antar perbedaan *anchors* yang dihasilkan maka digunakan satu nilai rasio IoU yang tinggi [13]. Dalam penelitian tersebut digunakan *threshold* IoU bernilai 0,65 untuk memaksimalkan deteksi sel [13]. Selain itu, penelitian tersebut menggunakan fungsi *loss multi-taks* pada Persamaan (2.4) untuk memvisualisasikan dan mengevaluasi model *Mask R-CNN* selama fase pelatihan dan pengujian [13]. Fungsi *loss* ini menggabungkan klasifikasi *loss*, segmentasi *loss*, dan *mask loss* yang dihasilkan [13]. *Loss* klasifikasi dan *bounding-box*, L_{cls} dan L_{box} , sama dengan yang didefinisikan dalam [13,34]. Mengingat pada penelitian tersebut bahwa nilai *mask* yang dihasilkan adalah biner, fungsi *mask loss* L_{mk} dapat didefinisikan dengan *average binary cross-entropy loss* termasuk hanya *mask* k^{th} jika *region* dikaitkan dengan kelas *ground-truth* k^{th} sebagai Persamaan (2.5).

$$L = L_{cls} + L_{box} + L_{mk} \quad (2.4)$$

$$L_{mk} = -\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n [y_{ij} \log(\hat{y}_{ij}^k) + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)] \quad (2.5)$$

Di mana y_{ij} adalah label sel (i,j) pada *real mask* dari $n \times n$ *region* dan \hat{y}_{ij}^k adalah nilai prediksi dari sel yang sama pada *mask* dipelajari untuk kelas *ground-truth* k [13]. Evolusi fungsi *loss* L dan fungsi *loss* *Mask R-CNN* (L_{mk}) ditunjukkan pada Gambar 2.31. Pada evolusi fungsi *loss* tersebut tersebut dapat dilihat bahwa selama pelatihan dan pengujian, fungsi *loss* terus menurun dan konvergen ke nilai

mendekati nol. Hal tersebut membuktikan bahwa model *deep learning* yang dikembangkan sudah terlatih dengan baik tanpa adanya *overfitting* [13]. Pada Gambar 2.32., *Mask R-CNN* berhasil tidak hanya dalam indentifikasi dan segmentasi sel, tetapi juga dalam klasifikasi sel [13]. Selain itu, eksperimen menunjukkan bahwa model yang diusulkan secara efisien mendeteksi 92% RBC dan 96% WBC [13].



Gambar 2.32. Evolusi citra apusan darah selama proses segmentasi dan klasifikasi [13].

Yang-Hsien Lin, Ken Y.-K. Liao, dan Kung-Bin Sung pada tahun 2020 telah melakukan penelitian dengan judul “*Automatic detection and characterization of quantitative phase images of thalassemic red blood cells using a mask region-based convolutional neural network*” [35]. Pada penelitian tersebut model *deep learning* dikembangkan secara efisien untuk membedakan Sel Darah Merah *thalassemic* (tRBCs) dari Sel Darah Merah sehat (hRBCs) dalam citra fase kuantitatif dan segmentasi RBC untuk karakterisasi sel tunggal [35]. Citra fase kuantitatif dua dimensi dari hRBC dan tRBC diperoleh menggunakan mikroskop holografik digital [35]. Kemudian, model *Mask R-CNN* dilatih untuk membedakan tRBC dan mengelompokkan RBC secara individu [35].

Model yang diimplementasikan menggunakan *Mask R-CNN* tersebut mencapai akurasi 97,8% dalam mendeteksi tRBC (*test dataset*) [35]. Pada penelitian ini pemerikasaan apusan darah digunakan *Quantitative Phase Imaging* (QPI) karena mampu memetakan dalam dua dimensi pergeseran fasa yang disebabkan oleh konstituen seluler ketika berkas cahaya sumber mentransmisikan melalui sel transparan [35]. *Single-shot Digital Holographic Microscopy* (DHM) digunakan untuk memperoleh citra fase kuantitatif dua dimensi dari RBC yang dikumpulkan dari pasien thalassemia-minor dan subjek sehat [35]. Akurasi klasifikasi model dibandingkan dengan pengklasifikasi optimal yang dibangun dengan XGBoost dan dilatih dengan 15 fitur sel tunggal, termasuk statistik pergeseran fase, fitur morfologi 2D, dan fitur tekstur yang diekstraksi dari fase RBC yang tersegmentasi secara manual [35]. Daftar fitur yang digunakan ditampilkan pada Tabel 2.15.

Model *Mask R-CNN* juga didemonstrasikan keunggulannya oleh peneliti dengan mengekstrak fitur sel tunggal dari citra fase RBC yang tersegmentasi secara otomatis dan melatih pengklasifikasi XGBoost kedua untuk membedakan tRBC dari yang sehat [35]. Penelitian tersebut menggunakan 2.001 citra hRBCs dan 4.268 citra tRBCs [35]. Jumlah citra tRBC sekitar dua kali lebih banyak daripada citra hRBC, oleh karena itu penelitian tersebut mengadopsi augmentasi data dengan rotasi acak, untuk memperbesar *dataset* hRBC dan untuk menyeimbangkan data pelatihan [35]. Operasi rotasi acak juga meningkatkan variasi dalam *dataset* yang dapat mencegah *overfitting* selama pelatihan model *Mask R-CNN* [35].

Dataset tersebut dibagi menjadi data latih, validasi, dan pengujian dengan rasio 3:1:1 [35]. Parameter *learning rate* sebesar 0,002, *batch size* sebesar 64, dan *steps per epoch* sebesar 80 [35]. Pada Tabel 2.16. merangkum kinerja klasifikasi model *Mask R-CNN* dan pengklasifikasi XGBoost *benchmark* pada *dataset* pengujian [35]. Model *Mask R-CNN* mencapai akurasi 97,8% dan rata-rata IoU sebesar 0,945 dalam *test dataset* yang sangat dekat dengan akurasi yang diperoleh oleh *benchmark XGBoost classifier* (akurasi 99,9%) [35].

Tabel 2.15. Daftar fitur yang digunakan dalam XGBoost *classifier*.

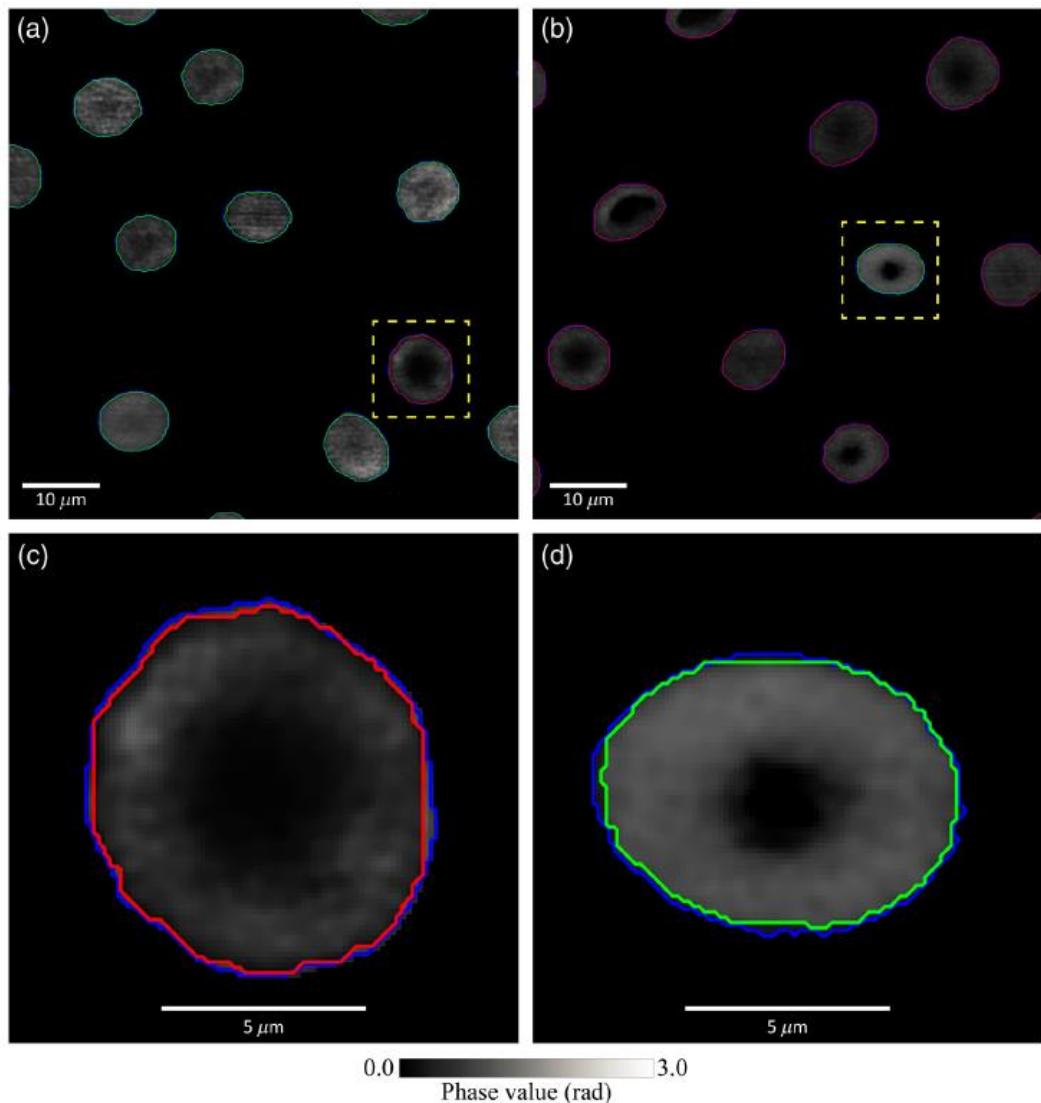
Kategori	Fitur
Statistik pergeseran fase (n = 5)	Volume optik, rata-rata, persentil kelima, persentil ke-95, dan standar deviasi pergeseran fasa
Sifat morfologis (n = 5)	Luas proyeksi, keliling, panjang sumbu mayor, panjang sumbu minor, dan eksentrisitas
Properti tekstur berdasarkan GLCM (n = 5)	Kontras, ketidakmiripan, homogenitas, energi, dan momen kedua sudut 49

Sumber: [35]

Tabel 2.16. Perbandingan kinerja klasifikasi pada RBC individu menggunakan *Mask R-CNN*, XGBoost dengan delineasi manual, dan XGBoost dengan hasil segmentasi *Mask R-CNN*.

Metode	<i>Sensitivity</i> (%)	<i>Specificity</i> (%)	F1-score	AUC	Akurasi (%)
<i>Mask R-CNN</i>	97,1	98,5	0,975	0,982	97,8
XGBoost dengan MD	99,9	100	0,992	0,999	99,9
XGBoost dengan <i>Mask R-CNN</i>	99,9	100	0,992	0,999	99,9

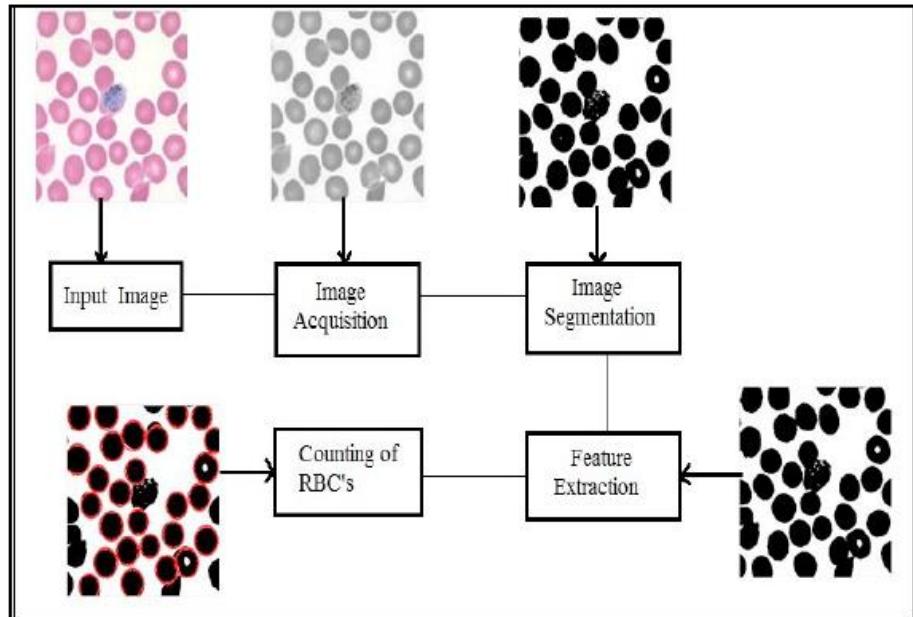
Sumber: [35]



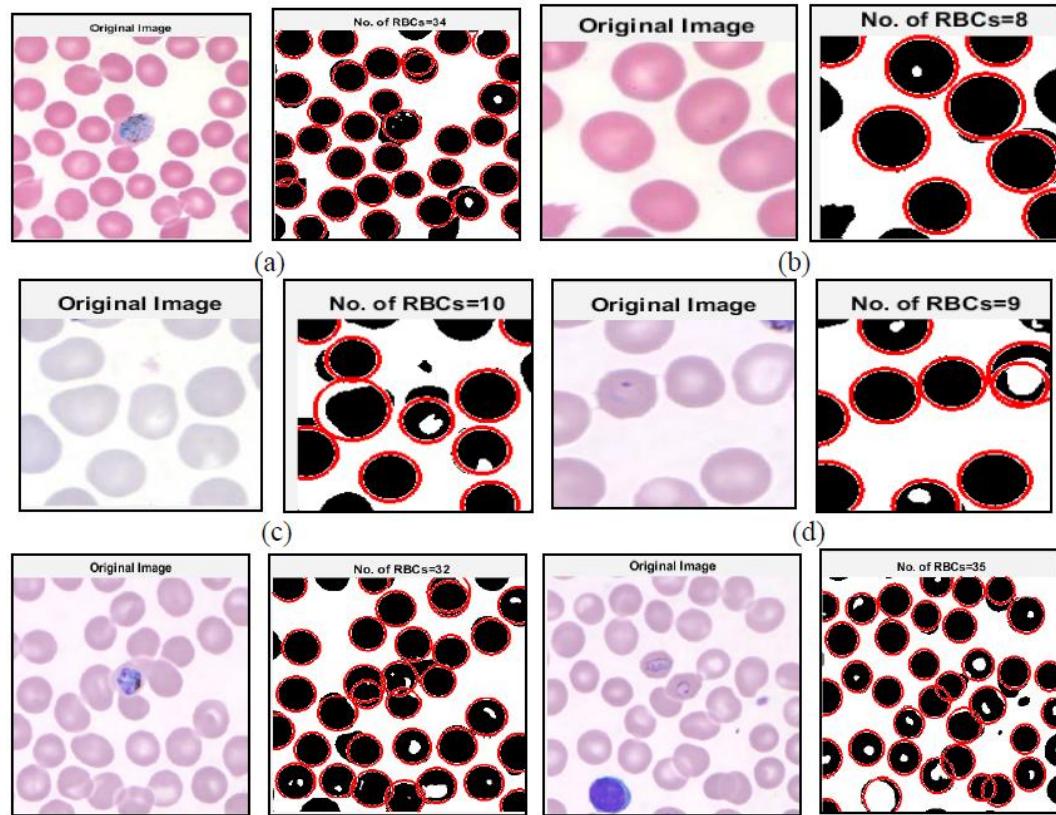
Gambar 2.33. Contoh hasil delineasi dan deteksi dengan model *Mask R-CNN*. Dua *slide* dengan hRBC dan tRBC ditunjukkan pada (a) dan (b). Garis biru adalah kontur penggambaran manual sel darah merah sebagai *ground truth*. Garis hijau dan merah menandai kontur yang diprediksi mode dari sel darah merah yang diklasifikasikan sebagai hRBC dan tRBC, masing-masing. Kasus yang salah diklasifikasikan dalam (a) dan (b) diperbesar dan ditunjukkan dalam (c) dan (d).

Kontur yang diprediksi model cocok dengan kontur penggambaran manual, menunjukkan bahwa kesalahan klasifikasi bukan karena segmentasi yang gagal [35].

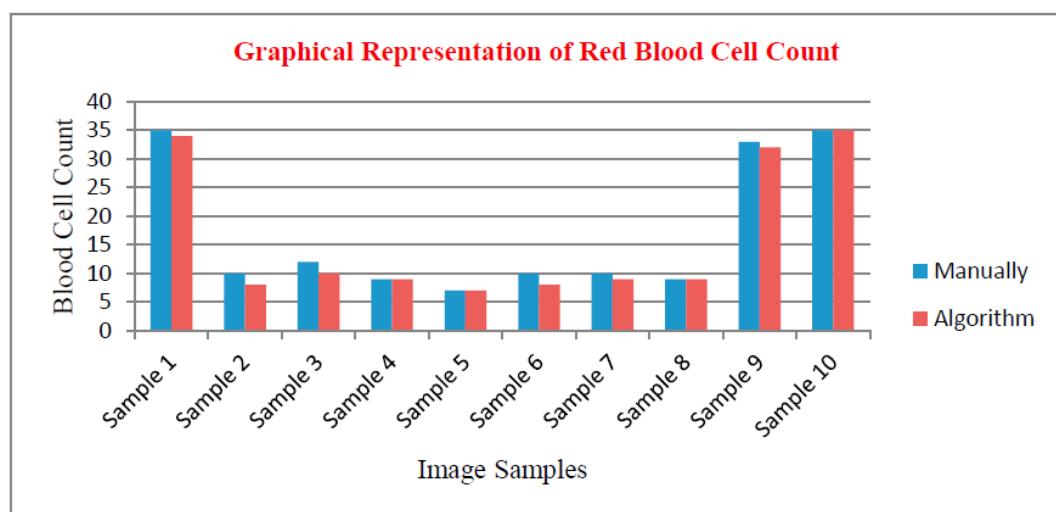
Gulpreet Kaur Chadha, Aakarsh Srivastava, Abhilasha Singh, Ritu Gupta, dan Deepanshi Singla pada tahun 2020 melakukan penelitian mengenai otomatisasi untuk mendeteksi dan menghitung RBC berdasarkan fungsi morfologi menggunakan ekstraksi fitur, *thresholding segmentation*, dan perhitungan sel menggunakan *Hough Transformation* [36]. Langkah yang paling signifikan dan rumit adalah segmentasi sel darah karena jumlah sel yang tepat didasarkan pada segmentasi sel darah yang akurat [36]. Algoritma yang diusulkan mencapai akurasi keseluruhan 91,667% dan komputasi sangat efisien karena hanya membutuhkan 0,81432 detik untuk menghitung jumlah sel darah merah untuk sampel darah yang berbeda [36]. Proses tersebut terdiri dari akuisisi citra diikuti oleh berbagai operasi pemrosesan citra yaitu *thresholding segmentation*, *median filtering*, *edge detection*, *erosion*, dan perhitungan sel menggunakan *Hough Transformation* seperti yang ditunjukkan pada Gambar 2.34. dan Gambar 2.35. [36].



Gambar 2.34. Metodologi yang diusulkan peneliti [36].



Gambar 2.35. Beberapa gambar sampel (kiri) yang digunakan dalam eksperimen dan gambar keluaran masing-masing (kanan) [36].



Gambar 2.36. Perbandingan perhitungan manual dan otomatis [36].

Grafik pada Gambar 2.36. adalah perbandingan hasil antara perhitungan RBC menggunakan algoritma yang diusulkan dan melalui perhitungan manual menggunakan hemasitometer melalui slide mikroskopis [36]. Dari grafik tersebut terlihat bahwa hasil yang diperoleh dari algoritma yang diusulkan menawarkan konsistensi yang baik dibandingkan dengan perhitungan manual [36]. Hasil perhitungan akurasi, *sensitivity*, dan *specificity* ditunjukkan pada Tabel 2.17.

Tabel 2.17. Hasil perhitungan metrik.

Metrik	Hasil
<i>Accuracy</i>	91,667%
<i>Specificity</i>	87,50%
<i>Sensitivity</i>	100%

Sumber: [36]

Tabel 2.18. Rangkuman tinjauan pustaka.

Tahun	Nama peneliti	Judul penelitian	Metode yang diusulkan peneliti	Hasil dan kesimpulan	Dataset
2021	Pradeep Kumar Das dan Sukadev Meher	<i>An efficient deep Convolutional Neural Network based detection and classification of Acute Lymphoblastic Leukemia</i>	Menggunakan <i>probability-based weight factor</i> untuk menghibridisasi MobilenetV2 dan ResNet18.	Akurasi terbaik masing-masing nilainya adalah 99,39% untuk ALL-IDB1 dan 97,18% untuk ALL-IDB2.	ALL-IDB1 dan ALL-IDB2.
2020	Ahmed T. Sahlol, Philip Kollmannsberger, dan Ahmed A. Ewees	<i>Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features</i>	Menggunakan ekstraksi fitur VGGNet, fitur yang diekstraksi tersebut kemudian diseleksi menggunakan algoritma <i>Salp Swarm Algorithm</i> (SESSA) dan digunakan enam algoritma pengklasifikasi (Linear SVM, KNN, <i>Decision Trees</i> , Naive Bayes, Adaboost, dan <i>Multi-Layer Perceptron</i>).	Akurasi yang diperoleh 96,11% untuk ALL-IDB2 dan 83,3% untuk C-NMC.	ALL-IDB1, ALL-IDB2, dan C-NMC.
2020	Jonas Prellberg dan Oliver Kramer	<i>Acute Lymphoblastic Leukemia Classification from Microscopic Images</i>	Menggunakan model ResNeXt50 dan augmentasi data berupa pembalikan horizontal	Akurasi yang diperoleh selama 24 <i>training-runs</i> adalah $87,96 \pm 0,90$.	C-NMC

		<i>using Convolutional Neural Networks</i>	dan vertikal, rotasi, dan translasi.		
2021	Rohan Khandekar, Prakhyaa Shastry, Smruthi Jaishankar, Oliver Faust, dan Niranjana Sampathila	<i>Automated blast cell detection for Acute Lymphoblastic Leukemia diagnosis</i>	Menggunakan YOLOv4 dan augmentasi data berupa pembalikan horizontal dan vertikal, rotasi, rentang kecerahan, gaussian blur, dan noise.	Nilai mAP dan <i>F1-score</i> yang diperoleh sebesar 95,57% dan 92% untuk <i>dataset</i> ALL-IDB1 dan 98,57% dan 92% untuk <i>dataset</i> C-NMC 2019.	ALL-IDB1 dan C-NMC 2019.
2019	Arif Muntasar dan Muhammad Yusuf	<i>Modeling of the Acute Lymphoblastic Leukemia Detection based on the Principal Object Characteristics of the Color Image</i>	Menggunakan empat tahap berikut: <i>enhancement, segmentation, feature extraction, dan accuracy measurement.</i>	Akurasi maksimum masing-masing sebesar 81,54% untuk Euclidean Distance, 81,54% untuk Manhattan, 76,92% untuk Canberra, dan 82,31% untuk Chebyshev.	ALL-IDB2
2020	Muhamad Inska Fajri	Pendeteksian Sel Limfoblas pada Kasus <i>Acute Lymphoblastic Leukemia</i> Subtipe L1 Berbasis <i>Convolutional Neural Networks</i> dengan Metode <i>Faster Region-Based</i>	Menggunakan <i>Faster R-CNN</i> berbasis ResNet50 dan ResNet101 serta penggunaan augmentasi data berupa pencerminan sumbu-y.	Model terbaik diperoleh akurasi sebesar 86,2%, <i>F1-Score</i> sebesar 87,9%, dan mAP sebesar 90,9%”.	Citra sel darah dari RSUP Dr. Sardjito

		<i>Convolutional Neural Networks</i>			
2020	Arnanda Prabaswara	Deteksi Sel Limfoblas Pada Kasus <i>Acute Lymphoblastic Leukemia</i> Tipe L1 menggunakan Metode <i>Convolutional Neural Network</i>	Menggunakan metode pendekripsi objek YOLOv2.	Nilai akurasi sebesar 90,9% dan mAP sebesar 87,5%.	Citra sel darah dari RSUP Dr. Sardjito
2021	De Rong Loh, Wen Xin Yong, Jullian Yapeter, Karupppasamy Subburaj, dan Rajesh Chandramohanadas	<i>A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN</i>	Menggunakan <i>Mask R-CNN</i> dan <i>image processing</i> berupa peningkatan kontras, pemerataan histogram, dan operasi morfologi.	Akurasi sebesar 82,0% dan mAP sebesar 73,1%.	<i>Interstate blood bank (USA)</i>
2019	Najmeddine Dhib, Hakim Ghazzai, Hichem Besbes, dan Yehia Massoud	<i>An Automated Blood Cells Counting and Classification Framework using Mask R-CNN Deep Learning Model</i>	Menggunakan <i>Mask R-CNN</i> dengan <i>backbone</i> ResNet-101 dan augmentasi data berupa rotasi acak, <i>zooming</i> , pergeseran, dan transformasi membalik.	Dapat mendekripsi 92% RBC dan 96% WBC.	<i>Dataset pribadi peneliti</i>

2020	Yang-Hsien Lin, Ken Y.-K. Liao, dan Kung-Bin Sung	<i>Automatic detection and characterization of quantitative phase images of thalassemic red blood cells using a mask region-based convolutional neural network</i>	Menggunakan <i>Mask R-CNN</i> dan augmentasi data berupa rotasi acak.	Akurasi sebesar 97,5% dan F1-score sebesar 97,8%.	<i>Dataset pribadi peneliti</i>
2020	Gulpreet Kaur Chadha, Aakarsh Srivastava, Abhilasha Singh, Ritu Gupta, dan Deepanshi Singla	<i>An Automated Method for Counting Red Blood Cells using Image Processing</i>	Menggunakan pengolahan citra berupa <i>thresholding segmentation, median filtering, edge detection, erosion</i> , dan perhitungan sel menggunakan <i>Hough Transformation</i> .	Akurasi sebesar 91,667%, <i>Specificity</i> sebesar 87,50%, dan <i>Sensitivity</i> sebesar 100%.	<i>Dataset pribadi peneliti</i>

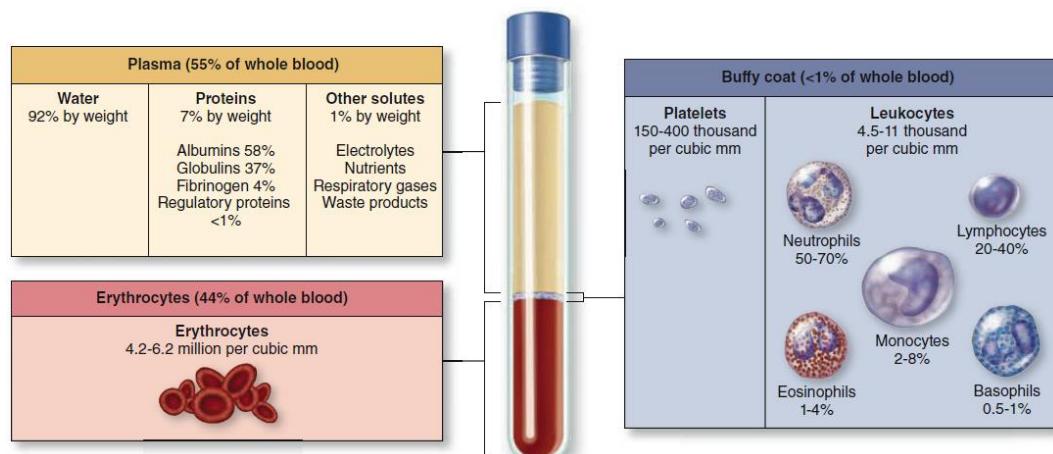
BAB III

DASAR TEORI

Pada bab ini akan diuraikan berbagai teori yang berkaitan dengan *Acute Lymphoblastic Leukemia L1* dan metode *Mask R-CNN*. Dasar teori mengenai sel darah, proses pematangan dan morfologinya, *Deep Learning*, *Computer Vision*, *Mask Regional-Convolutional Neural Network*, *Hyperparameter*, dan metrik evaluasi performa yang digunakan dalam penelitian ini dijelaskan pada bab berikut.

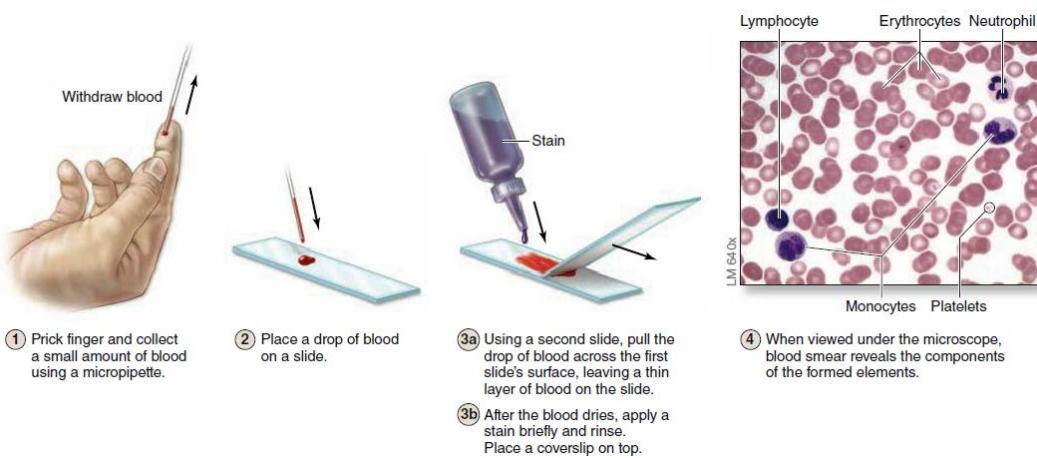
III.1. Sel Darah

Darah adalah jaringan ikat khusus yang terdiri dari sel dan bahan cairan ekstraseluler yang disebut plasma [1]. Unsur pembentuk yang beredar dalam plasma adalah eritrosit (RBC), leukosit (WBC), dan trombosit (PLT) [1]. Sebuah tabung darah setelah sentrifugasi (bagian tengah) memiliki hampir setengah dari volume yang diwakili oleh eritrosit (RBC) pada bagian bawah tabung pada Gambar 3.1. [1]. Di antara eritrosit (RBC) yang terendapkan dan plasma supernatan berwarna terang terdapat lapisan tipis leukosit (WBC) dan trombosit yang disebut *buffy coat* [1].



Gambar 3.1. Komposisi darah lengkap [36].

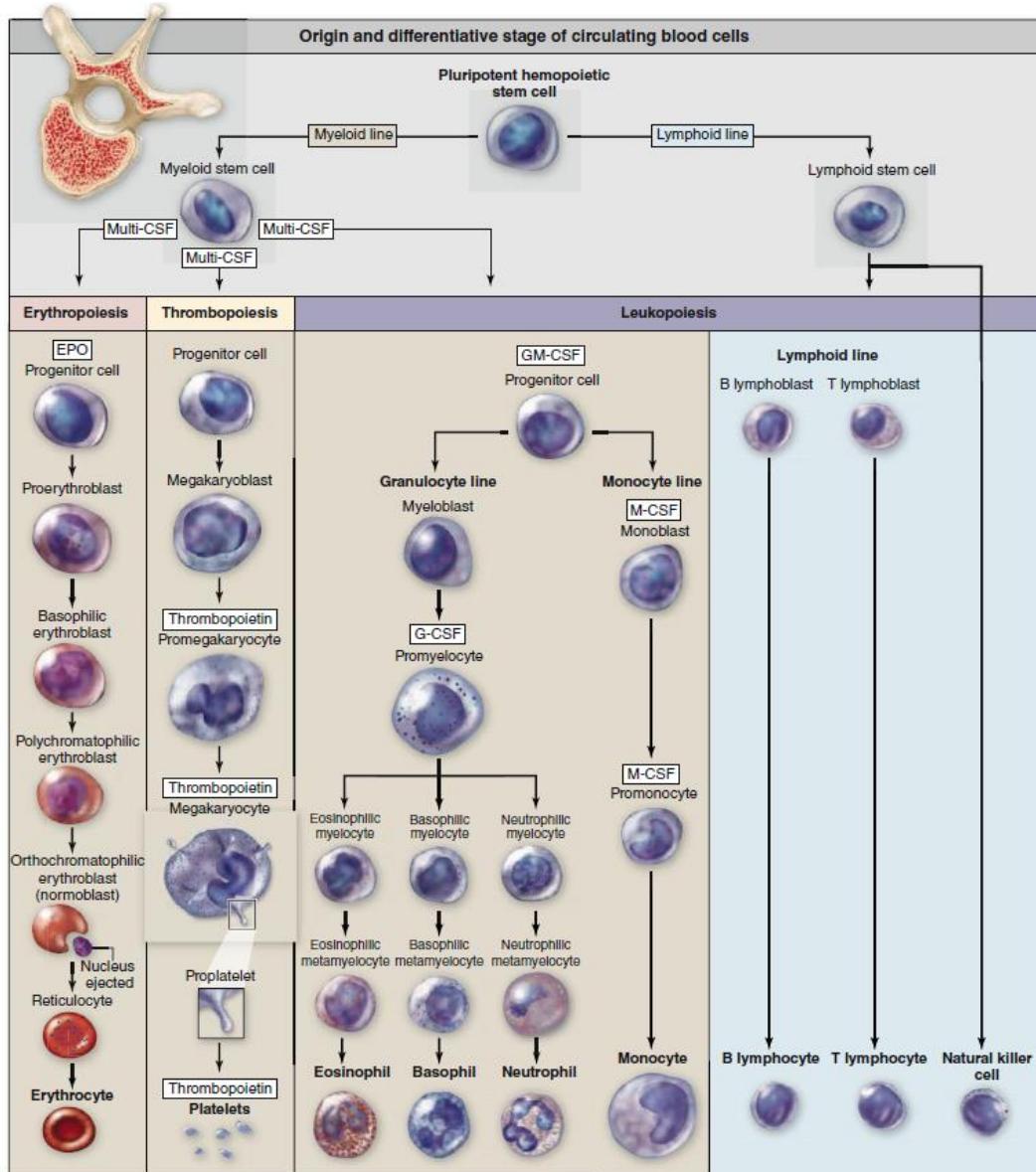
Sel darah dapat dipelajari secara histologis dalam sediaan apusan yang dibuat dengan menyebarkan setetes darah dalam lapisan tipis pada *slide* mikroskop ditunjukkan pada Gambar 3.2. [1]. Dalam film tersebut sel terlihat jelas dan berbeda satu sama lain, memfasilitasi pengamatan nuclei dan karakteristik sitoplasma [1].



Gambar 3.2. Penyiapan apusan darah [1].

III.1.1. Hemopoiesis

Semua sel darah muncul dari satu jenis sel punca hematopoietik pluripoten di sumsum tulang yang dapat menghasilkan semua jenis sel darah seperti yang ditunjukkan pada Gambar 3.3. [1]. Sel punca hemopoietik pluripoten membelah secara perlahan, mempertahankan populasinya sendiri, dan memunculkan dua garis keturunan sel utama sel progenitor: sel punca myeloid dan limfoid [1]. Garis keturunan myeloid adalah sel prekursor (*blasts*) untuk eritropoiesis, trombopoiesis, granulopoiesis, dan monocytopenies [1]. Garis keturunan limfoid membentuk limfosit B dan T dan sel terkait yang disebut sel *Natural Killer* [1].

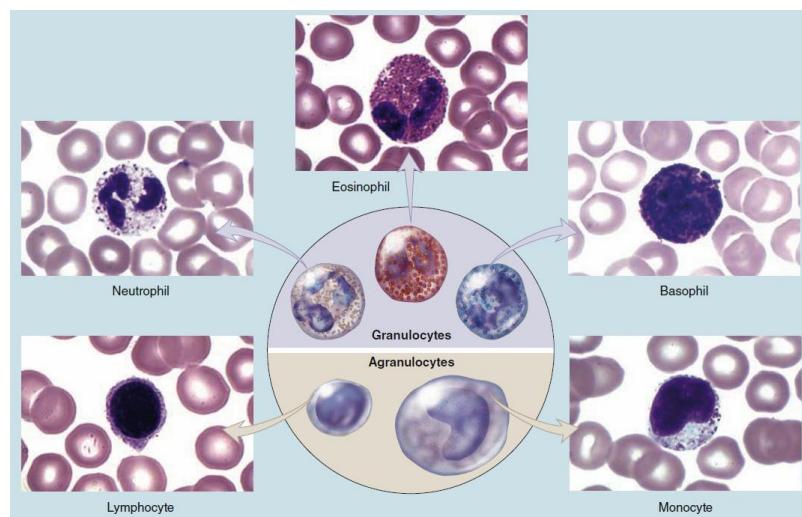


Gambar 3.3. Tahap asal dan diferensiasi sel darah [1].

III.1.2. Morfologi WBC

Leukosit (WBC) dibagi menjadi dua kelompok yaitu granulosit dan agranulosit didasarkan pada kepadatan granula sitoplasmiknya yang ditunjukkan pada Gambar 3.4. [1]. Granulosit memiliki inti polimorfik dengan dua atau lebih lobus yang berbeda (hampir terpisah) dan termasuk neutrofil, eosinofil, dan basofil [1]. Agranulosit tidak memiliki butiran spesifik, tetapi mengandung beberapa

butiran azurofilik (lisosom) dan nukleus berbentuk bulat atau menjorok tetapi tidak berlobus termasuk limfosit dan monosit [1]. Leukosit (WBC) adalah pertahanan utama melawan mikroorganisme yang menyerang dan dalam perbaikan jaringan yang terluka [1]. Fitur struktural pada kedua jenis tersebut ditunjukkan pada Tabel 3.1.



Gambar 3.4. Leukosit: jenis dan fitur struktural [1].

Tabel 3.1. Fitur struktural granulosit dan agranulosit.

Tipe	Nukleus	<i>Specific Granules</i>
Granulosit		
Neutrofil	3-5 lobus	Merah muda pucat/ringan
Eosinofil	<i>Bilobed</i> (berlobus ganda)	Merah/merah muda tua
Basofil	<i>Bilobed</i> atau berbentuk S	Biru tua/ungu
Agranulosit		
Limfosit	Agak bulat	(tidak ada)
Monosit	Indentasi atau berbentuk C	(tidak ada)

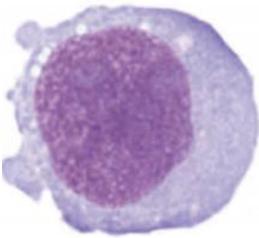
Sumber: [1]

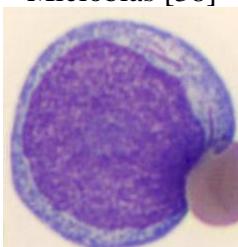
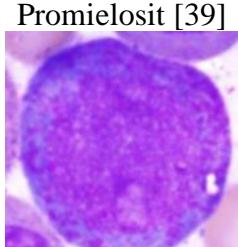
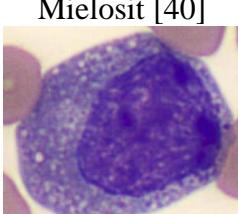
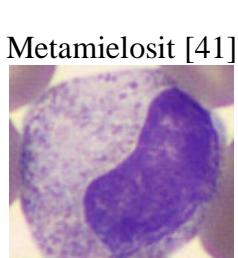
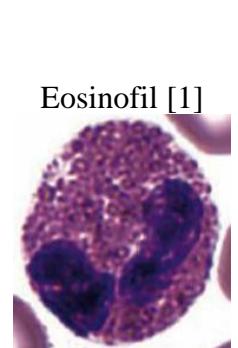
Berdasarkan Gambar 3.3. WBC dibentuk dari sel punca myeloid dan limfoid. Sel punca myeloid membentuk WBC berupa eosinofil, basofil, neutrofil, dan monosit. Namun, sel punca limfoid membentuk WBC berupa limfosit B, limfosit T, dan *Natural Killer*. Berikut adalah morfologi jenis-jenis WBC berdasarkan asal sel punca.

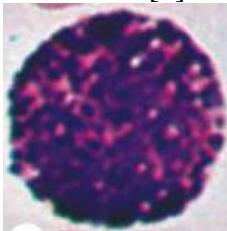
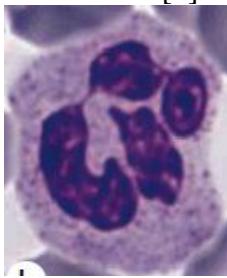
III.1.2.1. Morfologi WBC dari Sel Punca Myeloid

Berikut adalah morfologi WBC dari sel punca myeloid.

Tabel 3.2. Morfologi WBC dari sel punca myeloid.

No	Nama dan citra sel	Morfologi
1.	Monoblas [37] 	<ul style="list-style-type: none"> • Inti berbentuk bulat atau oval [37]. • Sitoplasma <i>basophilic rare azurophilic granules</i> [37]. • Nukleolus halus [37]. • Sel berdiameter 20-30 µm [37].
2.	Promonosit [37] 	<ul style="list-style-type: none"> • Inti berbelit-belit/menjorok [37]. • Nukleolus halus [37]. • Sitoplasma bervariasi basofilik, <i>variabel azurofilic granules</i> [37]. • Sangat mirip dengan monoblas kecuali bentuk intinya [37].
3.	Monosit [1] 	<ul style="list-style-type: none"> • Monosit berdiameter dari 12-20 µm [1]. • Monosit menunjukkan inti khas yang menjorok, berbentuk ginjal, atau berbentuk C [1].

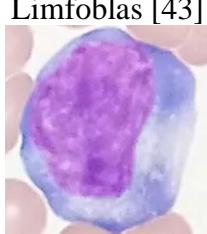
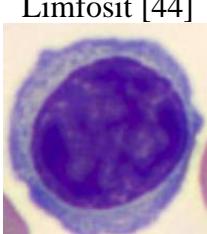
4.	<p>Mieloblas [38]</p> 	<ul style="list-style-type: none"> Mieloblas berukuran sekitar 15-20 µm dan memiliki rasio inti terhadap sitoplasma yang tinggi [38]. Nukleus biasanya bulat sampai oval, tetapi mungkin tidak teratur [38]. Kromatin halus dan terbuka, dan satu atau lebih nukleolus mungkin ada [38]. Sitoplasma berwarna biru pucat [38]. Granula sitoplasma biasanya tidak terlihat, meskipun granula kecil yang jarang mungkin ada [38].
5.	<p>Promielosit [39]</p> 	<ul style="list-style-type: none"> Promielosit lebih besar dari mieloblas [39]. Promielosit memiliki sitoplasma basofilik yang mengandung granula primer (azurofilik) [39]. Kromatin inti tersebar halus dan nukleolus dapat terlihat [39].
6.	<p>Mielosit [40]</p> 	<ul style="list-style-type: none"> Inti bulat sampai lonjong dan eksentrik, kromatin kasar, dan nukleolus kecil [42]. Kromatin lebih kental daripada promielosit [39].
7.	<p>Metamielosit [41]</p> 	<ul style="list-style-type: none"> Metamielosit neutrofilik memiliki kromatin yang kental dan nukleus yang sedikit menjorok (lekukannya kurang dari setengah diameter nukleus) [41]. Sitoplasma mengandung butiran spesifik yang melimpah dan butiran primer (azurofilik) yang langka [41].
8.	<p>Eosinofil [1]</p> 	<ul style="list-style-type: none"> Eosinofil berukuran kira-kira sama dengan neutrofil tetapi memiliki inti berlobus ganda dan granula sitoplasma kasar yang lebih banyak [1]. Sitoplasma sering diisi dengan butiran spesifik eosinofilik yang cerah, tetapi juga mencakup beberapa butiran azurofilik [1]. Eosinofil dengan granula yang mengisi sitoplasma, dua lobus nukleus eosinofil biasanya jernih [1].

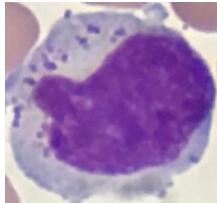
9.	Basofil [1] 	<ul style="list-style-type: none"> Basofil kira-kira berukuran sama dengan neutrofil dan eosinofil, tetapi mereka memiliki granula spesifik yang besar dan sangat basofilik yang biasanya menghalangi penampilan nukleus yang memiliki dua lobus besar dan tidak beraturan [1].
10.	Neutrofil [1] 	<ul style="list-style-type: none"> Inti <i>multi-lobulated</i> dengan lobus disatukan oleh untaian yang sangat tipis [1]. Sel memiliki diameter mulai dari 12-15 μm [1]. Granula sitoplasma relatif jarang dan memiliki sifat pewarnaan heterogen, meskipun umumnya pucat dan tidak menutupi nukleus [1].

III.1.2.2. Morfologi WBC dari Sel Punca Limfoid

Berikut adalah morfologi WBC dari sel punca limfoid.

Tabel 3.3. Morfologi WBC dari sel punca limfoid.

No	Nama dan citra sel	Morfologi
1.	Limfoblas [43] 	<ul style="list-style-type: none"> Kromatin inti halus [43]. Sitoplasma umumnya sedikit [43]. Nukleus yang mungkin berisi satu atau dua nukleolus [43]. Memiliki membran nukleus jelas [43]. Sel berdiameter 10-20 μm [43].
2.	Limfosit [44] 	<ul style="list-style-type: none"> Limfosit normal adalah sel-sel kecil dengan kromatin kental dan sejumlah sitoplasma kecil basofilik pucat [44]. Inti limfosit hanya sedikit lebih besar dari RBC [44].

3.	<i>Natural Killer</i> [46] 	<ul style="list-style-type: none"> • Sel NK memiliki penampilan limfosit granular besar [45]. • Inti matang eksentrik, sitoplasma berlimpah dan butiran azurofilik kasar [45].
----	---	--

III.1.3. Acute Lymphoblastic Leukemia Tipe L1

Acute Lymphoblastic Leukemia tipe L1 mewakili jenis leukemia akut yang umum terjadi pada masa kanak-kanak berdasarkan klasifikasi FAB [10]. Fitur yang dipertimbangkan dalam klasifikasi tipe leukemia ‘*lymphoblastic*’ adalah: ukuran sel, kromatin inti, bentuk inti, nukleolus, jumlah dan basofilia sitoplasma [10]. Namun, untuk fitur sel L1 adalah sebagian besar sel kecil, hingga dua kali diameter sel kecil limfosit [10]. Kromatin inti bervariasi dari kasus ke kasus tetapi homogen dalam satu kasus [10]. Kromatin inti tersebar halus tetapi tampak menggumpal di sel yang lebih kecil [10]. Bentuk inti sel teratur [10]. *Nucleoli* sering tidak terlihat; *nucleolus* biasanya kecil dan tidak menonjol [10]. Sitoplasma biasanya sedikit (rasio inti: sitoplasma tinggi) [10]. Basofilia sitoplasma tidak umum [10]. Pada Tabel 3.4. merupakan ringkasan karakteristik ALL L1 berdasarkan klasifikasi FAB.

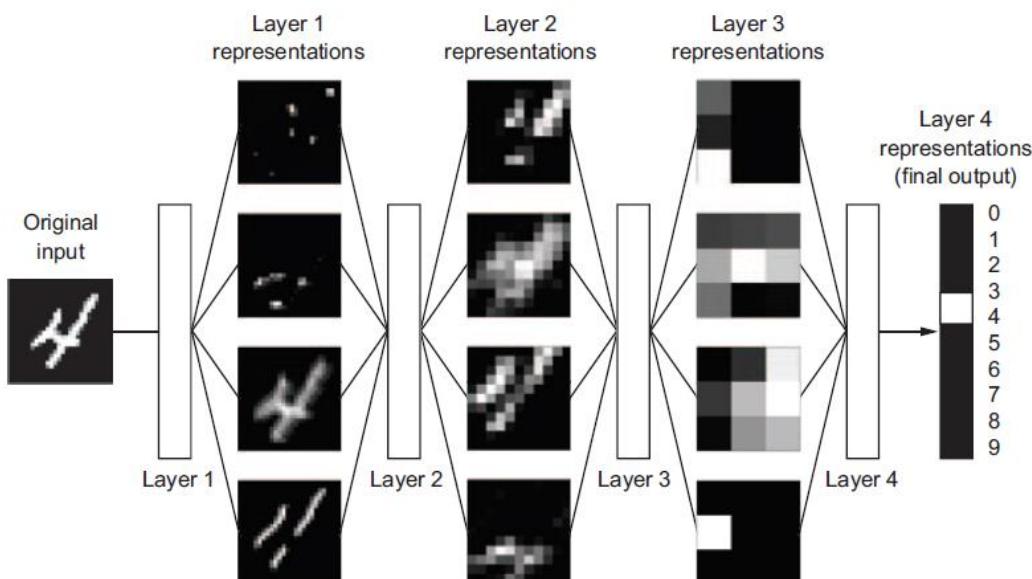
Tabel 3.4. Ringkasan karakteristik ALL L1.

Fitur sitologi	Deskripsi
Ukuran sel	Sel-sel kecil mendominasi
<i>Nuclear chromatin</i>	Homogen dalam setiap kasus
Bentuk inti	Celah atau lekukan yang teratur
<i>Nucleoli</i>	Tidak terlihat atau kecil dan tidak mencolok
Jumlah sitoplasma	Sedikit
<i>Basophilia of cytoplasm</i>	Sedikit atau sedang
<i>Cytoplasmic vacuolation</i>	Bervariasi

Sumber: [10]

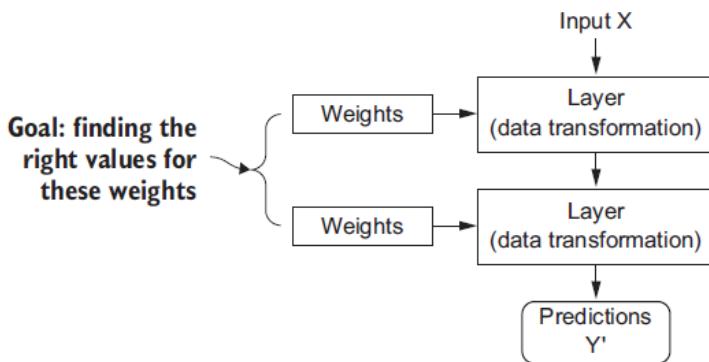
III.2. Deep Learning

Deep learning adalah subbidang khusus *machine learning* yang menggunakan kerangka matematika untuk mempelajari representasi dari data [47]. *Deep learning* melibatkan puluhan atau ratusan lapisan representasi yang berurutan dan semuanya dipelajari secara otomatis dari paparan data latih [47]. Metode *deep learning* memiliki basis jaringan saraf tiruan (*artificial neural networks*) dengan pembelajaran representasi (*feature engineering*) otomatis dari data.



Gambar 3.5. Representasi mendalam dipelajari oleh model klasifikasi digit [47].

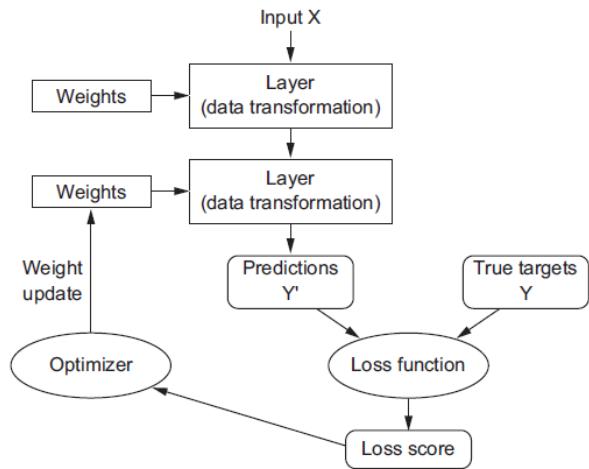
Jaringan pada Gambar 3.5. mengubah gambar digit menjadi representasi yang semakin berbeda dari gambar asli dan semakin informatif [47]. *Deep learning* dapat dianggap jaringan operasi penyulingan informasi bertingkat, di mana informasi melewati filter berturut-turut dan keluar semakin murni [47]. Dalam konteks ini, ‘*learning*’ adalah menemukan nilai-nilai bobot dan bias (*trainable parameters*) pada *artificial neural networks* yang dapat merepresentasi data latih untuk melakukan prediksi dari sekumpulan data serupa (bukan dari data latih), seperti ditunjukkan pada Gambar 3.6.



Gambar 3.6. Jaringan saraf diparameterisasi oleh nilai bobot dan bias-nya [47].

Jaringan saraf tiruan (*artificial neural networks*) yang terdiri dari bobot, bias, dan *activation function* dikendalikan dengan *loss function* atau *objective function* untuk mengukur seberapa jauh hasil prediksi dari hasil yang sebenarnya. Hasil dari *loss function* menunjukkan seberapa baik jaringan hasil dari ‘*learning*’ yang telah dilakukan. Pada *deep learning*, skor *loss function* digunakan sebagai umpan balik untuk menyesuaikan nilai bobot dan bias sedikit demi sedikit ke arah menurunkan skor *loss function* (ditunjukkan pada Gambar 3.7.) [47]. Penyesuaian tersebut adalah tugas *optimizer* yang mengimplementasikan algoritma *backpropagation* untuk pembaruan nilai bobot dan bias pada jaringan [47]. Melatih jaringan saraf tiruan melingkupi hal sebagai berikut [47]:

1. *Layers* yang digabungkan menjadi jaringan (atau model).
2. Data *input* dan target yang sesuai.
3. *Loss function* yang mendefinisikan sinyal umpan balik yang digunakan untuk ‘*learning*’. Kuantitas yang akan diminimalkan selama pelatihan. Ini mewakili ukuran keberhasilan untuk tugas yang ada.
4. *Optimizer* yang menentukan bagaimana ‘*learning*’ berlangsung. Menentukan bagaimana bobot dan bias jaringan akan diperbarui berdasarkan *loss function*. Ini mengimplementasikan varian spesifik dari *stochastic gradient descent* (SGD).



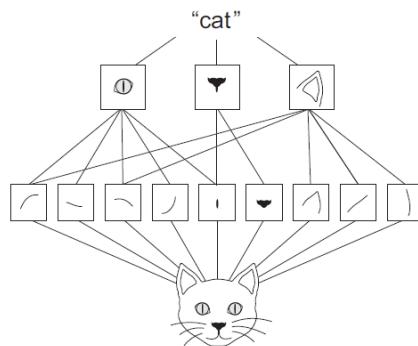
Gambar 3.7. *Loss function* mengukur kualitas *output* jaringan dan digunakan sebagai sinyal umpan balik untuk menyesuaikan nilai bobot [47].

III.2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs atau *ConvNets*) merupakan bagian dari *artificial neural networks* (ANNs) yang umum digunakan pada data visual. Jika arsitektur ANNs biasa digunakan untuk data visual, ukuran neuron yang dibutuhkan akan sangat besar (tidak efisien) dan mengakibatkan *overfitting*. Oleh karena itu, CNNs umum digunakan untuk data visual karena memiliki dua karakteristik yang membuat metode ini sangat efisien dalam aplikasi *computer vision* yaitu pola yang dapat dipelajari adalah *translation invariant* dan *spatial hierarchies*.

Translation merupakan perpindahan posisi dan *invariant* adalah sifat yang tidak berubah. Visual pada dasarnya adalah *translation invariant* [47]. Setelah model CNNs mempelajari pola objek pada bagian citra, model tersebut tetap dapat mengenalinya di bagian lain. Namun, jika ANNs di gunakan dalam permasalahan serupa, model tersebut harus mempelajari ulang karena adanya perbedaan posisi (ANNs menganggap objek tersebut berbeda). Hal tersebut membuat CNNs efisien karena membutuhkan lebih sedikit sampel pelatihan untuk mempelajari representasi citra dan memiliki kekuatan generalisasi yang cukup baik [47].

CNNs memiliki kemampuan *spatial hierarchies* yang mempelajari konsep visual kompleks dan abstrak secara efisien seperti ditunjukkan pada Gambar 3.8. karena pada dasarnya visual juga merupakan *spatial hierarchies* [47]. Sebagai contoh, *convolution layer* pertama akan mempelajari pola lokal kecil seperti *edges* (*low-level features*), kemudian *convolution layer* kedua akan mempelajari bentuk lebih besar yang dibuat dari fitur lapisan pertama (*higher-level features*), dan seterusnya [47]. CNNs terdiri dari dua bagian, yaitu *feature extraction layer* dan *fully-connected layer*. *Feature extraction layer* terdiri dari *convolution layer/kernel* dan *pooling layer*, selanjutnya *fully-connected layer* yang melakukan prediksi dari fitur yang telah diekstraksi dari *feature extraction layer*.



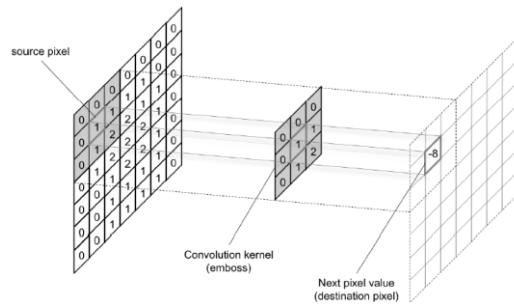
Gambar 3.8. Citra visual membentuk *spatial hierarchies*: *edges* hiperlokal bergabung menjadi objek lokal seperti mata atau telinga yang digabungkan menjadi konsep tingkat tinggi seperti "cat" [47].

III.2.1.1. Convolution Layer (Kernel)

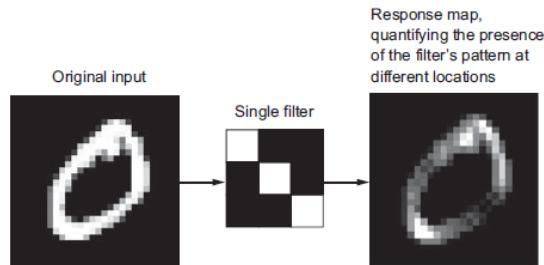
Kernel atau filter adalah *convolution layer* yang digunakan untuk melakukan konvolusi citra seperti yang ditunjukkan pada Gambar 3.9. Kernel memiliki tiga parameter, yaitu *stride*, *size*, dan *depth* (jumlah filter). *Stride* adalah jumlah piksel yang dipindahkan setiap melakukan konvolusi. *Stride* bernilai 1 akan menghasilkan citra berukuran hampir sama, sedangkan *stride* bernilai 2 menghasilkan ukuran setengah kalinya. *Size* merupakan dimensi dari persegi kernel. *Depth* merupakan jumlah filter yang digunakan. Namun, penggunaan

padding dapat mencapai ukuran *output* sama dengan *input*-nya (tanpa perubahan dimensi) setelah operasi konvolusi dilakukan. *Output* dari operasi konvolusi antara citra dan kernel merupakan *feature map* seperti ditampilkan pada Gambar 3.10. Proses pelatihan pada model menggunakan *convolution layer* adalah untuk menyesuaikan nilai bobotnya sehingga dapat memperoleh fitur-fitur citra yang terlihat jelas pada *feature map*. Konvolusi sering direpresentasikan secara matematis dengan tanda asteris (*). Jika masukan citra yang direpresentasikan sebagai X dan filter yang direpresentasikan dengan f , maka ekspresinya adalah Persamaan (3.1).

$$Z = X * f \quad (3.1)$$



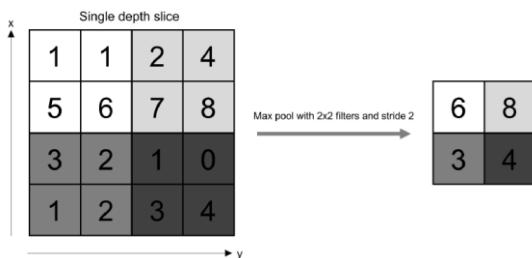
Gambar 3.9. Operasi konvolusi pada citra dengan kernel/filter 3×3 [48].



Gambar 3.10. *Feature map* yang dihasilkan oleh *single filter* 3×3 [47].

III.2.1.2. Pooling Layer

Pooling layer digunakan untuk mengurangi ukuran atau *downsample* pada *feature map* dengan cara *stride* seperti proses konvolusi. Proses *downsample* dapat dilakukan dengan cara *max pooling* atau *average pooling*. *Max pooling* melakukan *sampling* dengan mengambil nilai maksimum dari *window* dengan ukuran filter yang ditentukan seperti ditunjukkan pada Gambar 3.11., sedangkan *average pooling* mengambil nilai rata-ratanya. *Max pooling* biasa dilakukan dengan filter berukuran 2×2 dan *stride* bernilai 2 untuk melakukan *downsample* pada *feature map* dengan faktor 2 kalinya [47]. *Downsample* sangat penting untuk mengurangi nilai-nilai pada *feature map* karena dapat mengurangi banyak parameter latih tanpa menghilangkan fitur-fitur dominan untuk *layer* berikutnya yang dapat membuat CNNs efisien. Penggunaan *max pooling* bekerja lebih baik dibandingkan *average pooling* karena lebih informatif untuk menggunakan nilai maksimal fitur dibandingkan nilai rata-ratanya [47].

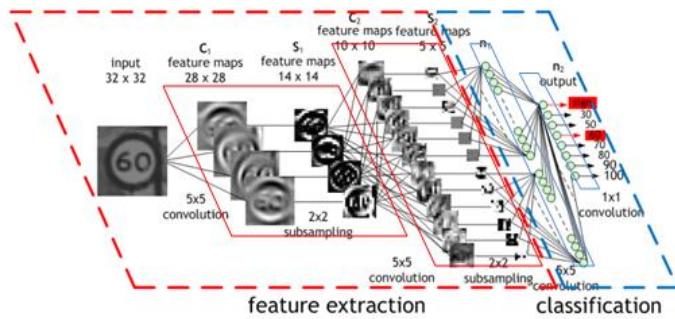


Gambar 3.11. *Max pooling* dengan filter 2×2 dan *stride* bernilai 2 [48].

III.2.1.3. Fully Connected Layer (FC Layer)

Fully connected layer adalah tahap akhir dari CNNs yang melakukan prediksi dari hasil *feature extraction layer* (*convolution layer* dan *pooling layer*). FC layer merupakan ANNs yang semua *input* dari satu *layer* terhubung ke setiap unit aktivasi pada *layer* berikutnya. Pada *layer* ini hanya dapat menerima *array* satu dimensi, sedangkan hasil pada *feature extraction layer* masih berbentuk multidimensi *array* oleh karena itu perlu dilakukan konversi dimensi *array* menjadi

satu dimensi atau disebut ‘*flatten layer*’ seperti yang ditunjukkan pada Gambar 3.12. Array yang sudah dikonversi akan masuk pada *feed-forward neural network* dan algoritma *backpropagation* untuk dilakukan iterasi pelatihan. Untuk beberapa kali iterasi, model akan dapat membedakan antara fitur yang mendominasi untuk setiap kelas dan melakukan klasifikasi dengan menggunakan *softmax activation function*.



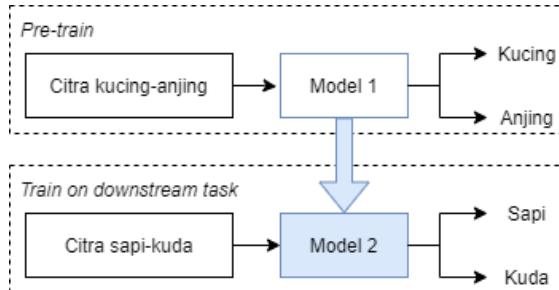
Gambar 3.12. Contoh arsitektur CNN yang merupakan gabungan dari *Convolution Layer*, *Pooling Layer*, dan *FC Layer* [49].

Pada Gambar 3.12. adalah contoh arsitektur CNN yang terdiri dari *Convolution Layer*, *Pooling Layer*, dan *FC Layer*. CNN tersebut memiliki masukan citra rambu lalu lintas yang difilter oleh 4 kernel berdimensi 5×5 dan menghasilkan 4 *feature maps*. *Feature maps* tersebut kemudian dilakukan *downsample* dengan *max pooling*. Pada lapisan berikutnya menerapkan 10 kernel berdimensi 5×5 kepada *feature maps* yang telah *downsample* sebelumnya dan kemudian dilakukan *downsample* kembali. Lapisan terakhir adalah *FC Layer*, dimana semua fitur yang dihasilkan akan digabungkan dan digunakan dalam proses klasifikasi.

III.2.2. Transfer Learning

Transfer learning dapat mengurangi waktu dan upaya saat membuat dan melatih model *deep learning* baru. Hal tersebut dapat dilakukan karena *transfer learning* melakukan transfer pengetahuan (*weights* dan *layers*) ke model yang baru.

Semakin mirip tugas yang ditransfer ke model baru, semakin mudah dalam melakukan pelatihan. Melatih model baru untuk kasus *computer vision* dari awal mengonsumsi banyak waktu dan *computationally expensive*. Oleh karena itu, penggunaan *transfer learning* sangat membantu dalam kasus ini. Sebagai contoh ditunjukkan pada Gambar 3.13., Model 1 dimulai dari nilai-nilai bobot acak sehingga membutuhkan banyak data dan komputasi latih untuk model dapat mengklasifikasi kucing atau anjing. Selanjutnya, *weights* dan *layers* Model 1 yang telah dilatih dapat di-transfer ke Model 2, sehingga Model 2 tidak perlu mengonsumsi banyak waktu dan komputasi seperti awal pelatihan Model 1 karena memiliki fitur yang mirip, seperti telinga, mata, dan lain-lain. Dalam kasus *transfer leaning*, model yang mentransfer *weights* dan *layers* disebut *pre-training task*, sedangkan yang menerima adalah *downstream task*.



Gambar 3.13. Contoh kasus *transfer learning*.

III.2.3. Distribusi Data

Dalam *machine learning*, tujuan utamanya adalah untuk mencapai model yang *generalize* dan berkinerja baik pada data *never-before-seen* dan *overfitting* adalah tantangan utamanya [47]. Pada *machine learning* umumnya *dataset* dibagi menjadi tiga bagian, yaitu data latih, data validasi, dan data uji. Model dilatih dengan data latih dan dievaluasi dengan data validasi. Setelah model bekerja baik pada data validasi, pengujian terakhir kalinya adalah dengan menggunakan data uji.

Pentingnya distribusi data dalam pengembangan model selau melibatkan *tuning* konfigurasi model yang disebut *hyperparameter tuning*. Data latih

digunakan untuk proses pembobotan pada *trainable parameters*, seperti *weights* dan *biases*. Namun, penggunaan data validasi adalah sebagai sinyal umpan balik performa model yang diakibatkan oleh *hyperparameter tuning*. *Tuning* konfigurasi berdasarkan performa data validasi saja dapat mengakibatkan *overfitting to the validation set*, walaupun data validasi bukan digunakan sebagai data latih. Model tersebut berkinerja sangat baik pada data validasi karena untuk itulah yang dioptimalkan oleh *hyperparameter*. Oleh karena itu, diperlukan data uji yang merupakan data *never-before-seen* untuk mengevaluasi model dari data yang benar-benar baru. Data uji adalah data yang dapat mengevaluasi hasil performa model yang sesungguhnya.

Distribusi data latih, data validasi, dan data uji dapat mempengaruhi kinerja model *machine learning*. Pada keadaan *real-world*, *dataset* yang diperoleh secara umum mengalami *class imbalance*, yaitu ketika masing-masing kelas *dataset* memiliki jumlah yang berbeda secara signifikan. *Class imbalance* dapat menyebabkan kinerja buruk dalam klasifikasi kelas minoritas walaupun kelas minoritas merupakan hal penting dalam proses klasifikasi. Sebagai contoh, model dapat memperoleh akurasi tinggi dengan hanya memprediksi kelas mayoritas untuk setiap saat melakukan prediksi (model mengalami bias). Permasalahan *class imbalance* tersebut dapat diatasi dengan dua metode, yaitu *undersampling* dan *oversampling* seperti ditunjukkan pada Gambar 3.14. Metode *undersampling* dilakukan dengan cara mengambil sampel dari kelas mayoritas, sehingga memiliki jumlah yang sama dengan kelas minoritas. Kelemahan dari metode tersebut adalah adanya kemungkinan informasi penting yang hilang. Namun, metode *oversampling* adalah dengan memperbanyak jumlah kelas minoritas, sehingga memiliki jumlah yang sama dengan kelas mayoritas. Proses *oversampling* sedikit lebih rumit daripada proses *undersampling*. Metode *oversampling* dapat dilakukan dengan banyak metode tergantung dari jenis datanya, seperti data citra dan data *tabular* memiliki metode yang berbeda. *Oversampling* hanya dilakukan pada data latih dan metode ini juga tidak menambah informasi baru hanya menyeimbangkan distribusi kelas. Proses *oversampling* dapat dilakukan dengan augmentasi data yang dijelaskan pada subbab berikutnya.



Gambar 3.14. *Undersampling* dan *oversampling* [50].

III.2.4. Augmentasi Data

Augmentasi data merupakan metode yang digunakan untuk meningkatkan jumlah data dan membuat performa ANNs lebih baik. Tujuan utama menerapkan augmentasi data adalah untuk meningkatkan generalisasi model. ANNs secara terus menerus melihat data baru dan sedikit modifikasi dari data tersebut, sehingga model ANNs dapat mempelajari fitur-fitur data lebih *robust* dan *generalize* yang dapat mengurangi *overfitting* saat melatih model. Peningkatan variasi data tersebut membantu model mengabaikan fitur-fitur yang tidak penting dan hanya memilih fitur-fitur yang benar-benar penting yang memungkinkan untuk menggeneralisasi lebih baik. Oleh karen itu, penurunan akurasi saat *training* biasa terjadi jika dibandingkan *training* tanpa augmentasi data. Augmentasi data tidak menambah informasi baru, tetapi hanya dapat mencampur informasi yang ada seolah-olah merupakan data baru. Augmentasi data citra pada data latih dapat dilakukan dengan menerapkan transformasi geometris sederhana, seperti pencerminan pada sumbu x, sumbu y, dan kedua sumbu.

III.3. Computer Vision

Computer vision adalah bidang kecerdasan buatan (AI) yang dapat memperoleh informasi dari citra visual dan dapat mengambil tindakan atau rekomendasi berdasarkan informasi tersebut [51]. Tugas dari *computer vision* meliputi metode seperti *object detection*, segmentasi, dan pengolahan citra yang dapat membantu memperoleh informasi citra visual. Pada bagian berikut akan dijelaskan metode-metode *computer vision* yang digunakan dalam penelitian ini.

III.3.1. Klasifikasi dan Lokalisasi Objek (*Object Detection*)

Klasifikasi citra merupakan metode memprediksi kelas dari objek yang terdapat pada citra, sedangkan lokalisasi objek merupakan metode menentukan lokasi pada satu objek atau lebih yang terdapat pada citra. Penggunaan kedua metode tersebut merupakan *object detection* yang melakukan klasifikasi dan lokalisasi pada satu objek atau lebih yang terdapat pada citra. Pada Tabel 3.5. merupakan ringkasan perbedaan *input* dan *output* dari ketiga metode tersebut.

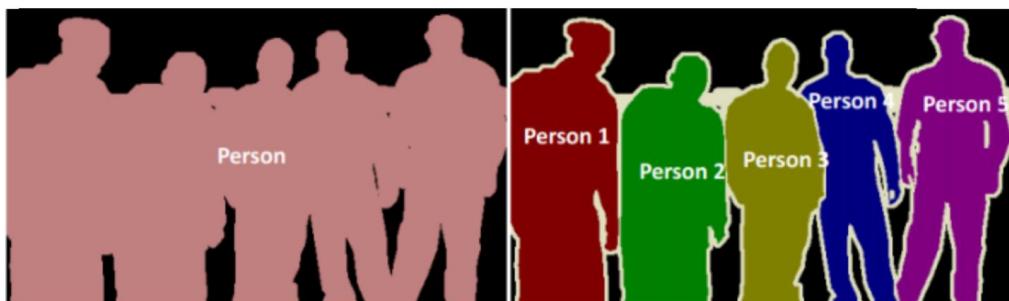
Tabel 3.5. Perbedaan klasifikasi, lokalisasi, dan deteksi objek.

Metode	Tugas	Input	Output
Klasifikasi Citra	Memprediksi kelas dari objek pada citra.	Satu citra dengan satu objek.	Label kelas dari citra.
Lokalisasi Objek	Menemukan lokasi objek dalam citra dengan menunjukkan <i>bounding box</i> .	Satu citra dengan satu atau lebih objek.	Satu atau lebih <i>bounding box</i> (diperoleh dari titik, lebar, dan tinggi objek)
Deteksi Objek	Menemukan lokasi objek dengan <i>bounding box</i> dan memprediksikan kelasnya.	Satu citra dengan satu atau lebih objek.	Satu atau lebih <i>bounding box</i> dan label kelas objek untuk setiap <i>bounding box</i> .

Sumber: [52]

III.3.2. Segmentasi Citra

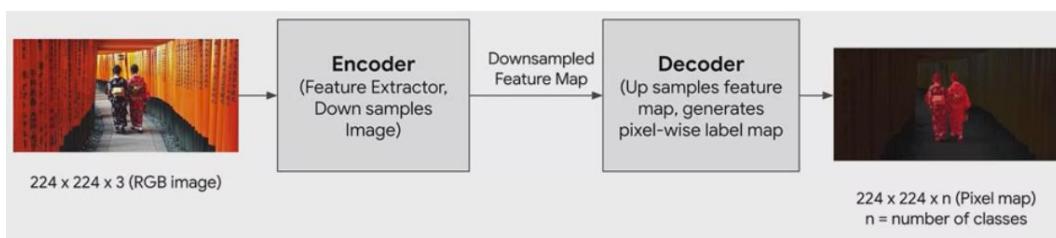
Segmentasi citra mempartisi citra menjadi beberapa bagian yang disebut segmen, bertujuan untuk analisis citra lebih lanjut, peningkatan efisiensi kompresi citra, atau hanya untuk efek visualisasi [53]. Segmentasi citra memberikan segmen piksel yang merupakan hasil dari analisis beberapa kriteria kesamaan. Segmentasi citra didukung dengan mendeteksi piksel yang menunjukkan beberapa perbedaan dengan piksel berdekatan (ketidaksejalan menciptakan batas antar segmen). Segmentasi citra merupakan hasil piksel yang membentuk objek terdeteksi. Terdapat dua jenis segmentasi citra, yaitu segmentasi semantik dan segmentasi *instance*. Segmentasi semantik merupakan semua objek dengan kelas yang sama membentuk klasifikasi tunggal, sedangkan segmentasi *instance* merupakan objek dengan tipe yang sama diperlakukan sebagai objek yang berbeda. Contoh kedua jenis segmentasi citra tersebut ditunjukkan pada Gambar 3.15.



Gambar 3.15. Contoh hasil segmentasi semantik (kiri) dan segmentasi *instance* (kanan) [54].

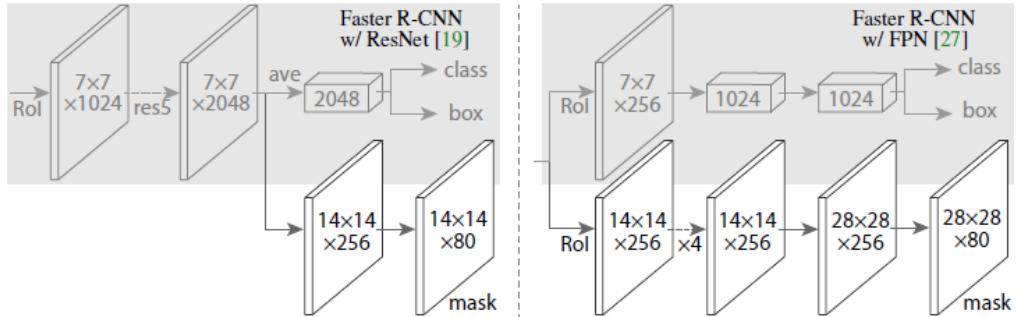
Arsitektur tingkat tinggi (*high level*) untuk algoritma segmentasi citra terletak pada *encoder* dan *decoder*, seperti ditunjukkan pada Gambar 3.16. Segmentasi bekerja dengan *input* citra dalam serangkaian dimensi, seperti $224 \times 224 \times 3$, dimana 3 untuk warna merah, hijau, dan biru. *Encoder* merupakan ekstraksi fitur yang biasanya menggunakan CNN untuk menghasilkan *feature map* dari citra masukan. CNN tersebut akan mempelajari untuk mengekstrak fitur citra

dan melakukan *pooling (downsampling)* untuk agregasi fitur tingkat rendah ke fitur tingkat tinggi, seperti dijelaskan pada bagian III.2.1. Pada bagian akhir CNN biasanya terdapat FC *layer* untuk melakukan klasifikasi dari hasil filter sebelumnya, sedangkan pada arsitekur segmentasi citra tidak menggunakan FC *layer*, sehingga hasil *feature map* yang telah di-*downsampling* langsung diteruskan pada *decoder*. *Decoder* juga merupakan CNN, *decoder* memberikan label kelas menengah ke setiap piksel *feature map* dan kemudian melakukan *upsampling* citra secara perlahan dan menambahkan kembali detail halus citra. *Decoder* tersebut kemudian mengulangi proses itu kembali hingga citra yang disampel kembali ke dimensi citra masukan aslinya. Hasil prediksi citra akhir memiliki label kelas yang ditetapkan untuk setiap pikselnya. Arsitektur CNN yang digunakan pada *encoder* dan *decoder* disebut sebagai *Fully Convolutional Network* (FCN). FCN merupakan CNN yang bagian FC *layers* digantikan oleh *convolutional layers* yang bertindak sebagai *decoder*.



Gambar 3.16. Arsitektur dasar segmentasi citra [55].

Secara umum, *encoder* mendekripsi fitur dan melakukan *downsampling* citra kemudian *decoder* melakukan *upsampling* hingga kembali ke ukuran aslinya untuk memberi label pada setiap pikselnya. FCN merupakan dasar dari berbagai arsitektur segmentasi citra, seperti Segnet, U-Net, PSPNet dan termasuk *Mask R-CNN* [55]. *Mask R-CNN* merupakan arsitektur untuk segmentasi *instance* dan merupakan ekstensi dari *Faster R-CNN*. *Mask R-CNN* memiliki cabang tambahan setelah ekstraksi fitur seperti pada Gambar 3.17. untuk melakukan *upsampling* dan menghasilkan segmentasi *instance* pada citra.



Gambar 3.17. Penambahan cabang pada *Faster R-CNN* untuk menghasilkan segmentasi *instance* [24].

III.3.3. Pengolahan Citra

Tahap pertama pada sebagian besar algoritma *computer vision* yaitu penggunaan pengolahan citra untuk memproses citra terlebih dahulu dan mengubahnya menjadi bentuk yang sesuai untuk analisis lebih lanjut [56]. Sebuah citra dapat dianggap sebagai fungsi $f(x, y)$ dari dua variabel kontinu x dan y [57]. Sebuah citra yang diproses secara digital harus diambil sampelnya dan diubah dalam bentuk matriks numeris [57]. Pemrosesan citra digital tersebut terdiri dari manipulasi matriks numeris tersebut [57]. Pengolahan citra digital dapat dibagi menjadi beberapa kelas: peningkatan citra, restorasi citra, analisis citra, dan kompresi citra [57]. Pada bagian berikut dijelaskan metode-metode pengolahan citra digunakan pada penelitian ini.

III.3.3.1. Filter Bilateral

Filter bilateral adalah filter yang menghaluskan citra dan mempertahankan tepi (*edges*), melalui kombinasi nonlinier dari nilai citra terdekat [58]. Filter ini digunakan jika kita ingin menghaluskan citra dan menjaga tepinya tetap utuh [59]. Filter bilateral adalah pendekatan menghalusi (*smoothing*) nonlinier yang mengambil rata-rata bobot dari piksel tetangga [59]. Secara matematis filter bilateral didefinisikan sebagai Persamaan (3.2) dan (3.3). Piksel tetangga tersebut didefinisikan dengan cara berikut: dua nilai piksel saling berdekatan dan dua nilai

piksel mirip satu sama lain [59]. Pada OpenCV fungsi *bilateralFilter* memiliki empat parameter yang didefinisikan sebagai berikut [59]:

1. Citra yang ingin kita haluskan.
2. Diameter *neighborhood* piksel (mendefinisikan diameter *neighborhood* untuk mencari tetangga).
3. Nilai sigma untuk warna (untuk menemukan piksel yang serupa).
4. Nilai sigma untuk ruang (untuk menemukan piksel yang lebih dekat).

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|) \quad (3.2)$$

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|) \quad (3.3)$$

Dimana,

$I^{filtered}$ = citra yang telah difilter

I = citra input asli yang akan difilter

x = koordinat piksel saat ini yang akan difilter

Ω = jendela yang berpusat di x , sehingga $x_i \in \Omega$ adalah piksel lain

f_r = adalah rentang kernel untuk menghaluskan perbedaan intensitas (fungsi ini dapat berupa fungsi Gaussian)

g_s = adalah kernel spasial (atau domain) untuk menghaluskan perbedaan koordinat (fungsi ini dapat berupa fungsi Gaussian)

Bobot W_p ditetapkan menggunakan kedekatan spasial (menggunakan kernel spasial g_s) dan perbedaan intensitas (menggunakan kernel rentang f_r) [58]. Pertimbangkan sebuah piksel yang terletak pada (i, j) yang perlu untuk di-*denoise* pada citra menggunakan piksel tetangganya dan salah satu piksel tetangganya terletak pada (k, l) . Kemudian, mengasumsikan rentang dan kernel spasial menjadi

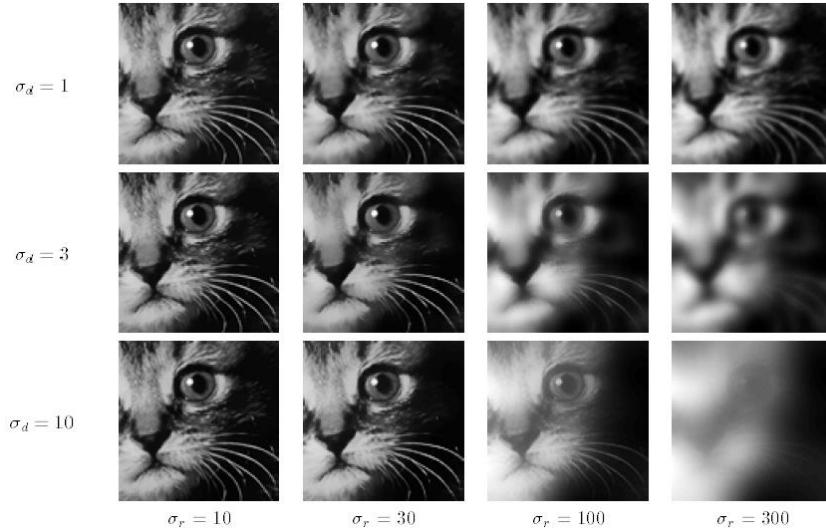
kernel Gaussian, bobot yang ditetapkan untuk piksel (k, l) untuk *denoise* piksel (i, j) diberikan pada Persamaan (3.4).

$$w(i, j, k, l) = \exp\left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} - \frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2}\right) \quad (3.4)$$

Dimana σ_d dan σ_r merupakan parameter *smoothing*, dan $I(i, j)$ dan $I(k, l)$ adalah intensitas piksel dari (i, j) dan (k, l) secara berturutan. Setelah menghitung bobot, menormalisasikan dengan Persamaan (3.5).

$$I_D(i, j) = \frac{\sum_{k, l} I(k, l) w(i, j, k, l)}{\sum_{k, l} w(i, j, k, l)} \quad (3.5)$$

Dimana I_D adalah intensitas piksel (i, j) yang telah di-*denoise*.



Gambar 3.18. Hasil proses filter bilateral dengan berbagai nilai parameter rentang dan domain [58].

III.3.3.2. *Thresholding* Citra

Thresholding citra biasa digunakan untuk melakukan segmentasi citra [59]. Objek pada citra dapat disegmentasi dengan cara menghapus latar belakangnya. Segmentasi tersebut dapat dilakukan dengan mengubah citra menjadi format biner (hanya terdiri dari hitam dan putih) dan kemudian dilakukan *thresholding* untuk

menggantikan nilai piksel yang melalui *threshold* (atas atau di bawahnya) menjadi hitam atau putih [59]. Berikut adalah lima jenis *thresholding* citra dalam notasi matematika:

1. *Binary*

$$J(x, y) = \begin{cases} \text{maxVal}, & \text{jika } I(x, y) > T \\ 0, & \text{lainnya} \end{cases}$$

2. *Binary inverted*

$$J(x, y) = \begin{cases} 0, & \text{jika } I(x, y) > T \\ \text{maxVal}, & \text{lainnya} \end{cases}$$

3. *Truncated*

$$J(x, y) = \begin{cases} \text{Threshold}, & \text{jika } I(x, y) > T \\ I(x, y), & \text{lainnya} \end{cases}$$

4. *To Zero*

$$J(x, y) = \begin{cases} I(x, y), & \text{jika } I(x, y) > T \\ 0, & \text{lainnya} \end{cases}$$

5. *To Zero Inverted*

$$J(x, y) = \begin{cases} 0, & \text{jika } I(x, y) > T \\ I(x, y), & \text{lainnya} \end{cases}$$

Dimana $J(x, y)$ adalah citra yang telah dilakukan *thresholding* dan T adalah nilai dari ambang batasnya.

III.3.3.3. Pemerataan histogram

Pemerataan histogram digunakan untuk penyesuaian kontras menggunakan histogram citra [59]. Distribusi intensitas piksel citra diplot pada histogram dan kemudian dimodifikasi. Pemerataan histogram digunakan untuk menemukan fungsi pemetaan intensitas $f(I)$, sehingga histogram yang dihasilkan rata [56]. Pemerataan dilakukan dengan secara efektif menyebarkan nilai intensitas yang tinggi dan kemudian meratakan intensitasnya keseluruh rentang histogram, seperti pada

Gambar 3.19. Penggunaan metode ini memungkinkan area dengan kontras lokal yang lebih rendah untuk mendapatkan kontras yang lebih tinggi.



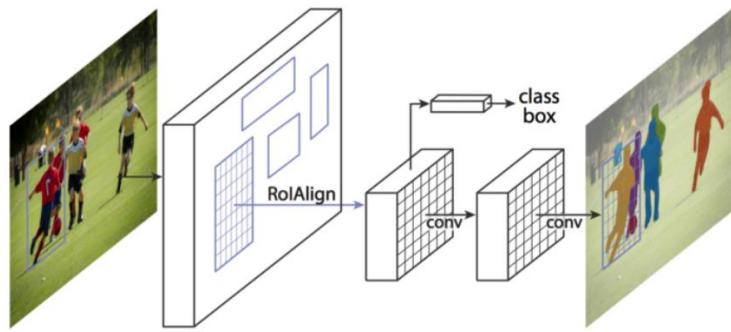
Gambar 3.19. Kiri: Masukkan citra *RagingBull* dengan histogram. Kanan: Gambar yang sama setelah pemerataan histogram [53].

III.4. *Mask Regional-Convolutional Neural Network*

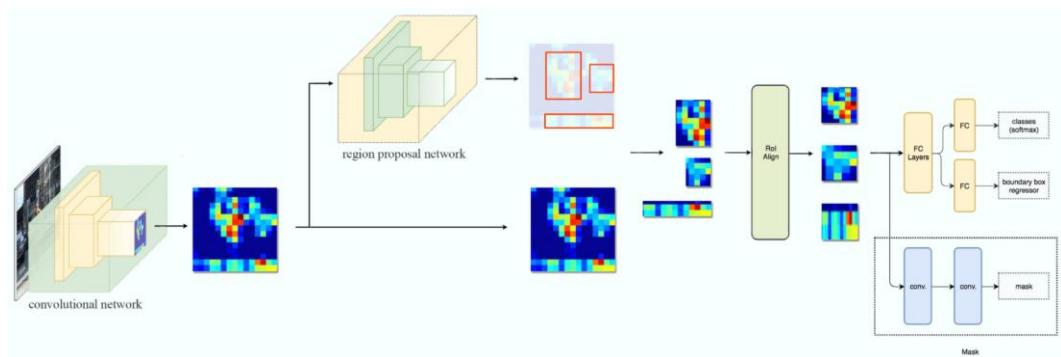
Mask R-CNN merupakan ekstensi dari *Faster R-CNN* dengan menambahkan cabang untuk memprediksi segmentasi *instance* pada setiap *Region of Interest* (RoI) secara paralel dari cabang yang telah ada untuk klasifikasi dan regresi *bounding box*, seperti pada Gambar 3.20 dan Gambar 3.21 [20]. *Mask R-CNN* merupakan deteksi objek dua tahap, di mana tahap pertama identik dengan *Faster R-CNN*, sedangkan pada tahap kedua ada penambahan cabang secara paralel untuk menghasilkan *mask* objek menggunakan FCN (dijelaskan pada bagian III.3.2.) yang diterapkan pada setiap RoI dengan cara piksel-ke-piksel [20].

Pada tahap pertama terdiri dari dua *networks*, yaitu arsitektur *backbone* (seperti ResNet-101 atau ResNet-50) dan *Region Proposal Network* (RPN). Kedua *networks* tersebut berjalan sekali per citra untuk memberikan satu set *region proposals*. *Region proposals* merupakan daerah citra pada *feature map* yang terdapat objek. Pada tahap kedua, *networks* memprediksi *bounding box*, kelas objek, dan *mask* objek untuk masing-masing daerah yang telah diusulkan pada tahap pertama. Setiap daerah yang diusulkan pada tahap pertama dapat memiliki

ukuran yang berbeda, sedangkan FC *layer* selalu memerlukan vektor berukuran tetap untuk melakukan prediksi. Ukuran daerah yang diusulkan tersebut ditetapkan dengan menggunakan metode RoIAlign untuk mengatasi masalah *misalignment*.



Gambar 3.20. Kerangka kerja Mask R-CNN untuk segmentasi *instance* [20].



Gambar 3.21. Arsitektur Mask R-CNN [60].

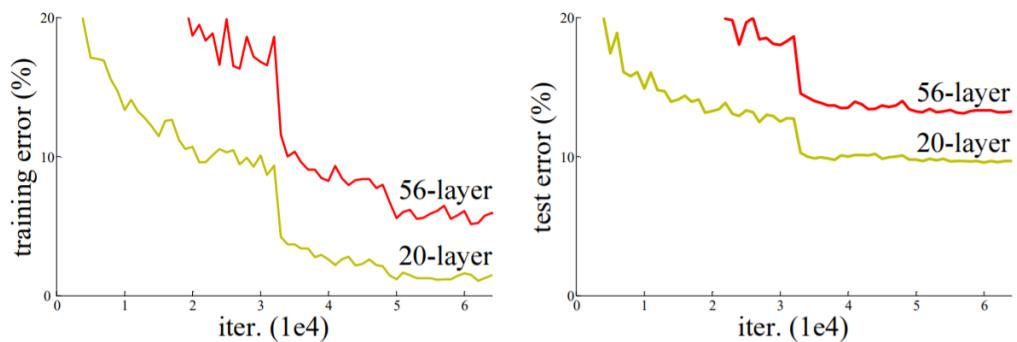
III.4.1. Backbone Network Architecture

Arsitektur *convolution backbone* digunakan untuk ekstraksi fitur di seluruh citra [20]. Arsitektur *backbone* ditunjukkan menggunakan *nomenclature network-depth-features* [20]. *Layers* awal mendeteksi fitur tingkat rendah (tepi dan sudut), dan lapisan selanjutnya berturut-turut mendeteksi fitur tingkat yang lebih tinggi (mata, hidung, dan telinga), seperti dijelaskan pada bagian III.2.1. Melewati arsitektur *backbone*, citra diubah dari $1024 \times 1024 \times 3$ menjadi *feature map* $32 \times 32 \times 2048$. *Feature map* tersebut menjadi masukan pada tahap selanjutnya, yaitu

Feature Pyramid Network (FPN). Arsitektur FPN tersebut adalah *top-down* dengan *lateral connection* [20]. ResNet-FPN *backbone* digunakan untuk ekstraksi fitur pada *Mask R-CNN* karena dapat menambah performa akurasi dan kecepatan [20].

III.4.1.1. Residual Neural Network (ResNet)

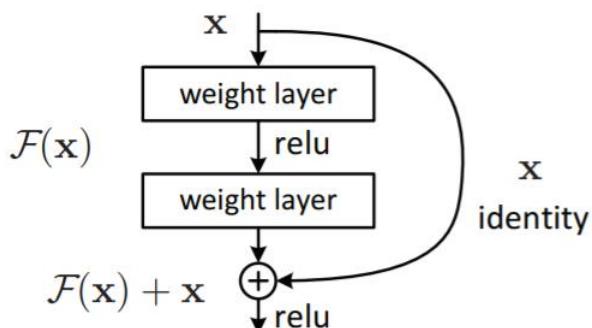
Residual Neural Network (ResNet) adalah salah satu jenis ANN khusus yang diperkenalkan pada tahun 2015. *Convolutional neural networks* mengintegrasikan fitur tingkat rendah/menengah/tinggi (dijelaskan pada III.2.1.) dan melakukan pengklasifikasi secara *end-to-end*, kemudian tingkat fitur dapat diperkaya dengan jumlah *layer* yang digunakan. Pada umumnya, jika ingin meningkatkan akurasi dan performa model dapat dilakukan dengan cara menambahkan jumlah *layer* yang digunakan. Intuisi di balik penambahan lebih banyak *layer* adalah untuk mempelajari fitur yang lebih kompleks, sebagai contoh *layer* pertama mendeteksi *edges*, *layer* kedua untuk mengidentifikasi tekstur, *layer* ketiga mendeteksi objek dan seterusnya. Namun, saat jumlah *layer* ditambahkan secara lebih dalam maka mulai muncul masalah degradasi: dengan jumlah *layer* meningkat, akurasi akan jenuh dan kemudian menurun dengan cepat [19]. Pada Gambar 3.22. dapat dilihat bahwa jumlah *layer* yang lebih besar memiliki *error* pelatihan yang tinggi dan demikian pula pada *error* uji.



Gambar 3.22. *Error* pelatihan (kiri) dan *error* uji (kanan) dengan 20-layer and 56-layer “plain” networks [19].

Degradasi tersebut tidak disebabkan oleh *overfitting* dan menambahkan *layer* ke model meningkatkan *error* pelatihan dan uji yang lebih tinggi [19]. Degradasi (akurasi pelatihan) menunjukkan bahwa tidak semua sistem sama mudahnya untuk dioptimalkan [19]. Dalam mengatasi permasalahan tersebut diperkenalkan *deep residual learning framework*. Alih-alih berharap setiap beberapa *layer* yang ditumpuk langsung cocok dengan *mapping* dasar diinginkan, maka secara eksplisit membiarkan lapisan ini sesuai dengan *residual mapping*. Formulasi $F(x) + x$ dapat diwujudkan dengan *feedforward neural networks* dengan koneksi *shortcut*, seperti ditampilkan pada Gambar 3.23. Koneksi *shortcut* adalah koneksi yang melewati satu atau lebih *layers*. Koneksi *shortcut* hanya melakukan pemetaan identitas, kemudian *output*-nya ditambahkan ke *output layers* yang ditumpuk. *Residual network* dapat mengatasi masalah seperti:

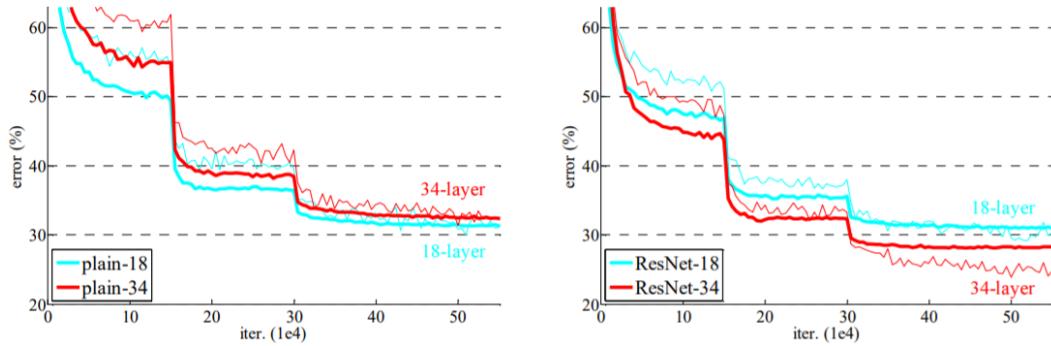
1. *Deep residual nets* yang sangat dalam lebih mudah dioptimalkan, dibandingkan “plain” networks (tanpa adanya *residual network*).
2. *Deep residual nets* dapat dengan mudah memperoleh peningkatan akurasi dari jumlah *layers* yang sangat dalam.



Gambar 3.23. Residual learning: a building block [19].

Arsitektur ResNet memiliki banyak jenis berdasarkan tingkat kedalaman *layer*. Semakin dalam tingkat *layer* pada ResNet akan menurunkan nilai *error* lebih baik dibandingkan *plain networks*. Kurva tipis pada Gambar 3.24. menunjukkan *error* pelatihan dan untuk kurva tebal menunjukkan *error* validasi. Penambahan

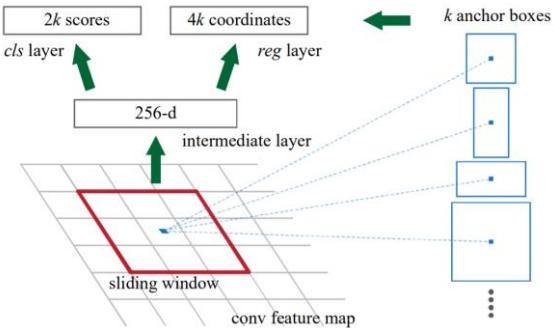
residual network tidak memiliki parameter tambahan dibandingkan dengan *plain networks*. Pada model *Mask R-CNN* hanya tersedia dua arsitektur *backbone*, yaitu ResNet-101 dan ResNet-50. Angka pada nama arsitektur ResNet menunjukkan tingkat kedalam *layer*.



Gambar 3.24. *Plain networks* dengan 18 dan 34 *layers*; ResNet dengan 18 dan 34 *layers* [19].

III.4.2. Region Proposal Networks (RPN)

Region Proposal Networks (RPN) mengambil citra (berapapun ukurannya) sebagai masukan dan keluarannya berupa satu set proposal objek persegi panjang, masing-masing dengan *objectness score* [34]. Proposal wilayah (*region proposal*) dihasilkan dengan cara menggeser *sliding window* berukuran $n \times n$ di atas keluaran *convolutional feature map* [34]. Setiap *sliding window* dipetakan pada *lower-dimensional feature* atau *intermediate layer* kemudian dimasukkan ke dalam dua FC *layer* yaitu, *box-regression layer* (*reg*) dan *box-classification layer* (*cls*), seperti ditampilkan pada Gambar 3.25.



Gambar 3.25. Region Proposal Network (RPN) [34].

III.4.3. Anchor Boxes

Pada setiap lokasi *sliding window*, secara simultan memprediksi beberapa proposal wilayah, di mana jumlah proposal maksimum yang mungkin untuk setiap lokasi dinotasikan sebagai k [34]. *Anchor* adalah kombinasi dari titik pusat *sliding window*, skala, dan rasio, sebagai contoh 3 skala + 3 rasio maka $k = 9$ *anchor boxes* pada setiap *sliding window*. *Reg layer* memiliki keluaran $4k$ yang meng-*encode* koordinat dari k *boxes*, dan keluaran *cls layer* adalah skor $2k$ (*objectness score*) yang memperkirakan probabilitas objek (*foreground*) atau bukan objek (*background*) untuk setiap proposal [34]. k proposal diparameterisasi relatif terhadap k *reference boxes* yang disebut *anchors* [34]. *Anchors* dipusatkan pada *sliding window* dan dikaitkan dengan skala dan *aspect ratio*, seperti pada Gambar 3.25. [34]. RPN menghasilkan dua keluaran untuk setiap *anchors*, yaitu *anchor class* dan *bounding box refinement*. *Anchor class* menentukan apakah dalam kotak tersebut *foreground* atau *background*, *foreground* mengartikan ada kemungkinan objek dalam kotak tersebut. Pada *bounding box refinement* terdapat *foreground anchor* atau *positive anchor* yang mungkin tidak terpusat sempurna pada objek, sehingga RPN melakukan estimasi delta dari % perubahan x , y , lebar, dan tinggi untuk memperbaiki *anchor box* dan menyesuaikan objek dengan lebih baik. *Anchors* yang tumpang tindih dapat dipilih dengan skor *foreground* tertinggi dan membuang sisanya, metode ini bernama *Non-maximum Suppression* (NMS) yang dijelaskan pada bagian III.4.5. Hasil proposal dari RPN adalah *Regions of Interest* (RoI).

III.4.4. Intersection over Union (IoU)

Intersection over union (IoU) merupakan metrik untuk mengukur *overlap* antara dua *bounding box* atau *mask*. IoU adalah rasio area yang *overlap* dari *ground truth* dan area yang diprediksi dengan total area (*union*), seperti pada Gambar 3.26. [48]. Prediksi yang sepenuhnya benar memiliki nilai $\text{IoU} = 1$, semakin rendah nilai IoU menunjukkan semakin buruk hasil prediksinya.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Gambar 3.26. *Intersection over union* [48].

III.4.5. Non-maximum Suppression (NMS)

Non-maximum suppression (NMS) merupakan algoritma pasca-pemrosesan yang bertanggung jawab untuk menggabungkan semua deteksi untuk objek yang sama [61]. Teknik *sliding window* seperti dijelaskan pada III.4.2. memberikan banyak *anchors* yang mendeteksi objek yang sama, seakan-akan satu objek yang sama tersebut dianggap lebih dari satu objek. Oleh karena itu, NMS digunakan untuk memperoleh *bounding box* terbaik dari satu objek yang terdeteksi. NMS bekerja dengan cara memilih *bounding box* yang memiliki *confidence score* tertinggi. *Confidence score* adalah probabilitas bahwa *anchor boxes* berisi objek yang diprediksi oleh *classifier*. *Bounding box* dengan *confidence score* tertinggi tersebut akan dihitung IoU-nya dengan *bounding box* lainnya. *Bounding box* selain *confidence score* tertinggi akan dihilangkan jika nilai IoU-nya melebihi *threshold* yang telah ditentukan. Algoritma pemilihan *bounding box* terbaik dapat disebut sesuai namanya, yaitu *Non-maximum suppression* atau melakukan *suppressing* pada *bounding box* yang tidak bernilai maksimum dan nilai IoU di atas *threshold* yang telah ditentukan. Pada Gambar 3.27. merupakan contoh penggunaan metode NMS.



Gambar 3.27. Contoh hasil *non-max suppression* [61].

III.4.6. Loss Function

Multi-task loss dari Mask R-CNN merupakan gabungan dari *loss function* klasifikasi (L_{cls}), lokalisasi (L_{box}), dan segmentasi *mask* (L_{mask}), seperti pada Persamaan (3.6) [20]. *Classification loss* (L_{cls}) dan *bounding-box loss* (L_{box}) identik dengan didefinisikan pada Faster R-CNN dimana *multi-task loss function* adalah $L = L_{cls} + L_{box}$ yang secara keseluruhan persamaan *loss function*-nya dijabarkan pada Persamaan (3.7) [34]. Pada cabang *mask* (dijelaskan pada III.4) menghasilkan *mask* berdimensi Km^2 untuk setiap RoI, dimana K *binary masks* dari resolusi $m \times m$, satu untuk setiap kelas K [20]. L_{mask} menghasilkan *mask* untuk setiap kelas, sehingga tidak ada persaingan antar kelas lainnya [20]. Fungsi sigmoid diterapkan per-piksel dan mendefinisikan L_{mask} sebagai *average binary cross-entropy loss* yang dijabarkan pada Persamaan (3.8) [20].

$$L = L_{cls} + L_{box} + L_{mask} \quad (3.6)$$

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (3.7)$$

$$L_{mask} = -\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)] \quad (3.8)$$

Dimana,

$L(\{p_i\}, \{t_i\}) = \text{multi-task loss function Faster R-CNN}$

$i = \text{indeks anchor pada mini-batch}$

$p_i = \text{probabilitas prediksi dari anchor } i \text{ adalah objek}$

$p_i^* = \text{label ground truth (biner) dari anchor } i \text{ adalah objek}$

$$p_i^* = \begin{cases} 1, & \text{jika anchor positif} \\ 0, & \text{jika anchor negatif} \end{cases}$$

$t_i = \text{vektor yang mewakili empat koordinat dari bounding box yang diprediksi}$

$t_i^* = \text{ground truth bounding box yang terkait dengan anchor positif}$

$N_{cls} = \text{nilai normalisasi } cls, \text{ ukuran mini-batch } (\sim 256) \text{ berdasarkan [34]}$

$N_{box} = \text{nilai normalisasi } reg, \text{ ukuran mini-batch } (\sim 2400) \text{ berdasarkan [34]}$

$\lambda = \text{parameter penyeimbang, sehingga } L_{cls} \text{ dan } L_{box} \text{ memiliki bobot yang sama}$

$y_{ij} = \text{label sel } (i, j) \text{ pada true mask untuk wilayah ukuran } m \times m$

$\hat{y}_{ij}^k = \text{nilai prediksi dari sel } (i, j) \text{ pada mask yang dipelajari untuk ground truth } K$

Classification loss (L_{cls}) adalah *log loss* atas dua kelas (objek dan non-objek) [34]. Fungsi *log loss* dijabarkan pada Persamaan (3.9).

$$L_{cls}(p_i, p_i^*) = - \sum_i p_i^* \log p_i \quad (3.9)$$

Regression loss digunakan persamaan $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$, di mana R adalah *robust loss function (smooth L₁)* [34]. *Smooth L₁* dijabarkan pada Persamaan (3.10).

$$\text{smooth}_{L_1}(x) = \begin{cases} 0,5x^2, & |x| < 1 \\ |x| - 0,5, & \text{sebaliknya} \end{cases} \quad (3.10)$$

Pada *bounding-box regression* digunakan parameterisasi empat koordinat dijabarkan pada Persamaan (3.11), (3.12), (3.13), dan (3.14). Persamaan-persamaan

tersebut dapat diperoleh *bounding-box regression* dari *anchor box* terhadap *ground truth box* terdekat [34].

$$t_x = \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a} \quad (3.11)$$

$$t_w = \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right) \quad (3.12)$$

$$t_x^* = \frac{x^* - x_a}{w_a}, \quad t_y^* = \frac{y^* - y_a}{h_a} \quad (3.13)$$

$$t_w^* = \log\left(\frac{w^*}{w_a}\right), \quad t_h^* = \log\left(\frac{h^*}{h_a}\right) \quad (3.14)$$

Dimana,

x = absis pusat prediksi *box*

x_a = absis pusat *anchor box*

x^* = absis pusat *ground-truth box*

y = ordinat pusat prediksi *box*

y_a = ordinat pusat *anchor box*

y^* = ordinat pusat *ground-truth box*

w = lebar prediksi *box*

w_a = lebar *anchor box*

w^* = lebar *ground-truth box*

h = tinggi prediksi *box*

h_a = tinggi *anchor box*

h^* = tinggi *ground-truth box*

t_x = hasil regresi absis pusat *box* prediksi

t_x^* = hasil regresi absis pusat *ground-truth box*

t_y = hasil regresi ordinat pusat *box* prediksi

t_y^* = hasil regresi ordinat pusat *ground truth box*

t_w = hasil regresi lebar *box* prediksi

t_w^* = hasil regresi lebar *ground truth box*

t_h = hasil regresi tinggi *box* prediksi

t_h^* = hasil regresi tinggi *ground truth box*

III.5. Hyperparameter

Parameter tingkat arsitektur disebut sebagai *hyperparameters* untuk membedakannya dari parameter model yang dilatih melalui *backpropagation* (seperti dijelaskan pada bagian III.2) [47]. *Hyperparameter tuning* dapat menghindari *overfitting* pada *dataset* validasi dan juga merupakan model *ensembling*, seperti penentuan berapa banyak *layer*, unit, filter, dll yang ada, [47]. *Hyperparameter* perlu diatur dan disesuaikan dengan *dataset* yang digunakan untuk memperoleh model *machine learning* yang optimal. Berikut adalah beberapa *hyperparameter* yang di-tuning dalam penelitian ini.

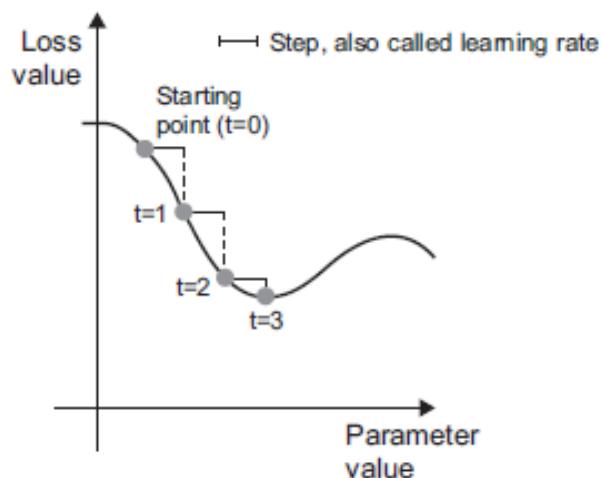
III.5.1. Epoch, Batch Size, dan Step

Jumlah *epoch* menentukan berapa kali pelatihan harus melalui *dataset* penuh [48]. Satu *epoch* adalah *dataset* latih penuh melalui *neural network* secara *forward* dan *backward* hanya sekali. *Batch size* adalah jumlah *dataset* pelatihan dalam satu *step*. Ukuran *batch* dapat dipilih berdasarkan RAM yang tersedia [48]. Semakin tinggi ukuran *batch*, semakin banyak RAM yang dibutuhkan [48]. Satu *epoch* biasanya terdiri dari banyak *step*, sebagai contoh jika *dataset* latih memiliki 2000 citra dan *batch size* sebesar 10, maka terdiri dari 200 *steps* per *epoch*-nya.

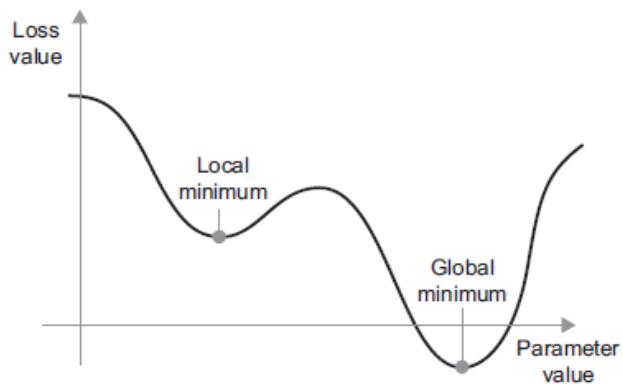
III.5.2. Learning Rate dan Learning Momentum

Learning rate merupakan *hyperparameter* yang menentukan seberapa besar *step* atau langkah untuk menurunkan nilai *loss*-nya mencapai minimum global atau minimum lokal, seperti ditunjukkan pada Gambar 3.28. *Learning rate* yang terlalu kecil, penurunan ke bawah kurva akan membutuhkan banyak iterasi dan bisa

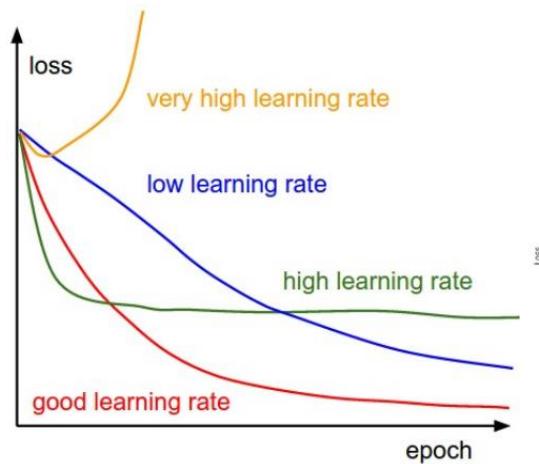
berhenti pada minimum lokal [47]. Pada Gambar 3.29. dapat dilihat bahwa di sekitar parameter tertentu terdapat minimum lokal dan pergerakan ke kiri akan mengakibatkan peningkatan nilai *loss*, begitu juga dengan bergerak ke kanan [47]. Nilai parameter yang sedang di-*optimize* dapat terjebak pada minimum lokal akibat dari *learning rate* yang terlalu kecil, sehingga tidak dapat menuju ke minimum global. Namun, *learning rate* yang terlalu besar pembaruan bobotnya akan membawa ke lokasi yang acak pada kurva [47]. Permasalahan terjebak pada minimum lokal dapat diatasi dengan *learning momentum* yang sama halnya pada momentum dalam fisika [47]. Nilai momentum yang cukup dapat membuat bola tidak akan terjebak di jurang dan akan berakhir di minimum global [47]. Dalam praktiknya, ini berarti memperbarui nilai parameter tidak hanya berdasarkan nilai gradien saat ini tetapi juga pada pembaruan parameter sebelumnya [47].



Gambar 3.28. *Stochastic Gradient Descent* 1D (satu parameter *learnable*) [47].



Gambar 3.29. Minimum lokal dan minimum global [47].



Gambar 3.30. Efek dari nilai *learning rate* [63].

Pada Gambar 3.30. adalah contoh dari efek nilai *learning rate* yang berbeda pada kurva *loss function* terhadap setiap *epoch*-nya. Nilai *learning rate* rendah memiliki peningkatan yang linier [63]. *Learning rate* tinggi memiliki bentuk yang lebih eksponensial [63]. *Learning rate* yang lebih tinggi lagi akan meluruhkan nilai *loss function* lebih cepat, tetapi terjebak pada *loss* yang lebih buruk (garis hijau) [63].

III.6. Evaluasi Performa

Metrik yang digunakan untuk mengukur performa model deteksi objek adalah nilai akurasi, presisi, *recall*, *F1-Score*, dan *mean average precision* [64]. Sebelum memperoleh nilai-nilai tersebut, kita harus meninjau beberapa konsep dasar terlebih dahulu. *Ground truth* adalah *output* yang diinginkan dari hasil deteksi objek (*bounding box* dan kelas objek). *Confidence score* adalah hasil prediksi nilai probabilitas bahwa *anchor box* berisi objek, seperti ditunjukkan pada Gambar 3.31. Namun, nilai IoU yang telah dijelaskan pada III.4.4. juga digunakan dalam metrik evaluasi performa. Nilai *confidence score* dan IoU digunakan sebagai kriteria yang menentukan suatu deteksi adalah *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) yang didefinisikan sebagai berikut:

1. *True Positive* (TP) merupakan hasil di mana model benar memprediksi kelas positif, secara teknis TP adalah objek yang memenuhi tiga kondisi sebagai berikut:
 - *Confidence score > threshold* [65].
 - Kelas objek yang diprediksi sama dengan kelas *ground truth* [65].
 - *Bounding box* yang diprediksi memiliki $\text{IoU} > \text{threshold}$ [65].
2. *False Positive* (FP) merupakan hasil di mana model salah memprediksi kelas positif, secara teknis FP adalah objek yang memiliki *confidence score > threshold*, tetapi kedua kondisi berikut tidak terpenuhi salah satunya atau keduanya:
 - Kelas objek yang diprediksi sama dengan kelas *ground truth* [65].
 - *Bounding box* yang diprediksi memiliki $\text{IoU} > \text{threshold}$ [65].
3. *False Negative* (FN) merupakan hasil di mana model salah memprediksi kelas negatif, secara teknis FN adalah objek *ground truth* yang seharusnya terdeteksi namun tidak terdeteksi karena nilai *confidence score < threshold*.
4. *True Negative* (TN) merupakan hasil di mana model memprediksi kelas negatif dengan benar, secara teknis TN adalah nilai *confidence score < threshold* karena tidak terdapat objek. Namun, nilai TN biasanya diabaikan

dalam deteksi objek karena ada banyak *bounding box* dalam jumlah tak terbatas yang seharusnya tidak terdeteksi dalam satu citra [65].



Gambar 3.31. Contoh hasil prediksi *confidence score* dan *bounding box* [64].

III.6.1. Akurasi

Akurasi merupakan persentase objek yang benar diprediksi (TP) dari seluruh prediksi objek (TP + FP + FN). Semakin tinggi nilai akurasi maka semakin banyak objek yang diprediksi benar [64]. Persamaan akurasi yang digunakan pada penelitian ini dijabarkan pada Persamaan (3.15).

$$Akurasi = \frac{TP}{TP + FP + FN} \quad (3.15)$$

III.6.2. Presisi

Presisi mengukur seberapa akurat prediksi model, yaitu persentase hasil prediksi objek benar (sesuai dengan *ground truth*) dari total objek yang telah diprediksi positif [65]. Presisi didefinisikan sebagai jumlah TP dibagi dengan jumlah TP dan FP yang dijabarkan pada Persamaan (3.16).

$$Presisi = \frac{TP}{TP + FP} = \frac{TP}{total\ deteksi} \quad (3.16)$$

III.6.3. Recall

Recall adalah kemampuan model untuk menemukan semua kasus yang relevan (semua *ground truth bounding box*) [65]. Ini adalah persentase prediksi positif yang benar di antara semua *ground truth* yang diberikan. Persamaan *recall* dijabarkan pada Persamaan (3.17).

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{total ground truth}} \quad (3.17)$$

III.6.4. F1-Score

F1-Score digunakan untuk mengukur akurasi model dengan cara menggabungkan nilai presisi dan *recall* yang telah diperoleh. *F1-Score* adalah *harmonic mean* dari presisi dan *recall*. Persamaan *F1-Score* dijabarkan pada Persamaan (3.18).

$$F_1 = \frac{2}{recall^{-1} + presisi^{-1}} = 2 \times \frac{presisi \times recall}{presisi + recall} \quad (3.18)$$

III.6.5. Average Precision (AP) dan mean Average Precision (mAP)

Deteksi objek yang baik adalah yang dapat menemukan semua objek *ground-truth* ($FN = 0 \equiv$ nilai *recall* tinggi) saat mengidentifikasi hanya objek yang relevan ($FP = 0 \equiv$ nilai presisi tinggi) [65]. Detektor objek dapat dianggap baik jika presisinya tetap tinggi seiring dengan peningkatan *recall*-nya yang berarti bahwa jika *confidence threshold* bervariasi, presisi dan *recall* akan tetap tinggi [65]. Oleh karena itu, *area under the curve* (AUC) yang tinggi cenderung menunjukkan presisi tinggi dan *recall* tinggi [65]. Dalam kasus praktis, plot presisi \times *recall* berupa kurva zig-zag yang menimbulkan tantangan untuk pengukuran AUC-nya yang akurat [65]. Zig-zag tersebut dapat dihilangkan dengan melakukan estimasi AUC, salah satunya adalah melakukan interpolasi 11 titik [65]. Contoh hasil plot presisi \times *recall* dan interpolasinya pada *confidence threshold* bervariasi dapat dilihat pada Gambar 3.32.

Dalam interpolasi 11 titik, bentuk kurva presisi \times *recall* diringkas dengan merata-ratakan nilai presisi maksimum pada sekumpulan 11 titik *recall* yang berjarak sama $[0, 0.1, 0.2, \dots, 1]$, seperti yang diberikan oleh Persamaan (3.19).

$$AP_{11} = \frac{1}{11} \sum_{R \in \{0, 0.1, \dots, 0.9, 1\}} P_{interp}(R), \quad (3.19)$$

Dimana,

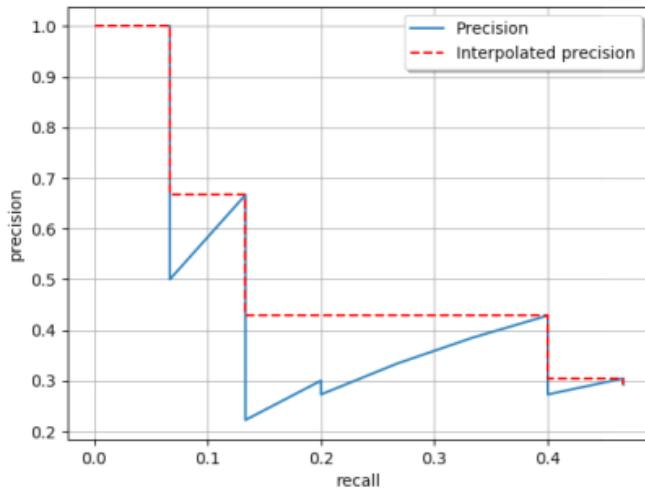
$$P_{interp}(R) = \max_{\tilde{R}: \tilde{R} \geq R} P(\tilde{R}) \quad (3.20)$$

Pada definisi dari AP, ketimbang menggunakan presisi $P(R)$ yang diamati pada setiap level *recall* R , AP diperoleh dengan mempertimbangkan presisi maksimum $P_{interp}(R)$ yang nilai *recall*-nya lebih besar dari R , seperti ditunjukkan Gambar 3.32. pada plot *original* dan plot interpolasi [65].

Mean AP (mAP) adalah metrik yang digunakan untuk mengukur keakuratan model deteksi objek [65]. Metrik mAP adalah produk presisi dan *recall* pada *bounding box* yang terdeteksi [48]. mAP memiliki nilai dari 0 hingga 100, semakin tinggi angkanya semakin baik modelnya. mAP hanyalah AP rata-rata untuk semua kelas yang dijabarkan pada Persamaan (3.21).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3.21)$$

Dimana AP_i merupakan AP di kelas ke- i dan N adalah jumlah total kelas yang dievaluasi.



Gambar 3.32. Kurva presisi \times recall [65].

III.6.6. Mean Absolute Error (MAE)

Mean absolute error (MAE) merupakan rata-rata perbedaan absolut antara nilai aktual dan prediksi. Metrik MAE digunakan untuk menentukan *error* antara hasil prediksi dan nilai aktualnya, sehingga performa model dapat ditentukan. Persamaan MAE dijabarkan pada Persamaan (3.22).

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (3.22)$$

Dimana,

y_i = nilai prediksi ke- i

x_i = nilai aktual ke- i

n = total data

III.6.7. Overfitting dan Underfitting

Model *machine learning* dapat terlalu besar atau terlalu kecil untuk permasalahan tertentu. Kedua hal tersebut dapat diklasifikasikan sebagai *underfitting* atau *overfitting* [48]. *Underfitting* terjadi ketika model terlalu kecil dan dapat diukur ketika *loss function* pada proses pelatihan yang kurang optimal. *Overfitting* terjadi ketika model terlalu besar dan ada perbedaan yang besar antara

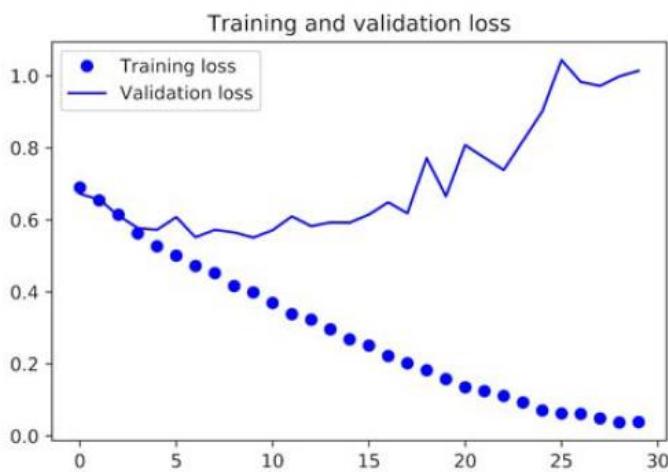
loss function pelatihan dan validasi. *Underfitting* dapat diselesaikan dengan metode berikut [48]:

- Memperoleh lebih banyak data.
- Menggunakan model yang lebih besar.
- Jika jumlah datanya kecil, dapat menerapkan teknik *transfer learning* atau melakukan augmentasi data.

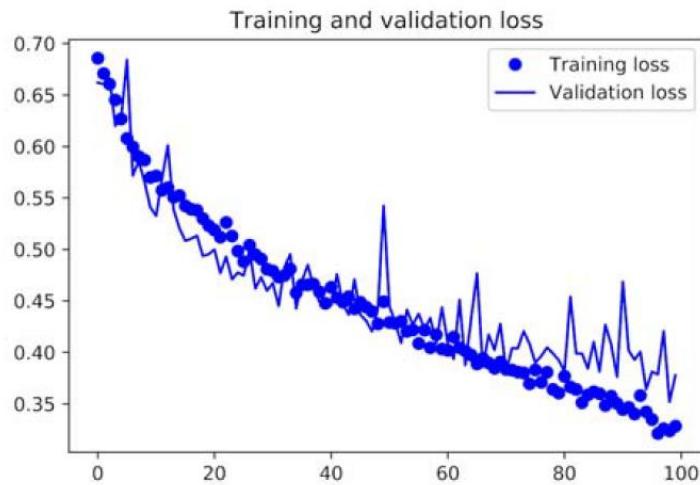
Overfitting dapat diselesaikan dengan metode berikut [48]:

- Regularisasi menggunakan teknik seperti *dropout* dan normalisasi *batch*.
- Melakukan augmentasi data.

Pada Gambar 3.33. merupakan contoh plot *loss function* dengan kasus *overfitting*. Pada plot grafik tersebut dapat dilihat bahwa model bekerja sangat baik pada data latih atau kurang *generalize*, sehingga prediksi pada data validasi tidak sebaik pada data latih. Namun, pada Gambar 3.34. plot *loss function* data latih dapat melacak dengan cermat plot *loss function* data validasi dikarenakan model lebih *generalize*, sehingga dapat memprediksi data validasi dengan baik.



Gambar 3.33. Contoh *loss function* dengan *overfitting* [47].



Gambar 3.34. Contoh *loss function* tanpa *overfitting* [47].

BAB IV

PELAKSANAAN PENELITIAN

IV.1. Alat dan Bahan Penelitian

IV.1.1. Alat Penelitian

Penelitian ini menggunakan beberapa alat berupa perangkat lunak dan perangkat keras sebagai berikut:

1. Komputer Personal Portabel:
 - a. Prosesor : AMD Ryzen 7 4800HS (2.9 GHz)
 - b. RAM : 16 GB
 - c. GPU : NVIDIA GeForce GTX1650Ti GDDR6 4GB
2. Perangkat lunak yang digunakan dalam penelitian:
 - a. *Google Colaboratory Pro+* yang dilengkapi Python 3.7.12 dan TensorFlow 2.6.0. *Google Colab Pro+* digunakan sebagai media pelatihan model *Mask R-CNN* dengan GPU NVIDIA Tesla P100.
 - b. OpenCV 4.5.3 digunakan sebagai pengolahan citra.
 - c. VGG *Image Annotator* (VIA 2.0.1.1) digunakan sebagai anotasi citra.
 - d. *Visual Studio Code* digunakan sebagai *code editor* untuk augmentasi citra, segmentasi citra, perhitungan jumlah sel berdasarkan label, pemisahan data, dan konfigurasi model *Mask R-CNN*.
 - e. *Microsoft Excel* 2019 digunakan sebagai pemisahan data dan perhitungan MAE.
 - f. *Google Drive* sebesar 2 TB digunakan sebagai penyimpanan model *Mask R-CNN*, data latih, data validasi, dan *logs* hasil latih model.

IV.1.2. Bahan Penelitian

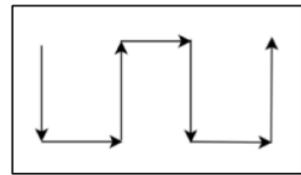
Bahan penelitian yang digunakan pada penelitian ini berupa citra digital sel darah dari pasien pengidap *Acute Lymphoblastic Leukemia* tipe L1. Citra sel darah telah diberi label oleh hematologis dari Rumah Sakit Umum Pusat (RSUP) Dr. Sardjito Yogyakarta. Sampel citra sel darah yang digunakan pada penelitian berjumlah 1.461 citra dengan resolusi 4100×3075 .

IV.1.2.1. Akuisisi Data

Sampel citra sel darah diperoleh dengan menggunakan mikroskop cahaya termodifikasi yang dilengkapi oleh *microstepper* dan kamera digital, seperti ditampilkan pada Gambar 4.1. Citra apusan darah pada mikroskop diambil citranya dengan kamera digital. Mikroskop dibantu dengan *microstepper* untuk mengatur fokus bayangan yang ditangkap lensa objektif dengan menaikkan dan menurunkan meja preparat. Kamera digital pada lensa okuler langsung dihubungkan pada komputer personal secara *real-time*. Citra digital yang telah diambil kemudian disimpan pada komputer personal dan kemudian data tersebut dilakukan pelabelan oleh hematologis. Citra digital yang telah diberi label tersebut digunakan untuk mengembangkan model *Mask R-CNN* sebagai pendekripsi sel limfoblas dan non-limfoblas. Rangkaian motor *microstepper* yang terhubung dengan mikroskop berfungsi untuk menggeserkan preparat secara konsisten. Preparat digerakkan secara mekanik-elektrik melalui hubungan komputer personal dengan *microstepper*-nya. Citra sel darah diambil dengan pola zig-zag, seperti pada Gambar 4.2. Citra diambil setiap menemukan minimal satu citra sel darah putih pada jendela kerja kamera yang tampak pada komputer personal.



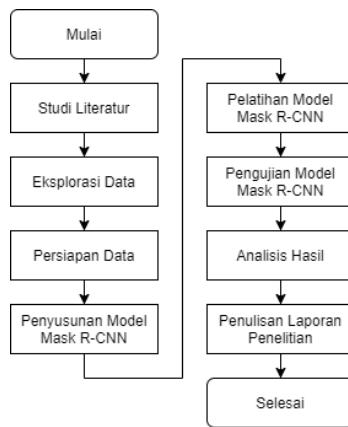
Gambar 4.1. Ilustrasi akuisisi data.



Gambar 4.2. Pola pergerakan preparat.

IV.2. Tata Laksana Penelitian

Proses penelitian dilakukan dengan tahapan sesuai dengan diagram alir pada Gambar 4.3. sebagai berikut.



Gambar 4.3. Diagram alir tata laksana penelitian.

IV.2.1. Studi Literatur

Studi literatur merupakan tahap awal peneliti untuk mengkaji penelitian-penelitian sebelumnya yang telah dilakukan terkait pengolahan citra, klasifikasi, dan deteksi objek pada sel darah maupun pada kasus *Acute Lymphoblastic Leukemia*. Tahap ini akan mengetahui kemajuan pengetahuan yang telah dicapai pada kasus penelitian ini, sehingga dapat membantu dalam pemilihan dan penyusunan model *deep learning* untuk mencapai tujuan penelitian. Sumber literatur yang digunakan berasal dari artikel ilmiah, buku, dan sumber lainnya yang terpercaya dan telah disitasi secara lengkap dalam daftar pustaka. Fokus kajian dalam penelitian ini adalah deteksi objek dan segmentasi *instance* berbasis *Mask R-CNN* untuk sel limfoblas dan non-limfoblas pada kasus *Acute Lymphoblastic Leukemia* tipe L1.

IV.2.2. Eksplorasi Data

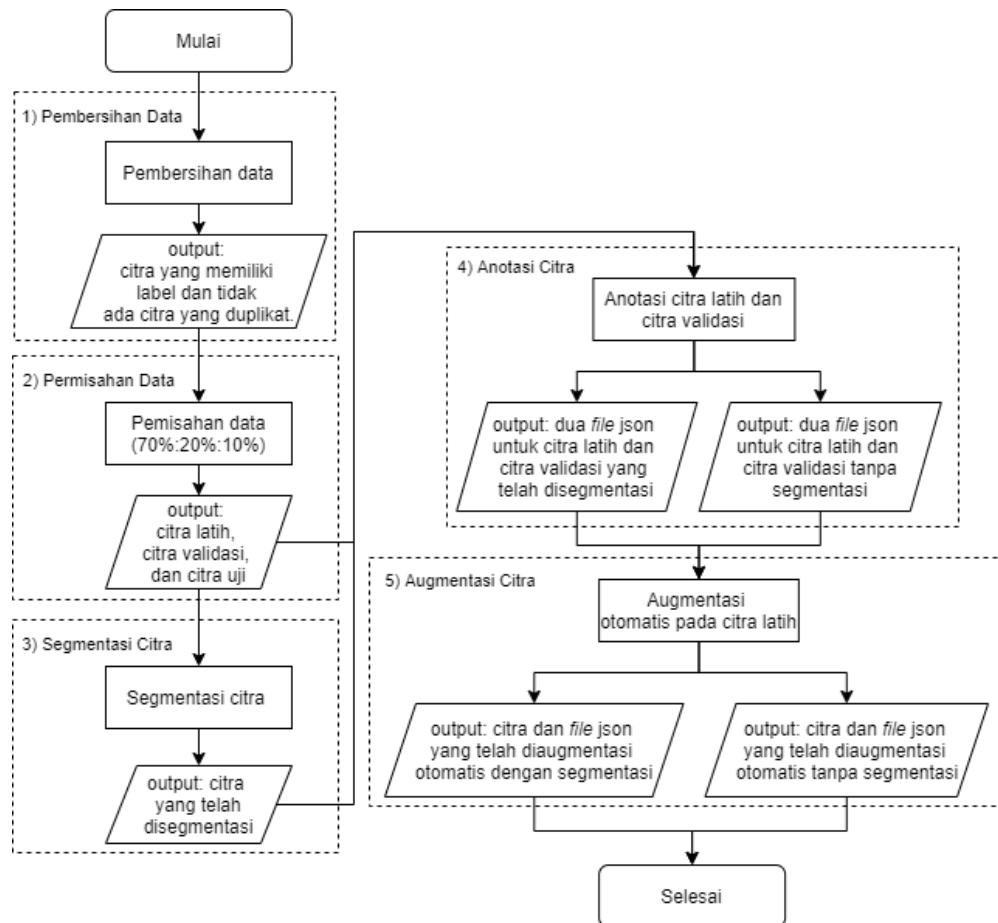
Pada 1.461 citra sel darah telah diberi label oleh hematologis RSUP Dr. Sardjito untuk setiap jenis WBC. Label untuk setiap jenis WBC terdapat pada nama *file* citra tersebut. Pemberian label dimulai dengan urutan dari kiri ke kanan. Jika WBC tersebut berada pada posisi atas atau bawah citra, maka akan ada keterangan tambahan dengan tanda kurung untuk mengetahui label tersebut untuk WBC di atas maupun di bawah pada citra. Pada Tabel 4.1. berikut adalah beberapa jenis WBC dan labelnya yang telah diberikan pada citra.

Tabel 4.1. Jenis WBC dan label pada citra.

No.	Jenis WBC	Label
1	Limfosit	L
2	Limfoblas	B
3	Segmen	S
4	Monosit	M

IV.2.3. Persiapan Data

Persiapan data dilakukan dengan lima tahap yang ditampilkan pada diagram alir Gambar 4.4. sebagai berikut.



Gambar 4.4. Diagram alir persiapan data.

IV.2.3.1. Pembersihan Data

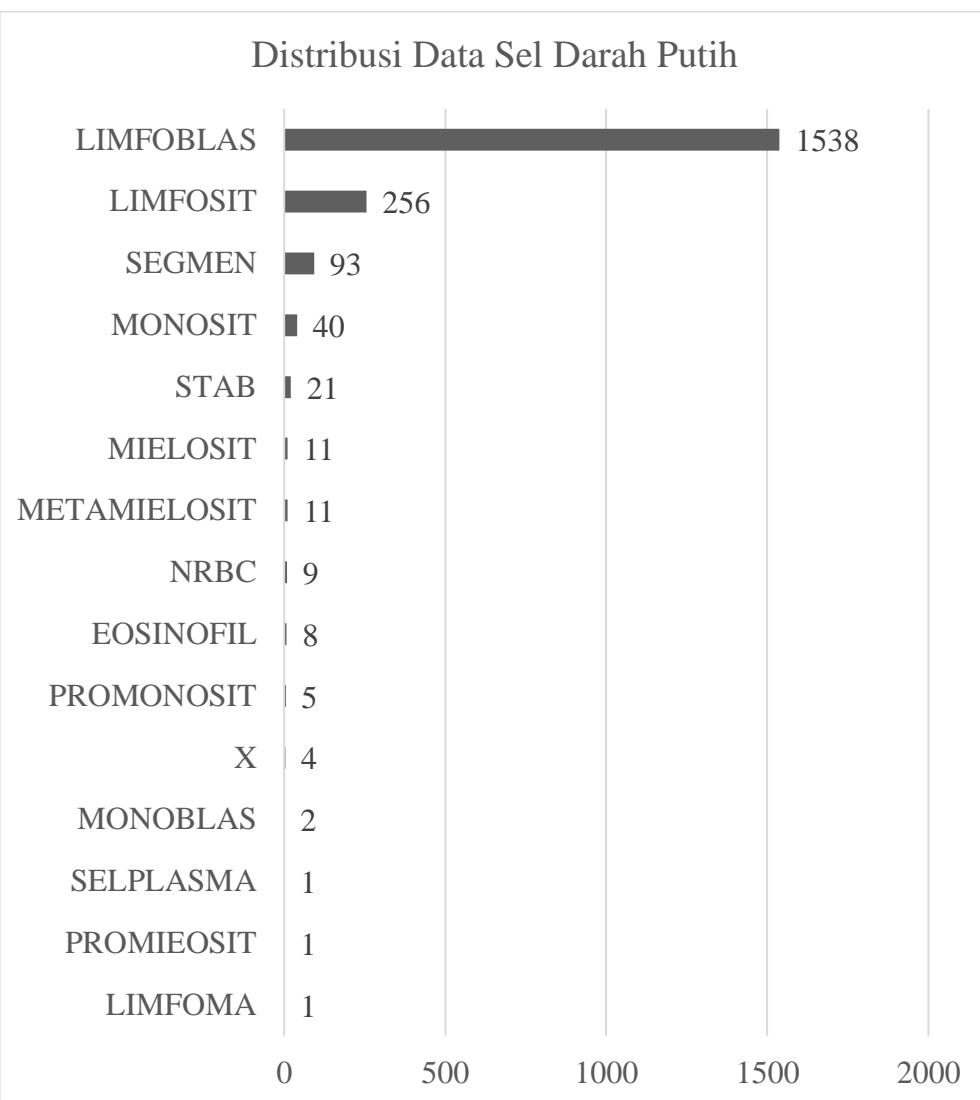
Sampel citra sel darah yang diperoleh dari hematologis Rumah Sakit Umum Pusat (RSUP) Dr. Sardjito Yogyakarta tidak sepenuhnya dapat digunakan untuk penelitian. Oleh karena itu diperlukan pembersihan data, sehingga hanya citra tertentu saja yang dapat digunakan pada penelitian ini. Sampel citra sel darah awalnya berjumlah jumlah 1.725 citra, setelah dilakukan pembersihan data diperoleh 1.461 citra yang dapat digunakan dalam penelitian ini. Hal yang

menentukan apakah citra dapat digunakan pada penelitian adalah citra yang memiliki label untuk setiap WBC dan tidak terjadinya duplikat untuk setiap citra. Setelah proses pembersihan data, maka dapat dilakukan perhitungan otomatis berdasarkan nama *file* citra tersebut menggunakan program Python sederhana untuk dilakukan proses pemisahan data.

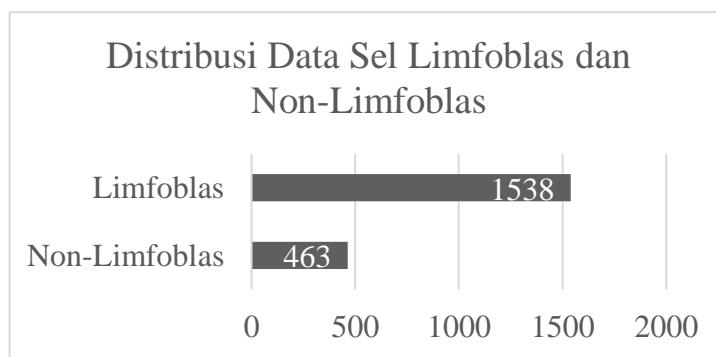
IV.2.3.2. Pemisahan Data

Sampel citra sel darah dilakukan pengelompokan sesuai dengan kebutuhan dari fokus penelitian, yaitu untuk mendeteksi sel limfoblas dan non-limfoblas. Jumlah citra yang telah diberi label oleh hematologis adalah 1.461 citra yang memiliki distribusi WBC pada Gambar 4.5. Fokus penelitian dipusatkan pada pendektsian limfoblas dan non-limfoblas dikarenakan untuk menghitung jumlah sel kanker (limfoblas) dari total semua jenis WBC. Jenis WBC selain sel limfoblas dijadikan satu label, yaitu non-limfoblas seperti yang ditampilkan pada Gambar 4.6.

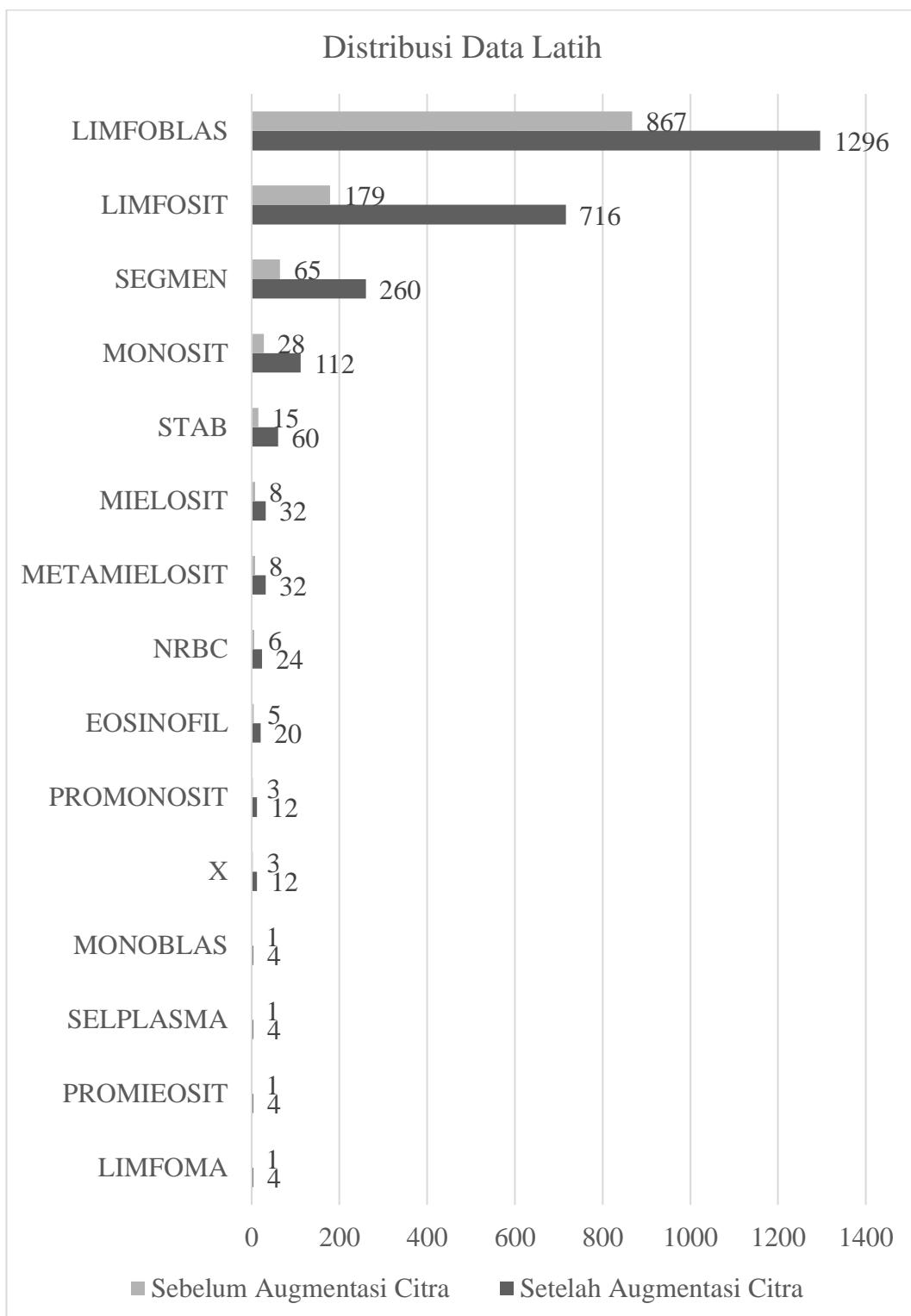
Pada *dataset* citra tersebut perlu dilakukan pemisahan data menjadi data latih, data validasi, dan data uji. Namun, untuk data latih dan data validasi perlu dilakukan pemisahan secara seimbang pada setiap jenis WBC (limfoblas dan non-limfoblas). *Dataset* pada penelitian ini dibagi menjadi 70% untuk data latih, 20% untuk data validasi, dan 10% untuk data uji yang didasarkan jumlah sel limfoblas dan non-limfoblas. Distribusi data setelah pemisahan ditampilkan pada Gambar 4.7. untuk data latih, Gambar 4.8. untuk data validasi, dan Gambar 4.11. untuk data uji. Pada penelitian ini hanya data latih yang dilakukan augmentasi atau *oversampling* agar tidak terjadi *class imbalance* dan *overfitting*, seperti yang telah dijelaskan pada III.2.3. Untuk menghindari *class imbalance* dilakukan *undersampling* pada citra yang terdiri hanya sel limfoblas, sehingga total citra yang digunakan dalam penelitian ini berjumlah 1.056 citra. Citra latih terdiri dari 865 citra, citra validasi terdiri dari 122 citra, dan citra uji terdiri dari 69 citra. Hasil pasca augmentasi citra pada data latih memiliki kelas yang seimbang, yaitu berjumlah 1.296 untuk masing-masing kelas, seperti ditampilkan pada Gambar 4.9. dan untuk metode augmentasi yang digunakan dijelaskan pada IV.2.3.5.



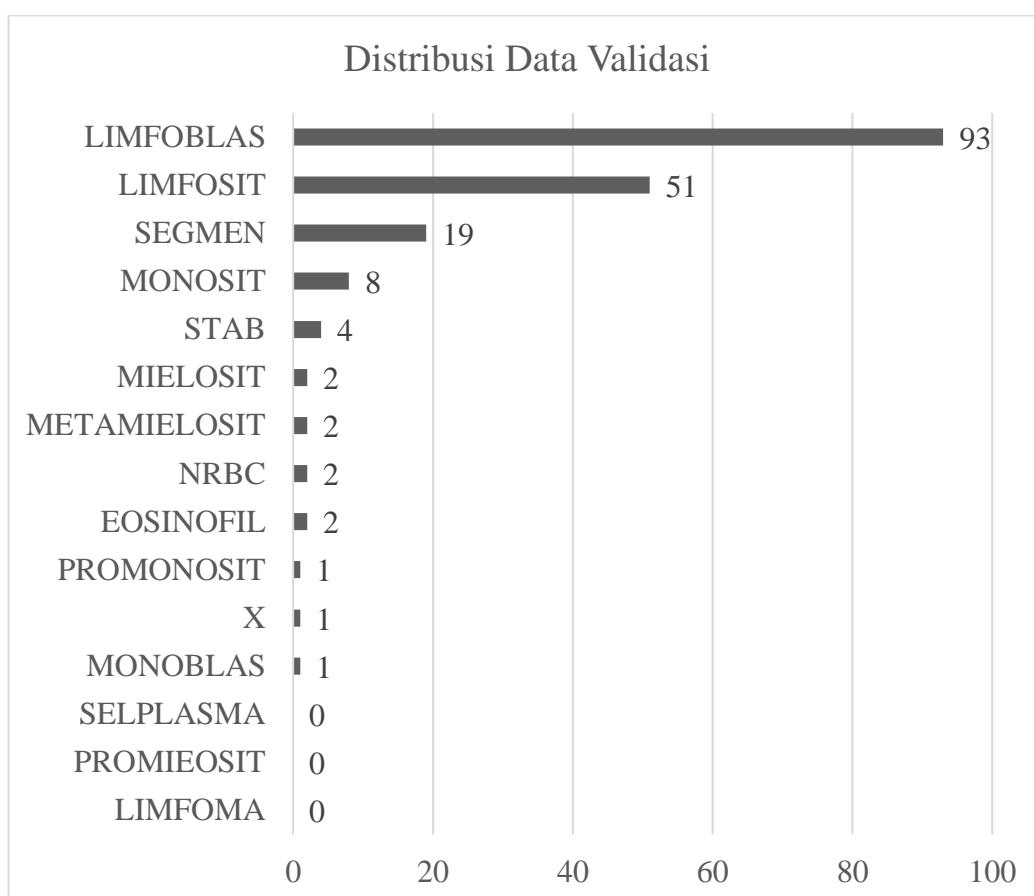
Gambar 4.5. Distribusi data WBC.



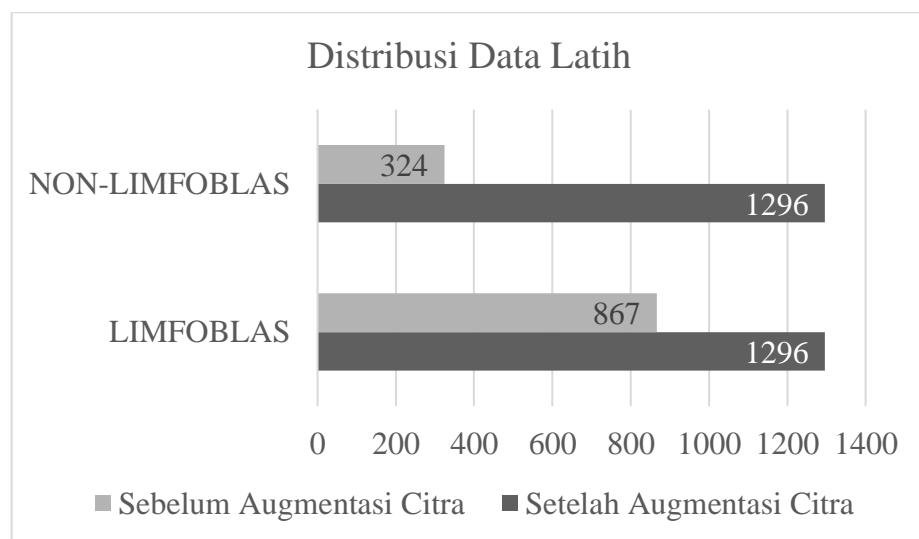
Gambar 4.6. Distribusi data sel limfoblas dan non-limfoblas.



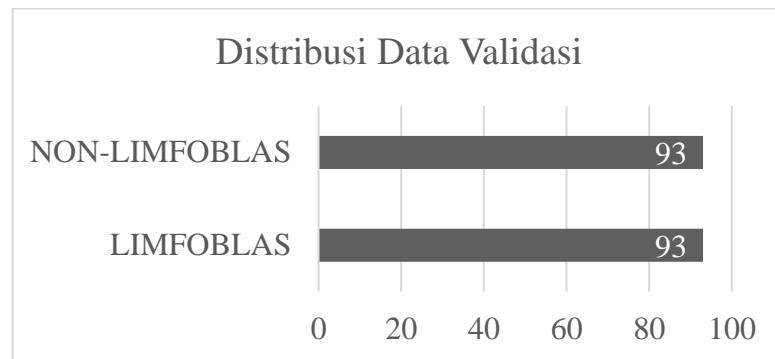
Gambar 4.7. Distribusi data latih.



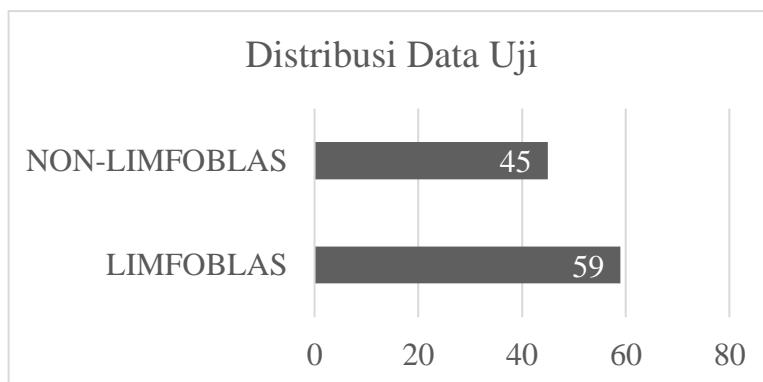
Gambar 4.8. Distribusi data validasi.



Gambar 4.9. Distribusi data latih biner (limfoblas dan non-limfoblas).



Gambar 4.10. Distribusi data validasi biner (limfoblas dan non-limfoblas).

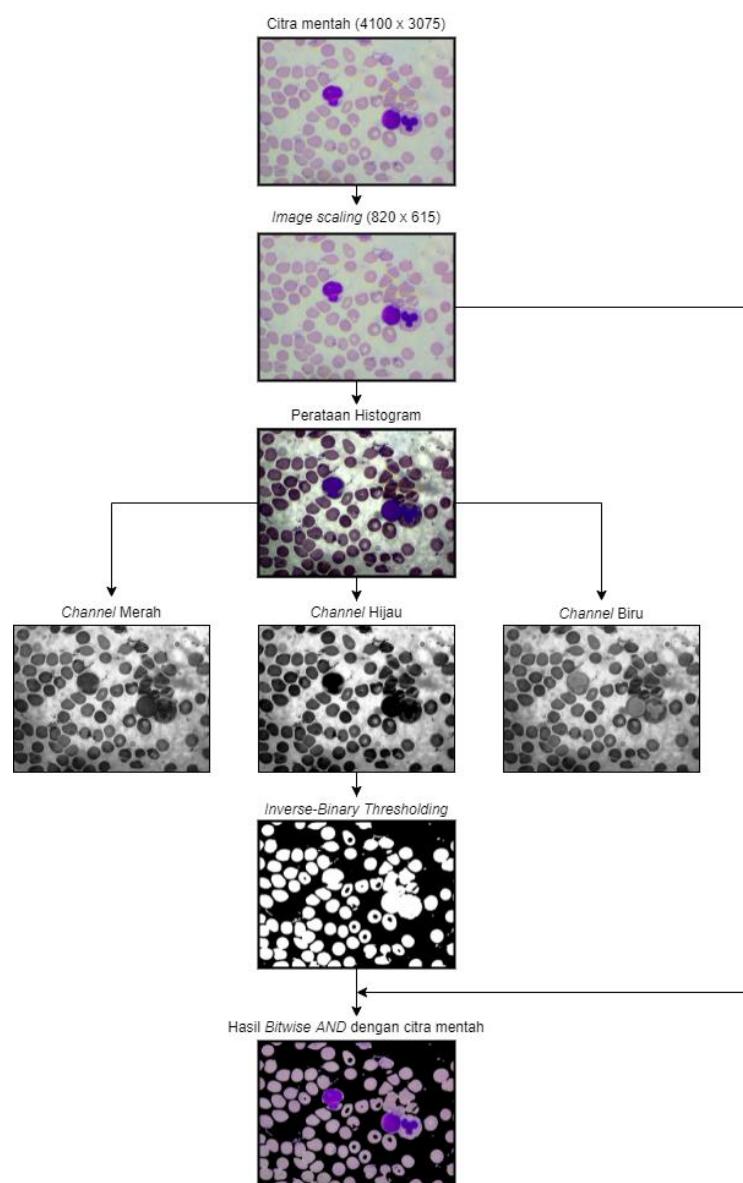


Gambar 4.11. Distribusi data uji biner (limfoblas dan non-limfoblas).

IV.2.3.3. Segmentasi Citra

Pada penelitian ini digunakan beberapa teknik pengolahan citra untuk melakukan segmentasi citra sel darah dari latar belakangnya. Tujuan dilakukan segmentasi ini adalah untuk membandingkan performa model *Mask R-CNN* dengan adanya segmentasi dan tanpa adanya segmentasi (citra mentah). Ilustrasi untuk proses segmentasi sel ditampilkan pada Gambar 4.12. Resolusi awal citra adalah 4100×3075 , kemudian dilakukan *image scaling* sebesar 0,2 kalinya dari citra asli untuk mengurangi komputasi dalam tahap pelatihan dan deteksi tanpa adanya mengurangi fitur WBC secara signifikan. Pada tahapan pengolahan citra dilakukan segmentasi sel terhadap latar belakangnya dengan melakukan *thresholding* pada *channel* hijau. Sebelum dilakukan *thresholding* perlu adanya pemerataan histogram untuk mendistribusikan intensitas pada citra, sehingga kontras antara sitoplasma

dan inti sel terhadap latar belakangnya lebih tinggi. Setelah dilakukan pemerataan histogram, *channel* hijau dari pemerataan tersebut dilakukan *inverse-binary thresholding* (metode dijelaskan pada III.3.3.2) agar nilai di atas *threshold* menjadi hitam (latar belakang citra) dan nilai di bawah *threshold* menjadi putih (WBC dan RBC). Hasil dari *thresholding* tersebut dapat digunakan sebagai *mask* pada operasi *Bitwise AND* untuk mengekstrak bagian WBC dan RBC dari citra mentah.



Gambar 4.12. Ilustrasi proses segmentasi sel darah dari citra mentah.

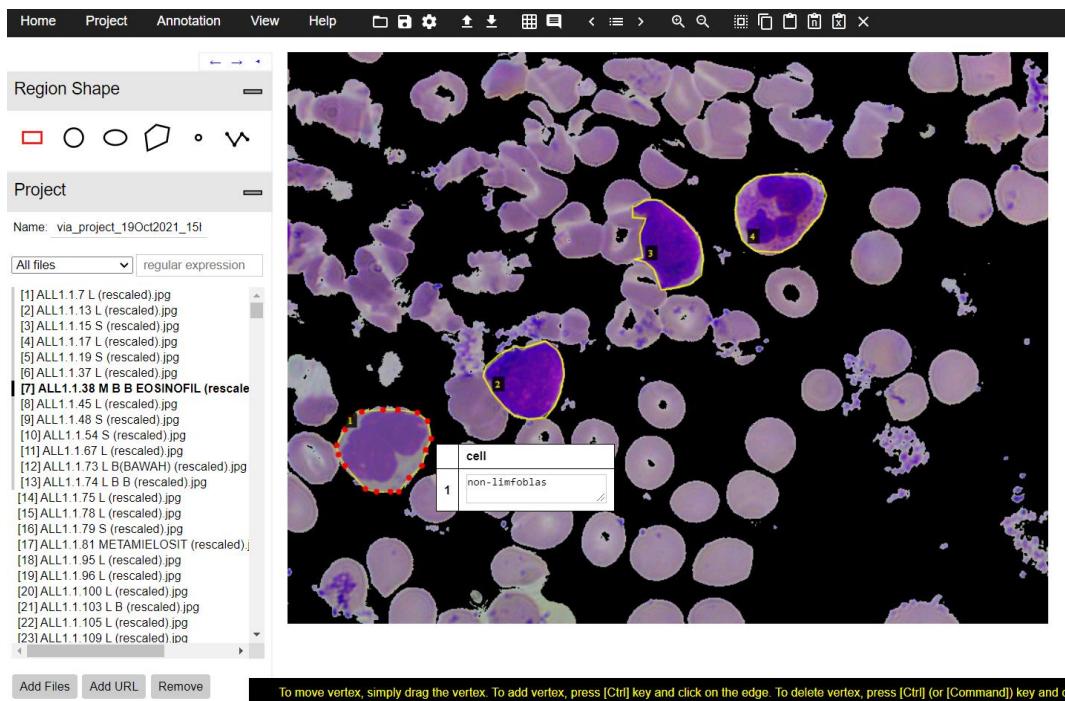
IV.2.3.4. Anotasi Citra

Model deteksi objek merupakan model yang melakukan lokalisasi dan klasifikasi pada citra (telah dijelaskan pada III.3.1). Oleh karena itu, diperlukan label dari data latih dan data validasi yang merepresentasikan lokasi dan kelas objek pada citra untuk proses pelatihan model deteksi objek. Proses anotasi citra pada model *Mask R-CNN* dilakukan dengan pembentukan poligon sesuai bentuk dari WBC yang kemudian diikuti pemberian nama label (limfoblas atau non-limfoblas). Poligon digunakan untuk melacak batas dan daerah WBC kemudian diberi label berdasarkan jenis WBC. Pada model deteksi objek *Mask R-CNN* tidak menggunakan *bounding box* pada objek, tetapi menggunakan poligon karena model ini merupakan segmentasi *instance* yang perlu melakukan klasifikasi secara piksel-ke-piksel yang telah dijelaskan pada III.3.2. dan III.4.

Hasil anotasi citra pada data latih dan data validasi merupakan *ground truth* dari deteksi objek. Pembentukan poligon dilakukan sesuai dengan bentuk WBC secara keseluruhan. Pada penelitian ini dilakukan anotasi citra manual pada data latih dan data validasi. Untuk data latih, anotasi manual hanya dilakukan pada citra asli tidak pada citra yang telah diaugmentasi. Gambar 4.13. adalah proses anotasi pada citra mentah kemudian dilakukan proses anotasi pada citra yang telah disegmentasi seperti pada Gambar 4.14. untuk menyesuaikan kembali bentuk poligon menjadi bentuk WBC yang telah disegmentasi. Pada tahap ini dihasilkan empat *file* (dengan *extension*: .json), yaitu dua untuk data latih dan dua untuk data validasi yang masing-masing untuk citra mentah dan citra yang telah disegmentasi. Proses anotasi dilakukan dengan perangkat lunak *VGG Image Annotator* (VIA 2.0.1.1) yang dilakukan secara manual untuk setiap WBC. *File* (.json) hasil proses anotasi terdapat variabel *regions* pada setiap citra yang berisis koordinat-koordinat poligon dan label kelas pada poligon tersebut.



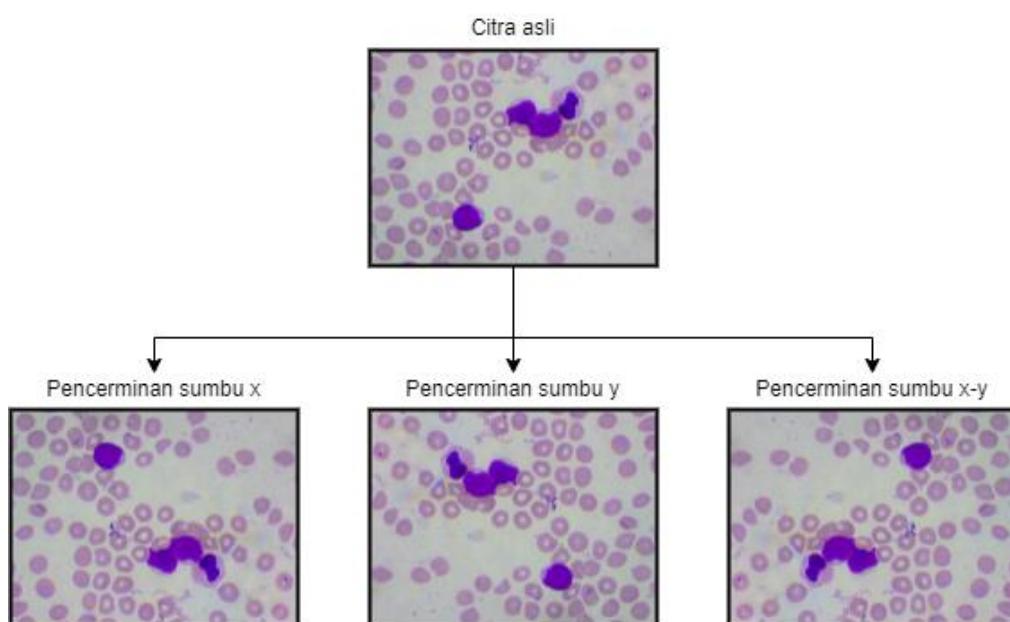
Gambar 4.13. Proses anotasi pada citra mentah.



Gambar 4.14. Proses anotasi pada citra yang telah disegmentasi.

IV.2.3.5. Augmentasi Citra

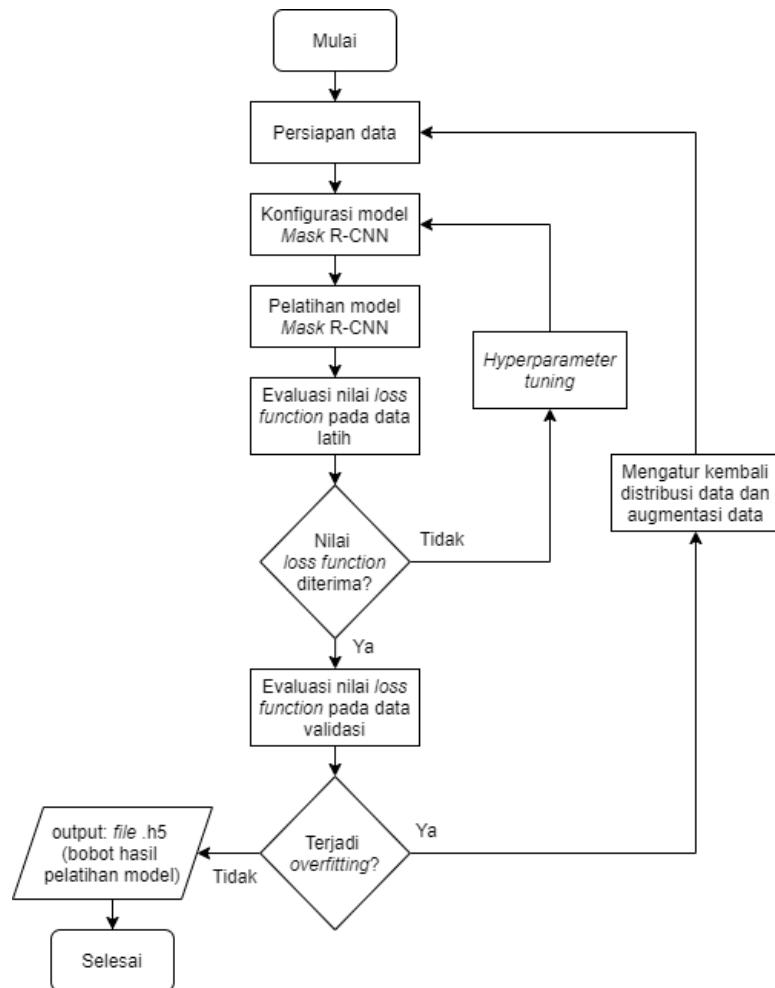
Pada data latih dilakukan augmentasi bertujuan untuk menyeimbangkan kelas dan menghindari *overfitting* pada data latih (telah dijelaskan pada III.2.3 dan III.2.4.). Augmentasi citra dilakukan dengan pencerminan pada sumbu x, sumbu y, dan kedua sumbu x-y, seperti ditampilkan pada Gambar 4.15. Pada proses augmentasi perlu dilakukan pengelompokan pada setiap citra yang terdiri dari jenis WBC apa saja agar tercapai kelas yang seimbang setelah dilakukan augmentasi. Kelas non-limfoblas adalah kelas minoritas pada kasus ini, sehingga augmentasi dilakukan pada citra yang terdiri setidaknya hanya satu non-limfoblas. Oleh karena itu setiap sel non-limfoblas akan menjadi empat kalinya dari jumlah sebelumnya, seperti ditampilkan pada Gambar 4.7. dan Gambar 4.9. Pada kedua *file* (.json), hasil proses anotasi manual sebelumnya, perlu ditambahkan koordinat poligon dan label kelas baru untuk citra hasil augmentasi. Penambahan koordinat poligon dan label kelas baru dapat dilakukan secara otomatis dengan menggunakan Python (*source code* terlampir) untuk melakukan pencerminan pada koordinat-koordinat poligon dan memberi label kelasnya pada citra hasil augmentasi.



Gambar 4.15. Augmentasi citra.

IV.2.4. Pelatihan Algoritma

Pelatihan model *Mask R-CNN* dilakukan pada *Google Colaboratory Pro+* menggunakan GPU NVIDIA Tesla P100 dan *Google Drive* 2TB sebagai penyimpanan model *Mask R-CNN*, data latih, data validasi, dan *logs* hasil latih model. Pelatihan model digunakan arsitektur *backbone* ResNet101 dikarenakan jumlah lapisan jaringan lebih dalam dibandingkan ResNet50 dan juga ketersediaan perangkat keras untuk melakukan pelatihan dengan arsitektur *backbone* ResNet101. *Pre-train weight* yang digunakan pada pelatihan ini berasal dari bobot COCO (*Common Object in Context*). Diagram alir proses pelatihan model *Mask R-CNN* pada penelitian ditampilkan pada Gambar 4.16.



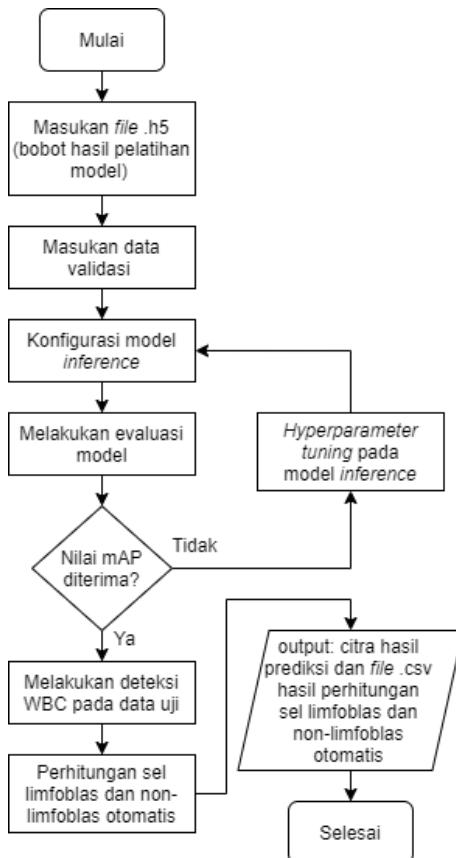
Gambar 4.16. Diagram alir proses pelatihan model *Mask R-CNN*.

Pelatihan model *Mask R-CNN* perlu dilakukan kofigurasi untuk menyesuaikan *dataset* yang digunakan pada penelitian ini. Konfigurasi tersebut berupa: arsitektur *backbone* yang digunakan, jumlah kelas, dan *hyperparameter* untuk pelatihan pada *dataset*. *Hyperparameter* yang diatur pada penelitian ini meliputi: *epochs*, *steps per epoch*, *validation steps*, *learning rate*, *detection minimum confidence*, *detection NMS threshold*, dan *RPN NMS threshold*. Kemudian proses pelatihan model *Mask R-CNN* dilakukan dengan menggunakan data latih dan data validasi. Hasil dari pelatihan ini berupa *file* dengan *extension*: .h5 yang merupakan bobot model hasil pelatihan. Hasil *loss function* pada data latih dan data validasi dapat dievaluasi menggunakan TensorBoard. Grafik pada TensorBoard dapat dilakukan pengamatan untuk *loss function* data latih maupun data validasi dan mengetahui apakah terjadi *underfitting* atau *overfitting* yang telah dijelaskan pada III.6.7. Jika model hasil pelatihan *underfitting* maka dapat dilakukan *hyperparameter tuning*. Namun, jika model mengalami *overfitting* dapat melakukan evaluasi ulang pada distribusi data dan augmentasi data untuk memperoleh model yang lebih *generalize* dan *robust* seperti yang telah dijelaskan pada III.2.3.

IV.2.5. Pengujian Algoritma Deteksi

Pengujian algoritma dilakukan berdasarkan hasil pelatihan algortima sebelumnya. Pada Gambar 4.17. merupakan diagram alir pengujian algoritma yang digunakan penelitian ini. Keluaran dari hasil pelatihan algoritma sebelumnya berupa *file* .h5 yang merupakan bobot hasil pelatihan. *File* .h5 tersebut digunakan sebagai proses *inference* pada data validasi dan data uji. Berdasarkan hasil pembobotan tersebut dapat dilakukan perhitungan mAP pada data validasi untuk melakukan evaluasi hasil pelatihan yang telah dijelaskan pada III.6. Jika nilai mAP yang diperoleh tidak maksimal maka dapat dilakukan *hyperparameter tuning* pada konfigurasi *inference* dengan *hyperparameter* sebagai berikut: *detection minimum confidence*, *detection NMS threshold*, dan *RPN NMS threshold*. Hasil konfigurasi *inference* dengan mAP terbaik akan dilakukan deteksi dengan data uji dan melakukan perhitungan sel limfoblas dan non-limfoblas secara otomatis. Keluaran

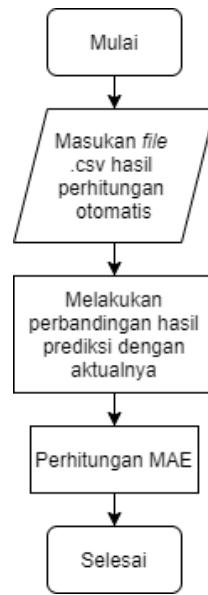
deteksi berupa citra hasil prediksi dan data dengan format *comma separated values* (.csv) yang berisi jumlah jenis WBC pada setiap citra.



Gambar 4.17. Diagram alir pengujian algoritma deteksi.

IV.2.6. Perhitungan Prediksi Objek Terdeteksi

Hasil proses pada tahap pengujian algoritma adalah berupa citra hasil prediksi (terdapat *bounding box* dan *object mask* pada sel terdeteksi) dan *file* .csv yang merupakan hasil perhitungan otomatis pada data uji. Kemudian hasil perhitungan otomatis akan dilakukan perbandingan dengan perhitungan aktualnya (berdasarkan pelabelan dari hematologis). Pada Gambar 4.18. merupakan diagram alir perbandingan antara perhitungan manual dengan hasil prediksi model yang telah dilatih. *Mean absolute error* (MAE) digunakan sebagai metrik evaluasi performa model pada data uji. Proses MAE dilakukan pada *Microsoft Excel* 2019.



Gambar 4.18. Diagram alir perbandingan perhitungan manual dengan hasil prediksi model.

IV.3. Analisis Hasil Penelitian

Pada pembahasan penelitian ini meliputi analisis performa model yang ditinjau dari *loss function* pada data latih dan data validasi berdasarkan *hyperparameter* saat proses pelatihan sebelumnya. Pengujian algoritma dilakukan pada data uji dengan menggunakan *hyperparameter* yang menghasilkan nilai mAP tertinggi pada data validasi dan melakukan evaluasi performa. Kemudian dilakukan perhitungan otomatis pada data uji dan melakukan perbandingan perhitungan secara manual dengan metrik MAE.

IV.4. Penulisan Laporan Penelitian

Penulisan laporan penelitian pada tugas akhir ini berdasarkan format penulisan yang telah ditetapkan oleh Departemen Teknik Nuklir dan Teknik Fisika, Fakultas Teknik, Universitas Gadjah Mada. Penulisan laporan yang berdasarkan format tersebut terdiri dari enam bab, yaitu pendahuluan, tinjauan pustaka, dasar teori, pelaksanaan penelitian, hasil dan pembahasan, serta kesimpulan dan saran.

BAB V

HASIL DAN PEMBAHASAN

V.1. Pelatihan Algoritma

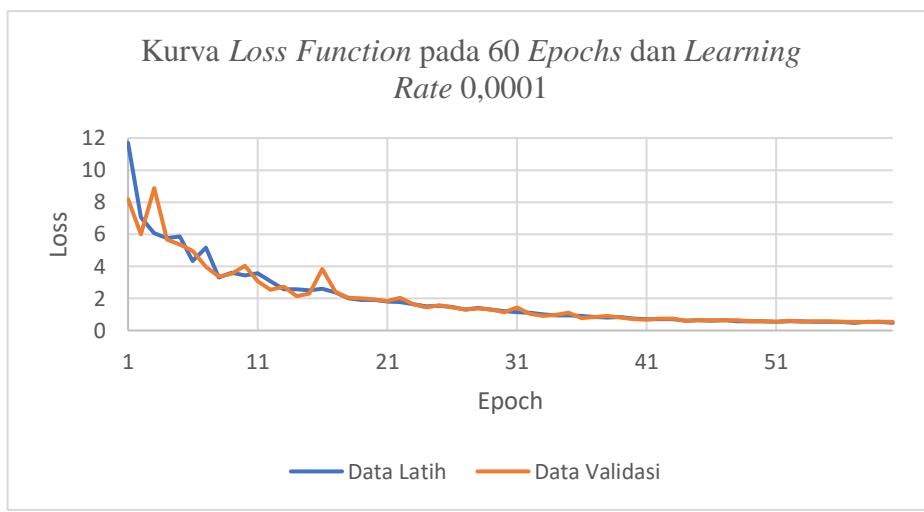
Proses pelatihan algoritma *Mask R-CNN* digunakan arsitektur *backbone* ResNet101. Pemilihan arsitektur tersebut didasarkan dengan jumlah *layers* terbanyak yang tersedia pada algoritma tersebut, sehingga dapat memperoleh nilai *loss* lebih baik dibandingkan jumlah *layers* yang lebih kecil. Alur kerja pelatihan algoritma dilakukan sesuai pada IV.2.4. Dalam proses pelatihan terjadi proses pembobotan dari konfigurasi yang telah ditentukan. Hasil proses pembobotan tersebut berupa *file* dengan *extension* .h5 yang berisi nilai bobot hasil pelatihan dan kemudian digunakan untuk melakukan prediksi pada data uji. Proses evaluasi hasil pelatihan atau pembobotan dapat dilihat dari kurva *loss function* pada data latih dan data validasi. Evaluasi kurva *loss function* tersebut akan menentukan apakah hasil pelatihan terjadi *overfitting* atau *underfitting*. Pelatihan algoritma pertama kali dilakukan dengan 60 *epochs* yang membutuhkan waktu maksimal 3 jam 23 menit. Pelatihan dengan 60 *epochs* tersebut bertujuan untuk menentukan nilai *learning rate*, *steps per epoch*, dan *validation steps* terbaik dengan memvariasikan nilai-nilainya. Pelatihan ini juga menentukan apakah distribusi data yang digunakan mengalami *overfitting* atau tidak. Nilai *hyperparameter* pada tahap pelatihan awal ini divariasikan seperti berikut:

1. *Learning rate* bernilai 0,01; 0,001; dan 0,0001.
2. *Steps per epoch* bernilai 75, 100, 125, dan 150.
3. *Validation steps* bernilai 150, 200, dan 250.
4. *Detection minimum confidence* bernilai 0,7; 0,75; 0,8; 0,85; dan 0,9.

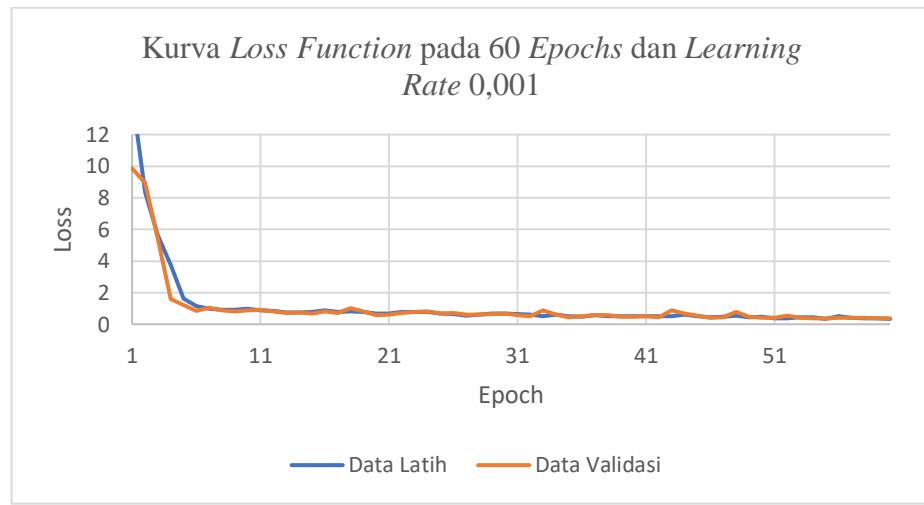
V.1.1. Penentuan Nilai *Learning Rate*

Nilai *learning rate* perlu divariasikan karena jika nilainya terlalu besar dapat terjadi ledakan pembobotan, sehingga kurva *loss function* tidak turun ke nilai minimum. Namun, jika nilai *learning rate* terlalu kecil akan membutuhkan waktu dan komputasi lebih banyak untuk mencapai nilai *loss* yang minimum. Hasil latih

dengan *learning rate* sebesar 0,0001 ditampilkan pada Gambar 5.1. dan untuk *learning rate* sebesar 0,001 ditampilkan pada Gambar 5.2. Dari kedua kurva *loss function* tersebut tidak terjadi *overfitting* yang mengartikan model cukup *generalize* dari distribusi data yang digunakan pada penelitian ini.



Gambar 5.1. Kurva *loss function* pada 60 epochs dengan *learning rate* sebesar 0,0001.



Gambar 5.2. Kurva *loss function* pada 60 epochs dengan *learning rate* sebesar 0,001.

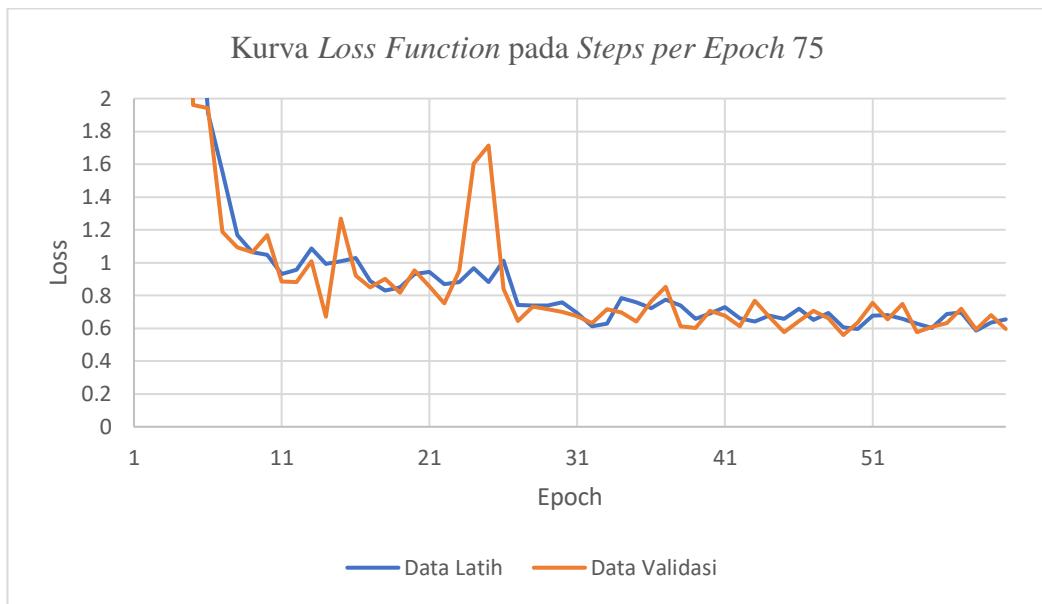
Kurva *loss* pada *learning rate* sebesar 0,01 tidak dapat ditampilkan karena terjadi ledakan pembobotan dan nilai *loss* sebagai besar berupa NaN (*Not a Number*). Oleh karena itu, penelitian ini cukup membandingkan hasil *loss* dengan *learning rate* sebesar 0,001 dan 0,0001. Pada *learning rate* sebesar 0,001 memberikan nilai *loss* akhir 0,3358 untuk data latih dan 0,3749 untuk data validasi. Namun, pada *learning rate* sebesar 0,0001 memberikan nilai *loss* akhir 0,4795 untuk data latih dan 0,5315 untuk data validasi. Pada kedua nilai *learning rate* tersebut dapat disimpulkan bahwa nilai 0,001 merupakan nilai yang optimal dan dapat menurunkan nilai *loss* lebih cepat dibandingkan 0,0001.

V.1.2. Penentuan Nilai *Steps per Epoch*

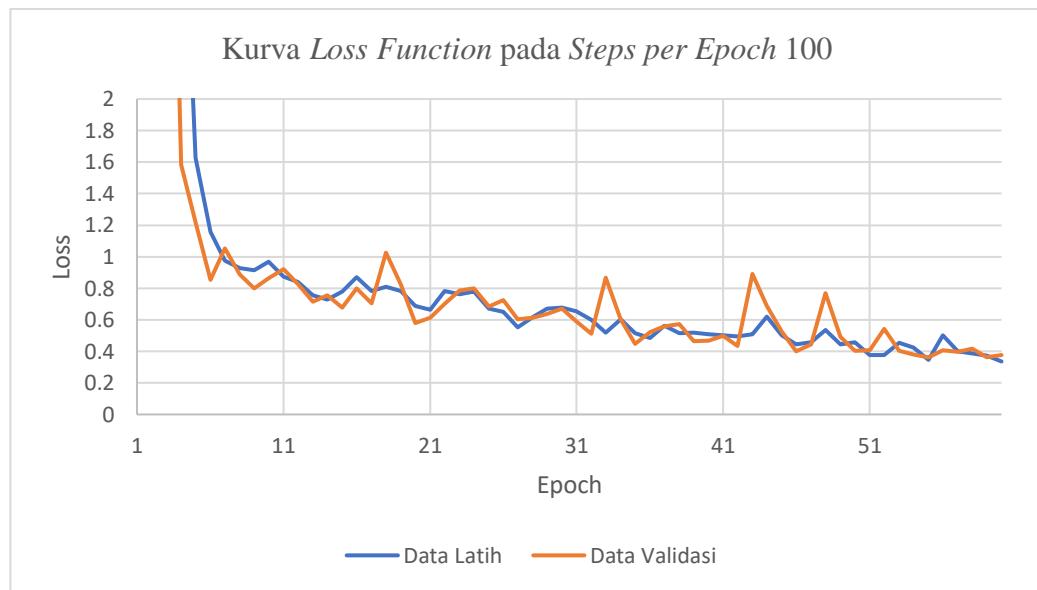
Hyperparameter selanjutnya yang ditentukan adalah *steps per epoch*. Pelatihan dengan variasi nilai *steps per epoch* dilakukan pada nilai *learning rate* 0,001. Hal yang menjadi pertimbangan dalam menentukan *steps per epoch* adalah nilai *loss* dan waktu pelatihannya. Kurva *loss function* hasil pelatihan dengan *steps per epoch* bernilai 75 ditampilkan pada Gambar 5.3, bernilai 100 ditampilkan pada Gambar 5.4., bernilai 125 ditampilkan pada Gambar 5.5., dan bernilai 150 ditampilkan pada Gambar 5.6. Kemudian, nilai *loss* akhir yang diperoleh dan waktu pelatihannya ditampilkan pada Tabel 5.1. Dari hasil variasi nilai *steps per epoch* menunjukkan semakin tinggi nilainya semakin baik nilai *loss* yang diperoleh, tetapi membutuhkan waktu lebih lama untuk pelatihan model. Namun, nilai *steps per epoch* yang lebih tinggi tidak menurunkan nilai *loss* secara signifikan. Oleh karena itu, nilai *steps per epoch* yang dipilih adalah bernilai 100.

Tabel 5.1. Hasil pelatihan dengan variasi nilai *steps per epoch*.

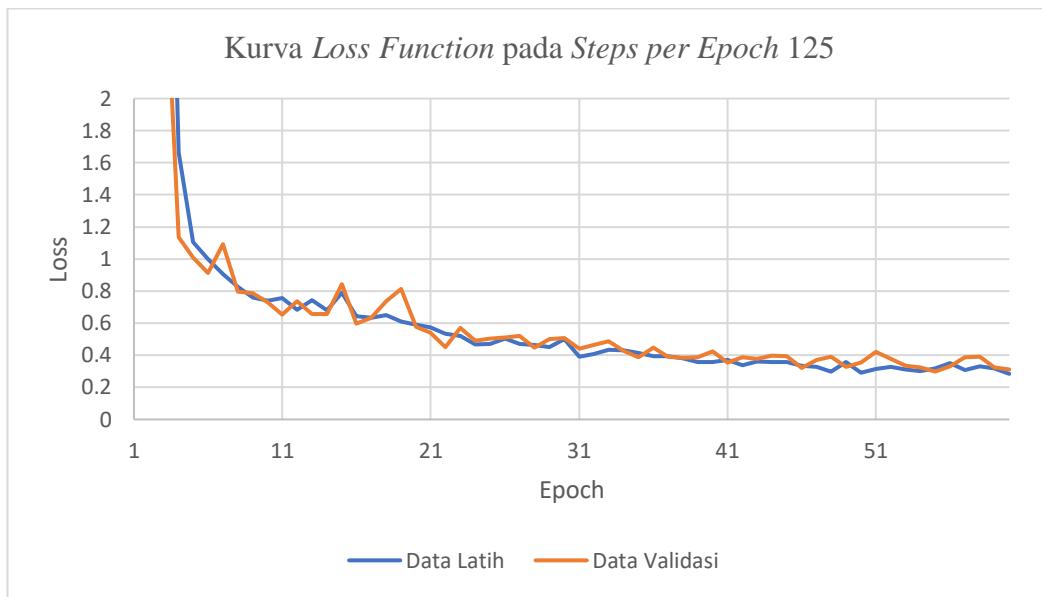
<i>Steps per epoch</i>	<i>Loss</i> data latih akhir	<i>Loss</i> data validasi akhir	Waktu pelatihan
75	0,6533	0,5946	2 jam 24 menit
100	0,3358	0,3749	2 jam 43 menit
125	0,2842	0,3110	2 jam 53 menit
150	0,3002	0,3387	3 jam 23 menit



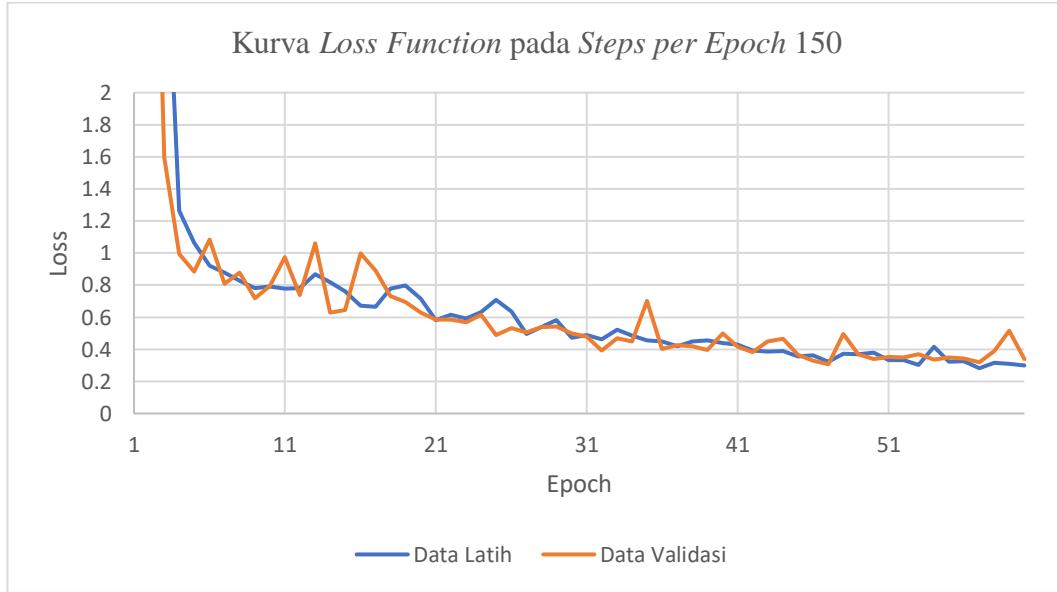
Gambar 5.3. Kurva *loss function* pada *steps per epoch* 75.



Gambar 5.4. Kurva *loss function* pada *steps per epoch* 100.



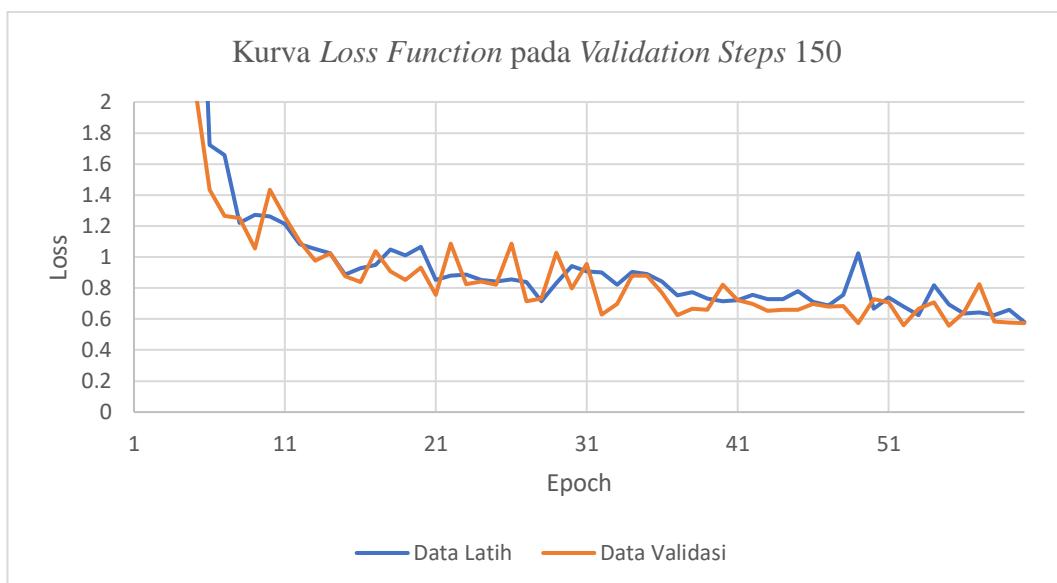
Gambar 5.5. Kurva *loss function* pada *steps per epoch* 125.



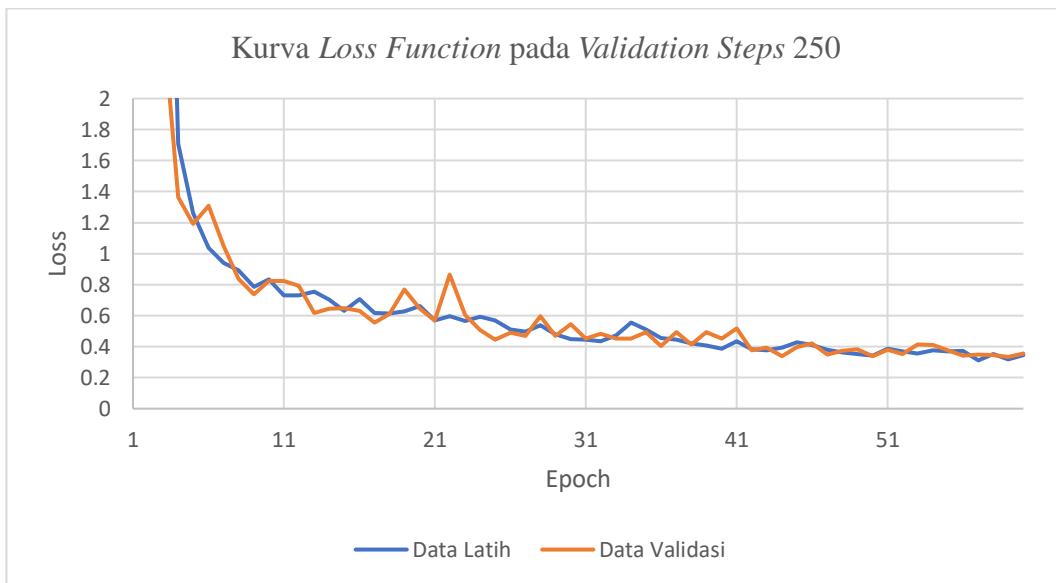
Gambar 5.6. Kurva *loss function* pada *steps per epoch* 150.

V.1.3. Penentuan Nilai *Validation Steps*

Nilai *validation steps* divariasikan pada *learning rate* sebesar 0,001 dan *steps per epoch* sebesar 100. Kurva *loss function* untuk *validation steps* sebesar 150 ditampilkan pada Gambar 5.7. dan sebesar 250 ditampilkan pada Gambar 5.8. Nilai *validation steps* 200 ditampilkan pada Gambar 5.4. Penentuan nilai *validation steps* ditentukan berdasarkan nilai *loss* dan waktu pelatihannya, seperti pada penentuan nilai *steps per epoch* sebelumnya. Variasi nilai *validation steps* memberikan dampak yang sama dengan variasi *steps per epoch*. Nilai *validation steps* yang lebih besar dapat menurunkan nilai *loss*, tetapi jika semakin besar nilainya maka tidak ada perubahan secara signifikan dan membutuhkan waktu yang lebih lama. Nilai *loss* akhir dan waktu pelatihan yang diperoleh pada setiap variasinya ditunjukkan pada Tabel 5.2. Oleh karena itu, nilai *validation steps* yang digunakan pada penelitian ini adalah 200.



Gambar 5.7. Kurva *loss function* pada *validation steps* 150.



Gambar 5.8. Kurva *loss function* pada *validation steps* 250.

Tabel 5.2. Hasil pelatihan dengan variasi nilai *validation steps*.

<i>Validation steps</i>	<i>Loss</i> data latih akhir	<i>Loss</i> data validasi akhir	Waktu pelatihan
150	0,5805	0,5737	2 jam 12 menit
200	0,3358	0,3749	2 jam 43 menit
250	0,3469	0,3539	3 jam 3 menit

Hasil pelatihan algoritma dengan jumlah *epoch* 60 dapat menentukan nilai *hyperparameter* yang optimal. *Hyperparameter* tersebut berupa *learning rate* sebesar 0,001, *steps per epoch* sebesar 100, dan *validation steps* sebesar 200. Nilai *hyperparameter* yang telah diperoleh tersebut akan digunakan pada pelatihan berikutnya dengan jumlah *epoch* yang lebih besar. Pelatihan algoritma tersebut juga membuktikan bahwa tidak terjadinya *overfitting* pada distribusi data yang digunakan, sehingga distribusi data tersebut dapat digunakan untuk tahap pelatihan selanjutnya.

V.1.4. Penentuan *Detection Minimum Confidence*

Penentuan jumlah *epoch* dilakukan dengan melihat kecenderungan konvergensi pada data latih dan data validasi. Pelatihan dengan jumlah *epoch* 60 menunjukkan bahwa model memiliki kecenderungan untuk menurunkan nilai *loss* lebih lanjut jika jumlah *epoch* ditingkatkan. Oleh karena itu, pada tahap pelatihan berikutnya jumlah *epoch* ditingkatkan menjadi 220 *epochs*. Jumlah *epoch* hanya dibatasi sampai dengan 220 *epochs* karena *epoch* di atas 175 tidak menurunkan nilai *loss* secara signifikan, seperti ditunjukkan pada Gambar 5.9 dan Gambar 5.10.

Penentuan nilai *detection minimum confidence* dilakukan dengan *hyperparameter* yang telah ditentukan pada V.1.1., V.1.2., dan V.1.3. Nilai *detection minimum confidence* divariasikan dengan nilai 0,7; 0,75; 0,80; 0,85; dan 0,90 untuk citra mentah dan citra disegmentasi. Seluruh pelatihan algoritma dilakukan dengan jumlah *epoch* 220. Hasil pelatihan dievaluasi berdasarkan mAP pada data validasi dengan nilai IoU sebesar 0,5 dan *detection minimum confidence* saat *inference* sebesar 0,4. *Detection minimum confidence* merupakan probabilitas minimum untuk menerima objek yang terdeteksi. Nilai tersebut berpengaruh saat melakukan pendekslan objek karena menentukan apakah objek akan terdeteksi atau tidak. Oleh karena itu, *detection minimum confidence* saat pelatihan perlu ditentukan secara optimal. Nilai *detection minimum confidence* yang ditentukan akan disesuaikan dengan *dataset* digunakan, yaitu untuk citra mentah dan citra disegmentasi.

Tabel 5.3. Performa hasil latih pada citra mentah.

<i>Detection minimum confidence</i>	mAP
0,70	0,9188
0,75	0,8060
0,80	0,8982
0,85	0,8974
0,90	0,8399

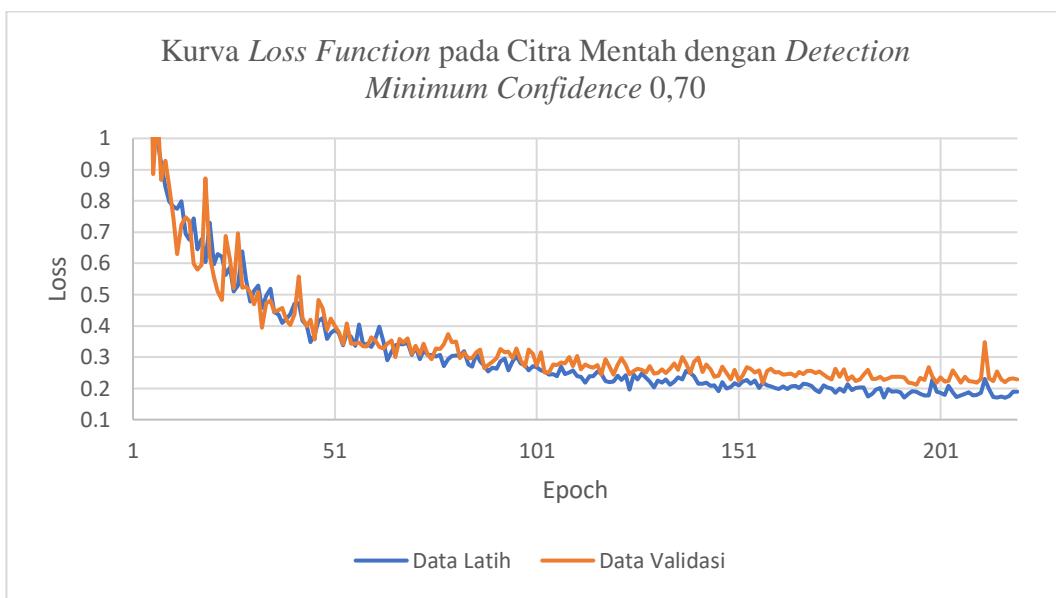
Tabel 5.4. Performa hasil latih pada citra disegmentasi.

<i>Detection minimum confidence</i>	mAP
0,70	0,8818
0,75	0,8171
0,80	0,8070
0,85	0,9225
0,90	0,8512

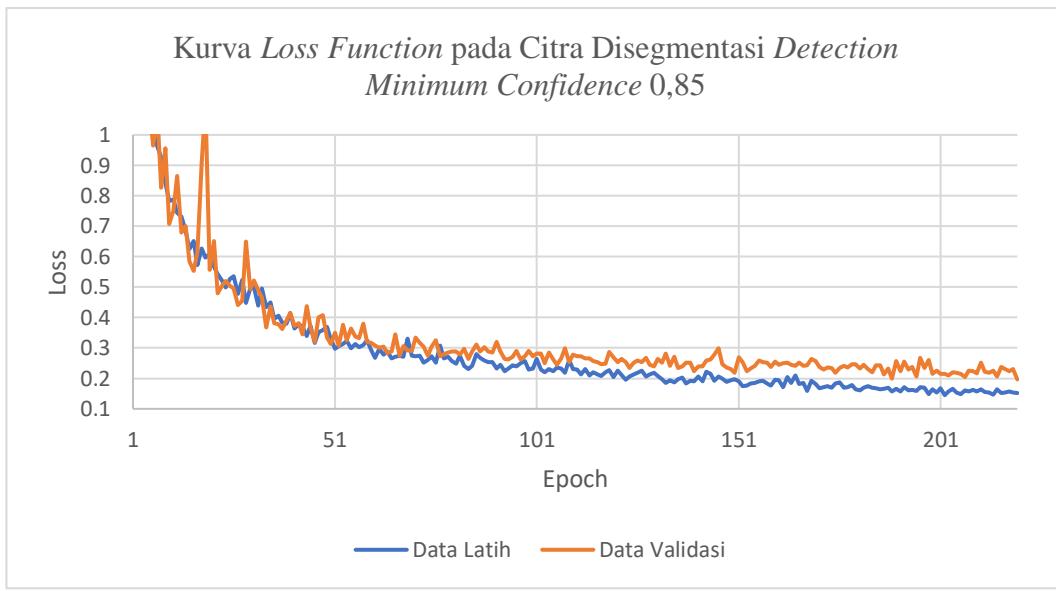
Berdasarkan hasil pelatihan pada Tabel 5.3. dan Tabel 5.4. untuk citra mentah dan citra disegmentasi dapat ditentukan nilai *detection minimum confidence* terbaik untuk masing-masing *dataset*. Hal tersebut dilakukan untuk meningkatkan peluang model terbaik saat melakukan pendekripsi pada data uji. Oleh karena itu, nilai *detection minimum confidence* terbaik untuk citra mentah adalah 0,70 dan untuk citra disegmentasi adalah 0,85. Dari hasil pelatihan ini menunjukkan bahwa nilai *detection minimum confidence* pada citra disegmentasi lebih unggul dibandingkan dengan citra mentah. Nilai *loss* akhir ditampilkan pada Tabel 5.5., *dataset* dengan citra disegmentasi memberikan nilai *loss* sedikit lebih baik dibandingkan citra mentah. Untuk waktu pelatihan, citra disegmentasi membutuhkan waktu lebih sedikit dibandingkan citra mentah, hal ini membuktikan metode-metode pengolahan citra yang digunakan pada penelitian ini dapat meningkatkan performa model lebih baik.

Tabel 5.5. Nilai *loss* akhir pada pelatihan.

<i>Dataset</i>	<i>Detection minimum confidence</i>	<i>Loss</i> data latih akhir	<i>Loss</i> data validasi akhir	Waktu pelatihan
Citra Mentah	0,70	0,1889	0,2286	11 jam 30 menit
Citra Disegmentasi	0,85	0,1510	0,1965	10 jam 7 menit



Gambar 5.9. Kurva *loss function* pada citra mentah dengan nilai *detection minimum confidence* 0,70.



Gambar 5.10. Kurva *loss function* pada citra disegmentasi dengan nilai *detection minimum confidence* 0,85.

V.1.5. Evaluasi Performa *Hyperparameter* Terpilih

Hyperparameter terbaik untuk kedua *dataset* diukur performanya pada data validasi. Rangkuman *hyperparameter* terbaik yang digunakan dalam pelatihan ditampilkan pada Tabel 5.6. Metrik evaluasi yang digunakan berupa presisi, *recall*, akurasi, *F1-Score*, dan mAP. Hasil perhitungan metrik evaluasi pada data validasi secara keseluruhan ditampilkan pada Gambar 5.11.

Tabel 5.6. Rangkuman *hyperparameter* terpilih.

<i>Dataset</i>	<i>Learning Rate</i>	<i>Steps per Epoch</i>	<i>Validation Steps</i>	<i>Detection Minimum Confidence</i>
Citra Mentah	0,001	100	200	0,70
Citra Disegmentasi	0,001	100	200	0,85

Tabel 5.7. Hasil deteksi pada data validasi.

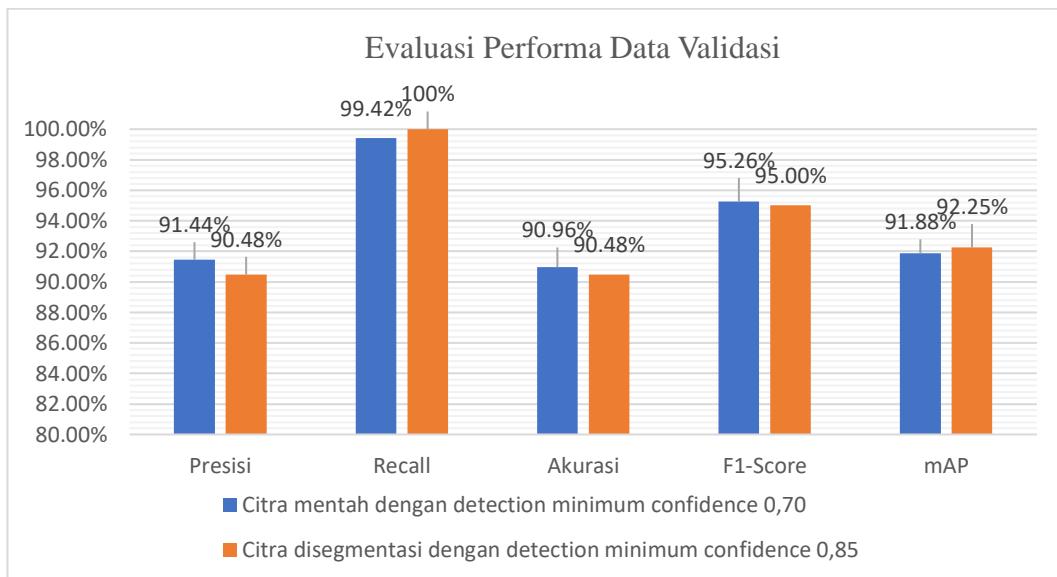
<i>Dataset</i>	TP	FP	FN
Citra Mentah	171	16	1
Citra Disegmentasi	171	18	0

Tabel 5.8. Hasil performa pada data validasi.

<i>Dataset</i>	Presisi	<i>Recall</i>	Akurasi	<i>F1-Score</i>	mAP
Citra Mentah	91,44%	99,42%	90,96%	95,26%	91,88%
Citra Disegmentasi	90,48%	100%	90,48%	95,00%	92,25%

Performa *hyperparameter* yang telah dipilih memberikan nilai *recall* yang hampir 100% untuk citra mentah dan 100% untuk citra disegmentasi. Hal tersebut menunjukkan hampir semua WBC dapat terdeteksi, tetapi dalam klasifikasi belum tentu menunjukkan kelas yang benar. Nilai presisi untuk citra mentah dan citra disegmentasi menunjukkan nilai yang hampir sama. Nilai presisi tersebut menunjukkan model dapat mendeteksi WBC dengan kelas yang sama dengan *ground truth* sebesar 91,44% untuk citra mentah dan 90,48% untuk citra disegmentasi. Untuk nilai akurasi, *F1-Score*, dan mAP masing-masing hampir

menunjukkan nilai yang sama, walaupun dilatih pada nilai *detection minimum confidence* yang berbeda.



Gambar 5.11. Evaluasi performa data validasi dengan *hyperparameter* terbaik.

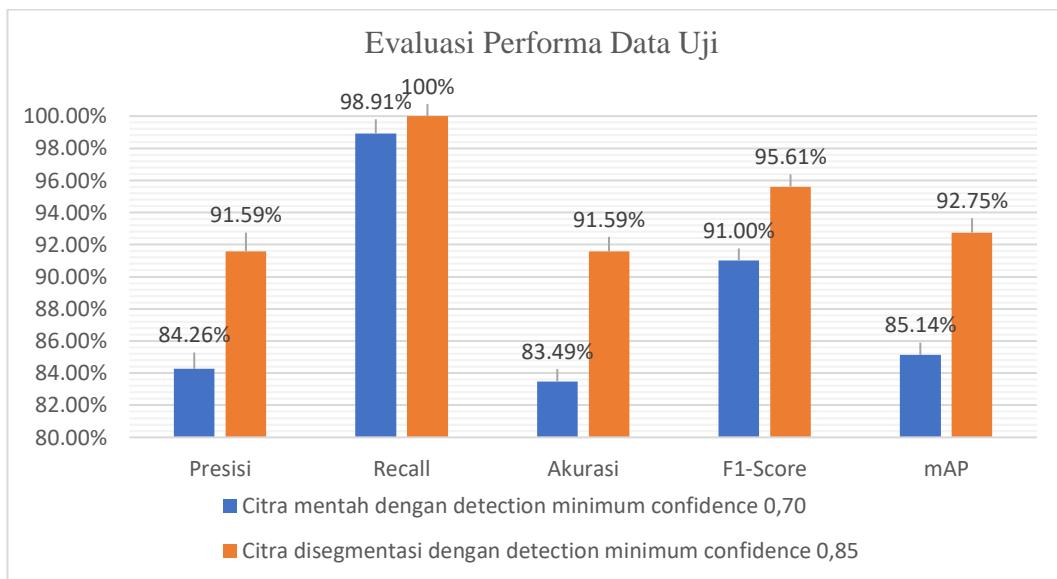
Evaluasi performa pada data validasi memberikan nilai-nilai metrik yang cukup baik. Hal tersebut dapat dicapai dengan baik karena penentuan *hyperparameter* dilakukan pada data latih dan data validasi, sehingga akan memberikan performa yang baik pula pada data validasi. Oleh karena itu, evaluasi performa selanjutnya digunakan pada data uji yang hasil evaluasi secara keseluruhannya ditampilkan pada Gambar 5.12. Data uji merupakan data yang tidak pernah dilihat oleh model sebelumnya, sehingga evaluasi performa pada data uji merupakan performa yang paling dekat pada keadaan sesungguhnya.

Tabel 5.9. Hasil deteksi pada data uji.

Dataset	TP	FP	FN
Citra Mentah	91	17	1
Citra Disegmentasi	98	9	0

Tabel 5.10. Rangkuman performa pada data uji.

Dataset	Presisi	Recall	Akurasi	F1-Score	mAP
Citra Mentah	84,26%	98,91%	83,49%	91,00%	85,14%
Citra Disegmentasi	91,59%	100%	91,59%	95,61%	92,75%



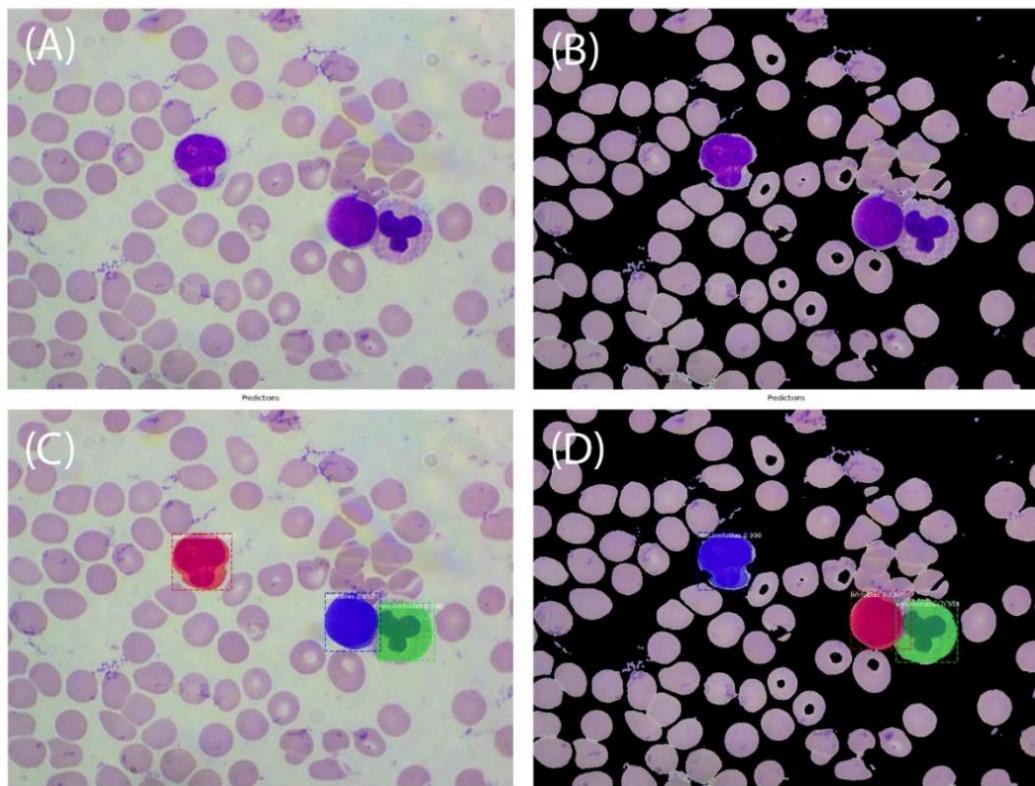
Gambar 5.12. Evaluasi performa data uji dengan *hyperparameter* terbaik.

Hasil metrik evaluasi pada data uji memberikan performa yang cukup berbeda untuk kedua *dataset* tidak seperti pada data validasi sebelumnya. Presisi pada citra mentah bernilai 84,26%, sedangkan pada citra disegmentasi bernilai 91,59%. Hal tersebut menunjukkan hanya 84,26% dari WBC yang terdeteksi untuk citra mentah merupakan kelas yang benar atau sesuai dengan *ground truth*. Namun, pada citra disegmentasi memberikan nilai 91,59% yang hampir sama performanya pada data validasi. Nilai *recall* untuk kedua *dataset* memberikan nilai yang hampir sama dengan data validasi. Nilai *recall* yang hampir 100% untuk citra mentah dan 100% pada citra disegmentasi mengakibatkan nilai akurasi hampir sama dengan nilai presisi. Berdasarkan nilai presisi, *recall*, akurasi, *F1-Score*, dan mAP *dataset* dengan citra disegmentasi memberikan performa yang hampir sama dengan data validasi. Namun, citra mentah memberikan performa lebih turun dibandingkan data validasi. Oleh karena itu, berdasarkan hasil performa data uji dan data validasi, citra

yang disegmentasi dapat menghasilkan performa lebih *robust* dan membutuhkan waktu lebih sedikit dalam pelatihan dibandingkan citra tanpa adanya segmentasi.

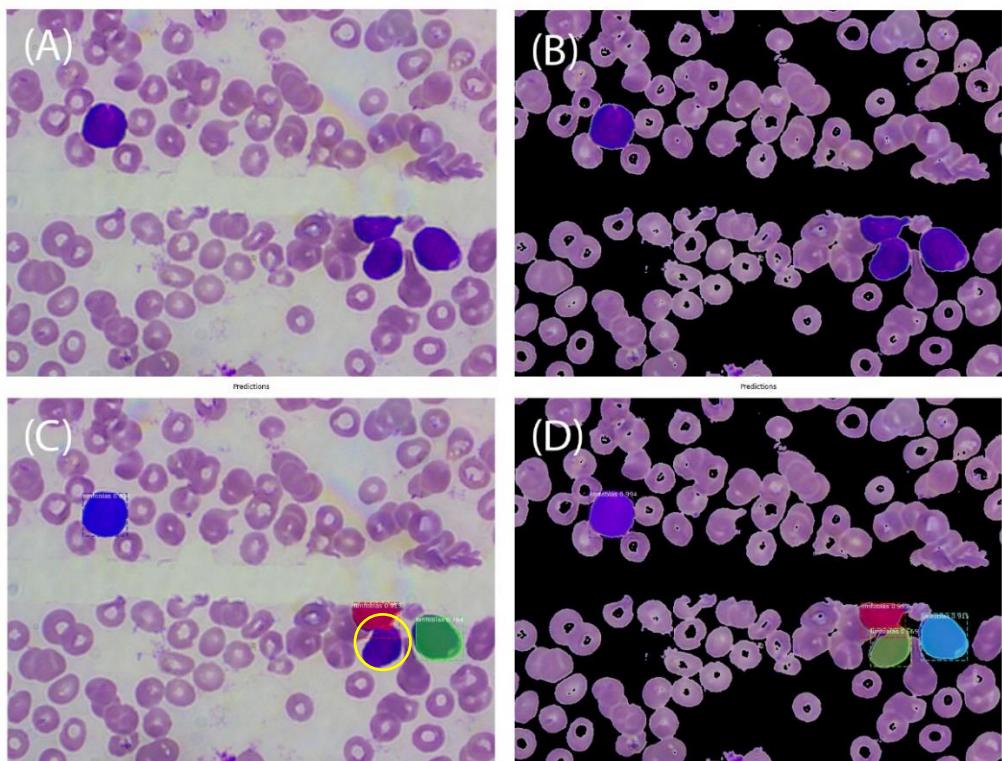
V.1.6. Evaluasi Citra Hasil Prediksi

Hasil proses pendekstrian pada data uji berupa citra yang telah diberi *bounding box*, *mask*, dan *confidence score* pada WBC yang terdeteksi. Berikut adalah beberapa contoh citra hasil prediksi untuk citra mentah dan citra disegmentasi.



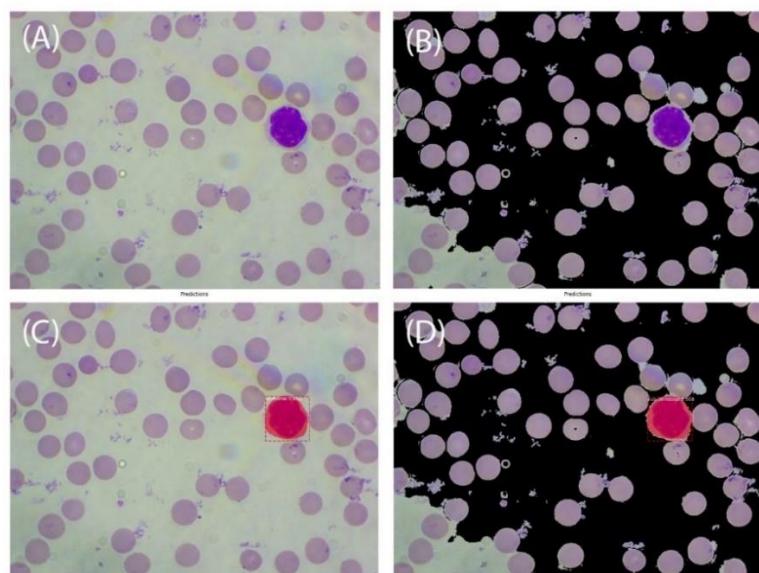
Gambar 5.13. Perbedaan hasil *confidence score* pada citra mentah dan citra disegmentasi. (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: non-limfoblas 88,7%, biru: limfoblas 65,2%, hijau: non-limfoblas 99,8%), (D) Citra disegmentasi setelah prediksi (biru: non-limfoblas 99,8%, merah: limfoblas 73,7%, hijau: non-limfoblas 99,8%).

Citra hasil prediksi pada Gambar 5.13. untuk citra mentah dan citra disegmentasi dapat mendeteksi WBC secara benar dalam hal kelas, *bounding box*, dan *mask*. Namun, hal yang membedakan untuk kedua *dataset* tersebut adalah *confidence score* yang dihasilkan. Dapat dilihat nilai *confidence score* untuk citra mentah yang dihasilkan adalah 0,887; 0,652; dan 0,998, sedangkan untuk citra disegmentasi adalah 0,998; 0,737; dan 0,998. Untuk mendeteksi kelas WBC yang sama citra disegmentasi memiliki *confidence score* yang lebih unggul dibandingkan citra mentah. Hal tersebut terjadi karena model prediksi untuk citra disegmentasi dilatih pada *detection minimum confidence* yang lebih tinggi dibandingkan citra mentah.

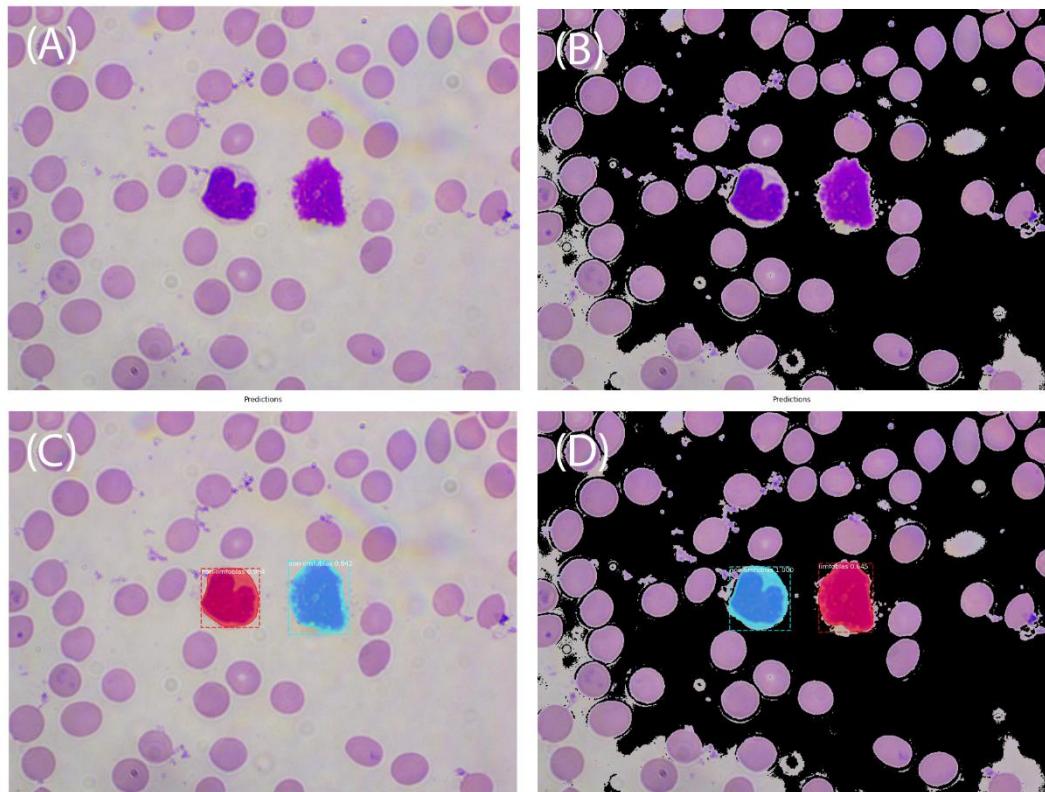


Gambar 5.14. Hasil prediksi citra mentah terdapat *false negative* (lingkaran kuning bagian C). (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (biru: limfoblas 81,7%, merah: limfoblas 91,3%, hijau: limfoblas 76,4%), (D) Citra disegmentasi setelah prediksi (ungu: limfoblas 99,4%, merah: limfoblas 99,5%, kuning: limfoblas 65,9%, biru muda: limfoblas 91,5%).

Hasil prediksi data uji pada citra mentah terdapat satu *false negative* yang ditampilkan pada Gambar 5.14 bagian (C) (lingkaran kuning). Pada citra tersebut terdapat empat sel limfoblas sebagai *ground truth*, tetapi dua dari sel tersebut berimpit. Model yang dilatih pada citra mentah hanya dapat mendekripsi tiga sel limfoblas dengan *confidence score* masing-masing 0,817; 0,913; dan 0,764. Namun, untuk model yang dilatih pada citra disegmentasi dapat mendekripsi empat sel limfoblas dengan lebih baik, *confidence score* masing-masing deteksi adalah 0,994; 0,995; 0,659; dan 0,915. Sel limfoblas yang tidak dapat didekripsi pada citra mentah, tetapi dapat didekripsi pada citra disegmentasi dengan *confidence score* sebesar 0,659. Hal tersebut dapat terjadi karena *confidence score* sel limfoblas pada citra mentah di bawah nilai 0,4 atau nilai *threshold* NMS terlalu kecil. Jika *detection minimum confidence* saat *inference* diturunkan kurang dari 0,4 untuk citra mentah maka ada kemungkinan munculnya *false positive* yang lebih banyak. Namun, jika nilai *threshold* NMS dinaikkan maka ada kemungkinan muncul *bounding box* yang lebih banyak pada objek lain.



Gambar 5.15. Hasil prediksi citra mentah berupa *false positive*. (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: limfoblas 60,2%), (D) Citra disegmentasi setelah prediksi (merah: non-limfoblas 96,8%).



Gambar 5.16. Hasil prediksi citra mentah dan disegmentasi memberikan *false positive* karena terdapat objek menyerupai WBC. (A) Citra mentah sebelum prediksi, (B) Citra disegmentasi sebelum prediksi, (C) Citra mentah setelah prediksi (merah: non-limfoblas 95,4%, biru muda: non-limfoblas 94,2%), (D) Citra disegmentasi setelah prediksi (biru muda: non-limfoblas 100%, merah: limfoblas 64,5%).

Berdasarkan hasil deteksi pada data uji, seperti ditampilkan pada Tabel 5.9. model yang dilatih pada citra mentah memberikan nilai *false positive* lebih tinggi dibandingkan model yang dilatih pada citra disegmentasi. Pada Gambar 5.15. merupakan salah satu contoh hasil prediksi citra mentah yang memberikan *false positive*, sedangkan pada citra disegmentasi memberikan *true positive*. Label *ground truth* pada sel WBC tersebut adalah sel non-limfoblas. Kedua hasil prediksi pada Gambar 5.15. bagian (C) dan (D) memberikan bentuk *mask* serupa yang menandakan bahwa sitoplasma ikut terdeteksi sebagai WBC. Kemudian, nilai *confidence score* pada citra mentah bernilai 0,602, sedangkan pada citra

disegmentasi memberikan nilai 0,968. Hal yang membuat klasifikasi kelas salah pada citra mentah belum dapat ditentukan dengan pasti penyebabnya. Namun, dapat dipastikan bahwa dengan adanya segmentasi citra dalam pelatihan dapat memberikan klasifikasi kelas yang benar dan *confidence score* yang lebih tinggi. Namun, pada Gambar 5.16. untuk kedua citra memberikan *false positive*. Hal tersebut terjadi karena terdapat objek yang menyerupai WBC. Nilai *confidence score* untuk objek menyerupai WBC tersebut lebih rendah pada citra disegmentasi dibandingkan citra mentah, yaitu bernilai 0,645 dan 0,942.

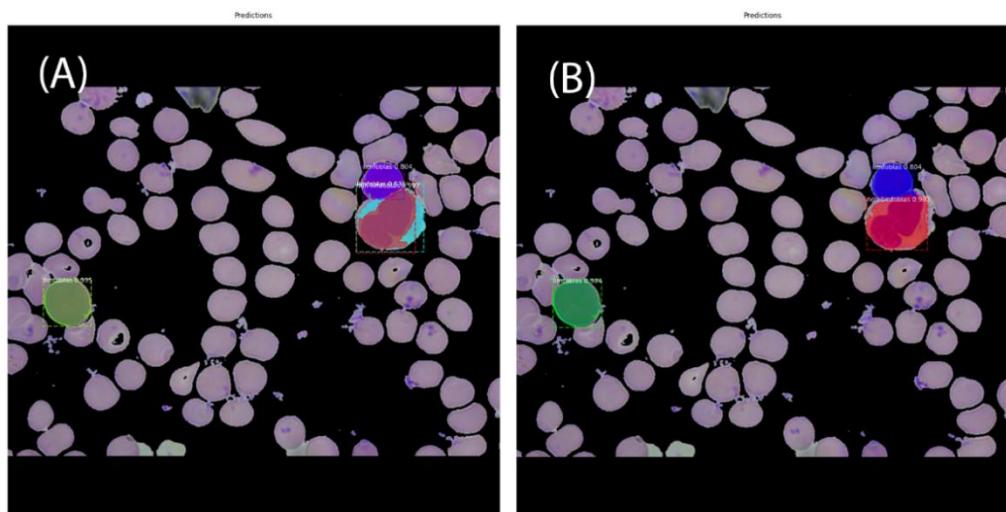
V.2. Pengujian Algoritma Deteksi

Setelah model dilatih, model dapat dijalankan pada data langsung dan memberikan hasil secara *real time* yang disebut sebagai *inference*. Pada penelitian ini *inference* dilakukan menggunakan data uji. Pada tahap pelatihan model sebelumnya, nilai *hyperparameter* digunakan *detection NMS threshold* sebesar 0,3 dan *RPN NMS threshold* sebesar 0,7 yang merupakan nilai *default* dari *Mask R-CNN*. Nilai *RPN NMS threshold* memiliki nilai yang lebih tinggi bertujuan untuk menghasilkan lebih banyak proposal saat pelatihan. Namun, nilai *detection NMS threshold* yang lebih rendah bertujuan agar tidak banyak *bounding box* yang dihasilkan untuk objek yang sama.

Hasil prediksi citra pada V.1.6. dilakukan dengan *detection NMS threshold* dan *RPN NMS threshold* masing-masing bernilai 0,2. Hal yang menjadi dasar pemilihan nilai *threshold* rendah saat *inference* adalah untuk memperoleh satu *bounding box* terbaik untuk satu objek. Berdasarkan citra hasil prediksi pada V.1.6. tidak terdapat *bounding box* lebih dari satu untuk setiap objek. Hal tersebut menunjukkan nilai *threshold* yang digunakan telah mendekati nilai optimal atau telah optimal. Namun, nilai *threshold* yang terlalu rendah dapat memberi dampak *false negative*. Hal tersebut dapat terjadi karena objek berdekatan yang seharusnya terdeteksi sebagai dua objek atau lebih, tetapi hanya terdeteksi satu objek karena nilai *threshold* yang terlalu kecil kemudian melakukan *suppression* untuk *bounding box* objek yang berdekatan. Oleh karena itu, diperlukan nilai *threshold* yang optimal pada *detection NMS threshold* dan *RPN NMS threshold* saat *inference*.

V.2.1. Penentuan *Detection NMS Threshold* dan *RPN NMS Threshold*

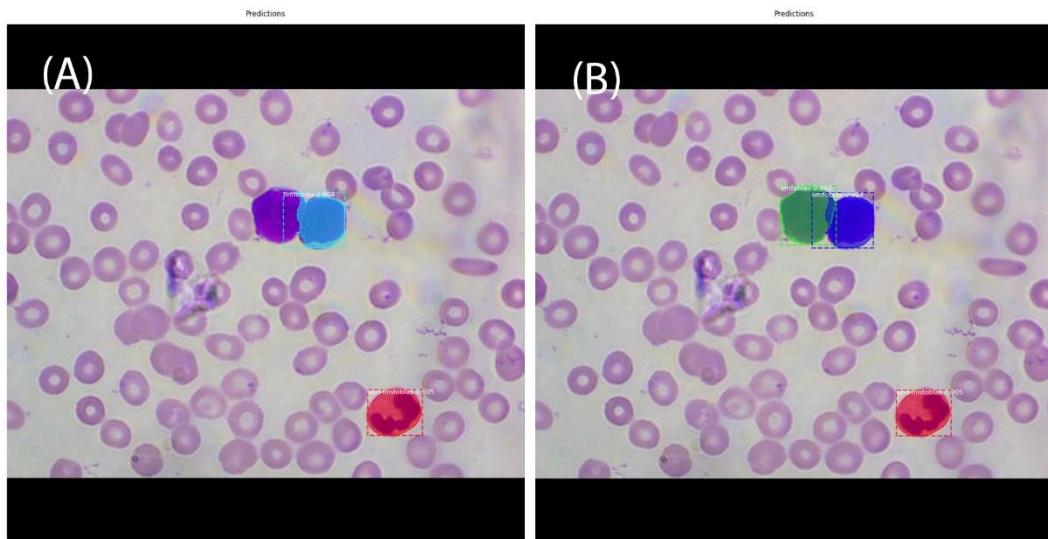
Penentuan nilai *detection NMS threshold* dan *RPN NMS threshold* dilakukan dengan cara melihat langsung hasil prediksi pada data validasi. Citra hasil prediksi tersebut akan terlihat apakah tepat lebih dari satu *bounding box* untuk setiap WBC atau tidak. Pada Gambar 5.17. merupakan contoh hasil prediksi pada data validasi dan proses penentuan *threshold* dilakukan berdasarkan citra hasil deteksi tersebut.



Gambar 5.17. Nilai *threshold* menentukan jumlah *bounding box* yang dihasilkan untuk setiap objek. (A) RPN NMS bernilai 0,7 dan *detection NMS* bernilai 0,2 (warna sel terdeteksi tumpang-tindih), (B) RPN NMS bernilai 0,2 dan *detection NMS* bernilai 0,2 (hijau: limfoblas 98,4%, biru: limfoblas 80,4%, merah: non-limfoblas 99,2%).

Hasil deteksi pada Gambar 5.17. memiliki pengaruh signifikan dalam perhitungan WBC. Nilai *threshold* yang terlalu besar dapat mendeteksi satu WBC sebagai lebih dari satu, seperti pada Gambar 5.17. bagian (A). Pada *dataset* citra disegmentasi dipilih nilai *threshold* untuk RPN NMS dan *detection NMS* sebesar 0,2. Hasil deteksi pada *hyperparameter* tersebut memberikan hasil seperti pada bagian V.1.5. Nilai *threshold* tersebut dipilih karena dapat mendeteksi setiap WBC

sebagai satu *bounding box* yang terbukti memiliki nilai *recall* 100% atau tidak terdapatnya *false negative*.



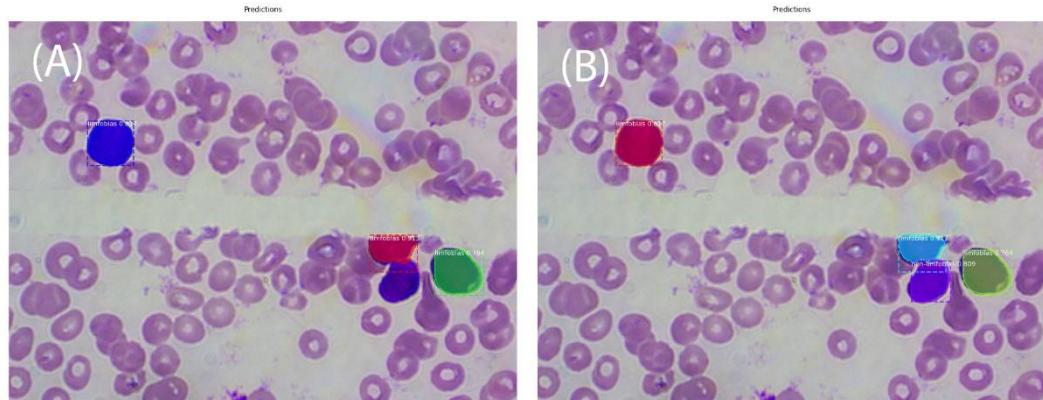
Gambar 5.18. Nilai *threshold* yang terlalu kecil menghilangkan *bounding box* untuk objek yang berdempetan. (A) RPN NMS bernilai 0,2 dan *detection* NMS bernilai 0,2 (biru muda: limfoblas 80,4%, merah: non-limfoblas 90,5%), (B) RPN NMS bernilai 0,25 dan *detection* NMS bernilai 0,25 (hijau: limfoblas 98,5%, ungu: limfoblas 80,4%, merah: non-limfoblas 90,5%).

Citra hasil prediksi pada V.1.5. untuk citra mentah memberikan satu *false negative* yang mengartikan terdapat satu objek yang seharusnya terdeteksi, tetapi terabaikan. Citra yang terdapat objek tidak terdeteksi tersebut ditampilkan pada Gambar 5.18. bagian (A). Berdasarkan citra hasil prediksi tersebut objek sangat berdempetan dengan objek sebelahnya, sehingga nilai *threshold* yang kecil dapat melakukan *suppression* pada objek berdekatan. Oleh karena itu, nilai *threshold* yang digunakan saat *inference* ditingkatkan menjadi 0,25 untuk *detection* NMS dan RPN NMS. Hasil peningkatan *threshold* tersebut membuat objek yang tidak terdeteksi menjadi terdeteksi, seperti ditampilkan pada Gambar 5.18. bagian (B).

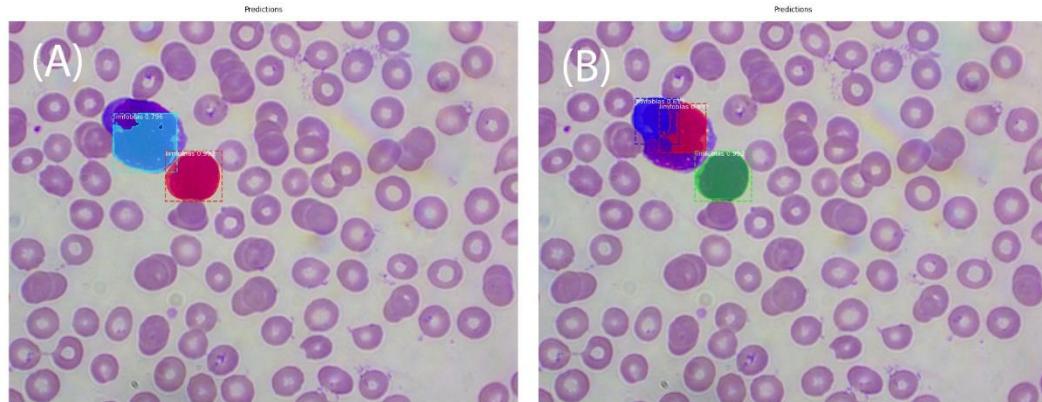
V.2.2. Evaluasi Performa Alogritma Deteksi

Evaluasi performa algoritma pada subbab ini hanya dilakukan pada *dataset* citra mentah karena performa algoritma pada *dataset* citra disegmentasi telah memberikan performa terbaik. Evaluasi performa dilakukan pada data validasi dan

data uji. Nilai RPN NMS dan *detection* NMS yang digunakan sebesar 0,25 dapat menghasilkan nilai *recall* 100% yang mengartikan bahwa semua sel dapat terdeteksi. Hal tersebut membuat model memiliki nilai presisi dan akurasi yang sama karena tidak terdapat *false negative*. Namun, kenaikan nilai *recall* dapat menurunkan nilai presisi atau jumlah *false positive* yang meningkat. Pada Gambar 5.19. merupakan hasil prediksi pada data uji. Nilai *threshold* sebesar 0,2 tidak dapat mendeteksi sel limfoblas yang berimpitan, tetapi dengan nilai *threshold* sebesar 0,25 sel limfoblas tersebut dapat terdeteksi dengan kelas yang salah (*false positive*). Peningkatan nilai *threshold* dapat meningkatkan nilai *recall* pada kasus ini, tetapi dapat menurunkan nilai presisi. Pada Gambar 5.20. sel limfoblas berukuran besar terdapat dua *bounding box*. Hal tersebut terjadi karena nilai *threshold* terlalu besar untuk objek tersebut, sehingga terdapat lebih dari satu *bounding box* yang dihasilkan.



Gambar 5.19. Hasil prediksi citra mentah pada data uji. (A) RPN NMS bernali 0,2 dan *detection* NMS bernali 0,2 terdapat *false negative* (biru: limfoblas 81,7%, merah: limfoblas 91,3%, hijau: limfoblas 76,4%), (B) RPN NMS bernali 0,25 dan *detection* NMS bernali 0,25 terdapat *false positive* (merah: limfoblas 81,7%, biru muda: limfoblas 91,3%, ungu: non-limfoblas 80,9%, kuning: limfoblas 76,4%).



Gambar 5.20. Hasil prediksi citra mentah pada data uji. (A) RPN NMS bernilai 0,2 dan *detection* NMS bernilai 0,2 memberikan satu *bounding box* (*mask* berwarna biru muda) pada sel di atas (biru muda: limfoblas 79,6%, merah: limfoblas 99,2%), (B) RPN NMS bernilai 0,25 dan *detection* NMS bernilai 0,25 memberikan dua *bounding box* (*mask* berwarna biru dan merah) pada sel di atas (biru: limfoblas 61,9%, merah: limfoblas 89,2%, hijau: limfoblas 99,2%).

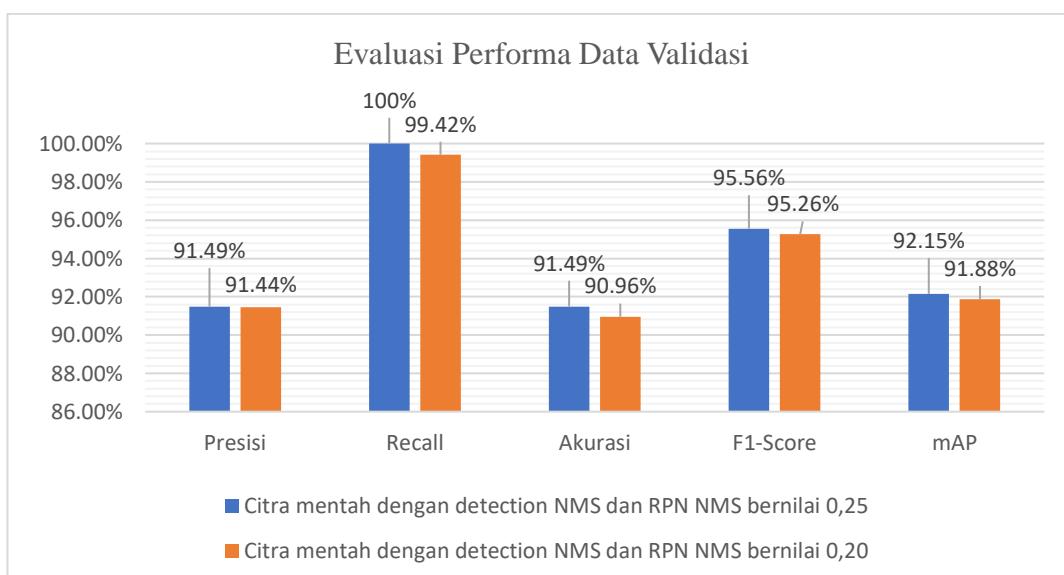
Berdasarkan hasil evaluasi performa data validasi pada Gambar 5.21. nilai *threshold* sebesar 0,25 memberikan performa sedikit lebih baik dibandingkan dengan *threshold* 0,20. Namun, performa pada data uji yang ditampilkan pada Gambar 5.22. nilai *threshold* 0,20 sedikit lebih unggul dibandingkan *threshold* 0,25. Nilai *recall* 100% yang dicapai pada *threshold* 0,25 sedikit menurunkan performa metrik lainnya. Oleh karena itu dapat disimpulkan bahwa performa model dengan *threshold* 0,20 memiliki performa yang lebih baik, walaupun terdapat *false negative*.

Tabel 5.11. Hasil deteksi pada data validasi.

Model	TP	FP	FN
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,25 dan <i>detection</i> NMS 0,25.	172	16	0

Tabel 5.12. Hasil performa pada data validasi.

Model	Presisi	Recall	Akurasi	F1-Score	mAP
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,25 dan <i>detection</i> NMS 0,25.	91,49%	100%	91,49%	95,56%	92,15%



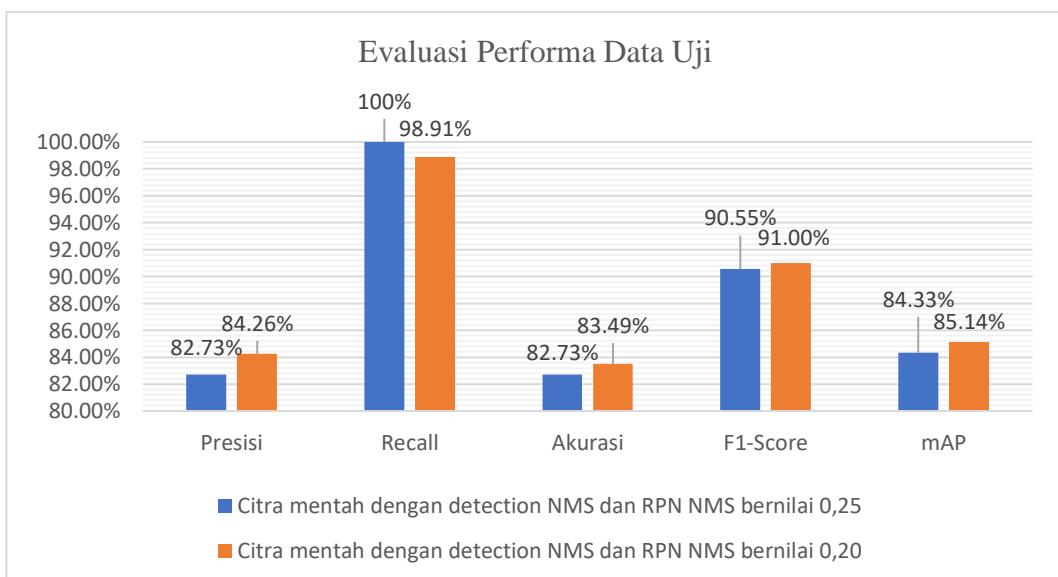
Gambar 5.21. Perbandingan performa data validasi pada NMS sebesar 0,25 dan 0,20.

Tabel 5.13. Hasil deteksi pada data uji.

Model	TP	FP	FN
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,25 dan <i>detection</i> NMS 0,25.	91	19	0

Tabel 5.14. Hasil performa pada data uji.

Model	Presisi	Recall	Akurasi	F1-Score	mAP
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,25 dan <i>detection</i> NMS 0,25.	82,73%	100%	82,73%	90,55%	84,33%



Gambar 5.22. Perbandingan performa data uji pada NMS sebesar 0,25 dan 0,20.

Hasil evaluasi performa algoritma deteksi menyimpulkan bahwa nilai NMS optimal saat *inference* adalah 0,20 untuk kedua *dataset*. Pada Tabel 5.15. adalah rangkuman performa model terbaik pada data validasi. Selanjutnya, pada Tabel 5.16. merupakan performa model terbaik pada data uji. Pada dua model terbaik tersebut yang dilatih pada dua jenis *dataset* yang berbeda, yaitu citra mentah dan citra disegmentasi menunjukkan bahwa dengan adanya segmentasi citra sebelum proses pelatihan dan *inference* dapat membuat model lebih *robust* pada data uji dan membutuhkan waktu lebih sedikit dalam pelatihan karena informasi yang tidak

diperlukan selama proses pelatihan telah dihilangkan dan dilatih pada *detection minimum confidence* yang lebih tinggi.

Tabel 5.15. Rangkuman performa model terbaik pada data validasi.

Model	Presisi	Recall	Akurasi	F1-Score	mAP
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,2 dan <i>detection</i> NMS 0,2.	91,44%	99,42%	90,96%	95,26%	91,88%
Model dilatih pada citra disegmentasi dengan nilai <i>threshold inference</i> RPN NMS 0,2 dan <i>detection</i> NMS 0,2.	90,48%	100%	90,48%	95,00%	92,25%

Tabel 5.16. Rangkuman performa model terbaik pada data uji.

Model	Presisi	Recall	Akurasi	F1-Score	mAP
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,2 dan <i>detection</i> NMS 0,2.	84,26%	98,91%	83,49%	91,00%	85,14%
Model dilatih pada citra disegmentasi dengan nilai <i>threshold inference</i> RPN NMS 0,2 dan <i>detection</i> NMS 0,2.	91,59%	100%	91,59%	95,61%	92,75%

Tabel 5.17. Waktu pelatihan yang dibutuhkan untuk kedua model terbaik.

Model	<i>Detection minimum confidence</i>	Waktu pelatihan
Model dilatih pada citra mentah dengan nilai <i>threshold inference RPN NMS</i> 0,2 dan <i>detection NMS</i> 0,2.	0,70	11 jam 30 menit
Model dilatih pada citra disegmentasi dengan nilai <i>threshold inference RPN NMS</i> 0,2 dan <i>detection NMS</i> 0,2.	0,85	10 jam 7 menit

V.3. Evaluasi Perhitungan Otomatis Sel Limfoblas dan Non-Limfoblas

Evaluasi perhitungan otomatis dilakukan pada kedua model terbaik untuk *dataset* citra mentah dan citra disegmentasi yang ditampilkan pada Tabel 5.18. Evaluasi perhitungan otomatis dilakukan pada dua kelas, yaitu limfoblas dan non-limfoblas dengan menghitung *mean absolute error* (MAE). Berdasarkan hasil evaluasi MAE pada sel limfoblas dan non-limfoblas tersebut membuktikan bahwa model yang dilatih pada citra disegmentasi memiliki MAE yang lebih kecil dibandingkan pada model yang dilatih pada citra mentah. Oleh karena itu, model terbaik pada penelitian ini adalah model yang dilatih dengan citra disegmentasi.

Tabel 5.18. Evaluasi MAE pada model terbaik.

Model	Kelas	Hitung Manual	Hitung Otomatis	MAE
Model dilatih pada citra mentah dengan nilai <i>threshold inference</i> RPN NMS 0,20 dan <i>detection</i> NMS 0,20.	Limfoblas	59	64	0,1884
	Non-limfoblas	45	44	0,2464
Model dilatih pada citra disegmentasi dengan nilai <i>threshold inference</i> RPN NMS 0,2 dan <i>detection</i> NMS 0,2.	Limfoblas	59	65	0,1159
	Non-limfoblas	45	42	0,1014

V.4. Perbandingan Hasil Deteksi WBC Metode *Mask R-CNN* dengan Penelitian-Penelitian Sebelumnya pada Pasien ALL Tipe L1

Pendeteksian sel kanker atau sel limfoblas pada pasien ALL tipe L1 merupakan penelitian yang telah dilakukan oleh Tim Riset Studio Sensor Visual, Departemen Teknik Nuklir dan Teknik Fisika (DTNTF), Fakultas Teknik, UGM. Penelitian-penelitian tersebut dilakukan dengan metode deteksi objek YOLOv2 dan *Faster R-CNN*. Pada Tabel 5.19. merupakan ringkasan perbandingan hasil pendekstian sel limfoblas pada pasien ALL tipe L1 menggunakan metode *Mask R-CNN* dengan penelitian sebelumnya. Berdasarkan Tabel 5.19. menunjukkan bawah metode *Mask R-CNN* yang telah dikembangkan (diajukan) memiliki nilai performa sedikit lebih baik jika dibandingkan metode deteksi objek yang dilakukan oleh Prabaswara [16] dan Fajri [26]. Nilai performa yang lebih baik tersebut berupa akurasi, *F1-Score*, dan mAP serta MAE secara keseluruhan. Kelebihan fitur deteksi objek yang dikembangkan pada penelitian ini dibandingkan penelitian-penelitian sebelumnya adalah menggunakan pengolahan citra untuk menfokuskan model terhadap sel darah (segmentasi) serta mengurangi waktu pelatihan dan dapat

mengklasifikasi sel dalam level piksel-ke-piksel (*object mask*) yang dapat mempermudah visualisasi sel terdeteksi.

Tabel 5.19. Perbandingan hasil penelitian menggunakan metode *Mask R-CNN* dengan penelitian yang telah dilakukan oleh Tim Riset Studio Sensor Visual DTNTF UGM.

Nama Peneliti	Tahun	Metode Penelitian	Hasil	Jumlah Sel Aktual		MAE	
				Limfoblas	Non-limfoblas	Limfoblas	Non-limfoblas
Arnanda Prabaswara [16]	2020	YOLOv2.	Nilai akurasi sebesar 90,9% dan mAP sebesar 87,5%.	93	93	0,192	0,062
M. Inska Fajri [26]	2020	<i>Faster R-CNN.</i>	Nilai akurasi sebesar 86,2%, <i>F1-Score</i> sebesar 87,9%, dan mAP sebesar 90,9%.	97	98	0,144	0,143
Mahatma Ageng Wisesa (diajukan)	2021	<i>Mask R-CNN</i> serta pengolahan citra.	Nilai akurasi sebesar 91,59%, <i>F1-Score</i> sebesar 95,61%, dan mAP sebesar 92,75%.	59	45	0,1159	0,1014

BAB VI

KESIMPULAN DAN SARAN

VI.1. Kesimpulan

1. Model *Mask R-CNN* terbaik untuk melakukan deteksi dan segmentasi *instance* sel limfoblas dan non-limfoblas pada penelitian ini diperoleh nilai akurasi sebesar 91,59%, *F1-Score* sebesar 95,61%, dan mAP sebesar 92,75%.
2. Nilai akurasi, *F1-Score*, dan mAP dapat ditingkatkan dengan melakukan *hyperparameter tuning* pada *detection minimum confidence*, *detection NMS threshold*, dan *RPN NMS threshold*. Pengolahan citra yang digunakan saat pelatihan dapat membuat performa model lebih *robust* serta megurangi waktu pelatihan.

VI.2. Saran

1. *Dataset* yang digunakan perlu diperbanyak untuk memperoleh model yang lebih *generalize*.
2. Teknik pengolahan citra yang digunakan dalam segmentasi sel darah perlu dilakukan penelitian lebih lanjut untuk memperoleh pelatihan *Mask R-CNN* yang lebih baik.
3. Perlu dilakukan penyesuaian warna pada sel terdeteksi agar lebih mempermudah visualisasi.
4. Perlu adanya penelitian lebih lanjut untuk melakukan perhitungan otomatis sel limfoblas dan non-limfoblas secara *real-time* menggunakan *Mask R-CNN*.

DAFTAR PUSTAKA

- [1] Anthony L. Mescher. *Junqueira's Basic Histology 14th Edition*. McGraw Hill Education, Bloomington, 2016.
- [2] Ahmed T. Sahlol, Philip Kollmannsberger, dan Ahmed A. Ewees. "Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features". 2020.
- [3] Andrea Biondi, Giuseppe Cimino, Rob Pieters, dan Ching-Hon Pui. "Biological and therapeutic aspects of infant leukemia". 2000.
- [4] Labati, R. D., Piuri, V., dan Scotti, F. "The acute lymphoblastic leukemia image database for image processing". 2011.
- [5] M.A. Alsalem, A .A . Zaidan, B.B. Zaidan, M. Hashim, H.T. Madhloom, dan N.D. Azeez. "A review of the automated detection and classification of acute leukaemia: Coherent taxonomy, datasets, validation and performance measurements, motivation, open challenges and recommendations". 2018.
- [6] Dipti Patra, Subrajeet Mohapatra, dan Sanghamitra Satpathi. "Image analysis of blood microscopic images for acute leukemia detection". 2010.
- [7] Jakkrich Laosai dan Kosin Chamnongthai. "Acute leukemia classification by using SVM and K-Means clustering". 2014.
- [8] Van-Nhan Tran, Waidah Ismail, Rosline Hassan, dan Atsuo Yoshitaka. "An automated method for the nuclei and cytoplasm of Acute Myeloid Leukemia detection in blood smear images". 2016.
- [9] Kirk D. Wyatt dan Richard J. Bram. "Immunotherapy in pediatric B-cell acute lymphoblastic leukemia". 2019.
- [10] J. M. Bennett, D. Catovsky, Marie-Therese, G. Flandrin, D. A. G. Galton, H. R. Gralnick, dan C. Sultan. "Proposals for the Classification of the Acute Leukaemias". 1976.
- [11] Gamal Abdul-Hamid. "Classification of Acute Leukemia". 2011.
- [12] Arif Muntasaa dan Muhammad Yusuf. "Modeling of the Acute Lymphoblastic Leukemia Detection based on the Principal Object Characteristics of the Color Image". 2019.
- [13] Najmeddine Dhib, Hakim Ghazzai, Hichem Besbes, dan Yehia Massoud. "An Automated Blood Cells Counting and Classification Framework using Mask R-CNN Deep Learning Model". 2019.

- [14] Dipti Patra dan Subrajeet Mohapatra. “Automated Cell Nucleus Segmentation and Acute Leukemia Detection in Blood Microscopic Images”. 2010.
- [15] Ilham Zulfikri Firdaus. *Klasifikasi Citra Sel Darah Putih pada Pasien Terjangkit Acute Lymphoblastic Leukemia Tipe L1 Menggunakan Metode Capsule Network*. Skripsi, Jurusan Teknik Fisika, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta, 2019.
- [16] Arnanda Prabaswara. *Deteksi Sel Limfoblas Pada Kasus Acute Lymphoblastic Leukemia Tipe L1 menggunakan Metode Convolutional Neural Network*. Skripsi, Jurusan Teknik Fisika, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta, 2020.
- [17] Nengliang Ouyang, Weijia Wang, Li Ma, Yanfang Wang, Qingwu Chen, Shanhong Yang, Jinye Xie, Shaoshen Su, Yin Cheng, Qiong Cheng, Lei Zheng, dan Yong Yuan. “Diagnosing acute promyelocytic leukemia by using convolutional neural network”. 2021.
- [18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, dan Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”. 2015.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, dan Jian Sun. “Deep Residual Learning for Image Recognition”. 2015.
- [20] Kaiming He, Georgia Gkioxari, Piotr Dollar, dan Ross Girshick. “Mask R-CNN”. 2018.
- [21] Ahmed T. Sahlol , Philip Kollmannsberger, dan Ahmed A. Ewees. “Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features”. 2020.
- [22] Pradeep Kumar Das dan Sukadev Meher. “An efficient deep Convolutional Neural Network based detection and classification of Acute Lymphoblastic Leukemia”. 2021.
- [23] Jonas Prellberg dan Oliver Kramer. “Acute Lymphoblastic Leukemia Classification from Microscopic Images using Convolutional Neural Networks”. 2019.
- [24] Rohan Khandekar, Prakhyा Shastry, Smruthi Jaishankar, Oliver Faust, dan Niranjana Sampathila. “Automated blast cell detection for Acute Lymphoblastic Leukemia diagnosis”. 2021.

- [25] Arif Muntasaa dan Muhammad Yusuf. “Modeling of the Acute Lymphoblastic Leukemia Detection based on the Principal Object Characteristics of the Color Image”. 2019.
- [26] Muhamad Inska Fajri. *Pendeteksian Sel Limfoblas pada Kasus Acute Lymphoblastic Leukemia Subtype L1 Berbasis Convolutional Neural Networks dengan Metode Faster Region-Based Convolutional Neural Networks*. Skripsi, Jurusan Teknik Fisika, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta, 2020.
- [27] De Rong Loh, Wen Xin Yong, Jullian Yapeter, Jullian Yapeter, Karupppasamy Subburaj, dan Rajesh Chandramohanadas. “A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN”. 2020.
- [28] Abhishek Dutta , Ankush Gupta, dan Andrew Zisserman. *VGG Image Annotator* (VIA). Diakses dari <https://www.robots.ox.ac.uk/~vgg/software/via/>, 15 September 2021.
- [29] Gary Bradski. “The OpenCV library”. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25: 120–125, 2000.
- [30] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuel Gouillart, dan Tony Yu. “scikit-image: image processing in Python”. 2014.
- [31] OpenCV. Histogram Equalization. Diakses dari https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html, 30 Agustus 2021.
- [32] OpenCV. Morphological Transformations. Diakses dari https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html, 31 Agustus 2021.
- [33] Scikit-image. Watershed segmentation. Diakses dari https://scikitimage.org/docs/dev/auto_examples/segmentation/plot_watershed.html, 31 Agustus 2021.
- [34] Shaoqing Ren, Kaiming He, Ross Girshick, dan Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. 2016.
- [35] Yang-Hsien Lin, Ken Y.-K. Liao, dan Kung-Bin Sung. “Automatic detection and characterization of quantitative phase images of thalassemic red blood cells using a mask region-based convolutional neural network”. 2020.

- [36] Gulpreet Kaur Chadha, Aakarsh Srivastava, Abhilasha Singh, Ritu Gupta, dan Deepanshi Singla. “An Automated Method for Counting Red Blood Cells using Image Processing”. 2019.
- [37] Jean E. Goasguen, John M. Bennett, Barbara J. Bain, Teresa Vallespi, Richard Brunning, dan Ghulam J. Mufti. “Morphological evaluation of monocytes and their precursors”. 2009.
- [38] American Society of Hematology. *Myeloblast with Auer rod*. Diakses dari <https://imagebank.hematology.org/image/60924/myeloblast-with-auer-rod>, 20 Agustus 2021.
- [39] American Society of Hematology. *Promyelocyte*. Diakses dari <https://imagebank.hematology.org/atlas/60399>, 20 Agustus 2021.
- [40] American Society of Hematology. *Myelocyte*. Diakses dari <https://imagebank.hematology.org/atlas/60506>, 20 Agustus 2021.
- [41] American Society of Hematology. *Metamyelocyte*. Diakses dari <https://imagebank.hematology.org/atlas/60398>, 20 Agustus 2021.
- [42] Ralph A. Bradshaw dan Philip D. Stahl. *Encyclopedia of Cell Biology*. Elsevier Inc, Amsterdam, 2016.
- [43] Microscope Master. *Lymphoblasts*. Diakses dari [https://www.microscopemaster.com/lymphoblasts.html#gallery\[pagegallery\]/1/](https://www.microscopemaster.com/lymphoblasts.html#gallery[pagegallery]/1/), 21 Agustus 2021.
- [44] American Society of Hematology. *Lymphocyte*. Diakses dari <https://imagebank.hematology.org/collection/60511>, 21 Agustus 2021.
- [45] Cellavision. *Natural killer cell proliferations*. Diakses dari <http://blog.cellavision.com/natural-killer-cell-proliferations/>, 21 Agustus 2021.
- [46] Dana-Farber Cancer Institute. *Putting Natural Killer Cells to the Test in Cancer Immunotherapy*. Diakses dari <https://blog.dana-farber.org/insight/2018/09/putting-natural-killer-cells-test-cancer-immunotherapy/>, 21 Agustus 2021.
- [47] Francois Chollet. *Deep Learning with Python*. Manning Publications Co, Shelter Island, 2018.
- [48] Adrian Rosebrock. *Deep Learning for Computer Vision*. PyImageSearch, 2017.
- [49] Maurice Peemen, Runbin Shi, Sohan Lal, Ben Juurlink, Bart Mesman, dan Henk Corporaal. “The Neuro Vector Engine: Flexibility to Improve Convolutional Net Efficiency for Wearable Vision”. 2016.

- [50] Roweida Mohammed, Jumanah Rawashdeh, dan Malak Abdullah. “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results”. 2020.
- [51] IBM. *What is computer vision?*. Diakses dari <https://www.ibm.com/topics/computer-vision>, 25 Agustus 2021.
- [52] Jason Brownlee. *A Gentle Introduction to Object Recognition With Deep Learning*. Diakses dari https://machinelearningmastery.com/object-recognition-with-deep-learning/?cf_chl_captcha_tk_=pmd_ipZNmAyep0_GeK0rkujAdXS8XYIGS.xmttmJff4Tvqc-1631069307-0-gqNtZGzNAuWjcnBszQu9, 26 Agustus 2021.
- [53] Reinhard Klette. *Concise Computer Vision*. Springer, Auckland, 2014.
- [54] Nick Colosimo, Antonios Tsourdos, Hyo-Sang Shin, dan Vinorth Varatharasan. “Improving Learning Effectiveness For Object Detection and Classification in Cluttered Backgrounds”. 2020.
- [55] Laurence Moroney. *Advanced Computer Vision with TensorFlow*. Coursera. Diakses dari <https://www.coursera.org/learn/advanced-computer-vision-with-tensorflow>, 17 Juli 2021.
- [56] Richard Szeliski. *Computer Vision: Algorithms and Applications*, 2nd ed. Springer, Seattle, 2021.
- [57] Eduardo A.B. da Silva dan Gelson V. Mendonça. *The Electrical Engineering Handbook*. Elsevier Inc, Amsterdam, 2005.
- [58] C. Tomasi dan R. Manduchi. “Bilateral Filtering for Gray and Color Images”. 1998.
- [59] Himanshu Singh. “Practical Machine Learning and Image Processing”. 2019.
- [60] Develop Paper. *Mask R-CNN*. Diakses dari <https://developpaper.com/mask-r-cnn/>, 1 September 2021.
- [61] Jan Hosang, Rodrigo Benenson, dan Bernt Schiele. “Learning non-maximum suppression”. 2017.
- [62] Yifan Ding, Yujia Yang, dan Yan Cui. “Deep Learning for Classifying of White Blood Cancer”. 2019.
- [63] Stanford. *Convolutional Neural Networks for Visual Recognition*. Diakses dari <http://cs231n.stanford.edu/index.html>, 2 September 2021.

- [64] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, dan Andrew Zisserman. “The PASCAL Visual Object Classes (VOC) Challenge”. 2009.
- [65] Rafael Padilla, Sergio L. Netto, dan Eduardo A. B. da Silva. “A Survey on Performance Metrics for Object-Detection Algorithms”. 2020.
- [66] Cecilia Di Ruberto, Andrea Loddo, dan Giovanni Puglisi. “Blob Detection and Deep Learning for Leukemic Blood Image Analysis”. 2020.
- [67] Jyoti Rawata, Annapurna Singha, H.S. Bhaduria, dan Jitendra Virmani. “Computer Aided Diagnostic System for Detection of Leukemia Using Microscopic Images”. 2015
- [68] Nizar Ahmed, Altug Yigit, Zerrin Isik, dan Adil Alpkocak. “Identification of Leukemia Subtypes from Microscopic Images Using Convolutional Neural Network”. 2019.
- [69] Jonas Prellberg dan Oliver Kramer. “Acute Lymphoblastic Leukemia Classification from Microscopic Images Using Convolutional Neural Networks”. 2019.
- [70] Christian Marzahl, Marc Aubreville, Jörn Voigt, dan Andreas Maier. “Classification of Leukemic B-Lymphoblast Cells from Blood Smear Microscopic Images with an Attention-Based Deep Learning Method and Advanced Augmentation Techniques”. 2019.

LAMPIRAN

LAMPIRAN A

HASIL PREDIKSI MODEL DILATIH PADA CITRA MENTAH DENGAN NILAI *THRESHOLD INFERENCE RPN NMS* DAN *DETECTION NMS* SEBESAR 0,2

Tabel A.1. Hasil prediksi otomatis pada data validasi.

filename	limfoblas	non-limfoblas	TP	FP	FN
ALL1.1.1001 L (rescaled).jpg	0	1	1		
ALL1.1.1003 S (rescaled).jpg	0	1	1		
ALL1.1.1011 L (rescaled).jpg	0	1	1		
ALL1.1.1016 L (rescaled).jpg	0	1	1		
ALL1.1.102 M (rescaled).jpg	1	0		1	
ALL1.1.1023 B L (rescaled).jpg	1	1	2		
ALL1.1.1025 L (rescaled).jpg	0	1	1		
ALL1.1.1056 S (rescaled).jpg	0	1	1		
ALL1.1.1058 B L (rescaled).jpg	1	1	2		
ALL1.1.107 STAB (rescaled).jpg	0	1	1		
ALL1.1.1077 S (rescaled).jpg	0	1	1		
ALL1.1.1079 S (rescaled).jpg	0	1	1		
ALL1.1.1091 L B (rescaled).jpg	1	1	2		
ALL1.1.1096 B L (rescaled).jpg	1	1	2		
ALL1.1.1097 S (rescaled).jpg	0	1	1		
ALL1.1.1118 L B (rescaled).jpg	1	1	2		
ALL1.1.1121 S (rescaled).jpg	0	1	1		
ALL1.1.1122 B L (rescaled).jpg	2	0	1	1	
ALL1.1.1127 S B (rescaled).jpg	1	1	2		
ALL1.1.1129 L B B B (rescaled).jpg	2	2	3	1	
ALL1.1.1137 METAMIELOSIT B (rescaled).jpg	1	1	2		
ALL1.1.1155 B L (rescaled).jpg	1	1	2		

ALL1.1.1160 B L (rescaled).jpg	1	1	2		
ALL1.1.1174 B L (rescaled).jpg	1	1	2		
ALL1.1.1194 EOSINOFIL B (rescaled).jpg	1	1	2		
ALL1.1.1196 B B L B (ATAS) (rescaled).jpg	3	1	4		
ALL1.1.1199 L B (rescaled).jpg	1	1	2		
ALL1.1.1224 B B S (rescaled).jpg	1	1	2		1
ALL1.1.1265 L(ATAS) B (rescaled).jpg	2	0	1	1	
ALL1.1.129 STAB (rescaled).jpg	0	1	1		
ALL1.1.132 B L M(BAWAH) (rescaled).jpg	2	1	2	1	
ALL1.1.1644 B B (rescaled).jpg	2	0	2		
ALL1.1.1649 B B (rescaled).jpg	2	0	2		
ALL1.1.1650 B B (rescaled).jpg	2	0	2		
ALL1.1.1651 B B (rescaled).jpg	2	0	2		
ALL1.1.1653 B B (rescaled).jpg	2	0	2		
ALL1.1.1655 B B B (rescaled).jpg	3	0	3		
ALL1.1.1656 B B B B B (rescaled).jpg	4	1	4	1	
ALL1.1.1658 B B (rescaled).jpg	2	0	2		
ALL1.1.1662 B B (rescaled).jpg	2	0	2		
ALL1.1.1663 B B (rescaled).jpg	2	0	2		
ALL1.1.1675 B B (rescaled).jpg	2	0	2		
ALL1.1.1678 B B (rescaled).jpg	2	0	2		
ALL1.1.1679 B B (rescaled).jpg	2	0	2		

ALL1.1.1685 NRBC B (rescaled).jpg	1	1	2		
ALL1.1.1686 B (rescaled).jpg	1	0	1		
ALL1.1.1687 B (rescaled).jpg	1	0	1		
ALL1.1.1689 B (rescaled).jpg	1	0	1		
ALL1.1.1690 B (rescaled).jpg	1	0	1		
ALL1.1.1691 B (rescaled).jpg	1	0	1		
ALL1.1.1692 B (rescaled).jpg	1	0	1		
ALL1.1.1693 B (rescaled).jpg	1	0	1		
ALL1.1.1694 B (rescaled).jpg	1	0	1		
ALL1.1.1697 B (rescaled).jpg	1	0	1		
ALL1.1.1698 B METAMIELOSIT L B (rescaled).jpg	2	2	2	2	
ALL1.1.1701 B (rescaled).jpg	1	0	1		
ALL1.1.1705 B (rescaled).jpg	1	0	1		
ALL1.1.1707 B (rescaled).jpg	1	1	1	1	
ALL1.1.1708 B (rescaled).jpg	1	0	1		
ALL1.1.1709 B (rescaled).jpg	1	0	1		
ALL1.1.1710 B (rescaled).jpg	1	0	1		
ALL1.1.1712 B (rescaled).jpg	1	0	1		
ALL1.1.1713 B (rescaled).jpg	1	0	1		
ALL1.1.1716 B (rescaled).jpg	1	0	1		
ALL1.1.1717 B (rescaled).jpg	1	0	1		
ALL1.1.1719 B (rescaled).jpg	1	0	1		
ALL1.1.1721 B (rescaled).jpg	1	0	1		
ALL1.1.1722 B (rescaled).jpg	1	0	1		
ALL1.1.1723 B (rescaled).jpg	1	0	1		
ALL1.1.1725 B (rescaled).jpg	1	0	1		
ALL1.1.204 M B (rescaled).jpg	2	0	1	1	
ALL1.1.21 B S (rescaled).jpg	1	1	2		
ALL1.1.234 M B (rescaled).jpg	1	1	2		
ALL1.1.244 NRBC (rescaled).jpg	0	1	1		
ALL1.1.317 B B STAB B (rescaled).jpg	3	1	4		
ALL1.1.33 B X (rescaled).jpg	1	1	2		
ALL1.1.39 M (rescaled).jpg	1	0		1	
ALL1.1.391 MIELOSIT L L (rescaled).jpg	0	3	3		
ALL1.1.409 EOSINOFIL (rescaled).jpg	0	1	1		

ALL1.1.492 MIELOSIT B (rescaled).jpg	2	0	1	1	
ALL1.1.598 L STAB(ATAS) B (BAWAH) (rescaled).jpg	1	2	3		
ALL1.1.6 M (rescaled).jpg	0	1	1		
ALL1.1.645 M (rescaled).jpg	0	1	1		
ALL1.1.676 MONOBLAS B (rescaled).jpg	1	2	2	1	
ALL1.1.730 L (rescaled).jpg	0	1	1		
ALL1.1.731 L (rescaled).jpg	0	1	1		
ALL1.1.734 L (rescaled).jpg	0	1	1		
ALL1.1.739 L (rescaled).jpg	0	1	1		
ALL1.1.740 L (rescaled).jpg	0	1	1		
ALL1.1.746 S B (BAWAH) (rescaled).jpg	1	1	2		
ALL1.1.755 L (rescaled).jpg	0	1	1		
ALL1.1.762 L (rescaled).jpg	0	1	1		
ALL1.1.765 L (rescaled).jpg	0	1	1		
ALL1.1.771 L S (rescaled).jpg	0	2	2		
ALL1.1.772 L (rescaled).jpg	0	1	1		
ALL1.1.783 L S (rescaled).jpg	0	2	2		
ALL1.1.786 L (rescaled).jpg	0	1	1		
ALL1.1.797 L L (rescaled).jpg	0	2	2		
ALL1.1.80 M (rescaled).jpg	0	1	1		
ALL1.1.816 L (rescaled).jpg	0	1	1		
ALL1.1.818 L (rescaled).jpg	0	1	1		
ALL1.1.822 L (rescaled).jpg	0	1	1		
ALL1.1.834 L (rescaled).jpg	0	1	1		
ALL1.1.838 L (rescaled).jpg	0	1	1		
ALL1.1.859 B PROMONOSIT (rescaled).jpg	2	0	1	1	
ALL1.1.869 L (rescaled).jpg	0	1	1		
ALL1.1.871 L (rescaled).jpg	0	1	1		
ALL1.1.872 S B(BAWAH) (rescaled).jpg	1	1	2		
ALL1.1.884 S (rescaled).jpg	0	1	1		
ALL1.1.890 L (rescaled).jpg	0	1	1		
ALL1.1.904 L (rescaled).jpg	0	1	1		
ALL1.1.907 S (rescaled).jpg	0	1	1		
ALL1.1.910 L (rescaled).jpg	0	1	1		
ALL1.1.913 S (rescaled).jpg	0	1	1		
ALL1.1.918 S (rescaled).jpg	0	1	1		
ALL1.1.921 S (rescaled).jpg	0	1	1		

ALL1.1.953 L (rescaled).jpg	0	1	1		
ALL1.1.954 S (rescaled).jpg	0	1	1		
ALL1.1.968 L (rescaled).jpg	1	0		1	
ALL1.1.969 L (rescaled).jpg	0	1	1		
ALL1.1.980 L (rescaled).jpg	1	0		1	
ALL1.1.998 L (rescaled).jpg	0	1	1		
Total	100	87	171	16	1

Tabel A.2. Hasil prediksi otomatis pada data uji.

filename	limfoblas	non-limfoblas	TP	FP	FN
ALL1.1.1026 L (rescaled).jpg	0	1	1		
ALL1.1.1050 L (rescaled).jpg	0	1	1		
ALL1.1.1062 L (rescaled).jpg	1	0		1	
ALL1.1.1066 L (rescaled).jpg	1	0		1	
ALL1.1.1068 L (rescaled).jpg	0	1	1		
ALL1.1.1071 L (rescaled).jpg	1	0		1	
ALL1.1.1085 L (rescaled).jpg	0	2	1	1	
ALL1.1.1092 L (rescaled).jpg	0	1	1		
ALL1.1.1099 B STAB (rescaled).jpg	1	1	2		
ALL1.1.1103 L (rescaled).jpg	0	1	1		
ALL1.1.1116 L (rescaled).jpg	0	1	1		
ALL1.1.1126 S (rescaled).jpg	0	1	1		
ALL1.1.1146 S (rescaled).jpg	0	1	1		
ALL1.1.1156 S (rescaled).jpg	0	1	1		
ALL1.1.1167 L (rescaled).jpg	0	1	1		
ALL1.1.1193 S (rescaled).jpg	0	1	1		
ALL1.1.1200 S (rescaled).jpg	0	1	1		
ALL1.1.1214 L (rescaled).jpg	0	1	1		
ALL1.1.1228 B(ATAS) L B (rescaled).jpg	2	1	3		
ALL1.1.1230 S (rescaled).jpg	0	1	1		
ALL1.1.1233 L (rescaled).jpg	0	2	1	1	
ALL1.1.1234 L (rescaled).jpg	1	0		1	
ALL1.1.1238 B L (rescaled).jpg	1	1	2		
ALL1.1.1239 L (rescaled).jpg	0	1	1		
ALL1.1.1622 L MIELOSIT (rescaled).jpg	1	1	1	1	
ALL1.1.1665 METAMIELOSIT B(BAWAH) (rescaled).jpg	2	0	1	1	

ALL1.1.1667 B M L (rescaled).jpg	1	2	3		
ALL1.1.1684 B B (rescaled).jpg	2	0	2		
ALL1.1.1688 B B (rescaled).jpg	1	1	1	1	
ALL1.1.1700 B B (rescaled).jpg	2	0	2		
ALL1.1.1711 B B (rescaled).jpg	2	0	2		
ALL1.1.1714 B B (rescaled).jpg	2	0	2		
ALL1.1.1724 B B (rescaled).jpg	2	0	2		
ALL1.1.1728 B B (rescaled).jpg	2	0	2		
ALL1.1.1729 B B (rescaled).jpg	2	0	2		
ALL1.1.1730 B (rescaled).jpg	1	0	1		
ALL1.1.1731 B B (rescaled).jpg	2	0	2		
ALL1.1.1732 B (rescaled).jpg	1	0	1		
ALL1.1.1735 B (rescaled).jpg	1	1	1	1	
ALL1.1.1736 B B (rescaled).jpg	2	1	2	1	
ALL1.1.1737 B (rescaled).jpg	1	0	1		
ALL1.1.1738 B (rescaled).jpg	1	0	1		
ALL1.1.1739 B (rescaled).jpg	1	0	1		
ALL1.1.1740 B (rescaled).jpg	1	0	1		
ALL1.1.1742 B B (rescaled).jpg	2	0	2		
ALL1.1.1743 B B (rescaled).jpg	2	0	2		
ALL1.1.1744 B (rescaled).jpg	1	0	1		
ALL1.1.1746 B (rescaled).jpg	0	1		1	
ALL1.1.1747 B B (rescaled).jpg	2	0	2		
ALL1.1.1750 B (rescaled).jpg	1	0	1		
ALL1.1.1752 B B B B (rescaled).jpg	3	0	3		1
ALL1.1.1753 B (rescaled).jpg	1	0	1		
ALL1.1.1756 B (rescaled).jpg	1	0	1		
ALL1.1.1757 B (rescaled).jpg	1	0	1		
ALL1.1.28 B S (rescaled).jpg	1	1	2		

ALL1.1.31 B B L(BAWAH) (rescaled).jpg	2	1	3		
ALL1.1.35 B L(BAWAH) (rescaled).jpg	2	0	1	1	
ALL1.1.46 L B S (rescaled).jpg	1	2	3		
ALL1.1.49 L B (rescaled).jpg	1	1	2		
ALL1.1.521 M (rescaled).jpg	1	0		1	
ALL1.1.543 STAB (rescaled).jpg	0	1	1		
ALL1.1.55 S B (rescaled).jpg	0	2	1	1	
ALL1.1.604 B EOSINOFIL (rescaled).jpg	1	1	2		
ALL1.1.614 M (rescaled).jpg	0	2	1	1	
ALL1.1.633 M (rescaled).jpg	0	1	1		
ALL1.1.636 NRBC (rescaled).jpg	0	1	1		
ALL1.1.66 B L (rescaled).jpg	1	1	2		
ALL1.1.831 PROMONOSIT (rescaled).jpg	1	0		1	
ALL1.1.937 B L (rescaled).jpg	1	1	2		
Total	64	44	91	17	1

LAMPIRAN B

HASIL PREDIKSI MODEL DILATIH PADA CITRA DISEGMENTASI DENGAN NILAI *THRESHOLD INFERENCE RPN NMS DAN DETECTION NMS SEBESAR 0,2*

Tabel B.1. Hasil prediksi otomatis pada data validasi.

filename	limfoblas	non-limfoblas	TP	FP	FN
ALL1.1.1001 L (rescaled).jpg	0	1	1		
ALL1.1.1003 S (rescaled).jpg	0	1	1		
ALL1.1.1011 L (rescaled).jpg	0	1	1		
ALL1.1.1016 L (rescaled).jpg	0	1	1		
ALL1.1.102 M (rescaled).jpg	1	0		1	
ALL1.1.1023 B L (rescaled).jpg	1	1	2		
ALL1.1.1025 L (rescaled).jpg	0	1	1		
ALL1.1.1056 S (rescaled).jpg	0	1	1		
ALL1.1.1058 B L (rescaled).jpg	1	1	2		
ALL1.1.107 STAB (rescaled).jpg	0	1	1		
ALL1.1.1077 S (rescaled).jpg	0	1	1		
ALL1.1.1079 S (rescaled).jpg	0	1	1		
ALL1.1.1091 L B (rescaled).jpg	1	1	2		
ALL1.1.1096 B L (rescaled).jpg	1	1	2		
ALL1.1.1097 S (rescaled).jpg	0	1	1		
ALL1.1.1118 L B (rescaled).jpg	1	1	2		
ALL1.1.1121 S (rescaled).jpg	0	1	1		
ALL1.1.1122 B L (rescaled).jpg	1	1	2		
ALL1.1.1127 S B (rescaled).jpg	1	1	2		
ALL1.1.1129 L B B (rescaled).jpg	2	2	1	3	
ALL1.1.1137 METAMIELOSIT B (rescaled).jpg	1	1	2		
ALL1.1.1155 B L (rescaled).jpg	1	1	2		
ALL1.1.1160 B L (rescaled).jpg	1	1	2		
ALL1.1.1174 B L (rescaled).jpg	1	1	2		
ALL1.1.1194 EOSINOFIL B (rescaled).jpg	1	1	2		
ALL1.1.1196 B B L B (ATAS) (rescaled).jpg	3	1	4		
ALL1.1.1199 L B (rescaled).jpg	1	1	2		
ALL1.1.1224 B B S (rescaled).jpg	2	1	3		

ALL1.1.1265 L(ATAS) B (rescaled).jpg	1	1	2		
ALL1.1.129 STAB (rescaled).jpg	0	1	1		
ALL1.1.132 B L M(BAWAH) (rescaled).jpg	2	1	2	1	
ALL1.1.1644 B B (rescaled).jpg	2	0	2		
ALL1.1.1649 B B (rescaled).jpg	2	0	2		
ALL1.1.1650 B B (rescaled).jpg	0	2		2	
ALL1.1.1651 B B (rescaled).jpg	2	0	2		
ALL1.1.1653 B B (rescaled).jpg	2	0	2		
ALL1.1.1655 B B B (rescaled).jpg	2	1	2	1	
ALL1.1.1656 B B B B B (rescaled).jpg	3	2	3	2	
ALL1.1.1658 B B (rescaled).jpg	2	0	2		
ALL1.1.1662 B B (rescaled).jpg	2	0	2		
ALL1.1.1663 B B (rescaled).jpg	2	0	2		
ALL1.1.1675 B B (rescaled).jpg	2	0	2		
ALL1.1.1678 B B (rescaled).jpg	2	0	2		
ALL1.1.1679 B B (rescaled).jpg	2	0	2		
ALL1.1.1685 NRBC B (rescaled).jpg	1	1	2		
ALL1.1.1686 B (rescaled).jpg	1	0	1		
ALL1.1.1687 B (rescaled).jpg	1	0	1		
ALL1.1.1689 B (rescaled).jpg	1	0	1		
ALL1.1.1690 B (rescaled).jpg	0	1		1	
ALL1.1.1691 B (rescaled).jpg	1	0	1		
ALL1.1.1692 B (rescaled).jpg	1	0	1		
ALL1.1.1693 B (rescaled).jpg	1	0	1		
ALL1.1.1694 B (rescaled).jpg	1	0	1		
ALL1.1.1697 B (rescaled).jpg	1	0	1		
ALL1.1.1698 B METAMIELOSIT L B (rescaled).jpg	2	2	4		
ALL1.1.1701 B (rescaled).jpg	1	0	1		
ALL1.1.1705 B (rescaled).jpg	1	0	1		
ALL1.1.1707 B (rescaled).jpg	1	0	1		
ALL1.1.1708 B (rescaled).jpg	1	0	1		
ALL1.1.1709 B (rescaled).jpg	1	1	1	1	
ALL1.1.1710 B (rescaled).jpg	1	0	1		
ALL1.1.1712 B (rescaled).jpg	1	0	1		
ALL1.1.1713 B (rescaled).jpg	1	0	1		
ALL1.1.1716 B (rescaled).jpg	1	0	1		

ALL1.1.1717 B (rescaled).jpg	1	0	1		
ALL1.1.1719 B (rescaled).jpg	1	0	1		
ALL1.1.1721 B (rescaled).jpg	1	0	1		
ALL1.1.1722 B (rescaled).jpg	1	0	1		
ALL1.1.1723 B (rescaled).jpg	1	0	1		
ALL1.1.1725 B (rescaled).jpg	1	0	1		
ALL1.1.204 M B (rescaled).jpg	2	0	1	1	
ALL1.1.21 B S (rescaled).jpg	1	1	2		
ALL1.1.234 M B (rescaled).jpg	1	1	2		
ALL1.1.244 NRBC (rescaled).jpg	0	1	1		
ALL1.1.317 B B STAB B (rescaled).jpg	3	1	4		
ALL1.1.33 B X (rescaled).jpg	1	1	2		
ALL1.1.39 M (rescaled).jpg	0	1	1		
ALL1.1.391 MIELOSIT L L (rescaled).jpg	0	3	3		
ALL1.1.409 EOSINOFIL (rescaled).jpg	0	1	1		
ALL1.1.492 MIELOSIT B (rescaled).jpg	2	0	1	1	
ALL1.1.598 L STAB(ATAS) B (BAWAH) (rescaled).jpg	1	2	3		
ALL1.1.6 M (rescaled).jpg	0	1	1		
ALL1.1.645 M (rescaled).jpg	0	1	1		
ALL1.1.676 MONOBLAS B (rescaled).jpg	1	2	2	1	
ALL1.1.730 L (rescaled).jpg	0	1	1		
ALL1.1.731 L (rescaled).jpg	0	1	1		
ALL1.1.734 L (rescaled).jpg	0	1	1		
ALL1.1.739 L (rescaled).jpg	0	1	1		
ALL1.1.740 L (rescaled).jpg	1	0		1	
ALL1.1.746 S B (BAWAH) (rescaled).jpg	1	1	2		
ALL1.1.755 L (rescaled).jpg	0	1	1		
ALL1.1.762 L (rescaled).jpg	0	1	1		
ALL1.1.765 L (rescaled).jpg	0	1	1		
ALL1.1.771 L S (rescaled).jpg	0	3	2	1	
ALL1.1.772 L (rescaled).jpg	0	1	1		
ALL1.1.783 L S (rescaled).jpg	0	2	2		
ALL1.1.786 L (rescaled).jpg	0	1	1		
ALL1.1.797 L L (rescaled).jpg	0	2	2		
ALL1.1.80 M (rescaled).jpg	0	1	1		
ALL1.1.816 L (rescaled).jpg	0	1	1		

ALL1.1.818 L (rescaled).jpg	0	1	1		
ALL1.1.822 L (rescaled).jpg	0	1	1		
ALL1.1.834 L (rescaled).jpg	0	1	1		
ALL1.1.838 L (rescaled).jpg	0	1	1		
ALL1.1.859 B PROMONOSIT (rescaled).jpg	2	0	1	1	
ALL1.1.869 L (rescaled).jpg	0	1	1		
ALL1.1.871 L (rescaled).jpg	0	1	1		
ALL1.1.872 S B(BAWAH) (rescaled).jpg	1	1	2		
ALL1.1.884 S (rescaled).jpg	0	1	1		
ALL1.1.890 L (rescaled).jpg	0	1	1		
ALL1.1.904 L (rescaled).jpg	0	1	1		
ALL1.1.907 S (rescaled).jpg	0	1	1		
ALL1.1.910 L (rescaled).jpg	0	1	1		
ALL1.1.913 S (rescaled).jpg	0	1	1		
ALL1.1.918 S (rescaled).jpg	0	1	1		
ALL1.1.921 S (rescaled).jpg	0	1	1		
ALL1.1.953 L (rescaled).jpg	0	1	1		
ALL1.1.954 S (rescaled).jpg	0	1	1		
ALL1.1.968 L (rescaled).jpg	0	1	1		
ALL1.1.969 L (rescaled).jpg	0	1	1		
ALL1.1.980 L (rescaled).jpg	0	1	1		
ALL1.1.998 L (rescaled).jpg	0	1	1		
Total	92	97	171	18	0

Tabel B.2. Hasil prediksi otomatis pada data uji.

filename	limfoblas	non-limfoblas	TP	FP	FN
ALL1.1.1026 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1050 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1062 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1066 L (rescaled) (Masked WC).jpg	1	0		1	
ALL1.1.1068 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1071 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1085 L (rescaled) (Masked WC).jpg	0	2	1	1	

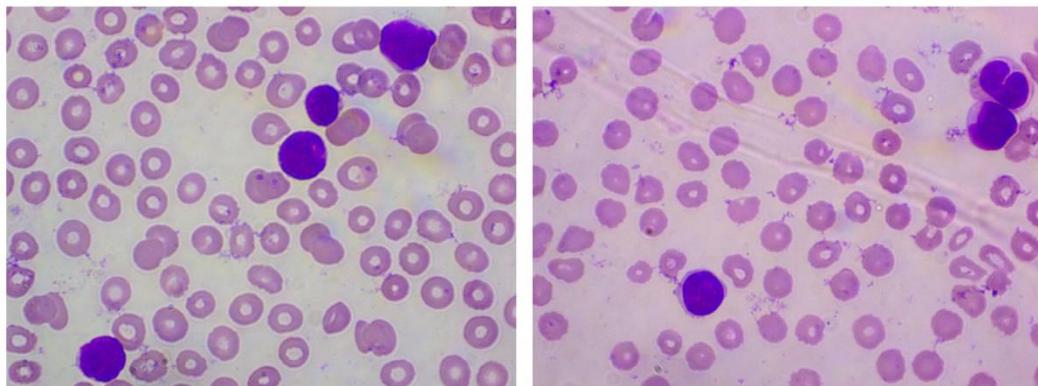
ALL1.1.1092 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1099 B STAB (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.1103 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1116 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1126 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1146 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1156 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1167 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1193 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1200 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1214 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1228 B(ATAS) L B (rescaled) (Masked WC).jpg	2	1	3		
ALL1.1.1230 S (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1233 L (rescaled) (Masked WC).jpg	1	0		1	
ALL1.1.1234 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1238 B L (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.1239 L (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.1622 L MIELOSIT (rescaled) (Masked WC).jpg	0	2	2		
ALL1.1.1665 METAMIELOSIT B(BAWAH) (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.1667 B M L (rescaled) (Masked WC).jpg	1	2	3		
ALL1.1.1684 B B (rescaled) (Masked WC).jpg	2	0	2		

ALL1.1.1688 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1700 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1711 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1714 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1724 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1728 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1729 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1730 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1731 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1732 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1735 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1736 B B (rescaled) (Masked WC).jpg	3	0	2	1	
ALL1.1.1737 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1738 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1739 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1740 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1742 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1743 B B (rescaled) (Masked WC).jpg	1	1	1	1	
ALL1.1.1744 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1746 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1747 B B (rescaled) (Masked WC).jpg	2	0	2		
ALL1.1.1750 B (rescaled) (Masked WC).jpg	1	0	1		

ALL1.1.1752 B B B B (rescaled) (Masked WC).jpg	4	0	4		
ALL1.1.1753 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1756 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.1757 B (rescaled) (Masked WC).jpg	1	0	1		
ALL1.1.28 B S (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.31 B B L(BAWAH) (rescaled) (Masked WC).jpg	2	1	3		
ALL1.1.35 B L(BAWAH) (rescaled) (Masked WC).jpg	2	0	1	1	
ALL1.1.46 L B S (rescaled) (Masked WC).jpg	1	2	3		
ALL1.1.49 L B (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.521 M (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.543 STAB (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.55 S B (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.604 B EOSINOFIL (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.614 M (rescaled) (Masked WC).jpg	1	1	1	1	
ALL1.1.633 M (rescaled) (Masked WC).jpg	1	0		1	
ALL1.1.636 NRBC (rescaled) (Masked WC).jpg	0	1	1		
ALL1.1.66 B L (rescaled) (Masked WC).jpg	1	1	2		
ALL1.1.831 PROMONOSIT (rescaled) (Masked WC).jpg	1	0		1	
ALL1.1.937 B L (rescaled) (Masked WC).jpg	1	1	2		
Total	65	42	98	9	0

LAMPIRAN C

NAMA FILE YANG TELAH DILABELI



ALL1.1.1196 B B L B (ATAS).jpg

ALL1.1.598 L STAB(ATAS) B (BAWAH).jpg

ALL1.1.1196 B B L B (ATAS).jpg

1 = Pasien 1

1 = Sampel darah ke-1

B = Limfoblas

L = Limfosit (Non-limfoblas)

(ATAS) = Sel terdapat pada bagian atas citra.

Pembacaan label objek diurutkan dari kiri ke kanan, sehingga pembacaan adalah Limfoblas, Limfoblas, Limfosit, Limfoblas.

ALL1.1.598 L STAB(ATAS) B (BAWAH).jpg

Pembacaan adalah Limfoblas, Non-limfoblas (di atas), Limfoblas (di bawah).

LAMPIRAN D

KODE SUMBER

Kode sumber pelatihan bersifat *open source* dikembangkan dari: https://github.com/matterport/Mask_RCNN yang diimplementasi dalam Python 3, Keras, dan TensorFlow ditulis oleh Waleed Abdulla. Kode tersebut dikembangkan untuk melakukan deteksi dan segmentasi *instance* sel limfoblas dan non-limfoblas. Namun, kode untuk pelatihan, *inference*, segmentasi citra, augmentasi citra, dan *auto-oriented* agumentasi serta implementasi yang telah dikembangkan dapat diakses pada:

<https://github.com/hatmahat/AutoCancerCellsCount>