

**KLASIFIKASI *MULTICLASS* STATUS ANEMIA MENGGUNAKAN
METODE *ARTIFICIAL NEURAL NETWORK* (ANN) BERBASIS
PARAMETER *COMPLETE BLOOD COUNT* (CBC)**

SKRIPSI

Oleh :
DITA SUCI YOFANA
NIM. 220605110117



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**KLASIFIKASI *MULTICLASS* STATUS ANEMIA MENGGUNAKAN
METODE *ARTIFICIAL NEURAL NETWORK* (ANN) BERBASIS
PARAMETER *COMPLETE BLOOD COUNT* (CBC)**

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
DITA SUCI YOFANA
NIM. 220605110117

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

**KLASIFIKASI MULTICLASS STATUS ANEMIA MENGGUNAKAN
METODE ARTIFICIAL NEURAL NETWORK (ANN) BERBASIS
PARAMETER COMPLETE BLOOD COUNT (CBC)**

SKRIPSI

Oleh :
DITA SUCI YOFANA
NIM. 220605110117

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 8 Desember 2025

Pembimbing I,



Okta Oomaruddin Aziz, M.Kom
NIP. 19911019 201903 1 013

Pembimbing II,



Dr. Cahyo Crysdian, M. Cs
NIP. 19740424 200901 1 008

Mengetahui
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M. Kom
NIP. 19841010 20193 1 012

HALAMAN PENGESAHAN

KLASIFIKASI *MULTICLASS* STATUS ANEMIA MENGGUNAKAN METODE *ARTIFICIAL NEURAL NETWORK* (ANN) BERBASIS PARAMETER *COMPLETE BLOOD COUNT* (CBC)

SKRIPSI

Oleh :

DITA SUCI YOFANA

NIM. 220605110117

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 17 Desember 2025

Susunan Dewan Penguji

Ketua Penguji : Dr. Irwan Budi Santoso, M. Kom
NIP. 19770103 201101 1 004

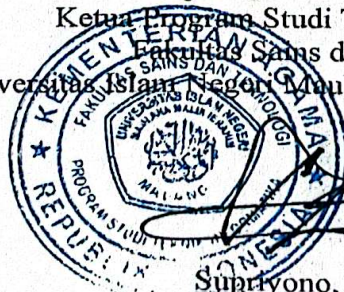
Anggota Penguji I : Tri Mukti Lestari, M.Kom
NIP. 19911108 202012 2 005

Anggota Penguji II : Okta Oomaruddin Aziz, M.Kom
NIP. 19911019 201903 1 013

Anggota Penguji III : Dr. Cahyo Crysdian, M. Cs
NIP. 19740424 200901 1 008

(
(
(
(

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Dita Suci Yofana
NIM : 220605110117
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Klasifikasi *Multiclass* Status Anemia Menggunakan Metode *Artificial Neural Network* (ANN) Berbasis Parameter *Complete Blood Count* (CBC)

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Desember 2025
Yang membuat pernyataan,



Dita Suci Yofana
NIM.220605110117

MOTTO

“Bentuk kesabaran yang paling sulit adalah kemampuan untuk bertahan dalam situasi yang tidak kamu inginkan”

-Gus Baha

“God has perfect timing, never early, never late. It takes a little patience and it takes a lot of faith, but it’s a worth to wait”

HALAMAN PERSEMBAHAN

Puji syukur kehadiran ALLAH SWT yang telah memberikan rahmat dan Hidayah-Nya sehingga penulis diberi kemudahan dalam Menyelesaikan penulisan skripsi ini dengan baik. Karya ini penulis persembahkan sebagai bentuk rasa syukur dan terima kasih yang tulus kepada:

Mama tercinta, Elvi Tiana, dan Ayah tercinta, Cahyo Andono, yang senantiasa memberikan kasih sayang, kesabaran, doa, serta dukungan tanpa henti kepada penulis. menjadikan kekuatan utama serta sumber motivasi bagi penulis dalam menempuh perjalanan akademik hingga skripsi ini dapat diselesaikan.

Abang, kakak, dan adik penulis, yaitu Krisna Aditya Putra, Dika Ramadhanti, dan Najwa Dinda Syahira, yang senantiasa menjadi sumber semangat, dukungan bagi penulis. Kehadiran serta perhatian kalian menjadi penguat bagi penulis dalam menjalani setiap perjalanan hidup.

Segenap keluarga besar yang senantiasa memberikan dukungan, perhatian, serta doa-doa terbaik sehingga penulis dapat menyelesaikan skripsi ini.

Para Dosen yang telah membimbing dan mengarahkan penulis melalui ilmu, kesabaran, serta keteladanan, sehingga proses pembelajaran berlangsung dalam ruang yang menumbuhkan nalar, kedisiplinan, dan tanggung jawab.

Teman-teman Angkatan 2022 Teknik Informatika
yang melalui kebersamaan, dukungan, dan perjalanan yang dilalui bersama,
menjadikan setiap langkah panjang selama masa studi terasa lebih ringan dan
bermakna.

KATA PENGANTAR

Assalamualaikum wr wb.

Segala puji dan syukur penulis panjatkan ke hadirat Allah *subhanahu wa ta'ala* atas limpahan rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Klasifikasi *Multiclass* Status Anemia Menggunakan Metode *Artificial Neural Network* (ANN) Berbasis Parameter *Complete Blood Count* (CBC)”. Sholawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW, semoga kita semua memperoleh syafaat beliau di hari akhir kelak. Aamiin.

Penulis menyadari bahwa penyusunan skripsi ini tidak dapat diselesaikan tanpa adanya bantuan, bimbingan, doa, serta dukungan dari berbagai pihak, baik secara langsung maupun tidak langsung, dalam bentuk moril maupun materiil. Setiap proses yang penulis jalani merupakan hasil dari perhatian, kesabaran, dan kepercayaan yang diberikan. Oleh karena itu, dengan segala kerendahan hati dan rasa syukur, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang

4. Okta Qomaruddin Aziz, M.Kom., selaku Dosen Pembimbing I, yang tidak hanya berperan sebagai pembimbing, tetapi juga sebagai sosok yang senantiasa memberi dukungan moral dan motivasi. Dengan kesabaran, ketulusan, serta perhatian yang luar biasa, beliau selalu meluangkan waktu untuk memberikan arahan, saran, dan masukan kapan pun dibutuhkan, tanpa memberikan tekanan kepada penulis. Pendampingan yang diberikan selama kurang lebih satu semester menjadi peran yang sangat berarti dan tidak ternilai dalam membantu penulis menyelesaikan skripsi ini dengan baik.
5. Dr. Cahyo Crysdiyan, M.Cs., selaku Dosen Pembimbing II, yang telah memberikan bimbingan, arahan, dan masukan yang sangat berarti dengan penuh ketelitian. Pandangan dan saran beliau membantu penulis memperbaiki serta menyempurnakan penyusunan skripsi ini hingga dapat diselesaikan dengan baik.
6. Dr. Irwan Budi Santoso, M.Kom., selaku Ketua Penguji dan Ibu Tri Mukti Lestari, dosen Penguji I yang telah menguji serta memberikan banyak saran untuk menyelesaikan skripsi ini dengan baik.
7. Nia Faricha S, Si., selaku admin Program Studi Teknik Informatika yang dengan penuh kesabaran selalu memberikan informasi, membantu proses administrasi, serta memberikan arahan selama masa perkuliahan hingga penyusunan skripsi ini.
8. Seluruh Dosen, Laboran dan jajaran Staf Program Studi Teknik Informatika yang telah memberikan banyak bantuan selama studi ini.

9. Dunia bagi penulis yaitu mama Elvi Tiana tercinta yang telah menjadi sosok penyabar, kuat dan semangat bagi penulis. Dukungan serta doa yang selalu diberikan untuk penulis sehingga penulis mampu menyelesaikan skripsi ini. Doanya lah yang hebat sehingga mempermudah jalan penulis mulai dari awal semester hingga akhir perkuliahan dan menjadi alasan terbesar penulis untuk terus bertahan dan berjuang.
10. Ayah Cahyo Andono, setiap ajaran, nasihat, doa, dan kasih sayang Ayah senantiasa mengiringi langkah penulis dalam menjalani kehidupan. Dukungan dan keteladanan Ayah menjadi sumber kekuatan bagi penulis untuk terus berusaha, bertumbuh, dan menyelesaikan skripsi ini dengan sebaik-baiknya.
11. Abang tercinta Krisna Aditya Putra, Kakak tersayang Dika Ramadhanti dan adik tersayang Najwa Dinda Syahira, yang selalu menjadi bagian penting dalam perjalanan hidup penulis. Terima kasih atas doa yang tak pernah terputus, semangat yang selalu menguatkan, serta kehadiran kalian yang menjadi tempat pulang dan sumber ketenangan di setiap lelah. Semoga langkah kita senantiasa diberi kelancaran, terus bertumbuh bersama, saling menjaga, dan dipertemukan dengan kesuksesan secepatnya.
12. Keluarga besar penulis, khususnya dari pihak keluarga Ayah, Alm. Bapak Handono dan Almh. Mama Sri suradjining (kakek dan nenek), serta Bapak Itik dan Nenek Suhayri (kakek dan nenek), yang tidak dapat disebutkan satu per satu. Penulis mengucapkan terima kasih atas doa, perhatian, dan dukungan tulus yang senantiasa diberikan. Dukungan tersebut menjadi

sumber kekuatan bagi penulis dalam menjalani proses penyusunan skripsi hingga dapat diselesaikan tepat waktu.

13. Teman-teman terdekat penulis sejak masa SMA yang tergabung dalam grup WhatsApp “*Bude Yamaha*”, yaitu Ghea, Wawa, dan Dito. Terima kasih atas kebersamaan yang masih menemani penulis hingga akhir kuliah ini atas bantuan semangat maupun materi. walaupun kadang persahabatan ini hampir runtuh tapi terima kasih untuk masih saling memperbaiki, kehadiran kalian tak akan pernah penulis lupakan karna kalian sangat berarti bagi penulis.
14. Desi Adrianti Awaliyah, terima kasih telah menjadi saksi bisu perjalanan panjang penulis sejak bangku SMP hingga detik ini. Sosok yang memahami tanpa perlu banyak kata, yang hadir dalam setiap fase kehidupan penulis baik di saat suka maupun duka, mau jauh dimanapun kita selalu mengingat satu sama lain, selalu ada tempat kembali untuk kita berdua, Terima kasih aci selalu kebersamai penulis sampai sekarang.
15. Ken Ayu Sintha Monica, merupakan sosok yang tak pernah sangka-sangka penulis temukan di dunia rantau ini, ken yang merupakan kawan kamar di mahad bahkan kawan sekasur, yang merupakan sosok baik, lucu dan orang yang sangat penulis syukuri bisa temui di dunia. Terima kasih ken atas semua bantuan mulai dari hal kecil dan besar mulai dari mahad sampai sekarang menjadi teman kos penulis, jasamu tak akan pernah penulis lupakan.

16. Teman-teman satu kamar penulis di ma'had (Firda, Sakhiya, Khusnul, Cindy, Hida, Laily, Syifa, Rachel), yang pernah menjadi rumah dalam kebersamaan di mahad. Kebersamaan kalian menghadirkan kenangan yang akan selalu dikenang dan menjadi bagian dari perjalanan hidup penulis. Terima Kasih atas dukungan dan ada di saat sidang penulis terutama untuk Khusnul dan juga Sakhiya dan tak lupa juga untuk Meise kawan ken yang sudah hadir dan mensupport penulis.
17. Kak Cantika Andista Putri atau Kak Meycan, sosok yang selalu ada ketika penulis mulai merasa bingung dan kehilangan arah. Dari obrolan ringan sampai cerita hidup, Terima kasih atas kesediaan mendengar tanpa menghakimi dan atas kata-kata sederhana yang sering kali justru paling menenangkan dan berarti bagi penulis.
18. Qonita Nashifa Anabila, yang penulis anggap sebagai saudara di perantauan. Kamu hadir di waktu yang tepat, membawa cahaya ketika penulis berada di titik paling redup. Kebaikanmu terasa melalui hal-hal sederhana yang kamu lakukan tanpa perlu banyak kata. Terima kasih telah menemani hari-hari yang berat dan membuat penulis merasa tidak sendirian dalam menjalaninya di dunia perantauan ini.
19. Aisyah Nur Fitriyah, sosok yang dengan caranya sendiri selalu peduli pada sekitar. Kebaikanmu tidak pernah berisik, tapi terasa dari perhatian kecil, kesediaan mendengar, hingga caramu hadir saat orang lain membutuhkan. Bersamamu, penulis belajar bahwa kepedulian yang tulus mampu menenangkan dan membuat proses panjang ini terasa lebih ringan.

20. Firna, kawan senasib dalam fase *life after break up* di tengah perjalanan skripsi ini. Terima kasih telah berjalan berdampingan dalam luka yang sama-sama sedang dipelajari walaupun sama-sama tersesat. Dari obrolan larut, tawa kecil disela lelah mengerjakan skripsi, hingga saling menguatkan tanpa perlu banyak penjelasan, kebersamaan kita menjadi pengingat bahwa patah tidak selalu berarti sendirian. Terima kasih telah menemani proses ini dengan sabar dan penuh keikhlasan.
21. Radifan Roihanul Fiqri, terima kasih karena selalu hadir di waktu yang penulis butuhkan. Kebaikan dan pengorbananmu sering datang tanpa banyak kata, lewat tindakan sederhana yang justru sangat tepat waktu. Di saat keadaan tidak mudah, terima kasih sudah mengembalikan semangat penulis. Kehadiranmu menjadi pengingat bagi penulis bahwa hidup harus tetap berlanjut mau bagaimanapun dunia terus membuatmu ingin berhenti.
22. Moh. Musa Al Kadzim, terima kasih atas keceriaan yang selalu kamu bawa, bahkan ketika kamu sendiri sedang memikul banyak beban. Tawamu, candamu, dan caramu tetap menyebarkan lelucon menjadi cahaya kecil yang menghangatkan sekitar. Bersamamu, penulis belajar bahwa kekuatan juga bisa hadir dalam bentuk senyum dan tawa.
23. Teman-teman seperjuangan dengan dosen pembimbing yang sama, yaitu Atsila, Anissa, Yoza, Gavrila, Kila, dan Iim, yang telah berjalan bersama penulis dalam setiap proses bimbingan, saling memberikan dukungan, serta menghadirkan suasana kebersamaan di laboratorium. Kehadiran kalian

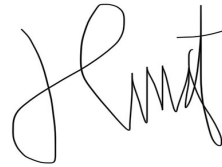
menjadikan perjalanan panjang dalam menyelesaikan skripsi ini terasa lebih ringan karna berjuang bersama-sama.

24. Teman-teman “Infinity” Teknik Informatika angkatan 2022 yang telah kebersamai penulis sepanjang masa studi, menghadirkan kebersamaan, dukungan, dan semangat, serta meninggalkan kenangan berharga yang akan selalu dikenang dalam perjalanan akademik ini.
25. Seluruh pihak yang telah memberikan bantuan, perhatian, dan kebaikan dalam berbagai bentuk selama proses penyusunan skripsi ini, yang tidak dapat disebutkan satu per satu, penulis menyampaikan terima kasih yang sebesar-besarnya.
26. Untuk diri sendiri, terima kasih sudah bertahan sejauh ini. Terima kasih untuk setiap air mata yang jatuh di malam-malam sepi, untuk setiap kali bangkit meski rasanya ingin menyerah, untuk setiap keraguan yang akhirnya berubah jadi langkah kecil hingga akhirnya kamu sadari hal-hal sakit lah yang membuat mu lebih kuat dari yang kamu kira. Perjalanan ini berat, tapi kamu membuktikan bahwa kamu bisa bukan karena sempurna, tapi karena terus mencoba dan doa orang tua saya yang tak pernah terputus.
Long story short, You survived Dita, you can and you can do it better, always

Penulis sadar bahwa skripsi ini masih sangat jauh dari kata sempurna dan mungkin terdapat kesalahan di dalamnya. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk mengembangkan skripsi ini agar lebih bermanfaat bagi dirinya dan pembaca pada umumnya.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 22 Desember 2025

A handwritten signature in black ink, consisting of a large, stylized initial 'R' followed by a series of loops and a vertical line at the end.

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGANTAR.....	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN.....	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	ix
DAFTAR ISI	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xx
ABSTRAK.....	xxi
ABSTRACT.....	xxii
مستخلص البحث.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah	6
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
BAB II STUDI PUSTAKA.....	8
2.1 Sistem Klasifikasi	8
2.2 Anemia.....	13
2.3 <i>Artificial Neural Network</i>	16
2.4 SMOTE.....	18
BAB III DESAIN DAN IMPLEMENTASI SISTEM.....	20
3.1 Pengumpulan Data.....	20
3.2 Desain Sistem	22
3.3 Normalisasi Data.....	24
3.4 Desain <i>Artificial Neural Network</i>	26
3.4.1 <i>Forward Propagation</i>	28
3.4.2 <i>Loss Function</i>	30
3.4.3 <i>Backward Propagation</i>	31
3.5 Implementasi <i>Artificial Neural Network</i>	34
BAB IV UJI COBA DAN PEMBAHASAN.....	37
4.1 Skenario Pengujian	37
4.1.1 SMOTE.....	38
4.1.2 <i>Split Data</i>	40
4.1.3 Menghitung Kinerja Sistem.....	40
4.1.3.1 <i>K-Fold Cross Validation</i>	42
4.2 Hasil Pengujian.....	43
4.2.1 Hasil Training Skenario Pengujian.....	44
4.2.2 Hasil Evaluasi Model.....	47
4.2.2.1 Model 1	47
4.2.2.2 Model 2	50
4.2.2.3 Model 3	52
4.2.2.4 Model 4	54

4.2.2.5 Model 5	56
4.2.2.6 Model 6	59
4.2.2.7 Model 7	61
4.2.2.8 Model 8	63
4.3 Pembahasan	65
4.3.1 Rangkuman Hasil Evaluasi Model	66
4.3.2 Analisis Pengaruh <i>Hyperparameter</i>	68
4.3.2.1 Pengaruh <i>Balancing data</i>	68
4.3.2.2 Pengaruh <i>Neuron</i> pada <i>Hidden Layer</i>	71
4.3.2.3 Pengaruh <i>Learning Rate</i>	74
BAB V KESIMPULAN DAN SARAN.....	79
5. 1 Kesimpulan.....	79
5. 2 Saran	80
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 Desain model <i>Artificial Neural Network</i>	16
Gambar 2. 2 Model Dasar Neuron	17
Gambar 3. 1 Desain Sistem.....	23
Gambar 3. 2 Arsitektur ANN.....	27
Gambar 3. 3 Alur Proses Training	35
Gambar 4. 1 <i>Source Code Split Data</i>	40
Gambar 4. 2 Ilustrasi <i>5-Fold Cross Validation</i>	43
Gambar 4. 3 Grafik <i>Training & Validation Accuracy</i> Model 1.....	44
Gambar 4. 4 Grafik <i>Training & Validation Accuracy</i> Model 2.....	44
Gambar 4. 5 Grafik <i>Training & Validation Accuracy</i> Model 3.....	45
Gambar 4. 6 Grafik <i>Training & Validation Accuracy</i> Model 4.....	45
Gambar 4. 7 Grafik <i>Training & Validation Accuracy</i> Model 5.....	45
Gambar 4. 8 Grafik <i>Training & Validation Accuracy</i> Model 6.....	45
Gambar 4. 9 Grafik <i>Training & Validation Accuracy</i> Model 7.....	45
Gambar 4. 10 Grafik <i>Training & Validation Accuracy</i> Model 8.....	45
Gambar 4. 11 Boxplot Performa Model ANN (5-Fold) Model 1	50
Gambar 4. 12 Boxplot Performa Model ANN (5-Fold) Model 2	52
Gambar 4. 13 Boxplot Performa Model ANN (5-Fold) Model 3	54
Gambar 4. 14 Boxplot Performa Model ANN (5-Fold) Model 4	56
Gambar 4. 15 Boxplot Performa Model ANN (5-Fold) Model 5	59
Gambar 4. 16 Boxplot Performa Model ANN (5-Fold) Model 6	61
Gambar 4. 17 Boxplot Performa Model ANN (5-Fold) Model 7	63
Gambar 4. 18 Boxplot Performa Model ANN (5-Fold) Model 8	65
Gambar 4. 19 Box Plot Perbandingan Metrik Evaluasi Model ANN pada Data SMOTE dan Non-SMOTE	70
Gambar 4. 20 Box Plot Perbandingan Metrik Evaluasi Model ANN pada jumlah neuron (16-32 dan 32-64)	73
Gambar 4. 21 Box Plot Perbandingan Metrik Evaluasi Model ANN pada perbandingan <i>Learning Rate</i> 0,01 dan 0,001	75

DAFTAR TABEL

Tabel 2. 1 Penelitian Penelitian Terkait Klasifikasi Penyakit.....	11
Tabel 3. 1 Fitur Dataset	20
Tabel 3. 2 Klasifikasi Kelas Diagnosis Anemia.....	21
Tabel 3. 3 Dataset Complete Blood Count (CBC).....	22
Tabel 4. 1 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 1	48
Tabel 4. 2 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 1	49
Tabel 4. 3 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 2	50
Tabel 4. 4 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 2.....	51
Tabel 4. 5 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 3	53
Tabel 4. 6 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 3.....	53
Tabel 4. 7 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 4	55
Tabel 4. 8 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 4.....	56
Tabel 4. 9 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 5	57
Tabel 4. 10 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 5.....	58
Tabel 4. 11 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 6	60
Tabel 4. 12 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 6.....	60
Tabel 4. 13 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 7	62
Tabel 4. 14 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 7.....	62
Tabel 4. 15 Rata-Rata Performa Klasifikasi per Kelas <i>5-Fold Cross-Validation</i> Model 8	64
Tabel 4. 16 Hasil Evaluasi <i>5-Fold Cross-Validation</i> Model 8.....	65
Tabel 4. 17 Hasil dari Pengujian Metode ANN <i>Backpropagation</i>	66
Tabel 4. 18 Hasil Evaluasi <i>Confusion Matrix</i> berdasarkan preprocessing menggunakan SMOTE dan Non-SMOTE	69
Tabel 4. 19 Hasil Evaluasi <i>Confusion Matrix</i> berdasarkan 2 <i>hidden layer</i> , neuron 16-32	71
Tabel 4. 20 Hasil Evaluasi <i>Confusion Matrix</i> berdasarkan 2 <i>hidden layer</i> , neuron 32-64	71
Tabel 4. 21 Hasil Evaluasi <i>Confusion Matrix</i> berdasarkan <i>learning rate</i> 0.01 ..	74
Tabel 4. 22 Hasil Evaluasi <i>Confusion Matrix</i> berdasarkan <i>learning rate</i> 0.001	75

ABSTRAK

Yofana, Dita Suci. 2025. **Klasifikasi Multiclass Status Anemia Menggunakan Metode Artificial Neural Network (ANN) Berbasis Parameter Complete Blood Count (CBC)**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Okta Qomaruddin Aziz, M.Kom (II) Dr. Cahyo Crysdiyan, M.Cs.

Kata kunci: *Artificial Neural Network, multiclass classification, anemia, Complete Blood Count, SMOTE.*

Anemia adalah kondisi kesehatan yang ditandai dengan rendahnya kadar hemoglobin dalam darah sehingga mengganggu kemampuan darah dalam mengangkut oksigen ke seluruh tubuh. Diagnosis anemia dilakukan melalui pemeriksaan *Complete Blood Count* (CBC), namun proses interpretasinya masih bergantung pada analisis manual yang berpotensi menimbulkan kesalahan, terutama pada data berskala besar dengan karakteristik antar kelas yang saling tumpang tindih. Penelitian ini bertujuan untuk mengklasifikasikan status anemia secara *multiclass* berdasarkan parameter CBC menggunakan metode *Artificial Neural Network* (ANN). Dataset yang digunakan berasal dari Kaggle *Anemia Types Classification* yang terdiri dari 1.281 data dan 14 atribut CBC. Data diproses melalui tahapan *preprocessing*, normalisasi Min–Max, serta penyeimbangan data menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Model ANN diuji melalui delapan skenario dengan variasi jumlah neuron, nilai *learning rate*, serta penggunaan SMOTE dan Non-SMOTE. Evaluasi performa dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, serta divalidasi menggunakan metode *5-fold cross validation*. Hasil terbaik menunjukkan *accuracy* sebesar 83%, *precision* 79%, *recall* 81%, dan *F1-score* 79%.

ABSTRACT

Yofana, Dita Suci. 2025. **Multiclass Classification of Anemia Status Using the Artificial Neural Network (ANN) Method Based on Complete Blood Count (CBC) Parameters**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (I) Okta Qomaruddin Aziz, M.Kom. (II) Dr. Cahyo Crysdian, M.Cs.

Keywords: *Artificial Neural Network, multiclass classification, anemia, Complete Blood Count, SMOTE.*

Anemia is a health condition characterized by low hemoglobin levels in the blood, which reduces the ability of blood to transport oxygen throughout the body. Anemia diagnosis is commonly performed using Complete Blood Count (CBC) examination; however, the interpretation process still largely relies on manual analysis, which may lead to errors, especially in large-scale data with overlapping class characteristics. This study aims to classify anemia status in a multiclass setting based on CBC parameters using the Artificial Neural Network (ANN) method. The dataset used was obtained from the Kaggle Anemia Types Classification dataset, consisting of 1,281 samples with 14 CBC attributes. Data were processed through preprocessing, Min–Max normalization, and data balancing using the Synthetic Minority Oversampling Technique (SMOTE). The ANN model was evaluated under eight experimental scenarios with variations in the number of neurons, learning rate, and the use of SMOTE and Non-SMOTE. Model performance was assessed using accuracy, precision, recall, and F1-score, and validated using 5-fold cross validation. The best results achieved an accuracy of 83%, precision of 79%, recall of 81%, and F1-score of 79%.

مستخلص البحث

يوفانا، ديتا سوشي. ٢٠٢٥. التصنيف متعدد الفئات لفقر الدم باستخدام الشبكات العصبية الاصطناعية (ANN) بناءً على معايير تعداد الدم الكامل (CBC) أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرفون أوكتا قمر الدين عزيز املاجستري، امشرف الثاين : دكتور جتاهيو جتريس داين املاجستر

الكلمات المفتاحية : الشبكات العصبية الاصطناعية، التصنيف متعدد الفئات، فقر الدم، تعداد الدم الكامل، SMOTE

فقر الدم هو مشكلة صحية تتطلب تشخيصًا مبكرًا دقيقًا. يتم تحديد أنواع فقر الدم بشكل عام من خلال اختبارات تعداد الدم الكامل (CBC)، ولكن عملية التفسير لا تزال تعتمد بشكل كبير على التحليل اليدوي، مما قد يؤدي إلى حدوث أخطاء، خاصة في البيانات واسعة النطاق التي تتداخل خصائصها بين الفئات. تهدف هذه الدراسة إلى تصنيف حالات فقر الدم متعددة الفئات بناءً على معايير تعداد الدم الكامل باستخدام طريقة الشبكة العصبية الاصطناعية (ANN). تم استخدام مجموعة البيانات من تصنيف أنواع فقر الدم Kaggle Anemia Types Classification التي تضم ما مجموعه 1281 نقطة بيانات و 14 سمة. CBC. تمت معالجة البيانات من خلال المعالجة المسبقة، والتطبيع Min-Max، وموازنة البيانات باستخدام تقني (SMOTE). تم اختبار نموذج ANN من خلال ثمانية سيناريوهات مع اختلافات في عدد الخلايا العصبية ومعدل التعلم واستخدام SMOTE و Non-SMOTE. تم إجراء التقييم باستخدام مقاييس الدقة والإحكام والاسترجاع و F1-score، بالإضافة إلى التحقق المتقاطع الخماسي. أظهرت أفضل النتائج دقة بنسبة 83٪ وإحكام بنسبة 79٪ واسترجاع بنسبة 81٪ و F1-score بنسبة 79٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Anemia merupakan kondisi medis yang terjadi ketika tubuh kekurangan sel darah merah yang sehat atau kadar hemoglobin yang cukup untuk mengangkut oksigen ke seluruh jaringan tubuh. Kondisi ini dapat menimbulkan gejala seperti mudah lelah, sesak napas, dan wajah pucat yang dapat menurunkan produktivitas individu. Berdasarkan laporan terbaru dari *World Health Organization* (WHO) tahun 2025, anemia tetap menjadi salah satu permasalahan kesehatan masyarakat yang krusial. Jutaan wanita usia subur dan anak-anak di bawah lima tahun dilaporkan berada dalam kondisi berisiko. WHO juga menegaskan bahwa dunia saat ini belum berada pada jalur yang sesuai untuk mencapai target penurunan prevalensi anemia global sebesar 50% pada tahun 2030.

Di Indonesia, prevalensi anemia juga menunjukkan tren yang mengkhawatirkan. Berdasarkan data Survei Kesehatan Indonesia (SKI), sekitar satu dari lima anak mengalami anemia, dan pada beberapa wilayah seperti DKI Jakarta, prevalensinya bahkan mencapai 40%. Tingginya prevalensi ini umumnya disebabkan oleh pola konsumsi makanan yang tidak sehat serta kurangnya edukasi gizi, terutama pada keluarga yang rentan secara sosial ekonomi (Nafilah Sri Sagita K., 2025). Hal ini menunjukkan bahwa anemia merupakan masalah kesehatan yang kompleks, yang tidak hanya berkaitan dengan aspek medis, tetapi juga aspek sosial dan pendidikan masyarakat. Oleh karena itu, pencegahan sejak dini menjadi

langkah penting untuk menghindari terjadinya anemia, khususnya melalui perbaikan pola makan dan peningkatan kesadaran gizi pada masyarakat.

Upaya menjaga kesehatan tubuh, termasuk pencegahan penyakit, telah diisyaratkan dalam Al-Qur'an. QS. Al-Baqarah ayat 168 menegaskan perintah untuk mengonsumsi makanan halal dan baik (*thayyib*):

يَا أَيُّهَا النَّاسُ كُلُوا مِمَّا فِي الْأَرْضِ حَلَالًا طَيِّبًا وَلَا تَتَّبِعُوا خُطُوَاتِ الشَّيْطَانِ إِنَّهُ لَكُمْ عَدُوٌّ مُبِينٌ

“Wahai manusia, makanlah sebagian (makanan) di bumi yang halal lagi baik dan janganlah mengikuti langkah-langkah setan. Sesungguhnya ia bagimu merupakan musuh yang nyata.” (QS. Al-Baqarah : 168).

Ayat ini menekankan perintah Allah agar manusia mengonsumsi makanan halal dan *thayyib* sebagai kunci menjaga kesehatan jasmani dan rohani. Halal berkaitan dengan hukum agama, sedangkan *thayyib* menekankan kualitas dan manfaat makanan. Larangan mengikuti langkah setan dipahami sebagai ajakan menjauhi pola hidup merusak, seperti makanan haram, berlebihan, atau tidak seimbang. Jika dikaitkan dengan anemia, ayat ini mendorong kita untuk menjaga pola makan bergizi seimbang dengan mengonsumsi sayuran hijau, kacang-kacangan, serta daging yang kaya zat besi, karena kekurangan zat gizi tersebut dapat menurunkan hemoglobin dan menyebabkan anemia. Oleh karena itu, menjaga nutrisi merupakan bentuk ikhtiar sekaligus tanggung jawab dalam merawat amanah tubuh yang diberikan oleh Allah SWT.

Nilai-nilai kesehatan juga telah tersirat dalam surah QS. An-Nahl ayat 69 mengenai salah satu ciptaan-Nya yang mengandung obat bagi manusia:

ثُمَّ كُلِّي مِنْ كُلِّ الثَّمَرَاتِ فَاسْلُكِي سُبُلَ رَبِّكِ ذُلُلًا يَخْرُجُ مِنْ بَطُونِهَا شَرَابٌ مُخْتَلِفٌ أَلْوَانُهُ ۚ فِيهِ شِفَاءٌ لِلنَّاسِ إِنَّ فِي ذَلِكَ لَآيَةً لِّقَوْمٍ يَتَفَكَّرُونَ

“Kemudian, makanlah (wahai lebah) dari segala (macam) buah-buahan lalu tempuhlah jalan-jalan Tuhanmu yang telah dimudahkan (bagimu). Dari perutnya itu keluar minuman (madu) yang beraneka warnanya. Di dalamnya terdapat obat bagi manusia. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda (kebesaran Allah) bagi kaum yang berpikir.” (QS. An-Nahl: 69).

Ayat ini menguatkan bahwa selain menjaga pola makan sehat, manusia juga diberikan berbagai bahan alam sebagai obat penyembuhan penyakit, yang sangat relevan untuk upaya pencegahan dan pengobatan anemia melalui sumber nutrisi alami, seperti madu, sayur mayur, dan bahan pangan bergizi tinggi lainnya. Dengan demikian, menjaga asupan nutrisi yang baik merupakan wujud tanggung jawab manusia dalam merawat amanah tubuh yang dianugerahkan oleh Allah SWT.

Diagnosis anemia secara klinis umumnya dilakukan melalui pemeriksaan darah lengkap *Complete Blood Count* (CBC), yaitu tes standar yang mengukur berbagai parameter darah utama seperti kadar hemoglobin, distribusi ukuran sel darah merah (RDW), volume rata-rata sel darah merah (MCV), dan jumlah sel darah merah (RBC). Pemeriksaan ini dapat memberikan gambaran tentang status anemia seseorang. Namun demikian, interpretasi hasil CBC memerlukan keahlian medis yang mendalam dan terkadang dipengaruhi oleh subjektivitas tenaga kesehatan, sehingga proses diagnosis manual ini relatif memakan waktu dan berisiko terjadi kesalahan (Choudhary et al., 2020). Oleh karena itu, diperlukan solusi berbasis teknologi yang mampu membantu dalam analisis data medis secara lebih objektif dan efisien.

Seiring berkembangnya teknologi informasi dan machine learning, metode seperti *Artificial Neural Network* (ANN) mulai banyak diterapkan dalam bidang medis untuk membantu proses klasifikasi penyakit, termasuk anemia. ANN memiliki kemampuan untuk memodelkan hubungan yang kompleks dan non-linier dalam data, menjadikannya metode yang efektif dalam memproses dan menganalisis data medis yang bersifat multiclass. Penelitian oleh (Karagül Yıldız et al., 2021) menunjukkan bahwa Artificial Neural Network (ANN) mampu mengklasifikasikan status anemia dengan akurasi tinggi sebesar 79,6% pada dataset original dengan 25 fitur, yang lebih unggul dibandingkan metode lainnya seperti Naïve Bayes (74,0%) dan beberapa varian SVM yang menggunakan dataset dengan jumlah fitur lebih sedikit. Akurasi ini menegaskan potensi besar ANN dalam pengolahan data laboratorium kompleks seperti *Complete Blood Count* (CBC) untuk klasifikasi penyakit secara efektif dan akurat.

Selain itu, studi oleh (Pradana et al., 2022) mengenai klasifikasi penyakit jantung menggunakan ANN juga membuktikan efektivitas metode ini dengan akurasi sebesar 73,77%. Hal ini mengindikasikan bahwa ANN dapat diterapkan secara luas dalam berbagai jenis penyakit, termasuk anemia. Dalam praktiknya, anemia memiliki berbagai jenis yang berbeda, seperti normocytic normochromic anemia, microcytic hypochromic anemia, dan macrocytic anemia (Choudhary et al., 2020). Oleh karena itu, pendekatan klasifikasi *multiclass* menjadi relevan, karena memungkinkan setiap pasien dikategorikan ke dalam salah satu jenis anemia tertentu berdasarkan karakteristik data medis yang dimilikinya.

Penelitian ini akan menerapkan ANN untuk mengklasifikasikan status anemia berdasarkan parameter *Complete Blood Count* (CBC), dengan pendekatan *multiclass* yang memungkinkan setiap pasien dikategorikan ke dalam satu jenis anemia dominan. Model *Artificial Neural Network* (ANN) akan mempelajari hubungan kompleks antar fitur dalam data dan menghasilkan klasifikasi yang akurat untuk berbagai jenis anemia. Untuk memastikan efektivitas model dalam membedakan kelas anemia yang berbeda, evaluasi *multiclass* menjadi sangat penting.

Evaluasi *multiclass* digunakan untuk mengukur apakah model ANN dapat membedakan dengan tepat di antara berbagai kelas anemia dengan akurasi tinggi. Evaluasi yang umum digunakan dalam klasifikasi *multiclass* meliputi akurasi keseluruhan (overall accuracy), precision, *recall*, dan *F1-score* per kelas, serta rata-rata average untuk melihat performa di setiap kelas tanpa bias jumlah sampel (Kibria & Matin, 2022). Tanpa evaluasi *multiclass* yang komprehensif, kita tidak akan mengetahui seberapa baik model bekerja dalam membedakan setiap jenis anemia secara jelas, yang dapat menyebabkan kesalahan diagnosis jika kelas anemia tertentu selalu lebih dominan. Oleh karena itu, evaluasi *multiclass* yang lengkap adalah langkah krusial dalam penelitian ini untuk memastikan bahwa model dapat memberikan prediksi jenis anemia yang tepat dan berguna bagi diagnosis medis anemia.

Berdasarkan latar belakang tersebut, pengembangan model klasifikasi *multiclass* menggunakan *Artificial Neural Network* (ANN) dilakukan untuk mengidentifikasi status anemia secara lebih akurat dan efisien. Diharapkan model

ini dapat berkontribusi dalam meningkatkan kualitas diagnosis medis serta membantu tenaga kesehatan dalam menangani anemia secara lebih komprehensif.

1.2 Pernyataan Masalah

1. Berapa tingkat akurasi, presisi, dan *recall* yang dapat dihasilkan oleh model *Artificial Neural Network* dalam mengklasifikasikan status anemia secara *multiclass* berdasarkan data *Complete Blood Count* (CBC)?
2. Bagaimana pengaruh *Balancing* data pada metode *Artificial Neural Network* dalam klasifikasi *multiclass* penyakit anemia?

1.3 Batasan Masalah

1. Penelitian ini menggunakan dataset publik *Kaggle Anemia Types Classification* (Aboelnaga, 2024), yang terdiri atas 1.281 data hasil *Complete Blood Count* (CBC) dengan empat belas parameter darah, antara lain WBC, LYMp, NEUTp, LYMn, NEUTn, Hb, Hct, RBC, PLT, dan lain-lain.
2. Objek penelitian hanya mencakup data hasil pemeriksaan darah pasien yang terdapat dalam dataset tersebut, tanpa mempertimbangkan faktor klinis lain seperti riwayat penyakit, gaya hidup, atau pemeriksaan tambahan di luar parameter CBC.

1.4 Tujuan Penelitian

1. Mengukur nilai presisi, akurasi, dan *recall* dari metode *Artificial Neural Network* dalam klasifikasi penyakit *multiclass* anemia.

2. Mengukur pengaruh *Balancing* data pada performa metode *Artificial Neural Network* dalam klasifikasi penyakit *multiclass* anemia berdasarkan hasil akurasi.

1.5 Manfaat Penelitian

Manfaat pada penelitian ini adalah sebagai berikut:

1. Bagi Institusi Kesehatan, sistem yang dikembangkan dalam penelitian ini dapat diterapkan untuk mempercepat proses identifikasi anemia di laboratorium, sekaligus mendukung digitalisasi layanan pemeriksaan darah.
2. Bagi Tenaga Medis, penelitian ini dapat menjadi alat bantu dalam proses identifikasi dan klasifikasi anemia pasien, terutama untuk mempercepat diagnosis awal dan meminimalkan kesalahan analisis.
3. Bagi Individu atau Penderita Anemia, penelitian ini dapat mendorong kesadaran pentingnya pemeriksaan darah secara rutin untuk mencegah dan mengenali anemia sejak dini.

BAB II

STUDI PUSTAKA

2.1 Sistem Klasifikasi

Sistem klasifikasi penyakit adalah cara untuk mengelompokkan berbagai penyakit dan prosedur medis yang mirip ke dalam kelompok-kelompok dengan kode numerik tertentu, termasuk pengelompokan tindakan medis sejenis, sehingga mendukung pengkodean diagnosis dan prosedur medis secara akurat dan terstandarisasi berdasarkan The International Statistical Classification of Diseases and Related Health Problem Tenth Revisions (ICD-11) (Adawiyah *et al.*, 2023). Dalam bidang ilmu komputer, klasifikasi penyakit umumnya dilakukan dengan metode machine learning atau deep learning yang mampu mengenali pola kompleks dari data pasien untuk memprediksi jenis penyakit tertentu. Metode data mining yang digunakan dalam penelitian ini adalah klasifikasi, yaitu metode yang berfokus pada pengklasifikasian kelas target secara akurat. Tujuan dari metode ini adalah untuk mengklasifikasikan nilai suatu variabel yang tidak diketahui berdasarkan variabel lain yang telah ditentukan sebelumnya (Riany & Testiana, 2023).

Dalam penelitian yang dilakukan oleh Pradana *et al.* (2022) tentang penerapan metode *Artificial Neural Network* (ANN) untuk mengklasifikasikan penyakit jantung. Penelitian ini menggunakan 304 data pasien penyakit jantung yang dikelola oleh Bi Developer at Vision B.I. dan diperoleh dari dataset *Kaggle* dengan tujuan untuk mengukur performa metode ANN. Penelitian ini menunjukkan bahwa hasil akurasi dari metode ANN sebesar 73,77%, presisi sebesar 80,43%,

recall sebesar 84,09%, dan *f1-score* sebesar 82,22%. Dari hasil penelitian ini terlihat bahwa metode ANN cukup efektif digunakan dalam klasifikasi penyakit jantung.

Penelitian yang telah dilakukan Mardanus (2024) membandingkan performa algoritma SVM dan ANN dalam memprediksi risiko serangan jantung. Data yang digunakan berasal dari *Kaggle* tahun 2021 dengan jumlah 10 fitur. Setelah dilakukan proses *preprocessing* dan pembagian data, model diuji menggunakan metrik evaluasi akurasi, presisi, *recall*, dan *F1-score*. Hasil penelitian menunjukkan bahwa SVM hanya mampu mencapai akurasi 61%, sedangkan ANN memperoleh akurasi yang lebih tinggi yaitu 84%. Perbedaan utama terletak pada performa algoritma, di mana ANN terbukti lebih unggul dibandingkan dengan SVM dalam mengklasifikasikan risiko serangan jantung.

Berdasarkan penelitian Mohammad Soharto *et al.* (2024), pengembangan algoritma Artificial Neural Network (ANN) mampu mengklasifikasikan penyakit jantung dengan akurasi mencapai 70%. Dataset yang digunakan berupa 303 data medis pasien penyakit jantung yang diperoleh dari pusat dataset *Kaggle*, dengan atribut yang relevan terkait kondisi medis pasien. Penelitian ini bertujuan untuk memperbaiki hasil penelitian sebelumnya yang hanya memperoleh akurasi sebesar 73,77%, presisi 80,43%, *recall* 84,09%, dan *f-measure* 82,22%. Perbaikan dilakukan melalui penyesuaian parameter serta mengurangi jumlah epoch. Hasil penelitian menunjukkan bahwa algoritma *Artificial Neural Network* (ANN) berhasil mencapai akurasi sebesar 97,06%, presisi 92,30%, *recall* 92,30%, dan *f-*

measure 92,30%, sehingga membuktikan adanya peningkatan performa ANN dalam klasifikasi penyakit jantung.

Hasil penelitian yang dilakukan oleh Afrah (2023) berfokus pada penerapan metode ANN untuk mendeteksi penyakit liver menggunakan dataset Indian Liver Patient Dataset (ILPD) yang diambil dari UCI Machine Learning Repository yang berisi 11 fitur. Penelitian ini bertujuan untuk mengetahui keakuratan *model Artificial Neural Network* (ANN) dalam melakukan klasifikasi penyakit liver. Dari hasil penelitian, model ANN menghasilkan akurasi sebesar 71,79%, dengan nilai true negatif 70, true positif 14, false negatif 16, dan false positif 17. Hasil tersebut menunjukkan bahwa metode ANN cukup baik dalam melakukan klasifikasi pada dataset ILPD, meskipun masih terdapat tingkat error sebesar 21,37% sehingga diperlukan pengembangan lebih lanjut.

Penelitian yang diteliti oleh Purwono *et al.* (2022) mengembangkan model prediksi otomatis untuk klasifikasi jenis penyakit hipertensi menggunakan algoritma ANN. Dataset yang digunakan berasal dari *PPG-BP (photoplethysmography-blood pressure)* yang dikumpulkan dari 219 orang dewasa dengan rentang usia 21–86 tahun dan terdiri dari 8 fitur. Penelitian ini bertujuan untuk membangun model klasifikasi *Artificial Neural Network* (ANN) dalam memprediksi jenis penyakit hipertensi. Dari hasil penelitian, implementasi ANN menghasilkan akurasi prediksi sebesar 85%, yang menunjukkan bahwa model ini cukup efektif digunakan dalam proses klasifikasi jenis hipertensi.

Penelitian yang disampaikan oleh Niendy Alexandra Yosephine & Ratnadewi (2021) menerapkan metode ANN pada sinyal elektrokardiogram untuk

mendeteksi penyakit jantung aritmia supraventrikular. Dataset yang digunakan dalam penelitian ini berasal dari MIT-BIH Arrhythmia Supraventrikular Database (Physionet). Dari hasil penelitian, struktur algoritma *Multi-Layer Perceptron* (MLP) memberikan performa terbaik. Metode pelatihan Resilient Backpropagation menghasilkan nilai akurasi tertinggi sebesar 87,5%, dengan sensitivity 100%, sedangkan metode pelatihan Levenberg Marquard memberikan specificity tertinggi sebesar 83,3%. Hal ini menunjukkan bahwa pemanfaatan ANN mampu mendukung klasifikasi penyakit jantung aritmia supraventrikular secara efektif.

Sebagian besar penelitian terdahulu yang menggunakan metode *Artificial Neural Network (ANN)* lebih banyak diterapkan pada bidang medis lain, seperti klasifikasi penyakit jantung, diabetes, atau kanker, dan belum ada yang secara spesifik meneliti kasus anemia. Selain itu, penelitian sebelumnya umumnya tidak memanfaatkan data *Complete Blood Count (CBC)* sebagai dasar klasifikasi. Oleh karena itu, penelitian ini berupaya mengisi kekosongan tersebut dengan mengimplementasikan metode ANN untuk klasifikasi multiclass status anemia berbasis parameter CBC, sehingga menghasilkan pendekatan baru yang lebih spesifik dan berpotensi meningkatkan akurasi dalam mendukung diagnosis anemia.

Berdasarkan sejumlah studi yang telah dibahas sebelumnya, penulis merangkum hasil temuan tersebut dalam bentuk perbandingan yang terdapat pada Tabel 2.1.

Tabel 2. 1 Penelitian Penelitian Terkait Klasifikasi Penyakit

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan Penelitian
(Pradana <i>et al.</i> , 2022)	Penerapan metode Artificial Neural Network (ANN) untuk	Artificial Neural Network (ANN)	Model ANN yang digunakan berhasil mencapai akurasi 73,77%, dengan	Penelitian ini berfokus pada penggunaan data CBC untuk

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan Penelitian
	mengklasifikasikan penyakit jantung		presisi 80,43%, recall 84,09%, dan f1-score 82,22%, sehingga dapat dikatakan cukup efektif dalam klasifikasi penyakit jantung.	klasifikasi status anemia secara <i>multiclass</i> .
(Mardanus, 2024)	Klasifikasi Data Serangan Jantung Menggunakan Metode Super Vector Machine dan Artificial Neural Network	SVM, ANN	Perbandingan dua metode menunjukkan bahwa SVM memperoleh akurasi 61%, sedangkan ANN lebih unggul dengan akurasi 84%. Hasil ini menegaskan bahwa ANN memiliki performa lebih baik dalam memprediksi risiko serangan jantung	Dalam penelitian ini diterapkan metode ANN dengan berbagai variasi parameter untuk memperoleh konfigurasi terbaik.
(Mohammad Soharto <i>et al.</i> , 2024)	Klasifikasi Penyakit Jantung Menggunakan Algoritma ANN dengan Persentase diatas 70%	Artificial Neural Network (ANN)	Penelitian ini melaporkan hasil yang sangat baik, dengan akurasi 97,06%, presisi 92,30%, recall 92,30%, dan f1-score 92,30%. Hal ini membuktikan bahwa ANN sangat efektif dalam klasifikasi penyakit jantung.	Penelitian ini memanfaatkan dataset CBC sebagai dasar klasifikasi kategori status anemia.
(Afrah, 2023)	Sistem Diagnosa Penyakit Liver Menggunakan Metode Artificial Neural Network: Studi Berdasarkan Dataset Indian Liver Patient Dataset	Artificial Neural Network (ANN)	Kinerja ANN pada klasifikasi penyakit liver mencapai akurasi 71,79%. TN 70, TP 14, FN 16, FP 17, error 21,37%	Model ANN yang dikembangkan dalam penelitian ini berbasis pada parameter CBC.
(Purwono <i>et al.</i> , 2022)	Model Prediksi Otomatis Jenis Penyakit Hipertensi dengan Pemanfaatan Algoritma Machine Learning Artificial Neural Network	Artificial Neural Network (ANN)	Dengan akurasi sebesar 85%, ANN terbukti mampu memberikan hasil yang baik dalam mengklasifikasikan jenis hipertensi pada pasien.	Data laboratorium darah (CBC) digunakan sebagai sumber utama dalam klasifikasi status anemia.

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan Penelitian
(Niendy Alexandra Yosephine & Ratnadewi, 2021)	Penggunaan Artificial Neural Network pada Sinyal Elektrokardiogram untuk Mendeteksi Penyakit Jantung Aritmia Supraventrikular	Artificial Neural Network (ANN), MLP	Penggunaan ANN dengan struktur Multi-Layer Perceptron menghasilkan performa tinggi, yakni akurasi 87,5%, sensitivity 100%, dan specificity 83,3%,	Parameter numerik CBC pada penelitian ini diolah menggunakan teknik SMOTE untuk menyeimbangkan data.

Pada Tabel 2.1, penulis menyajikan rangkuman penelitian sebelumnya yang memanfaatkan algoritma *Artificial Neural Network* (ANN) dalam klasifikasi penyakit. Tabel ini memuat informasi mengenai judul penelitian, peneliti, metode yang digunakan, serta hasil yang diperoleh. Pada tabel tersebut ditampilkan sejumlah penelitian yang menerapkan ANN dan algoritma pembanding lainnya, seperti *Support Vector Machine* (SVM), untuk mendeteksi dan mengklasifikasikan berbagai jenis penyakit, di antaranya jantung, serangan jantung, liver, hipertensi, dan aritmia supraventrikular. Informasi yang tercantum dalam tabel ini memberikan gambaran mengenai efektivitas metode yang digunakan di bidang kesehatan, beserta tingkat akurasi, presisi, *recall*, dan *f1-score* yang dicapai oleh masing-masing penelitian.

2.2 Anemia

Anemia adalah keadaan yang ditandai dengan berkurangnya hemoglobin dalam tubuh. Hemoglobin adalah suatu metaloprotein yaitu protein yang mengandung zat besi di dalam sel darah merah yang berfungsi sebagai pengangkut oksigen dari paru-paru ke seluruh tubuh (Astuti & Kulsum, 2020). Pada dasarnya,

anemia dipengaruhi secara langsung oleh konsumsi makanan sehari-hari yang kurang mengandung zat besi. Secara umum, konsumsi makanan berkaitan erat dengan status gizi. Bila makanan yang dikonsumsi mempunyai nilai yang baik, maka status gizi juga baik, sebaliknya bila makanan yang dikonsumsi kurang nilai gizinya, maka akan menyebabkan kekurangan gizi dan dapat menimbulkan anemia (Nasruddin & Syamsu, 2021). Gejala anemia yang sering dijumpai adalah lesu, letih, lemah, lelah, lalai atau disingkat 5L (Kemenkes RI, 2018). Kaki dan tangan penderita terasa dingin sepanjang waktu sehingga mereka tidak melakukan pekerjaan rutin dan malas sepanjang waktu. Gejala-gejala ini muncul sebagai akibat berkurangnya zat besi dalam tubuh. Zat besi dibutuhkan tubuh untuk membentuk hemoglobin, molekul pengikat oksigen di sel darah merah (Danarsih *et al.*, 2024).

Anemia merupakan masalah kesehatan masyarakat global yang serius, khususnya menyerang anak kecil, remaja putri dan perempuan yang sedang menstruasi, serta perempuan hamil dan pascapersalinan. WHO memperkirakan bahwa 40% anak usia 6–59 bulan, 37% perempuan hamil, dan 30% perempuan usia 15–49 tahun di seluruh dunia mengalami anemia. Menurut Laporan Penelitian Kesehatan Dasar Indonesia dari tahun 2007 hingga 2018, dan Laporan Survei Kesehatan Nasional pada tahun 2023, prevalensi anemia telah mencapai tingkat yang cukup tinggi. Dari tahun 2007 hingga 2018, prevalensi anemia menunjukkan peningkatan dari 19,7% menjadi 38,5% pada anak di bawah lima tahun, dari 19,7% menjadi 27,2% pada wanita (≥ 15 tahun), dan dari 13,1% menjadi 20,3% pada pria (≥ 15 tahun). Laporan terbaru pada tahun 2023 menunjukkan prevalensi anemia

sebesar 23,8%, 18,0%, dan 14,4% pada anak-anak, wanita dewasa, dan pria dewasa, masing-masing (Lidwina Priliani *et al.*, 2025).

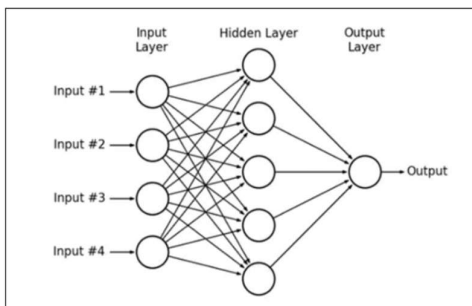
Anemia dapat menurunkan kinerja fisik, menghambat perkembangan tubuh, melemahkan daya tahan tubuh, dan bahkan berdampak jangka panjang pada kesehatan wanita usia subur. Penyebab anemia cukup beragam, antara lain kekurangan vitamin A, vitamin B12, asam folat, dan zat besi, serta kondisi lain seperti peradangan kronis, infeksi parasit, atau penyakit keturunan. Namun, penyebab yang paling sering terjadi di seluruh dunia adalah anemia defisiensi besi (IDA), yaitu kondisi ketika tubuh kekurangan zat besi yang diperlukan untuk membentuk sel darah merah. Kekurangan ini sangat berbahaya, terutama pada ibu hamil, karena dapat meningkatkan risiko komplikasi seperti bayi lahir dengan berat badan rendah (BBLR), kelahiran prematur, hingga kematian bayi baru lahir (Permata *et al.*, 2023).

Pencegahan anemia dapat dilakukan melalui berbagai strategi, antara lain pemberian suplemen zat besi dan asam folat, edukasi gizi untuk meningkatkan pola makan seimbang, serta penggunaan kontrasepsi oral yang mengandung zat besi bagi wanita postpartum. Selain itu, pemeriksaan kesehatan secara rutin melalui layanan antenatal care penting untuk deteksi dini anemia pada ibu hamil, disertai skrining zat besi dan konseling gizi untuk mencegah komplikasi lebih lanjut. Upaya pencegahan juga mencakup perbaikan sanitasi air, diversifikasi dan fortifikasi makanan, serta tata laksana diet sehat dengan konsumsi makanan kaya zat besi. Pada kasus anemia akibat penyakit kronis seperti malaria, HIV-AIDS, atau thalasemia, diperlukan perawatan khusus seperti terapi ARV, transfusi darah, dan

pemberian suplemen asam folat untuk menjaga kadar hemoglobin tetap stabil (Pamela *et al.*, 2022).

2.3 *Artificial Neural Network*

Artificial Neural Network (ANN) atau Jaringan Saraf Tiruan merupakan model statistik nonlinier yang terinspirasi dari sistem saraf otak manusia. ANN digunakan untuk merepresentasikan hubungan kompleks antara input dan output guna mengidentifikasi pola dalam data. Struktur ANN digambarkan menyerupai graf yang terdiri atas node-node, seperti terlihat pada Gambar 2.1. Terdapat tiga komponen utama dalam ANN, yaitu arsitektur jaringan yang menggambarkan pola hubungan antar node, metode pelatihan yang digunakan untuk menentukan bobot penghubung, serta fungsi aktivasi yang berperan dalam menghasilkan output (Niendy Alexandra Yosephine & Ratnadewi, 2021).

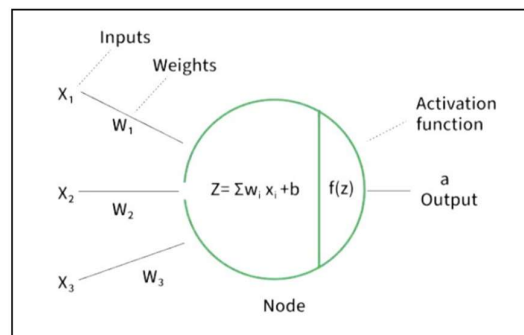


Gambar 2. 1 Desain model Artificial Neural Network

(Sumber: <https://medium.com/machine-learning-researcher/artificial-neural-network-ann-4481fa33d85a>)

Dalam neural network, neuron-neuron disusun dalam suatu struktur yang terdiri dari tiga lapisan yaitu *input layer*, *hidden layer*, dan *output layer* (Meilina *et al.*, 2021).

1. *Input Layer* adalah lapisan tempat data awal dimasukkan ke dalam jaringan, dan dari sini proses-proses pengolahan data lainnya dimulai.
2. *Hidden Layer* berperan mendukung pemrosesan data, semakin banyak jumlah *hidden layer*, hasil output yang diinginkan dapat dicapai lebih baik dan lebih cepat, meskipun waktu pelatihan jaringan akan menjadi lebih lama.
3. *Output Layer* adalah lapisan yang bertugas menampung hasil akhir dari proses yang dilakukan oleh jaringan syaraf tiruan.



Gambar 2. 2 Model Dasar Neuron

(Sumber: <https://medium.com/@suthetmilan/why-do-we-use-activation-functions-cant-we-just-stack-layers-cacc5a53f5ee>)

Artificial Neural Network memproses informasi secara bertahap melalui jaringan neuron yang saling berhubungan, sehingga sistem mampu mengenali pola kompleks dari data masukan. Seperti terlihat pada Gambar 2.2, Hubungan antar lapisan, khususnya antara *hidden layer* dan lapisan lainnya, dibentuk melalui bobot, bias, serta fungsi transfer. Untuk mengukur tingkat kesalahan (*error*), digunakan perbandingan antara keluaran jaringan dan target. Selanjutnya, pengurangan *error* dilakukan dengan metode optimasi melalui penyesuaian bobot dan bias. Proses pelatihan (*training*) berlangsung berulang kali dalam sejumlah *epoch* hingga

mencapai tingkat akurasi yang sesuai dengan tujuan. Setelah tahap pelatihan selesai, bobot dan bias yang telah diperoleh kemudian digunakan dalam proses validasi data pada jaringan (Hananti & Sari, 2021).

Artificial Neural Network dimulai dengan memasukkan data fitur pelatihan ke dalam model. Data dikirimkan melalui *input layer*, di mana tiap neuron input mewakili satu fitur. Dari situ, data dialirkan ke satu atau lebih *hidden layer*. Hidden layer sendiri terdiri dari neuron – neuron yang fungsinya menangkap pola-pola kompleks dan hubungan non-linear antar fitur dengan menggunakan fungsi aktivasi seperti ReLU. Setelah melewati hidden layer, output diteruskan ke *output layer*, yang memiliki neuron sebanyak jumlah kelas output. Fungsi aktivasi Softmax digunakan di output layer agar prediksi menjadi probabilitas untuk setiap kelas, dan kelas dengan probabilitas tertinggi dipilih sebagai output (Mustakim *et al.*, 2021).

2.4 SMOTE

Metode *Synthetic Minority Oversampling Technique* (SMOTE) merupakan metode dasar untuk mengatasi masalah data yang tidak seimbang (Hairani *et al.*, 2024). SMOTE bekerja dengan metode *k-nearest neighbor* untuk membuat sampel baru secara sintetik di dalam ruang fitur berdasarkan persentase tertentu untuk kelas minoritas (Swana *et al.*, 2022).

SMOTE adalah teknik yang menciptakan data sintetis baru dengan memanfaatkan data yang sudah ada pada kelas minoritas, tanpa menggandakan data tersebut, sehingga dapat menghindari masalah overfitting. Prosesnya dimulai dengan memilih secara berurutan setiap sampel dari kelas minoritas sebagai titik dasar untuk membuat sampel sintetis tambahan. Proses sintesis ini diulang

sebanyak n kali, dan sampel sintetis baru dihasilkan melalui interpolasi linier antara sampel dasar dan sampel tambahan tersebut (Hermaliani & Ernawati, 2024). Menurut (Sulistiyono et al., 2021), rumus *Euclidean* dapat dilihat pada persamaan 2.1.

$$dist = \sqrt{(X_1 - Y_1)^2 + \dots + (X_n - Y_n)^2} \quad (2.1)$$

Keterangan:

$dist$: Nilai jarak Euclidean antara dua data.
 X_i : Nilai fitur ke- i dari data pertama.
 Y_i : Nilai fitur ke- i dari data kedua (pembanding).
 n : Jumlah fitur (dimensi) yang digunakan dalam perhitungan jarak.

Misalkan atribut dengan data 2, maka jarak *euclidean distance* seperti dalam persamaan 2.2 yaitu:

$$dist = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2} \quad (2.2)$$

Setelah menghitung jarak antar instance menggunakan *Euclidean Distance*, dibuat data replika dari instance yang paling dekat dengan menggunakan persamaan 2.3.

$$X_{syn} = X_i + (X_{knn} - X_i) \times \sigma \quad (2.3)$$

Keterangan:

X_{syn} : Data sintetis (data baru) yang dihasilkan.
 X_i : Sampel asli dari kelas minoritas.
 X_{knn} : Salah satu tetangga terdekat (k-nearest neighbor) dari X_i .
 σ : Nilai acak antara 0 dan 1 yang menentukan posisi titik sintetis di antara X_i dan X_{knn} .

X_{syn} adalah data sintesis yang dihasilkan dari proses replikasi, X_i adalah data ke- i dalam kelas minoritas, X_{knn} merupakan data dari kelas minoritas yang jaraknya paling dekat dengan data X_i , dan σ adalah bilangan acak yang bernilai antara 0 dan 1.

BAB III

DESAIN DAN IMPLEMENTASI SISTEM

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari sumber terbuka di situs *Kaggle* yang berjudul *Anemia Types Classification* (Aboelnaga, 2024). Dataset tersebut berisi hasil pemeriksaan *Complete Blood Count* (CBC) yang digunakan untuk proses klasifikasi status anemia. Dataset ini terdiri atas 1.281 baris data dan 14 kolom fitur yang seluruhnya berupa data numerik bertipe float, serta satu kolom kategori yaitu Diagnosis sebagai label kelas anemia.

Sebelum dilakukan pengolahan data, terlebih dahulu disajikan penjelasan mengenai masing-masing fitur pada dataset. Tabel 3.1 memuat definisi setiap atribut yang digunakan, meliputi nama fitur dan keterangannya. Dataset ini kemudian dibagi menjadi data latih (training set) dan data uji (testing set) untuk digunakan dalam proses pembelajaran dan evaluasi model. Contoh potongan data yang digunakan disajikan pada Tabel 3.3, dan diagnosis anemia pada Tabel 3.2 serta definisi lengkap tiap fitur terdapat pada Tabel 3.1.

Tabel 3. 1 Fitur Dataset

No	Parameter	Keterangan
1	White Blood Cell Count (WBC)	Jumlah sel darah putih
2	Lymphocyte Percentage (LYMp)	Persentase limfosit
3	Neutrophil Percentage (NEUTp)	Persentase neutrofil
4	Absolute Lymphocyte Count (LYMn)	Jumlah absolut limfosit
5	Absolute Neutrophil Count (NEUTn)	Jumlah absolut neutrofil
6	Red Blood Cell Count (RBC)	Jumlah sel darah merah
7	Hemoglobin (HGB)	Kadar hemoglobin dalam darah
8	Hematocrit (HCT)	Persentase hematokrit
9	Mean Corpuscular Volume (MCV)	Volume rata-rata sel darah merah
10	Mean Corpuscular Hemoglobin (MCH)	Rata-rata hemoglobin per sel darah merah

No	Parameter	Keterangan
11	Mean Corpuscular Hemoglobin Concentration (MCHC)	Konsentrasi hemoglobin rata-rata
12	Platelet Count (PLT)	Jumlah trombosit dalam darah
13	Platelet Distribution Width (PDW)	Variasi ukuran trombosit
14	Plateletcrit (PCT)	Persentase volume darah yang ditempati trombosit

Tabel 3. 2 Klasifikasi Kelas Diagnosis Anemia

Kelas	Jumlah	Diagnosis
0	336	Healthy
1	189	Iron deficiency anemia
2	47	Leukemia
3	11	Leukemia with thrombocytopenia
4	18	Macrocytic anemia
5	279	Normocytic hypochromic anemia
6	269	Normocytic normochromic anemia
7	59	Other microcytic anemia
8	73	Thrombocytopenia
Total	1.281	-

Klasifikasi diagnosis anemia pada penelitian ini mengacu pada *International Classification of Diseases 11th Revision (ICD-11, 2022)*, yang diterbitkan oleh World Health Organization (WHO). Dalam ICD-11, gangguan darah dan organ pembentuk darah dikategorikan dalam bab *Diseases of the blood or blood-forming organs* dengan rentang kode 3A00–3C0Z. Beberapa contoh kode yang relevan meliputi *iron deficiency anemia* (3A00), *macrocytic anemia* (3A01), *thrombocytopenia* (3B64), dan *leukemia* (2B33.0). Dengan demikian, klasifikasi diagnosis pada penelitian ini disesuaikan dengan standar terminologi medis internasional agar hasil analisis memiliki dasar validitas klinis yang kuat.

Tabel 3.3 menampilkan contoh data yang akan digunakan. Data ini merupakan dataset CBC yang berisi 14 kolom. Kolom-kolom tersebut mencakup hasil tes darah yang mengukur dan mengevaluasi sel-sel darah yang telah dijelaskan pada tabel sebelumnya 3.1.

Tabel 3. 3 Dataset Complete Blood Count (CBC)

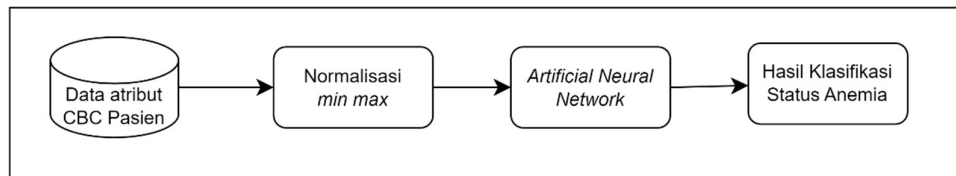
No	1	2	3	4	5	6	7	8	9
	WBC	LYMp	NEUTp	LYMn	NEUTn	RBC	HGB	HCT	MCV
1	10	43.2	50.1	4.3	5	2.77	7.3	24.2	87.7
2	10	42.4	52.3	4.2	5.3	2.84	7.3	25	88.2
3	7.2	30.7	60.7	2.2	4.4	3.97	9	30.5	77
4	6	30.2	63.5	1.8	3.8	4.22	3.8	32.8	77.9
5	4.2	39.1	53.7	1.6	2.3	3.93	0.4	316	80.6
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
1277	4.4	25.845	77.511	1.88076	5.14094	4.86	13.5	46.1526	80.7
1278	5.6	25.845	77.511	1.88076	5.14094	4.85	15	46.1526	91.7
1279	9.2	25.845	77.511	1.88076	5.14094	4.47	13.1	46.1526	88.7
1280	6.48	25.845	77.511	1.88076	5.14094	4.75	13.2	46.1526	86.7
1281	8.8	25.845	77.511	1.88076	5.14094	4.95	15.2	46.1526	89.7

No	10	11	12	13	14	Diagnosis
	MCH	MCHC	PLT	PDW	PCT	
1	26.3	30.1	189	12.5	0.17	Normocytic hypochromic anemia
2	25.7	20.2	180	12.5	0.16	Normocytic hypochromic anemia
3	22.6	29.5	148	14.3	0.14	Iron deficiency anemia
4	23.2	29.8	143	11.3	0.12	Iron deficiency anemia
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
1277	27.7	34.4	180	14.31251	0.26028	Healthy
1278	31	33.8	215	14.31251	0.26028	Healthy
1279	29.3	33	329	14.31251	0.26028	Healthy
1280	27.9	32.1	174	14.31251	0.26028	Healthy
1281	30.6	34.2	279	14.31251	0.26028	Healthy

3.2 Desain Sistem

Desain sistem disusun secara terstruktur untuk memastikan setiap tahapan penelitian dapat berjalan dengan efisien serta menghasilkan identifikasi status anemia yang akurat. Gambar 3.1 berikut menunjukkan rancangan sistem yang

dikembangkan untuk melakukan deteksi anemia menggunakan metode *Artificial Neural Network*.



Gambar 3. 1 Desain Sistem

Sistem pada Gambar 3.1 menggunakan data atribut CBC pasien yang berisi 14 parameter pemeriksaan darah. Setelah data masuk ke dalam sistem, langkah awal yang dilakukan adalah normalisasi *min max*. Pada tahap ini, seluruh parameter CBC dinormalisasi menggunakan metode min-max agar seluruh nilai berada pada rentang yang seragam. Tahap berikutnya adalah implementasi metode *Artificial Neural Network*. Pada proses ini, bobot dan bias diinisialisasi secara acak, kemudian model dilatih menggunakan mekanisme propagasi maju dan propagasi mundur. Pada propagasi maju, data yang telah dinormalisasi diproses melalui jaringan hingga menghasilkan prediksi awal. Selanjutnya, propagasi mundur digunakan untuk menghitung kesalahan dan memperbarui bobot agar model semakin akurat. Proses pelatihan berlangsung berulang hingga nilai kesalahan berada pada tingkat yang dapat diterima. Tahap terakhir adalah menghasilkan klasifikasi status anemia berdasarkan pola yang telah dipelajari oleh model. Output ini berupa prediksi kelas anemia yang sesuai dengan sampel CBC pasien.

3.3 Normalisasi Data

Normalisasi data merupakan proses penyesuaian skala nilai pada setiap atribut agar berada dalam rentang yang seragam dan sesuai untuk proses analisis atau pemodelan data. Salah satu metode transformasi yang umum digunakan adalah normalisasi data, yaitu teknik untuk menskalakan ulang nilai data menjadi rentang yang lebih kecil (Ambarwari *et al.*, 2020). Penerapan normalisasi sangat penting ketika fitur dalam dataset memiliki skala yang berbeda-beda. Tanpa melakukan normalisasi, performa model data mining dapat menurun karena nilai akurasi yang kecil, serta fitur dengan skala kecil akan memiliki kontribusi yang minimal dalam proses analisis berbasis pengukuran jarak. Sebaliknya, hasil pemodelan akan lebih dominan dipengaruhi oleh fitur dengan skala besar, meskipun fitur berskala kecil memiliki peranan yang sama pentingnya (Allorerung *et al.*, 2024). Metode normalisasi yang akan digunakan di penelitian ini adalah Min–Max Normalization.

Normalisasi Min–Max adalah metode pengolahan data yang menggunakan nilai maksimum dan minimum dari suatu atribut untuk mengubah skala data ke rentang 0 sampai 1. Proses ini dilakukan agar setiap atribut memiliki skala yang seimbang, sehingga nilai antar data dapat dibandingkan secara proporsional tanpa ada atribut yang memiliki pengaruh lebih besar dari yang lain. Rumus normalisasi Min–Max ditunjukkan pada Persamaan 3.1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

Keterangan:

- X : nilai asli dari atribut.
- X_{min} : nilai minimum atribut pada dataset.
- X_{max} : nilai maksimum atribut pada dataset.
- X' : nilai hasil normalisasi (setelah dinormalisasi, nilainya akan berada

pada rentang 0 sampai 1).

Tabel 3.4 Contoh data CBC setelah dilakukan normalisasi

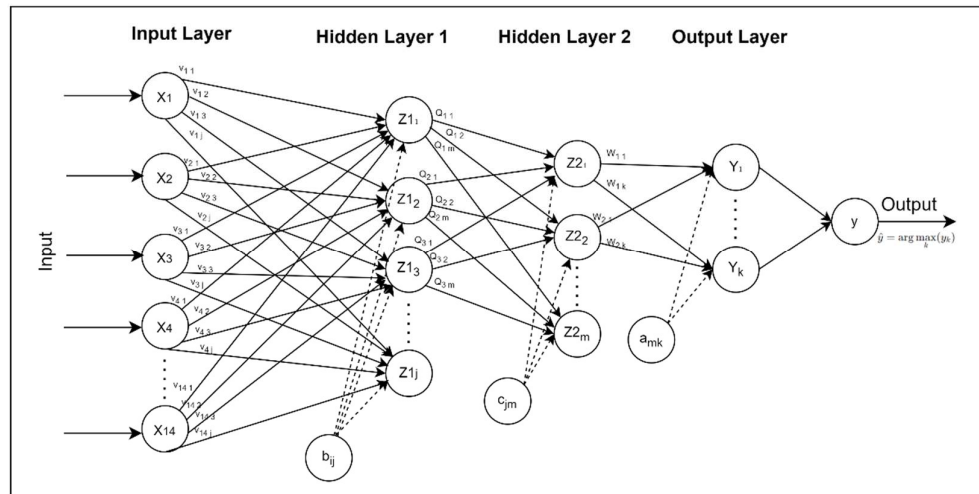
NO	1	2	3	4	5	6	7
	WBC	LYMp	NEUTp	LYMn	NEUTn	RBC	HGB
1	0.2048997 8	0.43427230	0.0092921 8	0.098557 69	0.057324 84	0.015764 76	0.178166 84
2	0.2048997 8	0.42488263	0.0097060 0	0.096153 85	0.061146 50	0.016547 41	0.178166 84
3	0.1425389 8	0.28755869	0.0112860 4	0.048076 92	0.049681 53	0.029181 57	0.195674 56
4	0.1158129 2	0.28169014	0.0118127 3	0.038461 54	0.042038 22	0.031976 74	0.142121 52
5	0.0757238 3	0.38615023	0.0099693 4	0.033653 85	0.022929 94	0.028734 35	0.107106 08
6	0.1291759 5	0.24765258	0.0121701 2	0.038461 54	0.048407 64	0.029069 77	0.193614 83
7	0.3541202 7	0.15140845	0.0126968 0	0.072115 38	0.138853 50	0.042374 78	0.249227 60
8	0.1893095 8	0.24882629	0.0119067 8	0.057692 31	0.068789 81	0.033877 46	0.226570 55
9	0.0979955 5	0.15845070	0.0134868 2	0.019230 77	0.042038 22	0.039020 57	0.238928 94
10	0.2160356 3	0.07276995	0.0147282 9	0.026442 31	0.099363 06	0.035889 98	0.230690 01

NO	8	9	10	11	12	13	14
	HCT	MCV	MCH	MCHC	PLT	PDW	PCT
1	0.0059789 9	0.15617694	0.0049579 9	0.228782 29	0.275384 62	0.046275 40	0.011773 36
2	0.0061944 5	0.15664453	0.0047648 2	0.107011 07	0.261538 46	0.046275 40	0.011037 53
3	0.0076757 3	0.14617039	0.0037667 8	0.221402 21	0.212307 69	0.066591 42	0.009565 86
4	0.0082951 8	0.14701206	0.0039599 5	0.225092 25	0.204615 38	0.032731 38	0.008094 19
5	0.0845677 3	0.14953708	0.0041853 1	0.223862 24	0.347692 31	0.049661 40	0.015452 54
6	0.0074602 7	0.14448705	0.0036380 0	0.837638 38	0.303076 92	0.034988 71	0.012509 20
7	0.0115270 7	0.15561582	0.0053443 2	0.247232 47	0.216923 08	0.049661 40	0.009565 86
8	0.0096687 3	0.15496119	0.0052799 3	0.247232 47	0.283076 92	0.084650 11	0.013245 03
9	0.0105036 4	0.15337136	0.0052477 4	0.253382 53	0.263076 92	0.018058 69	0.010301 69
10	0.0099380 6	0.15393248	0.0052155 4	0.249692 50	0.236923 08	0.032731 38	0.009565 86

Tabel 3.4 merupakan contoh data *Complete Blood Count* (CBC) yang telah melalui proses normalisasi menggunakan metode Min–Max. Setiap atribut, seperti *WBC*, *RBC*, *HGB*, *HCT*, dan parameter lainnya, ditransformasikan ke dalam rentang nilai 0 hingga 1 dengan menggunakan nilai minimum dan maksimum pada masing-masing atribut. Proses normalisasi ini bertujuan untuk menyetarakan skala antar fitur sehingga tidak terdapat atribut yang mendominasi proses pembelajaran model. Dengan skala data yang seragam, model *Artificial Neural Network* dapat mempelajari pola data secara lebih stabil dan optimal, serta meningkatkan efektivitas proses pelatihan dan klasifikasi.

3.4 Desain *Artificial Neural Network*

Artificial Neural Network dirancang meniru cara kerja otak manusia, dengan banyak node yang disebut neuron yang saling terhubung membentuk pola seperti jaring laba-laba. Neuron adalah satuan dasar dalam otak yang berjumlah miliaran, yang berfungsi mengolah informasi. Setiap neuron memiliki badan sel yang menerima dan mengirimkan sinyal informasi ke neuron lain, mirip dengan fungsi input dan output di otak manusia (Dastres & Soori, 2021). Arsitektur jaringan yang digunakan dalam penelitian ini ditampilkan pada Gambar 3.2. Gambar tersebut menunjukkan susunan layer yang digunakan, mulai dari input layer yang menerima seluruh parameter CBC hingga output layer yang menghasilkan prediksi kelas anemia. Desain arsitektur ini menjadi dasar alur pemrosesan data pada model yang dikembangkan dalam penelitian ini.



Gambar 3. 2 Arsitektur ANN

Keterangan:

- x : variabel masukan
- v_{ij} : bobot dari neuron input ke neuron hidden
- b_{ij} : bias pada neuron hidden layer 1
- c_{jm} : bias pada neuron hidden layer 2
- Q_{jm} : bobot dari hidden layer 1 ke hidden layer 2
- w_{mk} : bobot dari neuron hidden ke neuron output
- a_{mk} : bias pada neuron output

Gambar 3.2 menunjukkan arsitektur *Artificial Neural Network* (ANN) yang terdiri atas empat lapisan, yaitu *input layer*, *hidden layer 1*, *hidden layer 2*, dan *output layer*. Pada *input layer*, terdapat 14 neuron yang masing-masing merepresentasikan satu variabel masukan (x) berupa fitur hasil praproses dan normalisasi data. Setiap neuron pada lapisan ini meneruskan nilai input ke lapisan berikutnya melalui bobot penghubung.

Hidden layer 1 menerima sinyal dari *input layer* melalui bobot v_{ij} dan bias b_{ij} . Pada lapisan ini, setiap neuron melakukan proses penjumlahan terbobot antara nilai masukan dan bias, kemudian hasilnya diteruskan ke *hidden layer 2*. Selanjutnya, *hidden layer 2* menerima keluaran dari *hidden layer 1* melalui bobot

Q_{jm} dan bias c_{jm} . Keberadaan dua *hidden layer* ini bertujuan untuk menangkap pola nonlinier yang lebih kompleks pada data, sehingga model mampu mempelajari representasi fitur secara lebih mendalam.

Lapisan terakhir adalah *output layer* yang terdiri dari 9 neuron, di mana setiap neuron merepresentasikan satu kelas status anemia. Neuron pada *output layer* menerima sinyal dari *hidden layer* 2 melalui bobot w_{jk} dan bias a_{jk} . Keluaran akhir jaringan berupa nilai aktivasi pada masing-masing neuron output, kemudian kelas prediksi ditentukan berdasarkan neuron dengan nilai keluaran terbesar menggunakan fungsi *arg max*.

3.4.1 *Forward Propagation*

Dalam *Artificial Neural Network* (ANN), proses pelatihan dimulai dengan forward propagation. Forward propagation merupakan fase pertama dalam algoritma backpropagation di mana sinyal masukan x_i dipropagasikan secara maju dari lapisan input ke lapisan tersembunyi menggunakan fungsi aktivasi, kemudian keluaran lapisan tersembunyi z_j diteruskan lagi ke lapisan berikutnya hingga menghasilkan keluaran jaringan y_k .

Berikut ini akan diuraikan proses forward propagation:

1. Inisialisasi bobot dan bias dilakukan terlebih dahulu sebelum memulai proses feedforward. Bobot pada setiap koneksi antara neuron-neuron di lapisan input, lapisan tersembunyi, dan lapisan output diinisialisasi secara acak, biasanya dengan nilai kecil atau distribusi tertentu.

2. Langkah pertama dalam *feedforward* adalah menghitung Z_{inj} , yaitu nilai total input yang diterima oleh neuron-neuron di lapisan tersembunyi. Nilai ini dihitung dengan menjumlahkan hasil perkalian antara nilai input dan bobot yang sesuai, serta menambahkan bias. Rumus perhitungannya ditunjukkan pada persamaan 3.2.

$$Z_{inj} = b_{ij} + \left(\sum_{i=1}^n X_i V_{i,j} \right) \quad (3.2)$$

Keterangan:

- Z_{inj} : Nilai total input yang diterima oleh neuron pada lapisan tersembunyi.
 b_{ij} : Nilai bias pada neuron ke- j di lapisan tersembunyi.
 $X_{i,j}$: Nilai input dari neuron ke- i pada lapisan sebelumnya.
 $V_{i,j}$: Bobot koneksi antara neuron input ke- i dan neuron tersembunyi ke- j .
 n : Jumlah neuron pada lapisan input.

Setelah menghitung Z_{inj} , nilai tersebut diteruskan ke fungsi aktivasi ReLU.

Mengaktifkan bobot pada lapisan tersembunyi dengan fungsi aktivasi ReLU pada persamaan 3.3.

$$z_j = f(z_{inj}) = \max(0, z_{inj}) \quad (3.3)$$

Keterangan:

- z_j : Keluaran neuron ke- j setelah fungsi aktivasi ReLU.
 Z_{inj} : Penjumlahan berbobot input ke neuron ke- j .
 $\max(0, z_{inj})$: Fungsi aktivasi *ReLU* yang menghasilkan nilai z_{inj} jika bernilai positif, dan 0 jika bernilai nol atau negatif.

Langkah berikutnya adalah menghitung Y_{ink} , yaitu nilai total input yang diterima oleh neuron-neuron di lapisan output. Perhitungan ini melibatkan penjumlahan hasil perkalian antara keluaran neuron di lapisan tersembunyi dan bobot yang menghubungkannya dengan neuron output. Rumusnya ditunjukkan pada persamaan 3.4.

$$y_{ink} = a_{jk} + \left(\sum_{j=1}^m Z_j W_{j,k} \right) \quad (3.4)$$

Keterangan:

- y_{ink} : Nilai input total yang masuk ke neuron pada lapisan output.
- a_{jk} : Nilai bias pada neuron ke- j di lapisan output.
- Z_j : Nilai keluaran dari neuron ke- i pada lapisan tersembunyi.
- W_{jk} : Bobot yang menghubungkan neuron ke- i pada lapisan tersembunyi dengan neuron ke- j pada lapisan output.
- n : Jumlah neuron pada lapisan tersembunyi.

Fungsi softmax diterapkan pada setiap neuron output untuk menghitung probabilitas yang sesuai dengan setiap kelas output. Fungsi softmax mengubah nilai input menjadi rentang antara 0 dan 1, sehingga dapat diinterpretasikan sebagai probabilitas untuk setiap kelas menggunakan persamaan 3.5.

$$y_k = \frac{e^{y_{ink}}}{\sum_{j=1}^C e^{y_{inj}}} \quad (3.5)$$

Keterangan:

- y_k : nilai keluaran (*output*) neuron ke- k pada lapisan output yang merepresentasikan probabilitas kelas ke- k .
- y_{ink} : nilai masukan bersih (*net input*) neuron ke- k pada lapisan output.
- y_{inj} : nilai masukan bersih (*net input*) neuron ke- j pada lapisan output.
- e : bilangan eksponensial (*Euler number*).
- C : jumlah neuron pada lapisan output atau jumlah kelas pada proses klasifikasi.

3.4.2 Loss Function

Loss function adalah fungsi yang digunakan untuk mengukur seberapa baik model dalam memprediksi output yang benar dengan membandingkan hasil prediksi model dan label yang sebenarnya. Dalam klasifikasi multi-kelas, fungsi loss yang paling umum digunakan adalah Categorical Cross-Entropy Loss, yang mengukur seberapa baik probabilitas prediksi model sesuai dengan distribusi probabilitas yang sebenarnya (label asli). Fungsi ini digunakan ketika model

memprediksi kelas dari beberapa kelas yang mungkin, dan setiap data hanya memiliki satu kelas yang benar. Menghitung error menggunakan Categorical Cross-Entropy Loss dapat dilihat pada persamaan 3.6.

$$\mathcal{L} = - \sum_{k=1}^C t_k \log(y_k) \quad (3.6)$$

Keterangan:

C	: Jumlah kelas.
t_k	: Label asli untuk kelas ke-k (dalam one-hot encoding).
$\hat{y}_k$	: Probabilitas prediksi untuk kelas ke-k.
$\log(\hat{y}_k)$	: Logaritma natural dari probabilitas prediksi untuk kelas ke-k.

3.4.3 Backward Propagation

Backward Propagation adalah proses pembelajaran pada *neural network* yang digunakan untuk memperbarui bobot jaringan berdasarkan nilai kesalahan (*error*) yang dihasilkan. Kesalahan tersebut diperoleh dari selisih antara nilai keluaran prediksi dan nilai target yang sebenarnya. Selanjutnya, error ini disebarkan secara bertahap dari *output layer* menuju *input layer* untuk menyesuaikan bobot pada setiap lapisan jaringan. Melalui proses ini, model dapat mempelajari pola kesalahan yang terjadi dan melakukan pembaruan bobot secara berulang, sehingga nilai *loss* semakin berkurang dan performa prediksi model meningkat hingga mencapai kondisi konvergen.

Berikut ini akan diuraikan proses backward propagation:

1. Menghitung nilai *error* pada lapisan output

Setelah proses *feedforward* selesai, langkah pertama dalam *backpropagation* adalah menghitung error pada lapisan output. Error dihitung sebagai selisih antara output yang dihasilkan oleh jaringan dan target yang diinginkan.

Persamaan yang digunakan untuk menghitung error pada neuron output dituliskan pada rumus 3.7.

$$\delta_k = y_k - t_k \quad (3.7)$$

Keterangan:

y_k : output yang dihasilkan oleh neuron k di lapisan output.
 t_k : nilai target atau nilai yang diinginkan untuk kelas output tersebut.
 δ_k : error yang dihitung untuk neuron k di lapisan output.

Selanjutnya δ_k ini digunakan untuk menghitung besarnya koreksi bobot error untuk memperbaiki nilai W_{jk} dengan menggunakan persamaan 3.8.

$$\Delta W_{jk} = \eta \times \delta_k \times Z_j \quad (3.8)$$

Tahap selanjutnya adalah menghitung koreksi bias. Hasilnya kemudian digunakan untuk menyesuaikan nilai a_{mk} berdasarkan persamaan 3.9.

$$\Delta a_{mk} = \eta \times \delta_k \quad (3.9)$$

Setiap unit di lapisan tersembunyi Z_j akan menjumlahkan semua sinyal eror (delta) yang masuk dari lapisan berikutnya, setelah setiap sinyal tersebut dikalikan dengan bobotnya masing-masing menggunakan persamaan 3.10.

$$\delta_{inj} = \sum_{k=1}^c \delta_k W_{jk} \quad (3.10)$$

Hasil penjumlahan dari langkah sebelumnya kemudian dikalikan dengan nilai turunan fungsi aktivasinya. Ini dilakukan untuk mendapatkan nilai akhir dari sinyal eror (delta) untuk unit tersebut dengan menggunakan persamaan 3.11.

$$f'(Z_j) = 1, Z_j \geq 0 \quad (3.11)$$

$$f'(Z_j) = 0, Z_j \leq 0$$

$$\delta_j = \delta_{in_j} f'(Z_j)$$

Delta j digunakan untuk menghitung koreksi error yang nantinya akan digunakan untuk memperbaiki nilai V_{ij} menggunakan persamaan 3.12

$$\Delta V_{ij} = \eta \times \delta_j \times X_i \quad (3.12)$$

Sebagai langkah terakhir dalam satu siklus, nilai penyesuaian untuk bias (b_{ij}) dihitung, yang akan menyelesaikan proses pembaruan untuk lapisan tersembunyi dengan menggunakan persamaan 3.13.

$$\Delta b_{ij} = \eta \delta_j \quad (3.13)$$

2. Pada langkah ini, semua bobot dan bias yang mengarah ke setiap unit di lapisan output akan diperbarui menggunakan persamaan 3.14:

$$W_{jk} (baru) = W_{jk} (lama) + \Delta W_{jk} \quad (3.14)$$

Untuk setiap unit di lapisan tersembunyi (Z_j), semua bobot dan bias yang datang dari lapisan input akan diperbarui nilainya dengan persamaan 3.15.

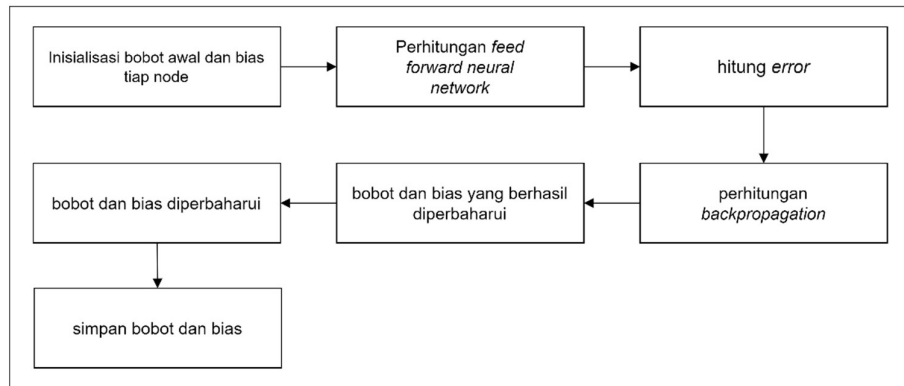
$$V_{ij} (baru) = V_{ij} (lama) + \Delta V_{ij} \quad (3.15)$$

Setelah bobot dan bias diperbarui, dilakukan pengecekan apakah kondisi penghentian (stopping condition) telah tercapai. Jika belum, maka proses perhitungan akan terus diulangi sampai kondisi tersebut terpenuhi. Apabila kondisi sudah tercapai, maka jaringan akan menghasilkan output berupa hasil klasifikasi.

3.5 Implementasi *Artificial Neural Network*

Implementasi algoritma *Artificial Neural Network Backpropagation* untuk klasifikasi multiclass status anemia dirancang dengan arsitektur tiga lapisan: *input*, *hidden*, dan *output*. Untuk mengoptimalkan performa model, pada *hidden layer* memerlukan proses pelatihan dan pengujian yang berulang untuk mengevaluasi berbagai konfigurasi jumlah *hidden layer*. Algoritma *backpropagation* terdiri dari tiga tahap utama: (1) *Feedforward*, dimana sinyal input diproses maju hingga menghasilkan output; (2) *Backward*, yaitu tahap propagasi balik untuk menghitung error dan (3) *Weight Update*, nilai bobot jaringan diperbarui guna meminimalkan error. Evaluasi kinerja model akhir dilakukan menggunakan *Confusion Matrix*.

Tahap awal pada proses pelatihan *Artificial Neural Network* dimulai dengan penentuan nilai awal untuk variabel-variabel yang berperan penting dalam proses perhitungan, seperti laju pembelajaran (*learning rate*) dan jumlah maksimum *epoch*. Selain itu, bobot awal dan *bias* awal jaringan diinisialisasi menggunakan nilai acak dengan magnitudo kecil. Inisialisasi ini bertujuan untuk mencegah terjadinya kondisi jenuh pada fungsi aktivasi serta membantu mempercepat proses konvergensi selama pelatihan model. Ilustrasi alur proses pelatihan pada metode *Artificial Neural Network* ditunjukkan pada Gambar 3.3.



Gambar 3. 3 Alur Proses Training

Setelah proses *feedforward* dilakukan dan nilai keluaran diperoleh, langkah selanjutnya adalah menghitung nilai *error* sebagai selisih antara output yang dihasilkan jaringan dan output yang seharusnya. Nilai *error* tersebut digunakan dalam proses *backpropagation* untuk memperbarui bobot dan *bias* jaringan berdasarkan nilai delta *error* dan *learning rate*. Proses pembaruan bobot ini dilakukan secara iteratif pada setiap lapisan jaringan dengan tujuan meminimalkan nilai *error* secara bertahap. Implementasi mekanisme *backpropagation* dan pembaruan bobot pada algoritma Artificial Neural Network direalisasikan dalam bentuk kode program yang disajikan pada bagian lampiran. Selanjutnya, nilai keluaran pada lapisan output yang direpresentasikan sebagai y_k dihitung menggunakan fungsi aktivasi *softmax* untuk menghasilkan nilai probabilitas setiap kelas hasil klasifikasi. Rumus perhitungan nilai keluaran y_k pada lapisan output dinyatakan pada persamaan 3.5 yaitu:

$$y_k = \frac{e^{y_{in k}}}{\sum_{j=1}^C e^{y_{in j}}}$$

Keterangan:

y_k : nilai keluaran (*output*) neuron ke- k pada lapisan output yang merepresentasikan

- probabilitas kelas ke - k .
 y_{in_k} : nilai masukan bersih (*net input*) neuron ke- k pada lapisan output.
 y_{in_j} : nilai masukan bersih (*net input*) neuron ke- j pada lapisan output.
 e : bilangan eksponensial (*Euler number*).
 C : jumlah neuron pada lapisan output atau jumlah kelas pada proses klasifikasi.

Selanjutnya, hasil klasifikasi ditentukan dengan memilih nilai keluaran y_k terbesar pada lapisan output menggunakan aturan *argmax* sebagai berikut:

$$\hat{y} = \arg \max_k (y_k) \quad (3.1)$$

Keterangan:

- $\hat{y}$: kelas hasil klasifikasi yang dipilih berdasarkan nilai probabilitas tertinggi pada lapisan output.
 y_k : nilai probabilitas keluaran neuron ke- k pada lapisan output.
 k : indeks neuron pada lapisan output.

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Skenario Pengujian

Tahap pengujian dilakukan untuk menilai kemampuan model *Artificial Neural Network* (ANN) dalam mengklasifikasikan tingkat anemia berdasarkan data CBC. Proses pengujian melibatkan beberapa variasi parameter utama, jumlah neuron pada tiap lapisan tersembunyi, nilai *learning rate* serta perbandingan penggunaan data hasil SMOTE dan Non-SMOTE. Seluruh model dilatih dengan mekanisme *early stopping*. Tabel 4.1 menampilkan keseluruhan kombinasi parameter yang diuji, sehingga setiap konfigurasi dapat dibandingkan untuk menentukan model ANN dengan performa terbaik.

4. 1 Tabel Skenario Pengujian

Preprocessing	Model	Hidden Layer	Neuron Hidden Layer	Learning Rate
SMOTE	Skenario 1	2	16–32	0.01
	Skenario 2	2	16–32	0.001
	Skenario 3	2	32–64	0.01
	Skenario 4	2	32–64	0.001
Non-SMOTE	Skenario 5	2	16–32	0.01
	Skenario 6	2	16–32	0.001
	Skenario 7	2	32–64	0.01
	Skenario 8	2	32–64	0.001

Tabel 4.1 merangkum 8 skenario pengujian ANN. Model 1–4 merupakan skenario dengan data SMOTE, sedangkan Model 5-8 merupakan skenario Non-SMOTE. Setiap model menggunakan 2 hidden layer, dengan variasi jumlah neuron 16–32 dan 32–64. Selain itu, setiap skenario diuji menggunakan dua nilai *learning*

rate, yaitu 0,01 dan 0,001 sehingga seluruh konfigurasi memberikan gambaran menyeluruh mengenai pengaruh parameter terhadap performa model ANN.

4.1.1 SMOTE

Pada bagian ini dijelaskan penerapan SMOTE sebagai salah satu skenario pengujian. Analisis awal menunjukkan bahwa distribusi kelas pada dataset tidak seimbang, di mana kelas Healthy memiliki 336 sampel sedangkan beberapa kelas langka seperti *Leukemia with Thrombocytopenia* hanya berjumlah 11 sampel. Ketimpangan ini dapat memengaruhi performa model karena model cenderung lebih mudah mempelajari kelas mayoritas. Oleh karena itu, pada skenario tertentu digunakan SMOTE untuk menambah sampel sintetis pada kelas minoritas sehingga data latih menjadi lebih seimbang. Perubahan distribusi setelah penerapan SMOTE ditunjukkan pada Tabel 4.2.

4. 2 *Synthetic Minority Over-sampling Technique Data*

Diagnosis	Kelas	Jumlah Sebelum SMOTE	Jumlah Sesudah SMOTE
Healthy	0	336	336
Iron deficiency anemia	1	189	336
Leukemia	2	47	336
Leukemia with thrombocytopenia	3	11	336
Macrocytic anemia	4	18	336
Normocytic hypochromic anemia	5	279	336
Normocytic normochromic anemia	6	269	336
Other microcytic anemia	7	59	336
Thrombocytopenia	8	73	336
Total		1.281	3.024

Proses pembuatan data sintetis dilakukan dengan memanfaatkan jarak antar data pada kelas minoritas menggunakan metode *Euclidean Distance*. Misalnya

terdapat dua data minoritas X_i dan Y_i , masing-masing memiliki dua fitur $F1$ dan $F3$ seperti pada contoh berikut:

$$X_i = (7.2, 60.7) \quad Y_i = (6.0, 63.5)$$

Langkah pertama adalah menghitung jarak Euclidean antara kedua data tersebut menggunakan persamaan 2.1:

$$\begin{aligned} dist &= \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2} \\ dist &= \sqrt{(7.2 - 6.0)^2 + (60.7 - 63.5)^2} = \sqrt{1.44 + 7.84} = \sqrt{9.28} \\ &= 3.046 \end{aligned}$$

Nilai $dist(X_i, Y_i)$ menunjukkan jarak antara dua data minoritas yang digunakan untuk menentukan tetangga terdekat (*nearest neighbor*). Setelah tetangga terdekat ditemukan, data sintetis dibentuk menggunakan persamaan 2.3:

$$\begin{aligned} X_{syn} &= X_i + (X_{knn} - X_i) \times \sigma \\ X_{knn} - X_i &= (6.0 - 7.2, 63.5 - 60.7) = (-1.2, 2.8) \\ (X_{knn} - X_i) \times \sigma &= (-1.2, 2.8) \times 0.6 = (-0.72, 1.68) \\ X_{syn} &= (7.2, 60.7) + (-0.72, 1.68) = (6.48, 62.38) \end{aligned}$$

Sehingga diperoleh satu data sintetis baru $X_{syn} = (6.48, 62.38)$ yang berada di antara dua data asli kelas minoritas. Proses ini dilakukan secara berulang pada setiap sampel minoritas hingga jumlah data pada setiap kelas menjadi seimbang. Dengan demikian, metode SMOTE mampu memperbaiki representasi kelas minoritas tanpa menyebabkan *overfitting* akibat duplikasi data

4.1.2 *Split Data*

Setelah penerapan SMOTE, tahap selanjutnya adalah membagi dataset menjadi data latih dan data uji. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk menilai kemampuan generalisasi model. Pada penelitian ini digunakan rasio 80% data latih dan 20% data uji. Dari total 1.281 data, sebanyak 1.024 data menjadi data latih dan 257 data menjadi data uji. Potongan *source code* pembagian data ditampilkan sebagai berikut.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split
(X_resampled, y_resampled, test_size=0.2, random_state=42)
```

Gambar 4. 1 *Source Code Split Data*

Gambar 4.1 memperlihatkan penggunaan fungsi *train_test_split* untuk memisahkan data sesuai proporsi tersebut. Tahap ini memastikan bahwa model memperoleh data yang cukup untuk proses pembelajaran sekaligus mampu dievaluasi secara objektif melalui data uji.

4.1.3 *Menghitung Kinerja Sistem*

Tahap menghitung kinerja sistem dilakukan untuk menilai kemampuan model *Artificial Neural Network* (ANN) dalam mengenali setiap kelas anemia pada data uji. Evaluasi ini memastikan bahwa model tidak hanya bekerja baik pada data latih, tetapi juga mampu melakukan generalisasi terhadap data baru.

Performa model diukur menggunakan *confusion matrix* serta metrik Accuracy, Precision, Recall, dan F1-score. *Confusion matrix* memperlihatkan jumlah prediksi benar dan salah pada tiap kelas berdasarkan empat komponen

utama, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Dari matriks ini, nilai-nilai evaluasi dihitung sebagai berikut:

1. Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Keterangan:

TP (True Positive)	: Prediksi benar pada kelas positif
TN (True Negative)	: Prediksi benar pada kelas negatif
FP (False Positive)	: Prediksi salah, model menganggap data positif padahal negatif
FN (False Negative)	: Prediksi salah, model menganggap data negatif padahal positif

2. Precision

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (4.2)$$

Rata-rata (Macro Average):

$$\text{Precision}_{avg} = \frac{1}{n} \sum_{i=1}^n \text{Precision}_i \quad (4.3)$$

3. Recall

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (4.4)$$

Rata-rata (Macro Average):

$$\text{Recall}_{avg} = \frac{1}{n} \sum_{i=1}^n \text{Recall}_i \quad (4.5)$$

4. F1-score

$$F1_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (4.6)$$

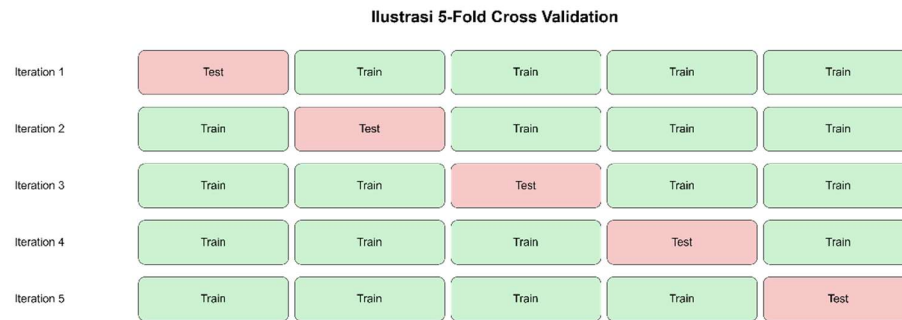
Rata-rata (Macro Average):

$$F1_{avg} = \frac{1}{n} \sum_{i=1}^n F1_i \quad (4.7)$$

Pendekatan *Macro Average* digunakan karena klasifikasi bersifat multiclass, sehingga setiap kelas anemia diberi bobot yang sama tanpa mempertimbangkan perbedaan jumlah sampel. Evaluasi ini memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan sembilan kelas anemia dan menjadi dasar penentuan skenario model dengan performa terbaik.

4.1.3.1 *K-Fold Cross Validation*

Pada penelitian ini juga diterapkan *K-Fold Cross Validation* dengan nilai k 5 untuk menilai konsistensi performa model. Dataset dibagi menjadi lima fold berukuran sama, kemudian proses pelatihan dan validasi dilakukan sebanyak lima kali. Pada setiap iterasi, satu fold digunakan sebagai data validasi sementara empat fold lainnya menjadi data latih. Hasil evaluasi dari kelima iterasi dirata-ratakan sehingga diperoleh gambaran performa model yang lebih stabil dan tidak bergantung pada satu pembagian data tertentu. Ilustrasi pembagian data *K-Fold Cross Validation* seperti gambar berikut.



Gambar 4. 2 Ilustrasi 5-Fold Cross Validation

Gambar 4.2 memperlihatkan alur kerja *5-Fold Cross Validation* dalam proses pelatihan dan pengujian model. Setiap iterasi menunjukkan bahwa satu *fold* digunakan sebagai data uji (*testing set*), sementara empat *fold* lainnya digunakan sebagai data latih (*training set*). Proses ini dilakukan secara bergantian hingga seluruh *fold* memperoleh peran yang sama. Dengan pendekatan ini, seluruh data digunakan baik untuk pelatihan maupun pengujian, sehingga hasil evaluasi model menjadi lebih objektif dan tidak bias terhadap pembagian data tertentu.

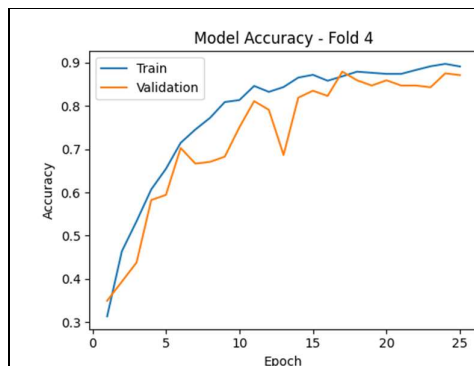
4.2 Hasil Pengujian

Bagian ini menyajikan hasil pengujian model Artificial Neural Network (ANN) berdasarkan 8 skenario konfigurasi hyperparameter yang telah dirancang. Pengujian dilakukan untuk menilai pengaruh variasi jumlah neuron, nilai learning rate, serta perbedaan penggunaan data SMOTE dan Non-SMOTE terhadap proses pembelajaran dan performa klasifikasi. Subbab 4.2.1 memaparkan hasil pelatihan setiap skenario melalui visualisasi training dan validation accuracy, sedangkan Subbab 4.2.2 menyajikan hasil evaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score, serta penilaian setiap metrik dilakukan dengan pendekatan

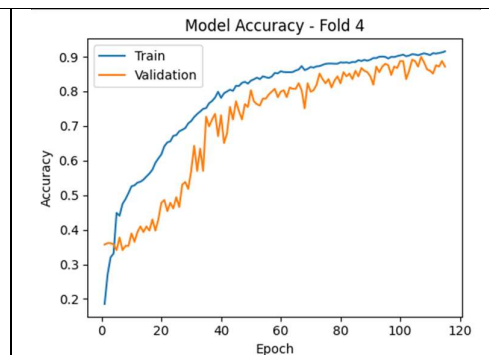
macro average. Kedua subbab tersebut digunakan untuk menentukan konfigurasi ANN yang memberikan performa paling optimal.

4.2.1 Hasil Training Skenario Pengujian

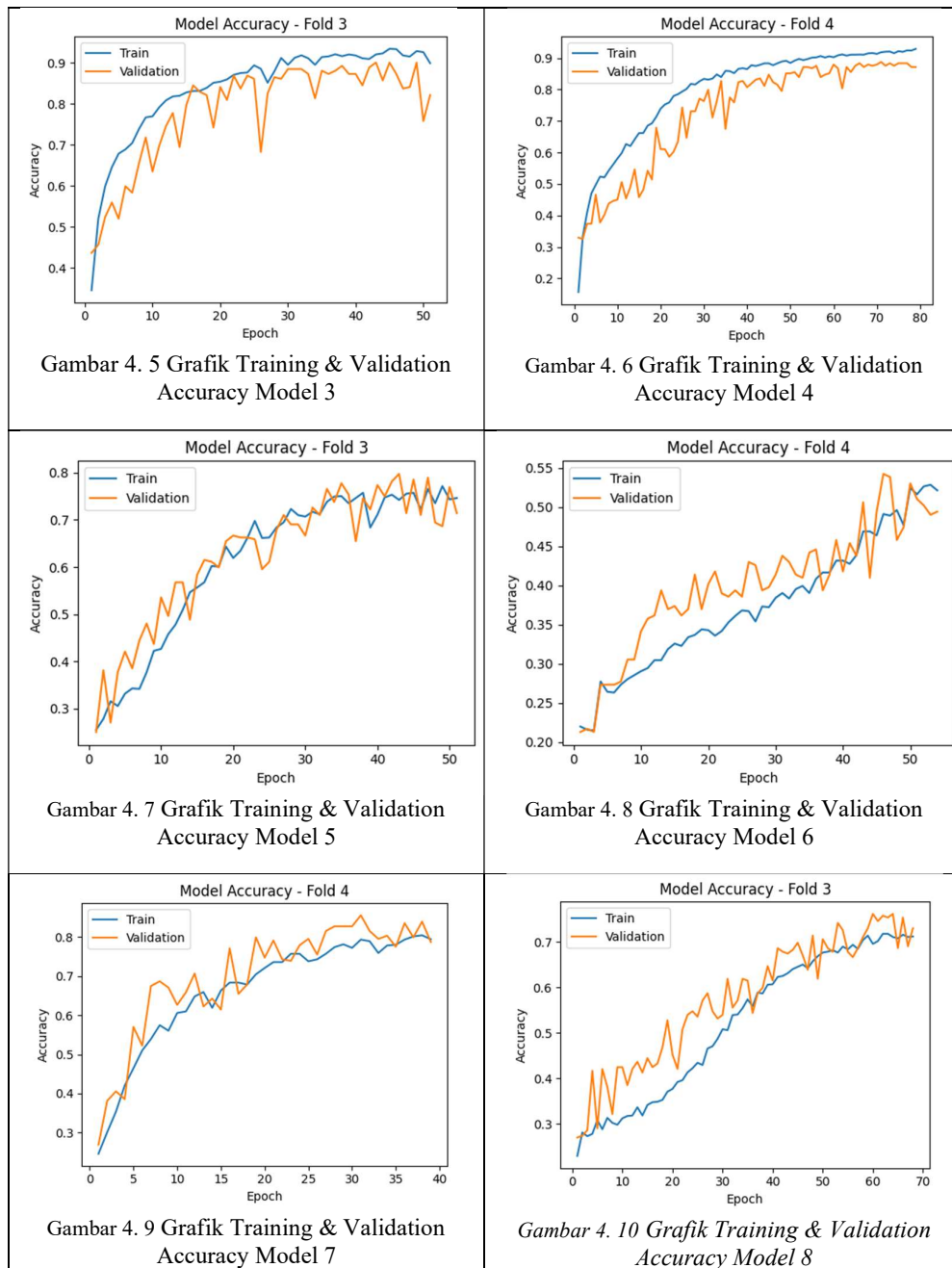
Hasil visualisasi yang merepresentasikan proses pembelajaran pada algoritma *Artificial Neural Network* (ANN) dibahas pada bagian ini. Analisis ini penting untuk menjelaskan bagaimana jaringan saraf melakukan proses pembaruan bobot secara bertahap melalui mekanisme *forward propagation* dan *backpropagation* selama pelatihan. Visualisasi yang ditampilkan berupa grafik *training validation accuracy*, yang memberikan gambaran mengenai stabilitas model, kemampuan generalisasi, serta pola peningkatan kinerja ANN pada setiap epoch selama proses training berlangsung yang dapat dilihat pada gambar sebagai berikut.



Gambar 4. 3 Grafik Training & Validation Accuracy Model 1



Gambar 4. 4 Grafik Training & Validation Accuracy Model 2



Berdasarkan keseluruhan grafik *training* dan *validation accuracy* pada Model 1 hingga Model 8, pola pembelajaran *Artificial Neural Network* (ANN) menunjukkan kecenderungan yang relatif konsisten pada seluruh skenario yang

diuji. Grafik-grafik tersebut merepresentasikan proses pembelajaran model pada salah satu fold dalam skema *K-Fold Cross-Validation*. Setiap model memperlihatkan fase peningkatan akurasi yang cukup tajam pada epoch awal, kemudian diikuti oleh fase perlambatan hingga mencapai kondisi konvergen atau dihentikan lebih awal oleh mekanisme *early stopping*. Pola ini mengindikasikan bahwa proses pembaruan bobot melalui mekanisme *backpropagation* berjalan efektif dalam mempelajari hubungan antarfitur pada data *Complete Blood Count* (CBC).

Sebagian besar model menampilkan jarak (gap) yang relatif kecil antara *training accuracy* dan *validation accuracy*, terutama pada skenario yang menggunakan konfigurasi neuron 16–32 dan 32–64 dengan learning rate 0,01. Kondisi ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik terhadap data validasi pada fold yang diuji dan tidak mengalami *overfitting* yang signifikan. Penerapan *early stopping* membantu menjaga kestabilan proses pelatihan dengan menghentikan iterasi ketika *validation accuracy* tidak lagi mengalami peningkatan yang berarti.

Beberapa model, khususnya yang menggunakan learning rate 0,001, memperlihatkan kurva *validation accuracy* yang lebih berfluktuasi sebelum mencapai kondisi stabil. Fluktuasi ini menyebabkan mekanisme *early stopping* menghentikan proses pelatihan pada epoch tertentu ketika performa validasi cenderung stagnan atau menurun. Kondisi tersebut menunjukkan bahwa laju pembelajaran yang lebih kecil menghasilkan proses penyesuaian bobot yang lebih

lambat dan sensitif terhadap variasi data pada fold tertentu, meskipun model tetap menunjukkan tren peningkatan akurasi secara keseluruhan.

Secara keseluruhan, visualisasi akurasi pada delapan model ini menunjukkan bahwa performa ANN dipengaruhi oleh kombinasi hyperparameter, terutama jumlah neuron dan nilai *learning rate*, serta penerapan *early stopping* sebagai mekanisme pengendalian *overfitting*. Model dengan kapasitas jaringan yang seimbang dan *learning rate* yang sesuai mampu menghasilkan peningkatan akurasi yang lebih stabil hingga proses pelatihan dihentikan secara optimal. Grafik *training* dan *validation accuracy* pada salah satu fold ini digunakan sebagai ilustrasi pembelajaran model, sementara penilaian performa akhir setiap skenario tetap didasarkan pada hasil evaluasi rata-rata seluruh fold yang dibahas pada bagian selanjutnya.

4.2.2 Hasil Evaluasi Model

Proses pembelajaran yang telah divisualisasikan melalui grafik *training validation accuracy* pada setiap skenario memberikan gambaran awal mengenai stabilitas dan pola peningkatan kinerja model. Tahap selanjutnya adalah melakukan evaluasi kinerja untuk menilai efektivitas masing-masing konfigurasi model. Evaluasi ini menggunakan empat metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*, serta dengan pedekatan *average* yang dihitung berdasarkan hasil prediksi model terhadap data uji.

4.2.2.1 Model 1

Berdasarkan Tabel 4.1, Model 1 dengan konfigurasi dua *hidden layer*, neuron 16–32, dan *learning rate* 0,01 menunjukkan performa klasifikasi yang cukup baik pada sebagian besar kelas. Kelas *Healthy* dan *Thrombocytopenia* memperoleh nilai *precision*, *recall*, dan *F1-score* yang tinggi, sedangkan kelas *Iron Deficiency Anemia*, *Normocytic Normochromic Anemia*, dan *Leukemia* menunjukkan performa menengah hingga baik. Sebaliknya, kelas *Macrocytic Anemia* dan *Other Microcytic Anemia* masih memiliki nilai *F1-score* yang lebih rendah, yang mengindikasikan keterbatasan model dalam membedakan pola pada kelas tersebut. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 69%, 76%, dan 71%, dengan nilai standar deviasi yang ditampilkan dalam tanda kurung menunjukkan adanya variasi performa antar *fold* yang masih dalam batas wajar, sehingga Model 1 tetap memiliki kemampuan generalisasi yang cukup baik dalam klasifikasi anemia berbasis parameter *Complete Blood Count* (CBC).

Tabel 4. 1 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 1

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	93%	90%	91%
2	Iron deficiency anemia	78%	83%	80%
3	Leukemia	71%	81%	74%
4	Leukemia with thrombocytopenia	53%	80%	62%
5	Macrocytic anemia	39%	48%	43%
6	Normocytic hypochromic anemia	78%	64%	69%
7	Normocytic normochromic anemia	85%	82%	82%
8	Other microcytic anemia	50%	70%	58%
9	Thrombocytopenia	78%	91%	82%
Rata-rata (Std)		69% (17%)	76% (21%)	71% (17%)

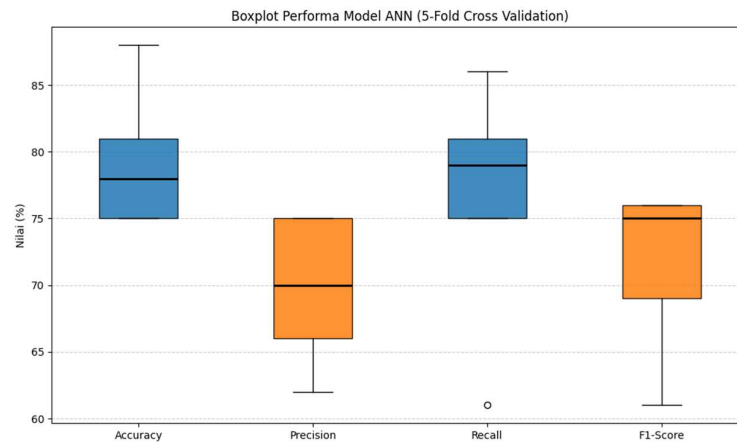
Hasil 5-Fold Cross-Validation pada Tabel 4.2 menunjukkan bahwa performa model relatif konsisten, dengan nilai *accuracy* berada pada kisaran 75%–88%. Fold ke-4 menghasilkan performa tertinggi, sedangkan Fold ke-3

menunjukkan nilai *precision macro*, *recall macro*, dan *F1-score macro* yang lebih rendah akibat perbedaan distribusi data. Nilai rata-rata yang disertai standar deviasi dalam tanda kurung menunjukkan variasi performa antar fold yang masih dalam batas wajar, sehingga model memiliki kemampuan generalisasi yang cukup baik.

Tabel 4. 2 Hasil Evaluasi 5-Fold Cross-Validation Model 1

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	75%	66%	75%	69%
2	78%	70%	86%	75%
3	81%	62%	61%	61%
4	88%	75%	79%	76%
5	75%	75%	81%	76%
Rata-rata (Std)	79% (5%)	70% (5%)	76% (9%)	71% (6%)

Boxplot performa model ANN pada 5-Fold Cross-Validation pada Gambar 4.11, menunjukkan bahwa nilai *accuracy* relatif stabil dengan sebaran yang sempit, menunjukkan konsistensi kinerja model antar fold. Metrik *precision macro* dan *recall macro* memiliki variasi yang lebih besar, yang menandakan adanya pengaruh perbedaan distribusi data pada setiap fold, sedangkan *F1-score macro* menunjukkan sebaran yang lebih seimbang. Secara keseluruhan, boxplot ini menunjukkan bahwa model memiliki kestabilan performa yang cukup baik dan mampu melakukan generalisasi dengan layak.



Gambar 4. 11 Boxplot Performa Model ANN (5-Fold) Model 1

4.2.2.2 Model 2

Berdasarkan Tabel 4.3, Model 2 dengan konfigurasi dua *hidden layer*, neuron 16–32, dan *learning rate* 0,001 menunjukkan performa klasifikasi yang bervariasi antar kelas. Kelas *Healthy*, *Leukemia with Thrombocytopenia*, dan *Thrombocytopenia* memperoleh nilai *recall* yang tinggi, sedangkan kelas *Iron Deficiency Anemia* dan *Normocytic Normochromic Anemia* menunjukkan performa menengah. Sebaliknya, kelas *Macrocytic Anemia* dan *Normocytic Hypochromic Anemia* masih memiliki nilai *F1-score* yang lebih rendah. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 59%, 71%, dan 61%, dengan standar deviasi yang ditampilkan dalam tanda kurung menunjukkan adanya variasi performa antar *fold* yang masih dalam batas wajar, sehingga model tetap memiliki kemampuan generalisasi yang cukup baik dalam klasifikasi anemia berbasis parameter *Complete Blood Count* (CBC).

Tabel 4. 3 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 2

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	83%	77%	77%
2	Iron deficiency anemia	65%	65%	65%

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
3	Leukemia	49%	92%	61%
4	Leukemia with thrombocytopenia	57%	90%	68%
5	Macrocytic anemia	44%	53%	46%
6	Normocytic hypochromic anemia	65%	44%	49%
7	Normocytic normochromic anemia	73%	59%	64%
8	Other microcytic anemia	43%	68%	51%
9	Thrombocytopenia	57%	89%	68%
Rata-rata (Std)		59% (20%)	71% (27%)	61% (21%)

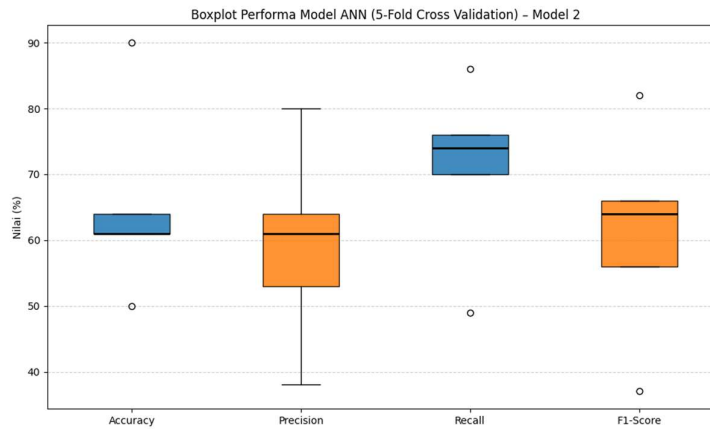
Hasil *5-Fold Cross-Validation* pada Tabel 4.4 menunjukkan bahwa performa model bervariasi antar fold, dengan nilai *accuracy* berada pada rentang 50% hingga 90%. Fold ke-4 menghasilkan performa terbaik, sedangkan Fold ke-3 menunjukkan performa terendah. Nilai *precision macro*, *recall macro*, dan *F1-score macro* juga mengalami fluktuasi, yang mengindikasikan bahwa performa model masih dipengaruhi oleh perbedaan distribusi data pada tiap fold. Meskipun demikian, sebagian besar fold menunjukkan hasil yang relatif sebanding, sehingga model masih memiliki kemampuan generalisasi yang cukup baik.

Tabel 4. 4 Hasil Evaluasi 5-Fold Cross-Validation Model 2

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	61%	53%	70%	56%
2	64%	61%	76%	64%
3	50%	38%	49%	37%
4	90%	80%	86%	82%
5	61%	64%	74%	66%
Rata-rata (Std)	65% (15%)	59% (15%)	71% (13%)	61% (17%)

Boxplot performa model ANN pada 5-Fold Cross-Validation pada Gambar 4.2. *Accuracy* menunjukkan sebaran yang relatif sempit sehingga kinerja model antar *fold* cukup konsisten, sementara *precision macro* memiliki variasi yang lebih besar. *Recall macro* cenderung lebih stabil dengan nilai median yang lebih tinggi, dan *F1-score macro* berada di antara *precision* dan *recall*, yang mencerminkan

keseimbangan kinerja model. Secara keseluruhan, boxplot ini menunjukkan bahwa model memiliki kestabilan dan kemampuan generalisasi yang cukup baik.



Gambar 4. 12 Boxplot Performa Model ANN (5-Fold) Model 2

4.2.2.3 Model 3

Berdasarkan Tabel 4.5, Model 3 dengan konfigurasi dua *hidden layer*, neuron 32–64, dan *learning rate* 0,01 menunjukkan performa klasifikasi yang relatif baik pada sebagian besar kelas. Kelas *Healthy*, *Iron Deficiency Anemia*, *Leukemia with Thrombocytopenia*, dan *Thrombocytopenia* memperoleh nilai *precision*, *recall*, dan *F1-score* yang tinggi, yang menunjukkan kemampuan model dalam mengenali pola pada kelas-kelas tersebut secara konsisten. Kelas *Leukemia* serta *Normocytic Hypochromic Anemia* menunjukkan performa menengah dengan keseimbangan *precision* dan *recall* yang cukup baik. Sebaliknya, kelas *Macrocytic Anemia* dan *Other Microcytic Anemia* masih memiliki nilai *F1-score* yang lebih rendah, yang mengindikasikan bahwa model mengalami keterbatasan dalam membedakan pola pada kelas tersebut. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 79%, 81%, dan 79%,

dengan nilai standar deviasi yang ditampilkan dalam tanda kurung menunjukkan adanya variasi performa antar *fold* yang masih dalam batas wajar, sehingga model memiliki kemampuan generalisasi yang cukup baik dalam klasifikasi anemia berbasis parameter *Complete Blood Count* (CBC).

Tabel 4. 5 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 3

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	95%	87%	90%
2	Iron deficiency anemia	83%	88%	84%
3	Leukemia	74%	83%	76%
4	Leukemia with thrombocytopenia	83%	90%	83%
5	Macrocytic anemia	61%	63%	61%
6	Normocytic hypochromic anemia	78%	81%	79%
7	Normocytic normochromic anemia	90%	79%	83%
8	Other microcytic anemia	64%	73%	68%
9	Thrombocytopenia	84%	86%	84%
Rata-rata (Std)		79% (14%)	81% (17%)	79% (11%)

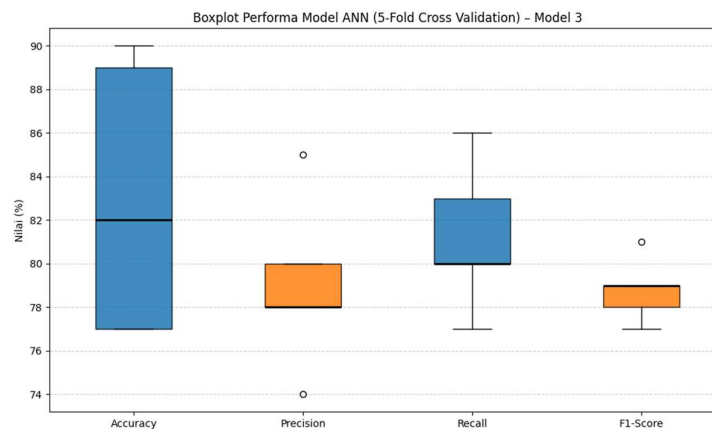
Hasil 5-Fold Cross-Validation pada Tabel 4.6 menunjukkan bahwa performa model relatif konsisten antar *fold*, dengan nilai *accuracy* berada pada kisaran 77%–90%. Fold ke-3 menghasilkan nilai *accuracy* tertinggi, sedangkan Fold ke-2 dan Fold ke-5 menunjukkan nilai *accuracy* yang lebih rendah dibandingkan fold lainnya. Secara umum, nilai *precision macro*, *recall macro*, dan *F1-score macro* pada setiap fold berada pada rentang yang saling berdekatan, yang mengindikasikan kestabilan performa model. Nilai rata-rata yang disertai standar deviasi dalam tanda kurung menunjukkan bahwa variasi performa antar *fold* relatif kecil, sehingga model dapat dikatakan memiliki kemampuan generalisasi yang baik.

Tabel 4. 6 Hasil Evaluasi 5-Fold Cross-Validation Model 3

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	82%	78%	80%	78%
2	77%	74%	86%	77%
3	90%	85%	77%	79%

4	89%	80%	83%	81%
5	77%	78%	80%	79%
Rata-rata (Std)	83% (6%)	79% (4%)	81% (3%)	79% (2%)

Boxplot performa model ANN pada 5-Fold Cross-Validation pada Gambar 4.3. *Accuracy* menunjukkan median yang relatif tinggi dengan sebaran yang masih terkendali, yang menandakan konsistensi kinerja model antar *fold*. Metrik *precision macro* dan *recall macro* memiliki sebaran yang cukup rapat, menunjukkan bahwa performa model dalam memprediksi dan mengenali kelas relatif stabil pada setiap pembagian data. Sementara itu, *F1-score macro* memperlihatkan variasi yang paling kecil, yang mencerminkan keseimbangan kinerja model secara keseluruhan. Secara umum, boxplot ini menunjukkan bahwa model ANN memiliki kestabilan performa dan kemampuan generalisasi yang baik.



Gambar 4. 13 Boxplot Performa Model ANN (5-Fold) Model 3

4.2.2.4 Model 4

Berdasarkan Tabel 4.7, Model 2 dengan konfigurasi dua *hidden layer*, neuron 32–64, dan *learning rate* 0,001 menunjukkan performa klasifikasi yang bervariasi antar kelas. Kelas *Healthy* dan *Thrombocytopenia* memiliki nilai *recall*

yang tinggi, sedangkan kelas *Iron Deficiency Anemia* dan *Normocytic Normochromic Anemia* menunjukkan performa menengah. Sebaliknya, kelas *Macrocytic Anemia*, *Normocytic Hypochromic Anemia*, dan *Other Microcytic Anemia* masih memiliki nilai *F1-score* yang lebih rendah. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 60%, 69%, dan 60%, yang menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik.

Tabel 4. 7 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 4

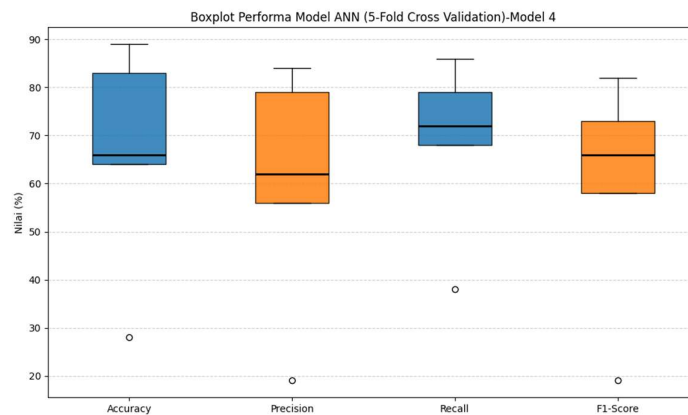
No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	79%	81%	78%
2	Iron deficiency anemia	67%	67%	67%
3	Leukemia	51%	85%	60%
4	Leukemia with thrombocytopenia	56%	83%	59%
5	Macrocytic anemia	42%	50%	37%
6	Normocytic hypochromic anemia	69%	48%	53%
7	Normocytic normochromic anemia	63%	56%	58%
8	Other microcytic anemia	45%	55%	49%
9	Thrombocytopenia	67%	92%	75%
Rata-rata (Std)		60% (29%)	69% (27%)	60% (27%)

Hasil 5-Fold Cross-Validation pada Tabel 4.8 menunjukkan bahwa performa model bervariasi cukup besar antar *fold*, dengan nilai *accuracy* berada pada kisaran 28%–89%. Fold ke-4 menghasilkan performa tertinggi pada seluruh metrik evaluasi, sedangkan Fold ke-5 menunjukkan nilai *accuracy*, *precision macro*, *recall macro*, dan *F1-score macro* yang jauh lebih rendah dibandingkan fold lainnya. Perbedaan ini mencerminkan adanya variasi distribusi data latih dan data uji pada setiap pembagian *fold*. Nilai rata-rata yang disertai standar deviasi dalam tanda kurung menunjukkan bahwa variasi performa antar *fold* cukup besar, namun secara umum model masih mampu menangkap pola data dan menunjukkan kemampuan generalisasi pada sebagian besar fold.

Tabel 4. 8 Hasil Evaluasi 5-Fold Cross-Validation Model 4

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	66%	56%	68%	58%
2	64%	62%	79%	66%
3	83%	84%	72%	73%
4	89%	79%	86%	82%
5	28%	19%	38%	19%
Rata-rata (Std)	66% (24%)	60% (25%)	69% (19%)	60% (24%)

Boxplot performa model ANN pada 5-Fold Cross-Validation pada Gambar 4.4. *Accuracy* dan *precision macro* menunjukkan sebaran yang cukup lebar, sedangkan *recall macro* cenderung memiliki median lebih tinggi. *F1-score macro* berada di antara keduanya, yang menunjukkan keseimbangan kinerja model. Secara umum, boxplot ini memperlihatkan adanya variasi performa antar *fold*, namun model masih mampu mempertahankan kinerja yang cukup baik.



Gambar 4. 14 Boxplot Performa Model ANN (5-Fold) Model 4

4.2.2.5 Model 5

Berdasarkan Tabel 4.9, Skenario 5 (Non-SMOTE) dengan konfigurasi dua *hidden layer*, neuron 16–32, dan *learning rate* 0,01 menunjukkan performa klasifikasi yang bervariasi antar kelas. Kelas *Healthy* dan *Thrombocytopenia* memiliki nilai *recall* yang tinggi, masing-masing sebesar 91%, yang menunjukkan

kemampuan model dalam mengenali kelas tersebut dengan baik. Kelas *Iron Deficiency Anemia*, *Leukemia*, serta *Normocytic Normochromic Anemia* menunjukkan performa menengah dengan nilai F1-score berada pada kisaran 69%–76%. Sebaliknya, kelas *Macrocytic Anemia* dan *Other Microcytic Anemia* memiliki nilai F1-score yang rendah, yang mengindikasikan kesulitan model dalam mengklasifikasikan kelas minoritas. Secara keseluruhan, nilai *macro average* precision, recall, dan F1-score masing-masing sebesar 67%, 64%, dan 64%, yang menunjukkan bahwa model pada skenario Non-SMOTE ini memiliki kemampuan generalisasi yang cukup baik, meskipun performa pada beberapa kelas masih perlu ditingkatkan.

Tabel 4. 9 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 5

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	89%	91%	90%
2	Iron Deficiency Anemia	79%	62%	69%
3	Leukemia	80%	66%	69%
4	Leukemia with Thrombocytopenia	73%	63%	67%
5	Macrocytic Anemia	16%	20%	18%
6	Normocytic Hypochromic Anemia	64%	72%	68%
7	Normocytic Normochromic Anemia	73%	79%	76%
8	Other Microcytic Anemia	45%	29%	33%
9	Thrombocytopenia	81%	91%	84%
Rata-rata (Std)		67% ($\pm 19\%$)	64% ($\pm 20\%$)	64% ($\pm 17\%$)

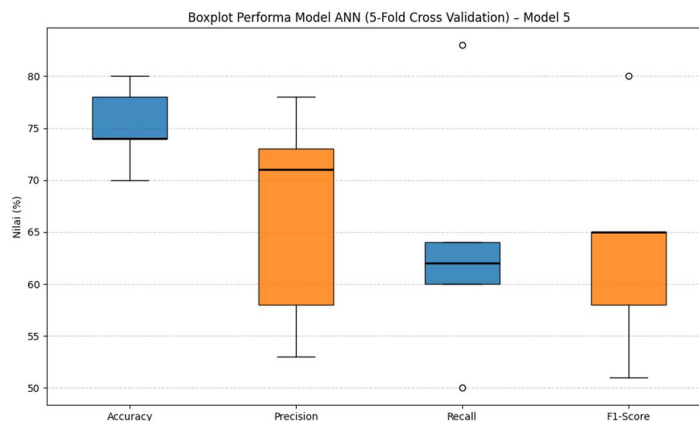
Hasil 5-Fold Cross-Validation pada Skenario 5 (Non-SMOTE) yang ditunjukkan pada Tabel 4.10 memperlihatkan bahwa performa model relatif konsisten antar fold, dengan nilai *accuracy* berada pada kisaran 70% hingga 80%. Fold ke-3 menghasilkan performa terbaik dengan *accuracy* tertinggi sebesar 80%, sedangkan Fold ke-1 menunjukkan nilai *accuracy* terendah sebesar 70%. Nilai rata-rata yang disertai standar deviasi menunjukkan adanya variasi performa antar fold,

namun perbedaannya tidak terlalu besar. Hal ini mengindikasikan bahwa model pada skenario Non-SMOTE masih memiliki kemampuan generalisasi yang cukup baik.

Tabel 4. 10 Hasil Evaluasi 5-Fold Cross-Validation Model 5

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	70%	58%	60%	58%
2	78%	78%	83%	80%
3	80%	73%	62%	65%
4	74%	53%	50%	51%
5	74%	71%	64%	65%
Rata-rata (Std)	75% (4%)	67% (10%)	64% (12%)	64% (11%)

Boxplot performa model ANN pada 5-Fold Cross-Validation tanpa SMOTE pada Gambar 4.5. Terlihat bahwa *accuracy* memiliki sebaran yang relatif lebih stabil dibandingkan metrik lainnya, namun *precision macro*, *recall macro*, dan *F1-score macro* menunjukkan variasi yang lebih besar antar *fold*. Variasi ini mengindikasikan bahwa ketidakseimbangan data memengaruhi konsistensi performa model pada skenario Non-SMOTE. Secara keseluruhan, boxplot ini menunjukkan bahwa tanpa penerapan SMOTE, kestabilan dan konsistensi performa model masih lebih rendah dibandingkan skenario dengan data yang telah diseimbangkan.



Gambar 4. 15 Boxplot Performa Model ANN (5-Fold) Model 5

4.2.2.6 Model 6

Berdasarkan Tabel 4.11, Skenario 6 (Non-SMOTE) dengan konfigurasi dua *hidden layer*, neuron 16–32, dan *learning rate* 0,001 menunjukkan performa klasifikasi yang sangat bervariasi antar kelas. Kelas *Healthy* memiliki nilai *recall* yang sangat tinggi, yang menunjukkan bahwa model mampu mengenali sebagian besar data pada kelas tersebut, meskipun nilai *precision* dan *F1-score* masih terbatas. Kelas *Iron Deficiency Anemia* dan *Normocytic Hypochromic Anemia* menunjukkan performa rendah hingga menengah dengan keseimbangan *precision* dan *recall* yang belum optimal. Sebaliknya, beberapa kelas seperti *Leukemia*, *Leukemia with Thrombocytopenia*, *Macrocytic Anemia*, dan *Other Microcytic Anemia* memiliki nilai *precision*, *recall*, dan *F1-score* yang sangat rendah atau mendekati nol, yang mengindikasikan bahwa model belum mampu mengenali pola pada kelas-kelas tersebut. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 17%, 19%, dan 15%, yang

menunjukkan bahwa pada skenario Non-SMOTE ini kemampuan generalisasi model masih terbatas akibat ketidakseimbangan distribusi data antar kelas.

Tabel 4. 11 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 6

No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	45%	96%	60%
2	Iron deficiency anemia	47%	16%	21%
3	Leukemia	0%	0%	0%
4	Leukemia with thrombocytopenia	0%	0%	0%
5	Macrocytic anemia	0%	0%	0%
6	Normocytic hypochromic anemia	32%	47%	36%
7	Normocytic normochromic anemia	14%	5%	7%
8	Other microcytic anemia	0%	0%	0%
9	Thrombocytopenia	12%	10%	11%
Rata-rata (Std)		17% (19%)	19% (27%)	15% (19%)

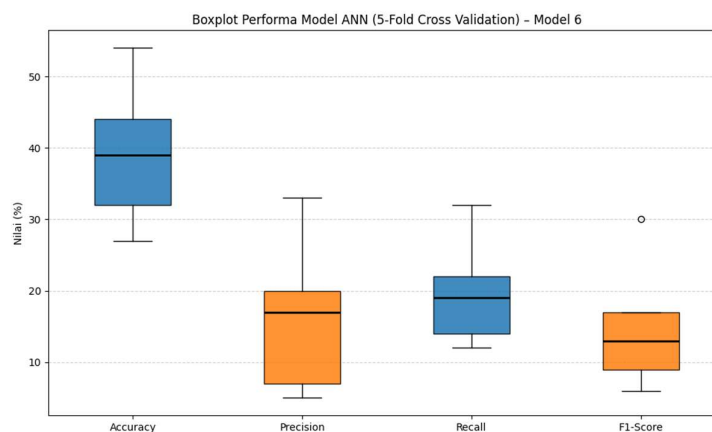
Hasil 5-Fold Cross-Validation pada Tabel 4.12 menunjukkan bahwa performa model relatif rendah dan bervariasi antar *fold*, dengan nilai *accuracy* berada pada kisaran 27%–54%. Fold ke-4 menghasilkan performa tertinggi, sedangkan Fold ke-2 menunjukkan hasil terendah pada sebagian besar metrik. Nilai rata-rata yang disertai standar deviasi dalam tanda kurung menunjukkan bahwa variasi performa antar *fold* masih cukup besar, sehingga kemampuan generalisasi model pada skenario ini masih terbatas.

Tabel 4. 12 Hasil Evaluasi 5-Fold Cross-Validation Model 6

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	39%	20%	19%	13%
2	27%	5%	12%	6%
3	44%	17%	22%	17%
4	54%	33%	32%	30%
5	32%	7%	14%	9%
Rata-rata (Std)	39% (10%)	17% (11%)	19% (8%)	15% (9%)

Boxplot performa model ANN pada 5-Fold Cross-Validation tanpa SMOTE pada Gambar 4.6. *Accuracy* memiliki nilai median yang rendah dengan sebaran yang cukup lebar, sementara *precision macro*, *recall macro*, dan *F1-score macro*

berada pada kisaran nilai yang rendah dan menunjukkan variasi antar *fold*. Kondisi ini mengindikasikan bahwa tanpa penerapan SMOTE, model mengalami kesulitan dalam mengklasifikasikan kelas secara seimbang. Secara keseluruhan, boxplot ini menunjukkan bahwa kestabilan dan kemampuan generalisasi model pada skenario Non-SMOTE masih terbatas.



Gambar 4. 16 Boxplot Performa Model ANN (5-Fold) Model 6

4.2.2.7 Model 7

Berdasarkan Tabel 4.13, Model 5 (Non-SMOTE) dengan konfigurasi dua *hidden layer*, neuron 32–64, dan *learning rate* 0,01 menunjukkan performa klasifikasi yang bervariasi antar kelas. Kelas *Healthy* dan *Thrombocytopenia* memperoleh nilai *precision*, *recall*, dan *F1-score* yang tinggi, yang menunjukkan bahwa model mampu mengenali kedua kelas tersebut dengan baik. Kelas *Iron Deficiency Anemia*, *Leukemia*, serta *Normocytic Normochromic Anemia* menunjukkan performa menengah dengan keseimbangan *precision* dan *recall* yang cukup baik. Sebaliknya, kelas *Macrocytic Anemia* dan *Other Microcytic Anemia* masih memiliki nilai *F1-score* yang rendah, yang mengindikasikan keterbatasan

model dalam membedakan pola pada kelas tersebut. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 69%, 68%, dan 67%, yang menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik meskipun masih terdapat perbedaan performa antar kelas.

Tabel 4. 13 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 7

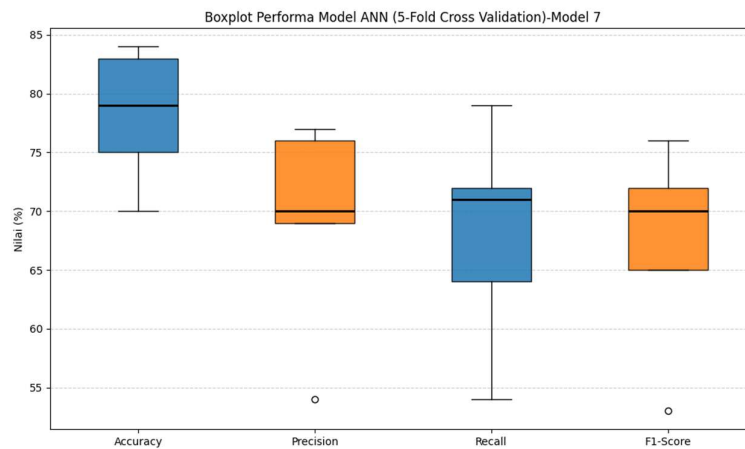
No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	92%	93%	92%
2	Iron Deficiency Anemia	77%	68%	72%
3	Leukemia	79%	76%	77%
4	Leukemia with Thrombocytopenia	80%	77%	75%
5	Macrocytic Anemia	12%	15%	13%
6	Normocytic Hypochromic Anemia	65%	73%	68%
7	Normocytic Normochromic Anemia	80%	84%	81%
8	Other Microcytic Anemia	51%	36%	41%
9	Thrombocytopenia	85%	90%	86%
Rata-rata (Std)		69% ($\pm 18\%$)	68% ($\pm 20\%$)	67% ($\pm 16\%$)

Hasil 5-Fold Cross-Validation pada Tabel 4.14 menunjukkan bahwa performa model berada pada kisaran *accuracy* 70% hingga 84%. Fold ke-4 menghasilkan *accuracy* tertinggi sebesar 84%, sedangkan nilai *precision macro*, *recall macro*, dan *F1-score macro* pada fold lainnya berada pada rentang yang relatif berdekatan. Nilai rata-rata yang disertai standar deviasi menunjukkan bahwa variasi performa antar fold masih dalam batas wajar, sehingga model memiliki kemampuan generalisasi yang cukup baik.

Tabel 4. 14 Hasil Evaluasi 5-Fold Cross-Validation Model 7

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	70%	54%	54%	53%
2	79%	77%	79%	76%
3	83%	76%	71%	72%
4	84%	69%	64%	65%
5	75%	70%	72%	70%
Rata-rata (Std)	78% ($\pm 6\%$)	69% ($\pm 9\%$)	68% ($\pm 10\%$)	67% ($\pm 9\%$)

Boxplot performa model ANN pada 5-Fold Cross-Validation tanpa SMOTE pada Gambar 4.7 menunjukkan bahwa metrik *accuracy* memiliki median yang relatif tinggi dengan sebaran yang tidak terlalu lebar, yang mengindikasikan konsistensi performa model antar fold. Nilai *precision macro* dan *recall macro* menunjukkan variasi yang lebih besar, menandakan adanya perbedaan kinerja model pada masing-masing fold. Sementara itu, *F1-score macro* memiliki sebaran yang relatif moderat, yang menunjukkan bahwa keseimbangan antara *precision* dan *recall* masih cukup terjaga. Secara keseluruhan, boxplot ini memperlihatkan bahwa pada skenario Non-SMOTE, model memiliki performa yang cukup stabil, meskipun masih terdapat variasi performa antar fold.



Gambar 4. 17 Boxplot Performa Model ANN (5-Fold) Model 7

4.2.2.8 Model 8

Berdasarkan Tabel 4.15, Skenario 8 (Non-SMOTE) dengan konfigurasi dua *hidden layer*, neuron 32–64, dan *learning rate* 0,001 menunjukkan performa klasifikasi yang bervariasi dan cenderung rendah antar kelas. Kelas *Healthy* memiliki nilai *recall* yang tinggi, sementara kelas *Iron Deficiency Anemia*,

Normocytic Hypochromic Anemia, dan *Normocytic Normochromic Anemia* menunjukkan performa rendah hingga menengah. Sebaliknya, kelas *Macrocytic Anemia* dan *Other Microcytic Anemia* memiliki nilai *precision*, *recall*, dan *F1-score* yang sangat rendah. Secara keseluruhan, nilai *macro average precision*, *recall*, dan *F1-score* masing-masing sebesar 33%, 35%, dan 33%, yang menunjukkan bahwa kemampuan generalisasi model pada skenario Non-SMOTE ini masih terbatas.

Tabel 4. 15 Rata-Rata Performa Klasifikasi per Kelas 5-Fold Cross-Validation Model 8

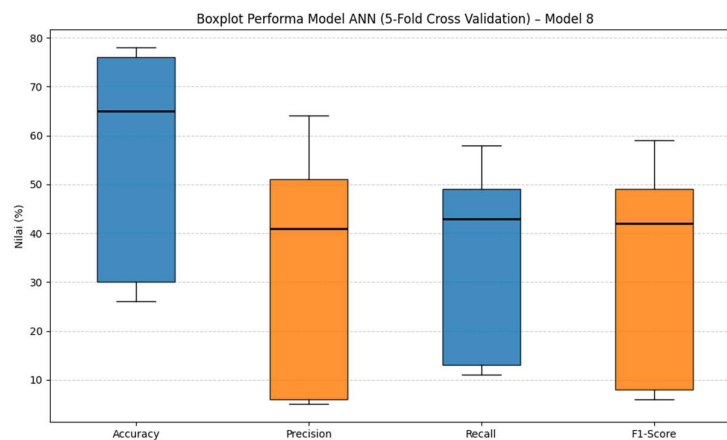
No	Kelas	Precision (%)	Recall (%)	F1-Score (%)
1	Healthy	65%	94%	74%
2	Iron deficiency anemia	46%	39%	42%
3	Leukemia	41%	28%	33%
4	Leukemia with thrombocytopenia	20%	10%	13%
5	Macrocytic anemia	0%	0%	0%
6	Normocytic hypochromic anemia	41%	49%	44%
7	Normocytic normochromic anemia	47%	49%	47%
8	Other microcytic anemia	0%	0%	0%
9	Thrombocytopenia	42%	45%	43%
Rata-rata (Std)		33% (21%)	35% (28%)	33% (23%)

Hasil 5-Fold Cross-Validation pada Tabel menunjukkan bahwa performa model bervariasi cukup besar antar *fold*, dengan nilai *accuracy* berada pada kisaran 26%–78%. Fold ke-4 menghasilkan performa terbaik pada seluruh metrik evaluasi, sedangkan Fold ke-2 dan Fold ke-5 menunjukkan performa terendah, terutama pada *precision macro* dan *F1-score macro*. Perbedaan ini menunjukkan bahwa model masih mengalami kesulitan dalam melakukan klasifikasi secara seimbang. Nilai rata-rata yang disertai standar deviasi dalam tanda kurung mengindikasikan bahwa variasi performa antar *fold* masih cukup besar, sehingga kemampuan generalisasi model pada skenario ini masih terbatas.

Tabel 4. 16 Hasil Evaluasi 5-Fold Cross-Validation Model 8

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	65%	41%	43%	42%
2	26%	5%	11%	6%
3	76%	51%	49%	49%
4	78%	64%	58%	59%
5	30%	6%	13%	8%
Rata-rata (Std)	55% (25%)	33% (27%)	35% (21%)	33% (24%)

Boxplot performa model ANN pada 5-Fold Cross-Validation tanpa SMOTE pada Gambar 4.18. *Accuracy* memiliki sebaran yang cukup lebar, menandakan variasi performa antar *fold*. *Precision macro* dan *F1-score macro* juga menunjukkan rentang nilai yang besar, sementara *recall macro* relatif lebih stabil meskipun masih berfluktuasi. Secara umum, boxplot ini mengindikasikan bahwa pada skenario Non-SMOTE, kestabilan dan kemampuan generalisasi model masih terbatas akibat ketidakseimbangan data.



Gambar 4. 18 Boxplot Performa Model ANN (5-Fold) Model 8

4.3 Pembahasan

Berdasarkan hasil uji coba yang telah dijabarkan secara rinci pada sub-bab 4.2.2.1 hingga 4.2.2.8, bagian ini menyajikan hasil evaluasi model dan analisis perbandingan menyeluruh. Analisis ini bertujuan untuk mengidentifikasi

konfigurasi hyperparameter yang paling optimal serta memahami pengaruh dari setiap parameter dan karakteristik dataset terhadap performa model.

4.3.1 Rangkuman Hasil Evaluasi Model

Bagian ini bertujuan merangkum dan membandingkan performa model *Artificial Neural Network* (ANN) pada 8 skenario pengujian, baik menggunakan data SMOTE maupun Non-SMOTE. Evaluasi dilakukan untuk mengidentifikasi pola performa pada setiap konfigurasi serta menentukan skenario terbaik berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-Score*.

Tabel 4. 17 Hasil dari Pengujian Metode ANN *Backpropagation*

Skenario	Dataset	Evaluasi			
		<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Skenario 1	SMOTE	79%	70%	76%	71%
Skenario 2		65%	59%	71%	61%
Skenario 3		83%	79%	81%	79%
Skenario 4		66%	60%	69%	60%
Skenario 5	Non-SMOTE	75%	67%	64%	64%
Skenario 6		39%	17%	19%	15%
Skenario 7		78%	69%	68%	67%
Skenario 8		55%	33%	35%	33%

Berdasarkan hasil pengujian pada Tabel 4.17, performa model *Artificial Neural Network* (ANN) *backpropagation* dipengaruhi secara signifikan oleh penyeimbangan data (SMOTE) serta pemilihan parameter model, khususnya jumlah neuron pada *hidden layer* dan nilai *learning rate*. Secara umum, skenario dengan SMOTE menunjukkan performa yang lebih baik dibandingkan skenario Non-SMOTE, yang terlihat dari nilai *accuracy*, *precision*, *recall*, dan *F1-score* yang lebih tinggi dan relatif stabil.

Pada kelompok SMOTE (Skenario 1–4), performa terbaik ditunjukkan oleh Skenario 3, yang menggunakan dua *hidden layer* dengan 32–64 neuron dan

learning rate 0,01. Skenario ini menghasilkan *accuracy* 83%, *precision* 79%, *recall* 81%, dan *F1-score* 79%, yang merupakan nilai tertinggi pada seluruh metrik evaluasi. Hasil ini menunjukkan bahwa kombinasi kapasitas jaringan yang lebih besar dan *learning rate* yang sesuai mampu meningkatkan kemampuan model dalam mempelajari pola parameter *Complete Blood Count* (CBC), termasuk pada kelas dengan karakteristik fitur yang lebih kompleks. Sebaliknya, Skenario 2 yang menggunakan konfigurasi neuron 16–32 dengan *learning rate* 0,001 menghasilkan performa terendah pada kelompok SMOTE, dengan *accuracy* hanya 65%, yang mengindikasikan bahwa kapasitas jaringan dan kecepatan pembelajaran yang kurang optimal membatasi kemampuan model dalam membentuk representasi fitur yang baik.

Pada kelompok Non-SMOTE (Skenario 5–8), performa terbaik diperoleh pada Skenario 7, yaitu konfigurasi dua *hidden layer* dengan 32–64 neuron dan *learning rate* 0,01, yang menghasilkan *accuracy* 78%, *precision* 69%, *recall* 68%, dan *F1-score* 67%. Meskipun hasilnya masih berada di bawah skenario berbasis SMOTE, konfigurasi ini menunjukkan bahwa peningkatan jumlah neuron dan pemilihan *learning rate* yang tepat tetap dapat memperbaiki performa model pada kondisi data yang tidak seimbang. Sebaliknya, Skenario 6 menunjukkan performa terendah dengan *accuracy* hanya 39% dan *F1-score* 15%, yang menegaskan bahwa tanpa penyeimbangan data, model cenderung bias terhadap kelas mayoritas dan kesulitan mengenali kelas minoritas.

Ditinjau dari pengaruh parameter, penggunaan jumlah neuron yang lebih besar (32–64) secara konsisten memberikan hasil yang lebih baik dibandingkan 16–

32 neuron, baik pada data SMOTE maupun Non-SMOTE. Jumlah neuron yang lebih besar membuat jaringan memiliki kapasitas representasi yang lebih banyak untuk memodelkan hubungan nonlinier antar fitur CBC. Selain itu, nilai *learning rate* 0,01 terbukti lebih efektif dibandingkan 0,001, karena mampu menghasilkan proses pembelajaran yang lebih stabil dan mendorong model mencapai performa yang lebih optimal.

Hasil penelitian ini menunjukkan bahwa kombinasi SMOTE, arsitektur dua *hidden layer* dengan 32–64 neuron, dan *learning rate* 0,01 merupakan konfigurasi yang paling optimal dalam penelitian ini. Hal ini menandakan bahwa keberhasilan model ANN tidak hanya ditentukan oleh algoritma yang digunakan, tetapi juga sangat bergantung pada keseimbangan data dan pemilihan parameter jaringan yang tepat dalam klasifikasi anemia berbasis parameter CBC.

4.3.2 Analisis Pengaruh *Hyperparameter*

Berikut penjelasan mengenai pengaruh dari setiap parameter yang digunakan dalam penelitian ini.

4.3.2.1 Pengaruh *Balancing data*

Balancing data merupakan salah satu faktor penting yang memengaruhi performa model pembelajaran mesin, terutama pada kasus klasifikasi dengan distribusi kelas yang tidak seimbang. Pada penelitian ini, metode *Synthetic Minority Oversampling Technique* (SMOTE) diterapkan untuk menyeimbangkan jumlah sampel pada setiap kelas sebelum proses pelatihan ANN. Pengaruh penggunaan

SMOTE dibandingkan data asli (Non-SMOTE) dianalisis menggunakan sepuluh skenario untuk masing-masing kondisi seperti pada tabel berikut.

Tabel 4. 18 Hasil Evaluasi Confusion Matrix berdasarkan preprocessing menggunakan SMOTE dan Non-SMOTE

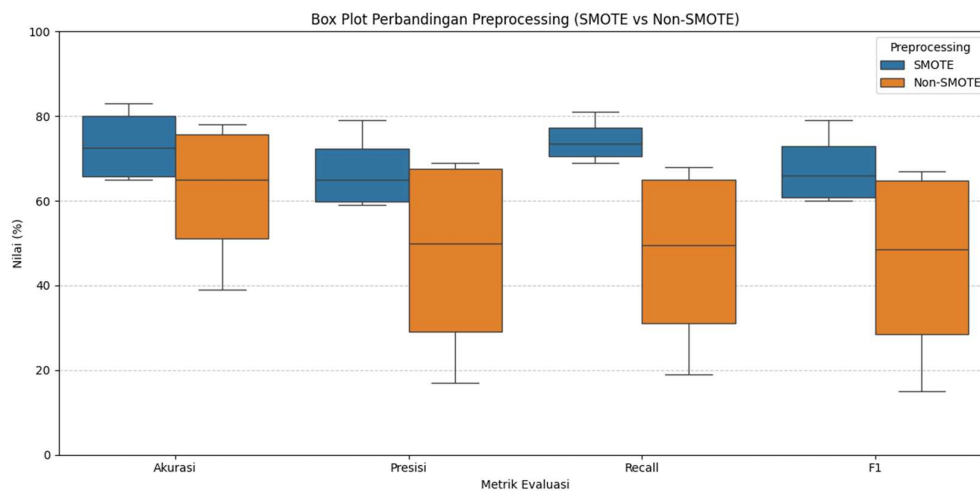
Preprocessing	Skenario	Accuracy	Precision	Recall	F1-Score
SMOTE	1	79%	70%	76%	71%
	2	65%	59%	71%	61%
	3	83%	79%	81%	79%
	4	66%	60%	69%	60%
Rata-Rata		73%	67%	74%	68%
Non-SMOTE	5	75%	67%	64%	64%
	6	39%	17%	19%	15%
	7	78%	69%	68%	67%
	8	55%	33%	35%	33%
Rata-Rata		62%	47%	47%	45%

Berdasarkan Tabel 4.18, terlihat bahwa model yang dilatih menggunakan data SMOTE menghasilkan performa yang lebih baik dibandingkan model tanpa penyeimbangan data. Pada kelompok SMOTE (Skenario 1–4), nilai *accuracy* berada pada rentang 65% hingga 83%, dengan rata-rata *accuracy* sebesar 73%. Nilai rata-rata *precision*, *recall*, dan *F1-score* masing-masing mencapai 67%, 74%, dan 68%, yang menunjukkan bahwa model mampu melakukan klasifikasi dengan cukup baik dan relatif seimbang antara ketepatan prediksi dan kemampuan mengenali data aktual.

Performa terbaik pada kelompok SMOTE ditunjukkan oleh Skenario 3, yang memperoleh *accuracy* 83%, *precision* 79%, *recall* 81%, dan *F1-score* 79%. Sebaliknya, performa terendah terdapat pada Skenario 2 dan Skenario 4, yang memiliki *accuracy* masing-masing 65% dan 66%, dengan nilai *precision* dan *F1-score* yang lebih rendah. Meskipun demikian, secara keseluruhan hasil pada kelompok SMOTE masih menunjukkan kestabilan performa yang cukup baik. Hal

ini mengindikasikan bahwa proses penyeimbangan data melalui SMOTE membantu model mempelajari pola fitur dari kelas minoritas secara lebih merata.

Sebaliknya, pada kelompok Non-SMOTE (Skenario 13–16), performa model cenderung lebih rendah. Nilai *accuracy* berada pada rentang 39% hingga 78%, dengan rata-rata *accuracy* hanya sebesar 62%. Nilai rata-rata *precision*, *recall*, dan *F1-score* juga lebih rendah, masing-masing sebesar 47%, 47%, dan 45%. Skenario dengan performa terendah adalah Skenario 6, yang hanya mencapai *accuracy* 39% dan *F1-score* 15%, menunjukkan bahwa model mengalami kesulitan besar dalam mengenali kelas minoritas ketika data tidak diseimbangkan.



Gambar 4. 19 Box Plot Perbandingan Metrik Evaluasi Model ANN pada Data SMOTE dan Non-SMOTE

Berdasarkan box plot yang ditampilkan pada Gambar 4.20, terlihat bahwa model dengan SMOTE memiliki nilai median yang lebih tinggi pada seluruh metrik evaluasi dibandingkan Non-SMOTE, serta rentang interkuartil yang lebih sempit. Hal ini menunjukkan bahwa hasil evaluasi lebih konsisten dan tidak terlalu bervariasi antar skenario. Sebaliknya, box plot Non-SMOTE memperlihatkan

sebaran nilai yang lebih lebar dengan beberapa nilai rendah, yang menandakan ketidakstabilan performa model akibat ketidakseimbangan data. Pada perbandingan jumlah neuron, konfigurasi 32–64 menunjukkan posisi median yang lebih tinggi dan sebaran yang lebih rapat dibandingkan konfigurasi 16–32, sehingga dapat disimpulkan bahwa model dengan jumlah neuron lebih besar menghasilkan performa yang lebih stabil dan andal berdasarkan visualisasi box plot

4.3.2.2 Pengaruh *Neuron* pada *Hidden Layer*

Pada model yang diujikan terdapat konfigurasi dengan 2 *hidden layer*. Setiap *hidden layer* berperan dalam menangkap serta mempelajari pola-pola kompleks pada data. Oleh karena itu, analisis terhadap variasi jumlah *hidden layer* diperlukan untuk menentukan arsitektur terbaik agar model dapat bekerja secara optimal.

Tabel 4. 19 Hasil Evaluasi Confusion Matrix berdasarkan 2 hidden layer, neuron 16-32

Skenario	Accuracy	Precision	Recall	F1-Score
1	79%	70%	76%	71%
2	65%	59%	71%	61%
5	75%	67%	64%	64%
6	39%	17%	19%	15%
Rata-Rata	65%	53%	58%	53%

Tabel 4. 20 Hasil Evaluasi Confusion Matrix berdasarkan 2 hidden layer, neuron 32-64

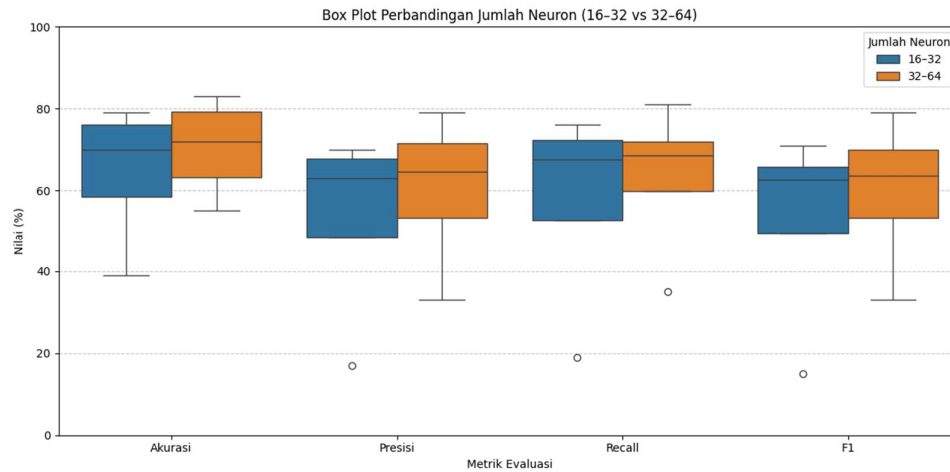
Skenario	Accuracy	Precision	Recall	F1-Score
3	83%	79%	81%	79%
4	66%	60%	69%	60%
7	78%	69%	68%	67%
8	55%	33%	35%	33%
Rata-Rata	71%	60%	63%	60%

Hasil pada Tabel 4.19 dan Tabel 4.20 menunjukkan bahwa peningkatan jumlah neuron pada dua *hidden layer* memberikan dampak positif terhadap performa model *Artificial Neural Network* (ANN). Pada konfigurasi 2 *hidden layer*

dengan neuron 16–32, model menghasilkan rata-rata accuracy sebesar 65%, dengan *precision* 53%, *recall* 58%, dan *F1-score* 53%. Nilai-nilai tersebut menunjukkan bahwa kapasitas jaringan dengan jumlah neuron yang lebih kecil masih memiliki keterbatasan dalam menangkap kompleksitas pola fitur *Complete Blood Count* (CBC), terutama pada kelas-kelas dengan karakteristik yang saling tumpang tindih.

Pada konfigurasi 2 *hidden layer* dengan neuron 32–64, performa model mengalami peningkatan yang cukup jelas. Rata-rata accuracy meningkat menjadi 71%, diikuti oleh peningkatan *precision* menjadi 60%, *recall* 63%, dan *F1-score* 60%. Peningkatan ini menunjukkan bahwa jumlah neuron yang lebih besar memberikan kemampuan representasi fitur yang lebih baik, sehingga model lebih efektif dalam membedakan antar kelas anemia yang memiliki pola CBC yang mirip.

Dengan demikian, penambahan jumlah neuron dari 16–32 menjadi 32–64 pada dua *hidden layer* mampu meningkatkan kualitas pembelajaran model ANN dan menghasilkan performa klasifikasi yang lebih optimal. Hasil ini menegaskan bahwa pemilihan kapasitas jaringan yang tepat merupakan faktor penting dalam meningkatkan kinerja model pada tugas klasifikasi anemia berbasis parameter CBC.



Gambar 4. 20 Box Plot Perbandingan Metrik Evaluasi Model ANN pada jumlah neuron (16-32 dan 32-64)

Berdasarkan box plot pada Gambar 4.20 perbandingan jumlah neuron 16–32 dan 32–64, terlihat bahwa konfigurasi 32–64 neuron memiliki nilai median yang lebih tinggi pada seluruh metrik evaluasi, yaitu akurasi, presisi, *recall*, dan *F1-score*. Selain itu, sebaran data pada konfigurasi 32–64 cenderung lebih rapat, yang menunjukkan performa model lebih stabil antar skenario pengujian. Sebaliknya, konfigurasi 16–32 neuron memperlihatkan rentang nilai yang lebih lebar serta beberapa nilai rendah, khususnya pada metrik presisi dan *F1-score*, yang menandakan variasi performa yang lebih besar. Visualisasi ini menunjukkan bahwa peningkatan jumlah neuron memberikan kemampuan representasi yang lebih baik bagi model ANN, sehingga menghasilkan kinerja yang lebih konsisten berdasarkan distribusi nilai pada box plot.

4.3.2.3 Pengaruh *Learning Rate*

Learning rate merupakan parameter yang menentukan seberapa cepat model melakukan proses pembelajaran pada setiap iterasi (*epoch*). Berikut disajikan perbandingan kinerja model pada beberapa nilai *learning rate*, yaitu 0.01 dan 0.001.

1. Perbandingan learning rate 0.01

Model pada Tabel 4.21 yang menggunakan learning rate sebesar 0,01 menunjukkan performa yang relatif stabil dan lebih tinggi dibandingkan konfigurasi dengan *learning rate* yang lebih kecil. Nilai *accuracy* pada kelompok ini berada pada rentang 75% hingga 83%, dengan performa terbaik ditunjukkan oleh Skenario 3 yang mencapai *accuracy* 83%. Skenario lainnya juga mampu mempertahankan nilai akurasi di atas 75%, yang menandakan konsistensi kinerja model. Nilai rata-rata *accuracy* sebesar 79%, disertai *precision* 71% dan *recall* sebesar 72%, serta *F1-score* 70%, menunjukkan bahwa *learning rate* 0,01 memberikan keseimbangan yang baik antara kecepatan pembelajaran dan stabilitas proses pelatihan. Kondisi ini memungkinkan model mempelajari pola data secara lebih efektif dan menghasilkan performa klasifikasi yang lebih optimal.

Tabel 4. 21 Hasil Evaluasi Confusion Matrix berdasarkan learning rate 0.01

Skenario	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	79%	70%	76%	71%
3	83%	79%	81%	79%
5	75%	67%	64%	64%
7	78%	69%	68%	67%
Rata-rata	79%	71%	72%	70%

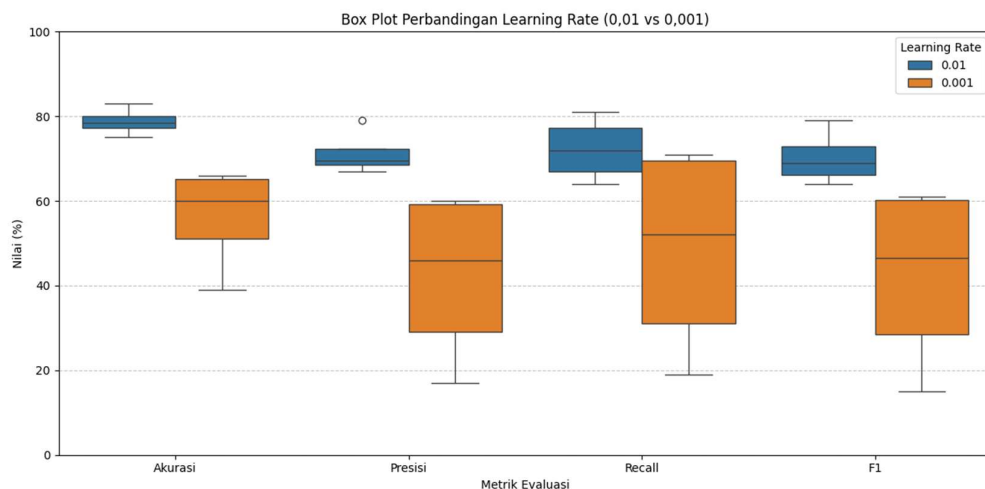
2. Perbandingan learning rate 0.001

Model pada Tabel 4.22 yang menggunakan learning rate sebesar 0,001 menunjukkan performa yang lebih rendah dan kurang stabil dibandingkan model

dengan *learning rate* 0,01. Nilai *accuracy* pada kelompok ini berada pada rentang 39% hingga 66%, dengan rata-rata *accuracy* hanya sebesar 56%. Beberapa skenario, khususnya Skenario 6, menunjukkan nilai *accuracy*, *precision*, *recall*, dan *F1-score* yang sangat rendah, yang mengindikasikan bahwa model mengalami kesulitan dalam mempelajari pola data secara optimal. Nilai rata-rata *precision*, *recall*, dan *F1-score* masing-masing sebesar 42%, 49%, dan 42% juga lebih rendah dibandingkan kelompok *learning rate* 0,01. Hasil ini menunjukkan bahwa penggunaan *learning rate* yang terlalu kecil menyebabkan proses pembelajaran berjalan lebih lambat, sehingga model sulit mencapai kondisi pembelajaran yang optimal dan kurang mampu mengenali pola pada beberapa kelas, terutama kelas dengan distribusi data yang lebih sedikit.

Tabel 4. 22 Hasil Evaluasi *Confusion Matrix* berdasarkan *learning rate* 0.001

Skenario	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
2	65%	59%	71%	61%
4	66%	60%	69%	60%
6	39%	17%	19%	15%
8	55%	33%	35%	33%
Rata-rata	56%	42%	49%	42%



Gambar 4. 21 Box Plot Perbandingan Metrik Evaluasi Model ANN pada perbandingan Learning Rate 0,01 dan 0,001

Berdasarkan box plot pada Gambar 4.21 perbandingan *learning rate* 0,01 dan 0,001, terlihat bahwa *learning rate* 0,01 menghasilkan nilai median yang lebih tinggi dan sebaran yang lebih sempit pada seluruh metrik evaluasi, yaitu akurasi, presisi, *recall*, dan *F1-score*. Hal ini menunjukkan bahwa proses pembelajaran model berlangsung lebih stabil dan konvergen pada nilai *learning rate* tersebut. Sebaliknya, *learning rate* 0,001 memperlihatkan rentang nilai yang lebih lebar dengan beberapa nilai rendah, khususnya pada metrik presisi dan *F1-score*, yang menandakan bahwa pembaruan bobot berjalan lebih lambat sehingga model belum mampu mencapai performa optimal. Visualisasi box plot ini menunjukkan bahwa pemilihan *learning rate* yang tepat berpengaruh langsung terhadap kestabilan dan kualitas kinerja model ANN.

Secara keseluruhan, hasil pengujian pada seluruh skenario menunjukkan bahwa konfigurasi terbaik terdapat pada Skenario 3, yaitu model ANN dengan penerapan SMOTE, arsitektur dua *hidden layer* berisi 32 dan 64 neuron, serta *learning rate* 0.01. Konfigurasi ini menghasilkan nilai evaluasi tertinggi pada seluruh metrik (accuracy, precision, recall, dan F1-score). Selain itu, proses validasi menggunakan metode *5-Fold Cross Validation* semakin memperkuat konsistensi performa model, karena hasil yang diperoleh stabil pada setiap fold. Hal ini menunjukkan bahwa performa terbaik dapat dicapai ketika pemilihan parameter dilakukan secara presisi. Model ANN yang bekerja dengan kombinasi parameter yang tepat menggambarkan bahwa suatu sistem akan menghasilkan output optimal. Prinsip ini sejalan dengan ajaran Islam bahwa segala sesuatu di dunia ini berjalan

berdasarkan ketentuan, ukuran, dan perhitungan yang sangat presisi. Sebagaimana firman Allah SWT dalam Q.S. Al-Qamar Ayat 49:

إِنَّا كُلَّ شَيْءٍ خَلَقْنَاهُ بِقَدَرٍ

“*Sesungguhnya Kami menciptakan segala sesuatu menurut ukuran (qadar).*”
(Q.S. Al-Qamar Ayat 49).

Ayat ini menjelaskan bahwa seluruh ciptaan Allah SWT tunduk pada sistem, aturan, perhitungan, dan ukuran yang pasti. Tidak ada satu pun yang terjadi secara acak atau tanpa ketetapan. *Qadar* dalam ayat ini merujuk pada ketentuan yang terukur, tersusun, dan presisi baik dalam hukum alam, pergerakan benda langit, siklus kehidupan, maupun struktur makhluk hidup. Para ulama seperti Ibn Kathir dan Al-Tabari menafsirkan ayat ini sebagai penegasan bahwa Allah menciptakan segala sesuatu dengan ukuran, takaran, dan proporsi tertentu, sehingga segala sesuatu berjalan harmonis sesuai fungsinya. Kemudian dalam Q.S. Al-Furqan Ayat 2:

إِلَٰهِ الَّذِي لَهُ ۖ مُلْكُ السَّمٰوٰتِ وَٱلْأَرْضِ ۖ وَلَمْ يَتَّخِذْ وَلَدًا ۖ وَلَمْ يَكُنْ لَهُ ۖ شَرِيكٌ فِى ٱلْمُلْكِ ۖ وَخَلَقَ كُلَّ شَيْءٍ فَقَدَرَهُ ۖ تَقْدِيرًا

" (Yaitu Zat) yang milik-Nyalah kerajaan langit dan bumi, (Dia) tidak mempunyai anak, dan tidak ada satu sekutu pun dalam kekuasaan(-Nya). Dia telah menciptakan segala sesuatu, lalu menetapkan ukuran-ukurannya dengan tepat.."(Q.S. Al-Furqan : 2).

Ayat ini menjelaskan bahwa Allah SWT menciptakan segala sesuatu di alam semesta disertai dengan ukuran, takaran, dan ketetapan yang sangat tepat. Frasa "*faqaddarahu taqdīrā*" menggambarkan bahwa setiap ciptaan memiliki

proporsi dan perhitungan yang terukur, baik dalam bentuk, fungsi, maupun proses yang mengaturnya. Para mufasir seperti Ibn Kathir menafsirkan ayat ini sebagai bukti bahwa Allah menetapkan seluruh makhluk dalam sistem yang presisi, tidak berlebihan dan tidak kekurangan, sehingga setiap bagian ciptaan berjalan harmonis sesuai tujuan yang telah ditentukan-Nya.

Konsep tersebut sejalan dengan hasil penelitian ini, di mana pemilihan parameter model Artificial Neural Network (ANN) juga menuntut ketepatan ukuran agar menghasilkan performa terbaik. Penentuan jumlah neuron, kedalaman *hidden layer*, nilai *learning rate*, rasio pembagian data, hingga proses penyeimbangan dataset menggunakan SMOTE harus dilakukan secara proporsional dan tidak berlebihan. Ketika seluruh parameter ditetapkan sesuai “ukuran” yang tepat, model mampu belajar secara optimal dan menghasilkan prediksi klasifikasi anemia yang lebih akurat. Hal ini terlihat pada Skenario 9, yang menjadi konfigurasi terbaik karena kombinasi parameternya sesuai dengan kebutuhan model, sehingga menghasilkan nilai akurasi, precision, recall, dan F1-Score yang paling tinggi.

Seperti halnya ciptaan Allah yang diciptakan berdasarkan ukuran yang sempurna, hasil pada penelitian ini menunjukkan bahwa proses ilmiah pun memerlukan ketelitian, keseimbangan, dan pemilihan ukuran yang sesuai. Dengan demikian, penelitian ini tidak hanya menunjukkan kinerja model ANN dalam ranah ilmiah, tetapi juga mencerminkan nilai-nilai Islam tentang pentingnya menetapkan setiap perkara menurut ukuran yang tepat agar tercapai hasil yang terbaik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian menggunakan metode Artificial Neural Network (ANN) *backpropagation*, dapat disimpulkan bahwa performa model sangat dipengaruhi oleh pemilihan arsitektur jaringan dan parameter pelatihan. Konfigurasi terbaik pada penelitian ini diperoleh dengan menggunakan dua *hidden layer* yang terdiri atas 32 dan 64 neuron, *learning rate* sebesar 0,01, serta rasio pembagian data 80% untuk data latih dan 20% untuk data uji. Konfigurasi tersebut mampu menghasilkan performa tertinggi dengan nilai *accuracy* sebesar 83%, *precision* 79%, *recall* 81%, dan *F1-score* 79%, yang menunjukkan kemampuan model dalam mengklasifikasikan status anemia berbasis parameter *Complete Blood Count* (CBC) secara cukup optimal.

Selain itu, hasil penelitian menunjukkan bahwa penerapan teknik penyeimbangan data menggunakan SMOTE memberikan dampak positif terhadap performa model dibandingkan skenario Non-SMOTE. Data latih yang lebih seimbang membantu model mengenali kelas minoritas dengan lebih baik, sehingga nilai *accuracy*, *precision*, *recall*, dan *F1-score* meningkat secara konsisten. Secara keseluruhan, kombinasi penerapan SMOTE, arsitektur dua *hidden layer* dengan 32–64 neuron, dan *learning rate* 0,01 dapat dinyatakan sebagai konfigurasi paling optimal dalam penelitian ini dan layak digunakan untuk klasifikasi status anemia.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan variasi metode penyeimbangan data selain SMOTE. Beberapa kelas minoritas masih sulit dikenali oleh model, sehingga perlu dieksplorasi teknik lain seperti ADASYN, Borderline-SMOTE, atau SMOTEENN.
2. Menambahkan lebih banyak fitur klinis selain parameter CBC. Penelitian berikut dapat menggabungkan parameter tambahan seperti riwayat medis, gejala klinis, atau data biokimia untuk meningkatkan kemampuan model dalam membedakan kelas anemia yang memiliki pola CBC mirip satu sama lain.
3. Mengoptimalkan pemilihan hyperparameter secara otomatis. Metode seperti *grid search*, *random search*, atau *Bayesian optimization* dapat diterapkan untuk memperoleh kombinasi *learning rate*, jumlah neuron, dan *batch size* yang lebih optimal.

DAFTAR PUSTAKA

- Aboelnaga, E. (2024). *Anemia Types Classification*. <https://www.kaggle.com/datasets/ehababoelnaga/anemia-types-classification>
- Adawiyah, R., Nita Budiyaniti, Syifa Meilinda, & Sandhy Fauzan Ramdansyah. (2023). Tinjauan Literatur Perbedaan Icd-10 Dan Icd-11: Literature Review the Differences of Icd-10 and Icd-11. *Intan Husada: Jurnal Ilmiah Keperawatan*, 11(02), 60–70. <https://doi.org/10.52236/ih.v11i2.269>
- Afrah, A. S. (2023). Sistem Diagnosa Penyakit Liver Menggunakan Metode Artificial Neural Network: Studi Berdasarkan Dataset Indian Liver Patient Dataset. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(3), 308–312. <https://doi.org/10.30591/jpit.v8i3.5346>
- Allorerung, P. P., Erna, A., Bagussahrir, M., & Alam, S. (2024). Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada Dataset Penyakit. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 9(3), 178–191. <https://doi.org/10.14421/jiska.2024.9.3.178-191>
- Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2020). Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman. *Vol ., 1*.
- Astuti, D., & Kulsum, U. (2020). Pola Menstruasi Dengan Terjadinya Anemia Pada Remaja Putri. *Jurnal Ilmu Keperawatan dan Kebidanan*, 11(2), 314. <https://doi.org/10.26751/jikk.v11i2.832>
- Choudhary, R., Koshti, A., & Malik, R. (2020). Study of Morphological Pattern of Anaemia at a Tertiary Care Centre. *Journal of Evolution of Medical and Dental Sciences*, 9(8), 468–472. <https://doi.org/10.14260/jemds/2020/106>
- Danarsih, D. E., Kusumawardani, A. M., & Ariningtyas, R. E. (2024). Deteksi Dini Anemia Dan Edukasi Pola Hidup Sehat Pada Remaja. *GEMAKES: Jurnal Pengabdian Kepada Masyarakat*, 4(3), 317–323. <https://doi.org/10.36082/gemakes.v4i3.1700>
- Dastres, R., & Soori, M. (2021). *Artificial Neural Network Systems*.
- Hairani, H., Widiyaningtyas, T., & Dwi Prasetya, D. (2024). Addressing Class Imbalance of Health Data: A Systematic Literature Review on Modified Synthetic Minority Oversampling Technique (SMOTE) Strategies. *JOIV: International Journal on Informatics Visualization*, 8(3), 1310. <https://doi.org/10.62527/joiv.8.3.2283>

- Hananti, H., & Sari, K. (2021). Perbandingan Metode Support Vector Machine (SVM) dan Artificial Neural Network (ANN) pada Klasifikasi Gizi Balita. *Seminar Nasional Official Statistics*, 2021(1), 1036–1043. <https://doi.org/10.34123/semnasoffstat.v2021i1.1014>
- Hermaliani, E. H., & Ernawati, M. (2024). Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujaran Kebencian. *Computer Science*, 4(1).
- ICD-11. (2022). <https://icd.who.int/en/>
- Karagül Yıldız, T., Yurtay, N., & Öneç, B. (2021). Classifying anemia types using artificial learning methods. *Engineering Science and Technology, an International Journal*, 24(1), 50–70. <https://doi.org/10.1016/j.jestch.2020.12.003>
- Lidwina Priliani, Alida R. Harahap, Ari W. Satyagraha, Rintis Noviyanti, Isabella Apriyana, Illene Nanine, Herawati Sudoyo, & Safarina G. Malik. (2025). Mapping anemia prevalence across Indonesia. *Asia Pacific Journal of Clinical Nutrition*, 34(3). [https://doi.org/10.6133/apjcn.202506_34\(3\).0017](https://doi.org/10.6133/apjcn.202506_34(3).0017)
- Mardanus, D. U. (2024). Klasifikasi Data Serangan Jantung Menggunakan Metode Super Vector Machine Dan Artificial Neural Network. *Jurnal Komputer Dan Informatika*, 19(2), 129–138.
- Meilina, L., Kumara, I. N. S., & Setiawan, I. N. (2021). Literature Review Klasifikasi Data Menggunakan Metode Cosine Similarity dan Artificial Neural Network. *Majalah Ilmiah Teknologi Elektro*, 20(2), 307. <https://doi.org/10.24843/MITE.2021.v20i02.P15>
- Mohammad Soharto, Mohammad Aldy Fermansyah Hadi, & Maulana Firdaus Al-Ayyubi. (2024). Klasifikasi Penyakit Jantung Menggunakan Algoritma ANN dengan Persentase diatas 70%. *Jurnal Kendali Teknik dan Sains*, 2(3), 249–255. <https://doi.org/10.59581/jkts-widyakarya.v2i3.3922>
- Mustakim, F., Fauziah, F., & Hayati, N. (2021). Algoritma Artificial Neural Network pada Text-based Chatbot Frequently Asked Question (FAQ) Web Kuliah Universitas Nasional. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5(4), 438–446. <https://doi.org/10.35870/jtik.v5i4.261>
- Nasruddin, H., & Syamsu, R. F. (2021). *Angka Kejadian Anemia Pada Remaja Di Indonesia*.
- Niendy Alexandra Yosephine & Ratnadewi. (2021a). Penggunaan Artificial Neural Network pada Sinyal Elektrokardiogram untuk Mendeteksi Penyakit Jantung Aritmia Supraventrikular. *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, 13(1), 14–23. <https://doi.org/10.37424/informasi.v13i1.69>

- Niendy Alexandra Yosephine & Ratnadewi. (2021b). Penggunaan Artificial Neural Network pada Sinyal Elektrokardiogram untuk Mendeteksi Penyakit Jantung Aritmia Supraventrikular. *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, 13(1), 14–23. <https://doi.org/10.37424/informasi.v13i1.69>
- Pamela, D. D. A., Nurmala, I., & Ayu, R. S. (2022). Faktor Risiko Dan Pencegahan Anemia Pada Wanita Usia Subur Di Berbagai Negara. *IKESMA*, 18(3), 161. <https://doi.org/10.19184/ikesma.v18i3.26510>
- Permata, I. I., Achyar, K., & Kusuma, I. R. (2023). Faktor-Faktor yang Mempengaruhi Anemia. *Jurnal Riset Kesehatan Masyarakat*, 3(3), 135–142. <https://doi.org/10.14710/jrkm.2023.18849>
- Pradana, D., Alghifari, M. L., Juna, M. F., & Palaguna, D. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55–60. <https://doi.org/10.56705/ijodas.v3i2.35>
- Purwono, P., Dewi, P., Wibisono, S. K., & Dewa, B. P. (2022). Model Prediksi Otomatis Jenis Penyakit Hipertensi dengan Pemanfaatan Algoritma Machine Learning Artificial Neural Network. *Insect (Informatics and Security): Jurnal Teknik Informatika*, 7(2), 82–90. <https://doi.org/10.33506/insect.v7i2.1828>
- Riany, A. F., & Testiana, G. (2023). Penerapan Data Mining untuk Klasifikasi Penyakit Jantung Koroner Menggunakan Algoritma Naïve Bayes. *MDP Student Conference*, 2(1), 297–305. <https://doi.org/10.35957/mdp-sc.v2i1.4388>
- Sulistiyono, M., Pristyanto, Y., Adi, S., & Gumelar, G. (2021). Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada Dataset Klasifikasi. *SISTEMASI*, 10(2), 445. <https://doi.org/10.32520/stmsi.v10i2.1303>
- Swana, E. F., Doorsamy, W., & Bokoro, P. (2022). Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset. *Sensors*, 22(9), 3246. <https://doi.org/10.3390/s22093246>

LAMPIRAN

LAMPIRAN-LAMPIRAN

Lampiran I. Source Code Bangun *Model Artificial Neural Network*

```
class ManualANN:
    def __init__(self, input_dim, hidden_layers, neurons, num_classes,
                 learning_rate=0.001, beta1=0.9, beta2=0.999, eps=1e-8):

        self.input_dim = input_dim
        self.hidden_layers = hidden_layers
        self.num_classes = num_classes
        self.learning_rate = learning_rate
        self.beta1 = beta1
        self.beta2 = beta2
        self.eps = eps

        # ===== INISIASI BOBOT DAN BIAS =====
        self.weights = []
        self.biases = []

        if isinstance(neurons, int):
            self.neuron_counts_per_layer = [neurons] * hidden_layers
        else:
            self.neuron_counts_per_layer = neurons

        prev = input_dim
        for n in self.neuron_counts_per_layer:
            self.weights.append(np.random.randn(prev, n) * np.sqrt(2.0 / prev))
            self.biases.append(np.zeros((1, n)))
            prev = n
        self.weights.append(np.random.randn(prev, num_classes) * np.sqrt(1.0 / prev))
        self.biases.append(np.zeros((1, num_classes)))

        # Adam moment vectors
        self.m_w = [np.zeros_like(w) for w in self.weights]
        self.v_w = [np.zeros_like(w) for w in self.weights]
        self.m_b = [np.zeros_like(b) for b in self.biases]
        self.v_b = [np.zeros_like(b) for b in self.biases]
        self.t = 0
```

```
# ===== FORWARD =====
#
# ===== FORWARD =====
def forward(self, X):
    self.layer_inputs = []
    self.layer_outputs = []

    a = X

    # hidden layers
    for i in range(self.hidden_layers):
        self.layer_outputs.append(a)
        z = np.dot(a, self.weights[i]) + self.biases[i]
        self.layer_inputs.append(z)
        a = relu(z)

    # output layer
    self.layer_outputs.append(a)
    z_out = np.dot(a, self.weights[-1]) + self.biases[-1]
    self.layer_inputs.append(z_out)
    out = softmax(z_out)

    self.layer_outputs.append(out)
    return out
```

```

# =====
# BACKPROPAGATION
# =====
def compute_gradients(self, X, y):
    m = X.shape[0]

    d = self.layer_outputs[-1] - y # softmax derivative

    dW = []
    dB = []

    # output layer gradient
    dW.insert(0, np.dot(self.layer_outputs[-2].T, d) / m)
    dB.insert(0, np.sum(d, axis=0, keepdims=True) / m)

    # hidden layers backward
    for i in range(self.hidden_layers - 1, -1, -1):
        d = np.dot(d, self.weights[i+1].T) * relu_deriv(self.layer_inputs[i])

        dW.insert(0, np.dot(self.layer_outputs[i].T, d) / m)
        dB.insert(0, np.sum(d, axis=0, keepdims=True) / m)

    return dW, dB

```

```

# =====
# TRAINING
# =====
def fit(self, X, y, epochs=100, batch_size=32, validation_data=None):
    history = {
        "train_loss": [], "val_loss": [],
        "train_acc": [], "val_acc": []
    }

    x_val, y_val = validation_data

    for epoch in range(epochs):

        # Shuffle
        idx = np.random.permutation(X.shape[0])
        X_shuff = X[idx]
        y_shuff = y[idx]

        # Mini batch
        for i in range(0, len(X_shuff), batch_size):
            X_batch = X_shuff[i:i+batch_size]
            y_batch = y_shuff[i:i+batch_size]

            # Training forward (WITH DROPOUT)
            _ = self.forward(X_batch, training=True)

            # Compute gradients
            dW, dB = self.compute_gradients(X_batch, y_batch)

```

```

# Adam update
self.t += 1
for l in range(len(self.weights)):

    self.m_w[l] = self.beta1 * self.m_w[l] + (1 - self.beta1) * dw[l]
    self.v_w[l] = self.beta2 * self.v_w[l] + (1 - self.beta2) * (dw[l] ** 2)

    m_w_hat = self.m_w[l] / (1 - self.beta1 ** self.t)
    v_w_hat = self.v_w[l] / (1 - self.beta2 ** self.t)

    self.weights[l] -= self.learning_rate * m_w_hat / (np.sqrt(v_w_hat) + self.eps)

    self.m_b[l] = self.beta1 * self.m_b[l] + (1 - self.beta1) * db[l]
    self.v_b[l] = self.beta2 * self.v_b[l] + (1 - self.beta2) * (db[l] ** 2)

    m_b_hat = self.m_b[l] / (1 - self.beta1 ** self.t)
    v_b_hat = self.v_b[l] / (1 - self.beta2 ** self.t)

    self.biases[l] -= self.learning_rate * m_b_hat / (np.sqrt(v_b_hat) + self.eps)

```

```

# ----- Evaluasi -----
y_pred_train = self.forward(X, training=False)
train_loss = cross_entropy_loss(y, y_pred_train)
train_acc = np.mean(np.argmax(y_pred_train, axis=1) ==
                    np.argmax(y, axis=1))

y_pred_val = self.forward(X_val, training=False)
val_loss = cross_entropy_loss(y_val, y_pred_val)
val_acc = np.mean(np.argmax(y_pred_val, axis=1) ==
                  np.argmax(y_val, axis=1))

history["train_loss"].append(train_loss)
history["val_loss"].append(val_loss)
history["train_acc"].append(train_acc)
history["val_acc"].append(val_acc)

print(f"Epoch {epoch+1}/{epochs} - "
      f"TrainLoss={train_loss:.4f} - TrainAcc={train_acc:.4f} | "
      f"ValLoss={val_loss:.4f} - ValAcc={val_acc:.4f}")

return history

# ===== PREDICT =====
def predict(self, X):
    y_pred = self.forward(X, training=False)
    return np.argmax(y_pred, axis=1)

```

Lampiran II. Graphical User Interface

AnemiaScope — Prediksi Status Anemia Berbasis ANN

Masukkan nilai Complete Blood Count (CBC) untuk memprediksi status anemia menggunakan Artificial Neural Network (ANN).

Tersedia mode **Input Manual** dan **Upload CSV (Batch)**. Hasil prediksi ditampilkan hingga **Top-9 kelas**.

Input Manual

Upload CSV (Batch)

Input Nilai CBC

WBC

Contoh WBC: 2-20 ($\times 10^3/\mu\text{L}$)

10

LYMp

Isi angka sesuai hasil CBC/dataset kamu.

43.2

NEUTp

Contoh Neutrofil: 30-80 (%)

50.1

LYMn

Isi angka sesuai hasil CBC/dataset kamu.

4.3

NEUTn

Contoh Neutrofil: 30-80 (%)

5

RBC

Contoh RBC: 2.5-6.5 ($\times 10^6/\mu\text{L}$)

2.77

HGB

Contoh Hb: 8-17 g/dL

7.3

HCT

Contoh HCT/PCV: 20-55 (%)

24.2

MCV

Contoh MCV: 60-110 (fL)

87.7

MCH

Contoh MCH: 18-36 (pg)

26.3

MCHC

Contoh MCH: 18-36 (pg)

30.1

PLT

Contoh PLT: 50-600 ($\times 10^3/\mu\text{L}$)

189

PDW

Isi angka sesuai hasil CBC/dataset kamu.

12.5

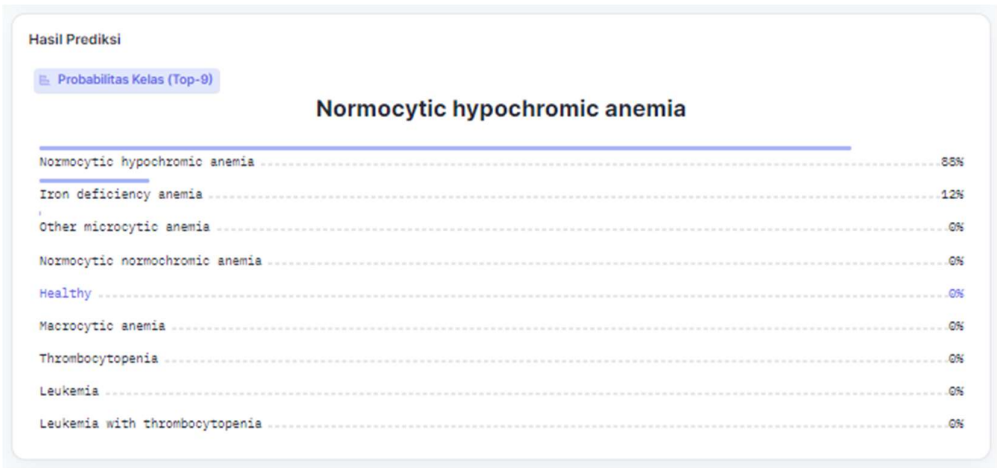
PCT

Isi angka sesuai hasil CBC/dataset kamu.

0.17

Prediksi

Reset



 **AnemiaScope — Prediksi Status Anemia Berbasis ANN**

Masukkan nilai Complete Blood Count (CBC) untuk memprediksi status anemia menggunakan Artificial Neural Network (ANN).

Tersedia mode **Input Manual** dan **Upload CSV (Batch)**. Hasil prediksi ditampilkan hingga **Top-9** kelas.

 **Input Manual**

 **Upload CSV (Batch)**

Upload CSV untuk Prediksi Massal

 Panduan Kolom CSV

Upload CSV

diagnosed_cbc_data_v4.csv122.6 KB

Proses CSV

Hasil Prediksi Batch

HCT	MCV	MCH	MC...	PLT	PDW	PCT	Diagnosis	pred_label	pred_confidence
42.9	87.2	28.6	32.8	160	14.9	0.16	Healthy	Healthy	0.9990534186363
37.1	87.1	29.1	33.6	155	12	0.14	Normocytic normochromic anem	Normocytic normochromic anem	0.6460832953453
43.5	85.2	27.6	32.4	181	15.9	0.19	Healthy	Healthy	0.9996230066217
41.9	88.2	28.6	32.4	132	16.4	0.13	Thrombocytopenia	Thrombocytopenia	0.9913406372076
42.8	84.6	27.6	32.7	120	19	0.13	Thrombocytopenia	Thrombocytopenia	0.9997611045837
44.4	85.1	26.6	31.5	189	17.2	0.19	Healthy	Healthy	0.9963021278381
39.5	75.2	23.2	31.1	171	12.8	0.16	Iron deficiency anemia	Iron deficiency anemia	0.9960891604423
38.6	80	24.6	31	170	12	0.15	Normocytic hypochromic anemia	Normocytic hypochromic anemia	0.7911916375166
41.4	82.6	26.2	31.6	171	14.9	0.17	Healthy	Healthy	0.9824364185333
37.7	80.2	24.6	30.7	176	15.6	0.18	Normocytic hypochromic anemia	Iron deficiency anemia	0.9052025079727
39.1	80.2	25	31.2	157	12.8	0.14	Normocytic hypochromic anemia	Normocytic hypochromic anemia	0.8444075584411
34.8	65.6	19.1	29.5	37	8.4	0.03	Iron deficiency anemia	Iron deficiency anemia	0.9998347759246

Catatan: Range nilai pada input hanya sebagai panduan. Nilai terbaik tetap mengikuti hasil pemeriksaan laboratorium atau dataset penelitian.

Lampiran III. Perhitungan Manual *Artificial Neural Network* (ANN)

Seluruh perhitungan pada percobaan ini menggunakan data *Complete Blood Count* (CBC). Arsitektur jaringan terdiri atas 14 neuron pada lapisan input, 16 dan 32 neuron pada lapisan tersembunyi (*hidden layer*), dan 9 neuron pada lapisan keluaran (*output layer*).

1. Inisialisasi Bobot dan Bias

Bobot awal ditentukan secara acak dengan nilai *learning rate* (α) sebesar 0,01. Nilai bobot acak dari lapisan input menuju lapisan tersembunyi ditunjukkan pada Tabel L.1, sedangkan bobot dari lapisan tersembunyi menuju lapisan keluaran ditunjukkan pada Tabel L.2. Nilai bias untuk setiap neuron pada lapisan tersembunyi disajikan pada Tabel L.3 dan hasil perhitungan zinj hidden layer 1.

Tabel L.1 Nilai bobot acak input menuju lapisan tersembunyi

Bobot	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
X_1	0.9862 42	- 0.2200 38	1.2909 53	0.71052 5	- 1.46096 2	1.71224 0	0.94010 3	1.02983 1
X_2	0.1965 05	- 1.5702 58	1.1794 72	0.47399 2	0.92911 6	- 0.52370 7	1.69451 3	1.41523 6
X_3	- 0.6270 29	- 0.4663 45	- 0.1095 99	- 1.11790 3	- 1.33228 3	- 0.08746 2	- 0.98312 6	0.61133 0
X_4	0.6569 84	- 1.2968 91	- 1.0803 30	- 1.77349 6	1.03292 8	0.59346 3	0.73859 5	1.01062 5
X_5	0.4849 86	0.1928 86	0.2131 46	- 0.70578 0	- 1.68905 6	- 0.22781 7	- 1.02749 5	- 0.32929 7
X_6	0.1765 50	1.5010 62	1.2553 94	- 0.75542 0	0.54750 5	0.21881 2	- 1.63220 0	0.88193 4
X_7	- 0.3041 26	- 1.0663 00	- 1.1092 53	- 0.37078 4	- 0.68202 0	0.61061 3	0.88375 4	- 0.50605 0
X_8	1.0536 95	0.0220 11	- 0.0492 53	- 1.85245 0	- 0.15103 5	1.24311 8	0.24098 2	- 1.21636 7

X_9	- 0.7880 07	- 1.3551 28	0.9293 02	0.14457 3	0.73703 2	1.28001 0	0.84940 1	- 0.06382 5
X_{10}	1.3807 65	- 0.5273 03	- 1.6699 19	1.54491 7	0.01845 2	1.01422 5	0.95613 0	- 0.54104 1
X_{11}	- 1.0658 89	0.8704 88	0.0259 61	- 0.88020 5	0.35280 3	- 1.40117 4	- 0.55145 3	1.65696 8
X_{12}	- 1.5196 46	- 0.8927 96	0.8742 81	- 0.43896 6	1.52125 7	1.55222 6	- 0.13491 6	- 0.79979 7
X_{13}	0.6301 35	- 0.9751 76	- 1.5150 10	- 1.01196 7	- 0.04751 9	0.07067 0	0.65326 6	1.63312 6
X_{14}	1.7079 29	0.8400 13	0.5094 70	- 0.90985 3	1.27711 7	1.43217 5	0.67367 5	0.18960 3

Bobot	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16
X_1	1.3387 91	- 0.1786 11	- 0.4651 27	1.53635 4	0.51791 4	1.16194 2	- 0.20370 9	- 0.98194 1
X_2	1.0021 81	- 1.0993 01	- 0.1198 04	- 1.64230 6	- 1.24455 8	0.65897 6	0.88114 4	1.68303 5
X_3	- 0.2262 53	1.1976 42	0.7209 54	- 0.67548 0	1.19613 5	1.09715 2	- 0.40507 8	- 0.76201 9
X_4	- 0.1479 03	0.2474 68	- 1.2967 31	- 1.38769 2	0.60625 1	- 0.10405 4	0.23485 0	0.95399 6
X_5	1.2722 51	- 0.9578 18	- 1.5901 10	- 0.78701 8	- 0.74306 2	0.58289 9	0.20531 6	1.02203 4
X_6	- 1.5103 85	0.7485 03	- 0.7778 75	0.19096 8	0.71026 7	- 1.12465 6	- 0.08230 8	- 1.25564 9
X_7	1.2741 50	0.5954 94	0.6045 60	0.79845 6	0.34635 9	- 0.26680 4	0.23121 4	0.03306 4
X_8	- 0.3428 69	0.7714 42	- 0.0633 76	0.31969 8	- 0.59745 4	1.30729 3	- 0.39630 8	0.89542 8
X_9	0.6550 08	- 1.1234 71	- 1.5118 62	- 0.57350 0	0.32224 1	- 0.67371 4	- 1.44982 3	1.24332 3
X_{10}	- 0.5469 63	0.2304 52	0.8658 81	- 0.75426 0	0.24594 8	- 0.61070 5	0.66343 6	1.32491 8
X_{11}	- 0.3715 80	- 1.1361 29	- 1.0978 77	1.49500 8	- 0.99282 0	- 0.47456 0	- 1.58733 5	0.80966 4

X_{12}	0.8879 46	- 0.3450 75	- 1.4216 63	1.60574 5	- 1.06221 9	1.75555 7	- 1.08230 0	0.61119 1
X_{13}	0.0442 85	0.4390 12	1.2529 93	0.26673 0	- 0.86048 5	1.57710 2	- 1.43699 8	- 0.10710 2
X_{14}	0.1316 91	- 0.6978 29	- 1.4463 18	- 0.21012 5	- 0.03903 6	- 0.78317 3	0.74573 2	1.33094 8

Tabel L.2 Nilai bobot acak lapisan tersembunyi menuju keluaran

Input	Bobot
$W_{1\ 1}$	0.986242
$W_{2\ 1}$	0.196505
$W_{3\ 1}$	-0.627029
$W_{4\ 1}$	0.656984
$W_{5\ 1}$	0.484986
$W_{6\ 1}$	0.176550
$W_{7\ 1}$	-0.304126
$W_{8\ 1}$	1.053695
$W_{9\ 1}$	-0.788007
$W_{10\ 1}$	1.380765
$W_{11\ 1}$	-1.065889
$W_{12\ 1}$	-1.519646
$W_{13\ 1}$	0.630135
$W_{14\ 1}$	1.707929

Tabel L.3 Nilai bias pada input menuju lapisan tersembunyi

Bias	Nilai
$b_{1\ 1}$	-0.384397
$b_{1\ 2}$	0.764655
$b_{1\ 3}$	-0.855253
$b_{1\ 4}$	0.099356
$b_{1\ 5}$	0.241155
$b_{1\ 6}$	0.192745
$b_{1\ 7}$	-0.211288
$b_{1\ 8}$	0.068035
$b_{1\ 9}$	0.132318
$b_{1\ 10}$	-0.418086
$b_{1\ 11}$	0.587761
$b_{1\ 12}$	-0.547180
$b_{1\ 13}$	0.114728
$b_{1\ 14}$	0.215038
$b_{1\ 15}$	-0.649676
$b_{1\ 16}$	-0.173955

Tabel L.4 hasil perhitungan zinj hidden layer 1

Neuron	Z_{inj}
Z_1	-0.783481
Z_2	0.579780
Z_3	-0.479581
Z_4	0.565049
Z_5	1.059353
Z_6	0.328588
Z_7	0.070665
Z_8	0.647236
Z_9	-0.798511
Z_{10}	-1.376942
Z_{11}	-1.527273
Z_{12}	-0.179607
Z_{13}	-0.105975
Z_{14}	1.058028
Z_{15}	-1.488693
Z_{16}	1.250904

2. Tahap I Propagasi Maju (Forward Propagation)

Perhitungan total input menuju neuron lapisan tersembunyi menggunakan Persamaan (3.2):

$$Z_{inj} = b_{ij} + \left(\sum_{i=1}^n X_{i,j} \times V_{i,j} \right)$$

Sebagai contoh perhitungan pada neuron pertama:

$$Z_{in\ 1} = b_{1,1} + (X_1 \times V_{1,1}) + (X_2 \times V_{2,1}) + (X_3 \times V_{3,1}) + (X_4 \times V_{4,1}) + (X_5 \times V_{5,1}) + (X_6 \times V_{6,1}) + (X_7 \times V_{7,1}) + (X_8 \times V_{8,1}) + (X_9 \times V_{9,1}) + (X_{10} \times V_{10,1}) + (X_{11} \times V_{11,1}) + (X_{12} \times V_{12,1}) + (X_{13} \times V_{13,1}) + (X_{14} \times V_{14,1})$$

$$z_{11} = -0.384397 + 0.202081 + 0.085351 - 0.005827 + 0.064764 + 0.027799 + 0.002783 - 0.054190 + 0.006300 - 0.123080 + 0.006846 - 0.243782 - 0.418642 + 0.029158 + 0.020100 = -0.783481$$

Tabel L.5 Tabel Nilai Aktivasi Hidden (ReLU)

Neuron	Z_{inj}	Aktivasi ReLU Z_j
Z_1	-0.783481	0
Z_2	0.579780	0.579780
Z_3	-0.479581	0
Z_4	0.565049	0.565049
Z_5	1.059353	1.059353
Z_6	0.328588	0.328588
Z_7	0.070665	0.070665
Z_8	0.647236	0.647236
Z_9	-0.798511	0
Z_{10}	-1.376942	0
Z_{11}	-1.527273	0
Z_{12}	-0.179607	0
Z_{13}	-0.105975	0
Z_{14}	1.058028	1.058028
Z_{15}	-1.488693	0
Z_{16}	1.250904	1.250904

Aktivasi pada lapisan tersembunyi menggunakan fungsi ReLU, sesuai Persamaan

(3.3):

$$f(x) = ReLU(x) = \max(0, x)$$

Tabel L.6 Z2k dan aktivasi ReLU pada hidden layer 2

Neuron	Z_2	Z_4	Z_5	Z_6	Z_7	Z_8	Z_{14}	Z_{16}	a_{jk}	Z_{inm}	Z_m (ReLU)
H2_1	0.274 245	- 0.461 018	0.640 104	- 1.190 430	- 0.57 5776	- 0.86 8605	- 0.34 4869	0.08 1441	0.21 7383	- 0.46 3064	0.00 0000
H2_2	- 0.935 986	- 0.732 395	0.714 013	- 0.897 522	- 0.83 3918	- 0.65 2703	1.31 5300	0.20 5405	- 0.18 8078	0.48 4078	0.48 4078
H2_3	0.675 406	0.554 381	- 0.313 853	0.359 797	- 0.35 0829	1.91 5623	- 0.99 6341	2.26 5727	- 0.06 6911	3.41 8791	3.41 8791
H2_4	0.846 508	1.016 075	- 0.416 117	- 0.814 931	- 1.23 9018	- 0.73 9248	- 0.80 5254	1.68 9160	- 0.68 7448	0.36 3854	0.36 3854
H2_5	- 1.755 932	- 0.102 553	0.772 178	- 0.340 346	0.57 1636	0.75 4640	0.57 8994	- 0.76 7919	- 0.11 9087	- 0.30 8089	0.00 0000
H2_6	- 1.171 899	- 0.353 441	0.140 806	0.047 560	1.64 4653	- 1.08 0448	- 0.44 7147	0.12 9308	0.35 6120	- 1.09 9629	0.00 0000

H2 _7	0.115 717	0.480 429	- 0.201 741	- 0.796 368	- 0.773 364	- 0.015 531	- 0.246 673	- 0.625 323	0.193 141	- 0.372 145	0.000 000
H2 _8	- 0.285 843	- 0.495 481	0.349 498	- 1.463 397	0.263 982	- 0.951 336	- 0.520 361	- 0.194 600	- 0.351 697	- 1.879 703	0.000 000
H2 _9	- 0.014 806	0.478 908	0.925 273	1.085 866	- 0.748 202	- 0.164 760	- 0.723 927	- 0.747 959	- 0.602 382	0.252 301	0.252 301
H2 _10	- 0.767 799	0.529 792	1.714 716	- 0.338 396	- 0.081 936	0.780 765	- 0.272 078	0.336 733	0.597 955	1.947 403	1.947 403
H2 _11	0.769 792	0.247 019	- 0.225 610	0.750 031	- 0.706 954	0.694 050	- 0.429 100	0.794 709	0.232 765	1.326 845	1.326 845
H2 _12	0.777 792	- 0.004 032	0.353 423	0.385 139	- 0.031 863	- 0.226 694	0.280 263	0.325 701	- 0.214 100	0.762 546	0.762 546
H2 _13	0.061 105	- 0.252 808	0.343 208	0.277 346	- 0.765 323	0.911 288	0.243 326	- 0.156 415	- 0.114 647	0.628 457	0.628 457
H2 _14	1.016 248	- 0.631 743	- 0.142 383	0.398 817	0.435 714	- 0.231 378	- 0.230 770	- 1.379 981	0.325 171	- 0.690 047	0.000 000
H2 _15	0.420 639	0.510 316	0.173 584	- 0.494 770	0.460 919	- 0.526 473	0.550 965	- 0.202 494	- 0.566 916	0.152 658	0.152 658
H2 _16	- 0.859 247	0.471 462	1.427 457	- 0.366 155	- 0.350 278	- 0.474 757	- 0.140 292	- 0.817 721	- 0.188 829	- 0.031 201	0.000 000
H2 _17	0.331 398	0.113 864	- 0.140 580	- 0.289 715	0.365 115	0.016 913	0.048 833	0.294 052	- 0.132 085	0.345 089	0.345 089
H2 _18	- 0.862 458	- 0.252 968	1.009 208	- 0.646 168	0.210 210	- 0.414 498	- 0.195 512	0.009 217	0.281 545	- 1.045 535	0.000 000
H2 _19	0.773 154	- 1.595 030	0.389 854	- 1.318 944	1.588 256	0.734 726	0.565 149	- 0.715 530	0.021 010	0.040 377	0.040 377
H2 _20	- 0.045 296	- 0.485 889	- 0.070 225	0.624 617	- 1.380 998	- 0.776 965	0.236 696	1.537 401	0.617 955	1.286 951	1.286 951
H2 _21	- 0.158 218	1.126 376	- 0.921 722	- 0.426 730	0.445 783	0.863 610	0.399 702	0.166 720	- 0.520 202	0.192 049	0.192 049
H2 _22	- 0.184 000	0.773 691	- 0.881 200	- 0.519 377	- 0.564 666	0.106 487	- 0.196 840	- 0.047 828	0.197 599	0.151 962	0.151 962
H2 _23	1.206 882	- 0.318 775	0.060 889	1.589 208	- 0.010 745	- 0.542 354	- 0.342 284	- 0.429 608	0.124 741	0.646 777	0.646 777

H2 _24	0.413 008	1.524 765	- 0.461 388	- 0.107 785	- 0.785 241	0.160 833	0.092 012	- 1.206 324	0.309 802	- 0.752 291	0.000 000
H2 _25	- 0.439 110	0.662 967	0.923 655	0.128 788	0.468 884	- 0.441 815	- 0.471 061	- 0.115 370	0.238 595	0.736 574	0.736 574
H2 _26	- 0.316 243	- 0.752 050	0.989 226	- 0.210 939	- 0.450 089	- 0.199 468	0.371 127	0.374 737	- 0.590 642	0.076 419	0.076 419
H2 _27	0.449 235	0.255 470	0.243 434	- 0.117 490	0.780 302	0.141 915	- 0.098 695	- 0.087 211	0.327 855	0.976 274	0.976 274
H2 _28	0.329 193	0.316 497	1.439 284	0.577 547	- 1.155 164	0.221 059	0.464 454	0.490 366	0.085 733	1.936 023	1.936 023
H2 _29	0.378 067	0.166 070	- 0.079 852	- 0.455 367	0.208 312	1.015 543	0.184 655	- 0.504 282	- 0.201 176	0.793 121	0.793 121
H2 _30	0.389 010	0.936 268	0.137 997	- 0.467 575	- 0.338 837	0.251 998	0.176 377	0.608 181	- 0.415 484	0.617 496	0.617 496
H2 _31	- 0.219 735	0.650 278	0.066 717	0.161 372	- 0.248 890	- 0.426 027	0.831 965	0.351 139	0.094 429	0.603 066	0.603 066
H2 _32	0.728 233	0.541 621	- 1.096 441	- 0.438 036	0.112 846	- 0.433 666	0.498 398	0.794 238	0.506 481	- 0.013 244	0.000 000

3. Tahap II Perhitungan Lapisan Keluaran

Total input ke neuron keluaran dihitung dengan Persamaan (3.4):

$$y_{ink} = a_{jk} + \left(\sum_{j=1}^m Z_j W_{j,k} \right)$$

$$\begin{aligned}
 y_{in5} = & b5 + (H21 \cdot W1) + (H22 \cdot W2) + (H23 \cdot W3) + (H24 \cdot W4) + (H25 \cdot W5) + (H26 \cdot W6) \\
 & + (H27 \cdot W7) + (H28 \cdot W8) + (H29 \cdot W9) + (H210 \cdot W10) + (H211 \cdot W11) + (H212 \cdot W12) \\
 & + (H213 \cdot W13) + (H214 \cdot W14) + (H215 \cdot W15) + (H216 \cdot W16) + (H217 \cdot W17) + (H218 \\
 & \cdot W18) + (H219 \cdot W19) + (H220 \cdot W20) + (H221 \cdot W21) + (H222 \cdot W22) + (H223 \cdot W23 \\
 & + (H224 \cdot W24) + (H225 \cdot W25) + (H226 \cdot W26) + (H227 \cdot W27) + (H228 \cdot W28) + (H229 \\
 & \cdot W29) + (H230 \cdot W30) + (H231 \cdot W31) + (H232 \cdot W32)
 \end{aligned}$$

$$y_{in5} = 0.10 + 3.108 = 3.208$$

Tabel L.7 Nilai pada neuron lapisan keluaran

Output	y_{ink}	y
Y ₁	1.962	0.121
Y ₂	0.884	0.041
Y ₃	2.365	0.181
Y ₄	0.722	0.035
Y ₅	3.208	0.336
Y ₆	0.966	0.045
Y ₇	0.557	0.027
Y ₈	0.794	0.039
Y ₉	1.452	0.075

Pengaktifan bobot pada output layer menggunakan fungsi softmax

$$y_k = \frac{e^{y_{ink}}}{\sum_{j=1}^C e^{y_{inj}}}$$

$$y_k = \frac{e^{y_{ink}}}{\sum_{j=1}^C e^{y_{inj}}} = \frac{1.000}{2.972} = 0.336$$

Berdasarkan hasil perhitungan *softmax* pada neuron kelima (Y₅), diperoleh nilai probabilitas sebesar 0.336. Nilai ini menunjukkan bahwa kelas Y₅ memiliki peluang terbesar dibandingkan kelas keluaran lainnya. Pada perhitungan *softmax*, digunakan pendekatan numerik stabil dengan mengurangi nilai logit maksimum sehingga nilai eksponensial Y_{net5} menjadi $\exp(0) = 1$. Sementara itu, total penjumlahan nilai eksponensial seluruh neuron keluaran bernilai 2.972. Perbandingan kedua nilai tersebut menghasilkan probabilitas *softmax* sebesar 0.336, yang menegaskan bahwa neuron Y₅ merupakan hasil prediksi utama model.

4. Tahap III Propagasi Mundur (Backpropagation)

Nilai error dihitung dengan Persamaan (3.6):

- Menghitung nilai error

$$\delta_k = y_k - t_k$$

$$\delta_5 = y_5 - t_5 = 0.373333 - 1.000000 = -0.626667$$

Tabel L.8 hasil perhitungan nilai error

Output	y_k	t_k	δ_k
Y ₁	0.134444	0	0.134444
Y ₂	0.045556	0	0.045556
Y ₃	0.201111	0	0.201111
Y ₄	0.038889	0	0.038889
Y ₅	0.373333	1	-0.626667
Y ₆	0.050000	0	0.050000
Y ₇	0.030000	0	0.030000
Y ₈	0.043333	0	0.043333
Y ₉	0.083333	0	0.083333

- b. Selanjutnya δ_k ini digunakan untuk menghitung besarnya koreksi bobot error untuk memperbaiki nilai W_{jk} dengan persamaan:

$$\Delta W_{jk} = \eta \times \delta_k \times Z_j$$

$$\Delta W_{3,5} = \eta \times \delta_5 \times Z_3 = 0.01 \cdot (-0.626667) \cdot (3.418791) = -0.021424$$

- c. Tahap selanjutnya adalah menghitung koreksi bias. Hasilnya kemudian digunakan untuk menyesuaikan nilai a_k berdasarkan rumus dibawah ini:

$$\Delta a_k = \eta \times \delta_k$$

$$\Delta a_5 = \eta \times \delta_5 = 0.01 \cdot (-0.626667) = -0.006267$$

Tabel L.9 hasil perhitungan koreksi bias pada lapisan output

Output	δ_k	$\eta \times \delta_k$
Y ₁	0.134444	0.001344
Y ₂	0.045556	0.000456
Y ₃	0.201111	0.002011
Y ₄	0.038889	0.000389
Y ₅	-0.626667	-0.006267
Y ₆	0.050000	0.000500
Y ₇	0.030000	0.000300
Y ₈	0.043333	0.000433

Y_9	0.083333	0.000833
-------	----------	----------

- d. Setiap unit di lapisan tersembunyi (Z_j) akan menjumlahkan semua sinyal error (delta) yang masuk dari lapisan berikutnya, setelah setiap sinyal tersebut dikalikan dengan bobotnya masing-masing menggunakan rumus dibawah ini:

$$\delta_{inj} = \sum_{k=1}^C \delta_k W_{jk}$$

$$\delta_{in5} = \sum_k \delta_5 \times W_{3,5} = (-0.626667)(0.40) = -0.250667$$

- e. Hasil penjumlahan dari langkah sebelumnya kemudian dikalikan dengan nilai turunan fungsi aktivasinya. Ini dilakukan untuk mendapatkan nilai akhir dari sinyal error (delta) untuk unit tersebut.

$$f'(Z_j) = 1, Z_j \geq 0$$

$$f'(Z_j) = 0, Z_j \leq 0$$

$$\delta_5 = \delta_{in5} \times f(Z_5) = -0.250667 \cdot 1 = -0.250667$$

- f. Delta j digunakan untuk menghitung koreksi error yang nantinya akan digunakan untuk memperbaiki nilai $V1j$:

$$\Delta V_{ij} = \eta \times \delta_j \times X_i$$

$$\Delta V_{2,5} = \eta \times \delta_1 \times X_1 = 0.01 \times (-0.250667) \times 0.579780 = -0.001453$$

- g. Sebagai langkah terakhir dalam satu siklus, nilai penyesuaian untuk bias (V_{0j}) dihitung, yang akan menyelesaikan proses pembaruan untuk lapisan tersembunyi.

$$\Delta b_{ij} = \eta \delta_j$$

$$\Delta V_{0,1} = \eta \times \delta_1 = 0.01 \times (-0.250667) = -0.0025066$$

5. Tahap IV Pembaruan Bobot

Pada langkah ini, semua bobot dan bias yang mengarah ke setiap unit di lapisan output akan diperbarui menggunakan rumus:

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

$$W_{[2,5]}(\text{baru}) = 0.400000 + (-0.021424) = 0.378576$$

Pada perhitungan di atas, bobot $W_{2,5}$ merepresentasikan hubungan antara neuron ke-2 pada lapisan tersembunyi kedua dengan neuron ke-5 pada lapisan output. Bobot awal sebesar 0.400000 diperbarui menggunakan nilai koreksi bobot $\Delta W_{2,5} = -0.021424$ yang diperoleh dari proses *backpropagation*. Dengan demikian, nilai bobot baru menjadi 0.378576. Pembaruan ini untuk menyesuaikan kontribusi neuron tersembunyi terhadap neuron keluaran untuk mengurangi kesalahan prediksi dan meningkatkan kinerja model secara bertahap. Untuk setiap unit di lapisan tersembunyi (Z_j), semua bobot dan bias yang datang dari lapisan input akan diperbarui nilainya.

$$V_{ij} (baru) = V_{ij} (lama) + \Delta V_{ij}$$

$$V_{[0,5]}(baru) = 0.772178 + (-0.001453) = 0.770725$$

Pada contoh pembaruan bobot $V_{0,5}$, bobot ini merepresentasikan hubungan antara neuron ke-5 pada lapisan tersembunyi dengan bias yang terhubung ke neuron tersebut. Nilai bobot awal sebesar 0.772178 diperbarui menggunakan nilai koreksi $\Delta V_{0,5} = -0.001453$ yang diperoleh dari proses *backpropagation*, sehingga menghasilkan bobot baru sebesar 0.770725. Pembaruan ini dilakukan untuk menyesuaikan tingkat aktivasi neuron tersembunyi berdasarkan sinyal kesalahan yang dihasilkan selama proses pelatihan. Setelah seluruh bobot dan bias diperbarui, sistem akan memeriksa apakah jumlah *epoch* maksimum telah tercapai. Jika belum, proses pelatihan dilanjutkan pada *epoch* berikutnya, sedangkan jika telah tercapai, proses pelatihan dihentikan dan model menghasilkan keluaran berupa hasil klasifikasi.