

**KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN
METODE *RECURRENT NEURAL NETWORK* (RNN)**



OLEH :

ANNISA SEPTIANI

09011181621026

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2020**

**KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN
METODE *RECURRENT NEURAL NETWORK* (RNN)**

SKRIPSI

**Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**



OLEH :

**ANNISA SEPTIANI
09011181621026**

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2020**

LEMBAR PENGESAHAN

**KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN
METODE *RECURRENT NEURAL NETWORK* (RNN)**

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

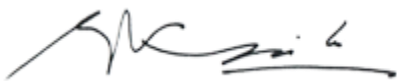
**ANNISA SEPTIANI
09011181621026**

Inderalaya, Januari 2020

Mengetahui,

Ketua Jurusan Sistem Komputer

Pembimbing Tugas Akhir



Dr. Ir. Sukemi, M.T
NIP. 196612032006041001



Prof. Dr. Ir. Siti Nurmaini, M.T.
NIP. 1196908021994012001

HALAMAN PERSETUJUAN



Telah diuji dan lulus pada :

Hari : Sabtu
Tanggal : 7 Maret 2020

Tim Penguji :

1. **Ketua** : Ahmad Zarkasih S.T., M.T.
2. **Sekretaris** : Prof. Dr. Ir. Siti Nurmaini, M.T.
3. **Anggota I** : Dr. Ir. Bambang Tutuko, M.T.
4. **Anggota II** : Rossi Passarella, S.T., M.Eng.





29/5/2020



APPROVED

**Mengetahui,
Ketua Jurusan Sistem Komputer**



Dr. Ir. Sukemi, M.T
NIP. 196612032006041001

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Annisa Septiani

NIM 09011181621026

Judul : Klasifikasi Penyakit Jantung Menggunakan Metode *Recurrent Neural Network* (RNN)

Hasil Pengecekan *Software iThenticate/Turnitin* : 7%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan / plagiat dari penelitian orang lain. Apabila ditemukan unsur penjiplakan / plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Inderalaya, Mei 2020

Yang menyatakan,



Annisa Septiani

HALAMAN PERSEMBAHAN

“Skripsi ini saya persembahkan untuk orang-orang yang selama ini telah meyakinkan saya bahwa saya dapat melalui perjuangan ini dan terus memberikan semangat serta kebahagiaan kepada saya. Terima kasih selalu menjaga saya dalam doa-doa serta memahami saya lebih dari saya sendiri. Saya akan memberikan yang terbaik dan tumbuh menjadi pribadi yang lebih baik lagi untuk kedepannya”

*Segenap hati berterima kasih dengan penuh rasa sayang
kepada :*

- *Ayah (Kornelis) dan Ibu (Yanti mala) tercinta*
- *Adik (Aulia Rahma, M. Okta Riansyah, M. Faris Syahputra) tersayang*
- *Teman-teman seperjuangan SKA dan Himasisko*
- *Keluarga Besar Sistem Komputer Universitas Sriwijaya*
- *Civitas Akademika Universitas Sriwijaya*

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah SWT, atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul **“Klasifikasi Penyakit Jantung Menggunakan Metode *Recurrent Neural Network* (RNN)”**.

Penulisan ini dilakukan untuk melengkapi salah satu syarat memperoleh gelar Sarjana Komputer di Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya. Dalam penulisan ini membahas mengenai klasifikasi penyakit jantung pada sinyal EKG menggunakan metode *Recurrent Neural Network* (RNN). Besar harapan untuk penelitian ini dapat bermanfaat untuk banyak orang dan menjadi referensi untuk peneliti lainnya yang tertarik pada bidang medis terkhusus penyakit jantung dan elektrokardiogram.

Pada kesempatan ini juga, saya sebagai penulis menyampaikan banyak ucapan terima kasih kepada semua pihak yang telah membantu baik dari ide, saran dan ilmunya sampai akhirnya penelitian ini dapat selesai dengan baik. Oleh karena itu, pada kesempatan ini saya mengucapkan rasa syukur kepada Allah SWT. dan mengucapkan terima kasih kepada yang terhormat :

1. Orang Tua dan keluarga yang saya sayangi, yang telah memberikan doa dan restu serta dukungan yang sangat besar selama penelitian ini dan selama masa perkuliahan di Fakultas Ilmu Komputer Jurusan Sistem Komputer Universitas Sriwijaya.
2. Bapak Jaidan Jauhari, S.Pd., M.T., selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
3. Bapak Rossi Passarella, S.T., M.Eng., selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Ibu Prof. Dr. Ir. Siti Nurmaini, M.T. selaku Dosen Pembimbing Tugas Akhir dan Akademik yang memberikan arahan serta nasihat pada penulisan tugas akhir serta masa perkuliahan di Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

5. Bapak Firdaus, S.T., M.Kom., Mba Nisa, dan Kak Naufal yang telah memberikan arahan dan nasihat dalam penulisan Tugas Akhir ini.
6. Mba Iis Oktaria, Kak Ahmad Reza Fahlevi, dan Mba Winda Kurnia Sari selaku admin Jurusan Sistem Komputer yang telah membantu mengurus seluruh berkas.
7. Seluruh bapak/ibu dosen yang ada di jurusan Sistem Komputer yang telah berbagi ilmunya selama masa perkuliahan ini.
8. Teman kelas SKA16 yang telah banyak membantu dalam masa perkuliahan selama ini, Khususnya Devi Maulitasari, Winda Maida, Dwi Ratna Ningsih, Tety Yuningsih telah menjadi teman curhat selama ini dan Ahmad Ilham Arismawan sebagai partner yang sangat banyak berjasa.
9. Seluruh teman-teman serta kakak tingkat Intelegent System Research Group (ISysRG) yang telah banyak membantu memberi saran dan berbagi ilmunya dalam membantu penulisan Tugas Akhir ini.
10. Teman-teman seperjuangan di Himasisko dan LDF Wifi.
11. Almamater

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan Tugas Akhir ini, oleh karena itu penulis mengharapkan segala saran dan kritik yang akan sangat bermanfaat bagi penulis untuk lebih baik kedepannya. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dan berguna bagi banyak orang khususnya pada mahasiswa/mahasiswi Fakultas Ilmu Komputer Universitas Sriwijaya..

Palembang, Mei 2020



Penulis

Heart Disease Classification with Recurrent Neural Network (RNN) Method

Annisa Septiani (0901181621026)

Departement of Computer Engineering, Faculty of Computer Science,
Sriwijaya University

Email: anissa.septiani325@gmail.com

Abstract

Heart disease is a blockage in the blood vessel system, which result in an imbalance between bood supply and demand. Heart signal can be detected by electrocardiogram record using an *Electrocardiograph* (ECG) tools. This study uses a *deep-learning* approach method, *Recurrent Neural Network* (RNN). With two cases, namely classifying unbalanced data (*Imbalance*) and balanced data (*Balance*). The first step is labeling the dataset *Dysrhythmia* (2), *Myocardical hypertrophy* (1), and *Healty controls*(0) using the *PTB Diagnostic ECG Database – Physionet* dataset. Next step is segmentation by *Windows size*. After that the resampling process is only done in the case of balance data using the *Random Oversampling* (ROS) method. The final step is to classify data with *Recurrent Neural Network* (RNN) and its classification model, namely SimpleRNN and LSTM. In this case, the best accuracy result is unbalanced data (*Imbalance*) with 99,81% training accuracy and 97,41% testing accuracy. As for the balanced data the training accuracy is 99,98% and and the testing accuracy is 99.96%. Validates values of the best tests on unbalanced data (*Imbalance*) averaged F1-Score 93.52%, precision 93.07%, sensitivity 93.99%, specificity 97.21%, and accuracy 98.09%. Validates values of the best tests on balanced data (average) are F1-Score 99.96, 99.96% precision, 99.96% sensitivity and 99.98% specificity, and 99.97% accuracy. The ROC curve and the Precision-Recall curve in the unbalanced data (*Imbalance*) and the balanced data (*Balance*) have gotten a good curve because the ROC curve is approaching the top left corner and the Precision-Recall Curve is approaching the top right corner which means each model classifier used is the right model.

Keywords: Heart Disease, *Recurrent Neural Network* (RNN), *Window Size*, *Random Oversampling* (ROS), *Imbalance*, SimpleRNN, LSTM.

Klasifikasi Penyakit Jantung menggunakan Metode Recurrent Neural Network (RNN)

Annisa Septiani (09011181621026)
Jurusan Sistem Komputer, Fakultas Ilmu Komputer,
Universitas Sriwijaya
Email: anissa.septiani325@gmail.com

Abstrak

Penyakit jantung ialah terjadinya penyumbatan pada sistem pembuluh darah yang mengakibatkan ketidakseimbangan antara suplai dan kebutuhan darah. Sinyal jantung dapat dideteksi dengan melakukan rekaman elektrokardiogram menggunakan alat *Electrocardiography* (ECG). Penelitian ini menggunakan metode pendekatan *deep-learning* yaitu *Recurrent Neural Network* (RNN). Menggunakan dua kasus yaitu mengklasifikasikan data tidak seimbang (*imbalance*) dan data seimbang (*balance*). Langkah pertama, pemberian label pada dataset yaitu *Dysrhythmia* (2), *Myocardial hypertrophy* (1), dan *Healthy controls* (0) dengan menggunakan dataset [The PTB Diagnostic ECG Database - PhysioNet](#). Selanjutnya dilakukan segmentasi yang dilakukan berdasarkan *window size*. Setelah itu untuk proses resampling hanya dilakukan untuk kasus data seimbang (*Balance*) menggunakan metode *Random Oversampling* (ROS). Langkah terakhir yaitu mengklasifikasikan data menggunakan *Recurrent Neural Network* (RNN) dengan model klasifikasinya yaitu SimpleRNN dan LSTM. Dalam hal ini didapatkan hasil akurasi terbaik yaitu data tidak seimbang (*Imbalance*) akurasi training 99.81% dan akurasi testing 97.14%. Sedangkan untuk data seimbang (*Balance*) akurasi training 99.98% dan akurasi testing 99.96%. Nilai validasi dari pengujian terbaik pada data tidak seimbang (*imbalance*) rata-ratanya yaitu F1-Score 93.52%, presisi 93.07%, sensitivitas 93.99%, spesifisitas 97.21%, dan akurasi 98.09%. Nilai validasi dari pengujian terbaik pada data seimbang (*balance*) rata-ratanya yaitu F1-Score 99.96%, presisi 99.96%, sensitivitas 99.96% dan spesifisitas 99.98%, dan akurasi 99.97%. Kurva ROC dan kurva Presisi-Recall pada data tidak seimbang (*Imbalance*) dan data seimbang (*Balance*) sudah mendapatkan kurva yang baik karna pada kurva ROC sudah mendekati pojok kiri atas dan Kurva Presisi-Recall mendekati pojok kanan atas yang berarti masing-masing model classifier yang digunakan merupakan model yang tepat.

Kata Kunci: Penyakit Jantung, Recurrent Neural Network (RNN), Window Size, Random Oversampling (ROS), Imbalance, SimpleRNN, LSTM.

DAFTAR ISI

	Halaman
Halaman Judul	i
Halaman Pengesahan	ii
Halaman Persetujuan	iii
Halaman Pernyataan	iv
Halaman Persembahan	v
Kata Pengantar	vi
Abstraction	viii
Abstrak	ix
Daftar isi	x
Daftar Gambar	xiii
Daftar Tabel	xv
Daftar Lampiran	
BAB I. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Rumusan Masalah	2
1.4. Sistematika Penulisan	3
BAB II. TINJAUAN PUSTAKA	
2.1. Pendahuluan	4
2.2. Elektrokardiogram	5
2.2.1. Gelombang EKG	5
2.3. PTB Diagnostic ECG Database	7
2.4. <i>Resampling</i> Data	8
2.5. <i>Machine Learning</i>	9
2.6. <i>Artificial Neural Network</i> (ANN)	9
2.6.1. Konsep Dasar <i>Artificial Neural Network</i> (ANN)	10

2.6.2. Permodelan <i>Artificial Neural Network</i> (ANN)	11
2.7. <i>Recurrent Neural Network</i> (RNN)	12
2.7.1. Vanilla RNN	13
2.7.2. <i>Long Short Term Memory</i> (LSTM)	14
2.7.3. <i>Gated Recurrent Unit</i> (GRU)	15

BAB III. METODOLOGI

3.1. Pendahuluan	16
3.2. Persiapan Data	17
3.3. <i>Pre-processing</i>	18
3.3.1. Segmentasi	18
3.3.2. Resampling Data	19
3.4. Klasifikasi <i>Recurrent Neural Network</i> (RNN)	21
3.5. Validasi	22
3.5.1. Akurasi	22
3.5.2. Sensitivitas	22
3.5.3. Spesifisitas	22
3.5.4. Presisi	23
3.5.5. F1-Score	23
3.5.6. Kurva ROC	23
3.5.7. Kurva Presisi-Recall	24

BAB IV. HASIL DAN PEMBAHASAN

4.1. Pendahuluan	25
4.2. Persiapan Dataset	25
4.2.1. Hasil Pengujian Segmentasi	26
4.2.2. Hasil Resampling Data	31
4.3. Klasifikasi Menggunakan <i>Recurrent Neural Network</i> (RNN)	32
4.3.1. Hasil Pengujian Klasifikasi Menggunakan <i>Recurrent Neural Network</i> (RNN) pada Data Tidak Seimbang (<i>Imbalance</i>)	32
4.3.1.1 Hasil Validasi Data Tidak Seimbang (<i>Imbalance</i>) pada Skenario133	
4.3.1.2 Hasil Validasi Data Tidak Seimbang (<i>Imbalance</i>) pada Skenario235	

433	Hasil Validasi Data Tidak Seimbang (<i>Imbalance</i>) pada Skenario337	
434	Hasil Validasi Data Tidak Seimbang (<i>Imbalance</i>) pada Skenario439	
4.3.2.	Hasil Pengujian Klasifikasi Menggunakan Recurrent Neural Network (RNN) pada Data Seimbang (<i>Balance</i>)	47
431	Hasil Validasi Data Seimbang (<i>Balance</i>) pada Skenario 1	48
432	Hasil Validasi Data Seimbang (<i>Balance</i>) pada Skenario 2	50
433	Hasil Validasi Data Seimbang (<i>Balance</i>) pada Skenario 3	52
434	Hasil Validasi Data Seimbang (<i>Balance</i>) pada Skenario 4	54
4.4.	Analisa	63
BAB V. KESIMPULAN DAN SARAN		
5.1.	Kesimpulan	69
5.2.	Saran	70
DAFTAR PUSTAKA		71

DAFTAR GAMBAR

Gambar 2.1. Morfologi Gelombang EKG	5
Gambar 2.2. Model <i>Arsitektur Neural Network</i>	11
Gambar 2.3. <i>Arsitektur Recurrent Neural Network</i>	13
Gambar 2.4. Struktur <i>Vanilla RNN</i>	13
Gambar 2.5. <i>Arsitektur pada Long Short Term Memory (LSTM)</i>	14
Gambar 3.1.a. Kerangka Kerja untuk data <i>Imbalance</i>	16
Gambar 3.1.b. Kerangka kerja untuk data <i>Balance</i>	16
Gambar 3.2. Pembagian Kelas	17
Gambar 3.3.a. Perbandingan dataset sebelum di-resampling	20
Gambar 3.3.b. Perbandingan dataset sebelum di-resampling	20
Gambar 3.4. Flowchart Model <i>Recurrent Neural Network (RNN)</i>	21
Gambar 4.1.a. Sinyal EKG berdasarkan window size 1 detik <i>Dysrhythmia</i>	27
Gambar 4.1.b. Sinyal EKG berdasarkan window size 1 detik <i>Hypertrophy</i>	27
Gambar 4.1.c. Sinyal EKG berdasarkan window size 1 detik <i>Healthy Control</i>	27
Gambar 4.2.a. Sinyal EKG berdasarkan window size 2 detik <i>Dysrhythmia</i>	27
Gambar 4.2.b. Sinyal EKG berdasarkan window size 2 detik <i>Hypertrophy</i>	27
Gambar 4.2.c. Sinyal EKG berdasarkan window size 2 detik <i>Healthy Control</i>	27
Gambar 4.3.a. Sinyal EKG berdasarkan window size 3 detik <i>Dysrhythmia</i>	27
Gambar 4.3.b. Sinyal EKG berdasarkan window size 3 detik <i>Hypertrophy</i>	27
Gambar 4.3.c. Sinyal EKG berdasarkan window size 3 detik <i>Healthy Control</i>	27
Gambar 4.4.a. Sinyal EKG berdasarkan window size 4 detik <i>Dysrhythmia</i>	27
Gambar 4.4.b. Sinyal EKG berdasarkan window size 4 detik <i>Hypertrophy</i>	27
Gambar 4.4.c. Sinyal EKG berdasarkan window size 4 detik <i>Healthy Control</i>	27
Gambar 4.5. Grafik data sebelum dan setelah dilakukan resampling	32
Gambar 4.6.a. Bentuk sinyal EKG skenario 1 pada data <i>Imbalance</i> akurasi	34
Gambar 4.6.b. Bentuk sinyal EKG skenario 1 pada data <i>Imbalance</i> loss	34
Gambar 4.7.a. Bentuk sinyal EKG skenario 2 pada data <i>Imbalance</i> akurasi	36
Gambar 4.7.b. Bentuk sinyal EKG skenario 2 pada data <i>Imbalance</i> loss	36
Gambar 4.8.a. Bentuk sinyal EKG skenario 3 pada data <i>Imbalance</i> akurasi	38

Gambar 4.8.b. Bentuk sinyal EKG skenario 3 pada data <i>Imbalance</i> loss	38
Gambar 4.9.a. Bentuk sinyal EKG skenario 4 pada data <i>Imbalance</i> akurasi	40
Gambar 4.9.b. Bentuk sinyal EKG skenario 4 pada data <i>Imbalance</i> loss	40
Gambar 4.10.a. Hasil pengujian kurva ROC terbaik dari skenario 1	46
Gambar 4.10.b. Hasil pengujian kurva ROC terbaik dari skenario 2	46
Gambar 4.10.c. Hasil pengujian kurva ROC terbaik dari skenario 3	46
Gambar 4.10.d. Hasil pengujian kurva ROC terbaik dari skenario 4	46
Gambar 4.11.a. Hasil pengujian presisi-recall terbaik dari skenario 1	47
Gambar 4.11.b. Hasil pengujian presisi-recall terbaik dari skenario 2	47
Gambar 4.11.c. Hasil pengujian presisi-recall terbaik dari skenario 3	47
Gambar 4.11.d. Hasil pengujian presisi-recall terbaik dari skenario 4	47
Gambar 4.12.a Grafik akurasi percobaan untuk data <i>Balance</i> skenario 1	49
Gambar 4.12.b Grafik loss percobaan untuk data <i>Balance</i> skenario 1	49
Gambar 4.13.a Grafik akurasi percobaan untuk data <i>Balance</i> skenario 2	51
Gambar 4.13.b Grafik loss percobaan untuk data <i>Balance</i> skenario 2	51
Gambar 4.14.a Grafik akurasi percobaan untuk data <i>Balance</i> skenario 3	53
Gambar 4.14.b Grafik loss percobaan untuk data <i>Balance</i> skenario 3	53
Gambar 4.15.a Grafik akurasi percobaan untuk data <i>Balance</i> skenario 4	55
Gambar 4.15.b Grafik loss percobaan untuk data <i>Balance</i> skenario 4	55
Gambar 4.16.a. Hasil pengujian kurva ROC terbaik dari skenario 1	62
Gambar 4.16.b. Hasil pengujian kurva ROC terbaik dari skenario 2	62
Gambar 4.16.c. Hasil pengujian kurva ROC terbaik dari skenario 3	62
Gambar 4.16.d. Hasil pengujian kurva ROC terbaik dari skenario 4	62
Gambar 4.17.a. Hasil pengujian presisi-recall terbaik dari skenario 1	63
Gambar 4.17.b. Hasil pengujian presisi-recall terbaik dari skenario 2	63
Gambar 4.17.c. Hasil pengujian presisi-recall terbaik dari skenario 3	63
Gambar 4.17.d. Hasil pengujian presisi-recall terbaik dari skenario 4	63
Gambar 4.18.a. Grafik hasil pengujian terbaik dari data tidak seimbang	67
Gambar 4.18.b. Grafik hasil pengujian terbaik dari data seimbang	67
Gambar 4.19. Kurva ROC pengujian data <i>Imbalance</i> dan <i>Balance</i>	67
Gambar 4.20. Kurva Presisi-Recall pengujian data <i>Imbalance</i> dan <i>Balance</i>	68

DAFTAR TABEL

Tabel 2.1. Dataset	8
Tabel 3.1. Label Dataset	18
Tabel 3.2. Jumlah data sebelum dan sesudah diresampling	19
Tabel 4.1. Pembagian label pada tiap kelas dataset	25
Tabel 4.2. Jumlah data sebelum dan sesudah diresampling	31
Tabel 4.3. Hasil pengujian data tidak seimbang (<i>Imbalance</i>) pada skenario 1	33
Tabel 4.4. Hasil pengujian data tidak seimbang (<i>Imbalance</i>) pada skenario 2	35
Tabel 4.5. Hasil pengujian data tidak seimbang (<i>Imbalance</i>) pada skenario 3	37
Tabel 4.6. Hasil pengujian data tidak seimbang (<i>Imbalance</i>) pada skenario 4	39
Tabel 4.7. Perbandingan nilai terbaik pengujian data <i>Imbalance</i> dari skenario pertama, skenario kedua, skenario ketiga, dan skenario keempat	41
Tabel 4.8. Hasil <i>confusion matrix testing</i> pada data tidak seimbang (<i>Imbalance</i>) skenario 1 percobaan ke-6	41
Tabel 4.9. Nilai <i>confusion matrix testing</i> dari data <i>Imbalance</i> untuk masing-masing kelas pada skenario pertama percobaan ke-6	42
Tabel 4.10. Hasil <i>confusion matrix testing</i> pada data tidak seimbang (<i>Imbalance</i>) skenario 2 percobaan ke-2	42
Tabel 4.11. Nilai <i>confusion matrix testing</i> dari data <i>Imbalance</i> untuk masing-masing kelas pada skenario kedua percobaan ke-2	42
Tabel 4.12. Hasil <i>confusion matrix testing</i> pada data tidak seimbang (<i>Imbalance</i>) skenario 3 percobaan ke-2	43
Tabel 4.13. Nilai <i>confusion matrix testing</i> dari data <i>imbalance</i> untuk masing-masing kelas pada skenario ketiga percobaan ke-2	43
Tabel 4.14. Hasil <i>confusion matrix testing</i> pada data tidak seimbang (<i>Imbalance</i>) skenario 4 percobaan ke-5	43
Tabel 4.15. Nilai <i>confusion matrix testing</i> dari data <i>imbalance</i> untuk masing-masing kelas pada skenario keempat percobaan ke-5	44
Tabel 4.16. Hasil performance data tidak seimbang (<i>Imbalance</i>) model RNN terbaik skenario 1	63

Tabel 4.17. Hasil performance data tidak seimbang (<i>Imbalance</i>) model RNN terbaik skenario 2	45
Tabel 4.18. Hasil performance data tidak seimbang (<i>Imbalance</i>) model RNN terbaik skenario 3	45
Tabel 4.19. Hasil performance data tidak seimbang (<i>Imbalance</i>) model RNN terbaik skenario 4	45
Tabel 4.20. Hasil pengujian data seimbang (<i>Balance</i>) pada skenario 1	48
Tabel 4.21. Hasil pengujian data seimbang (<i>Balance</i>) pada skenario 2	50
Tabel 4.22. Hasil pengujian data seimbang (<i>Balance</i>) pada skenario 3	52
Tabel 4.23. Hasil pengujian data seimbang (<i>Balance</i>) pada skenario 4	54
Tabel 4.24. Perbandingan nilai terbaik pengujian data <i>Balance</i> dari skenario pertama, skenario kedua, skenario ketiga, dan skenario keempat	56
Tabel 4.25. Hasil <i>confusion matrix testing</i> pada data seimbang (<i>Balance</i>) skenario 1 percobaan ke-2	57
Tabel 4.26. Nilai <i>confusion matrix testing</i> dari data <i>Balance</i> untuk masing-masing kelas pada skenario pertama percobaan ke-2	57
Tabel 4.27. Hasil <i>confusion matrix testing</i> pada data seimbang (<i>Balance</i>) skenario 2 percobaan ke-2	57
Tabel 4.28. Nilai <i>confusion matrix testing</i> dari data <i>Balance</i> untuk masing-masing kelas pada skenario kedua percobaan ke-2	58
Tabel 4.29. Hasil <i>confusion matrix testing</i> pada data seimbang (<i>Balance</i>) skenario 3 percobaan ke-2	58
Tabel 4.30. Nilai <i>confusion matrix testing</i> dari data <i>Balance</i> untuk masing-masing kelas pada skenario ketiga percobaan ke-2	59
Tabel 4.31. Hasil <i>confusion matrix testing</i> pada data seimbang (<i>Balance</i>) skenario 4 percobaan ke-2	59
Tabel 4.32. Nilai <i>confusion matrix testing</i> dari data <i>Balance</i> untuk masing-masing kelas pada skenario keempat percobaan ke-2	59
Tabel 4.33. Hasil performance data seimbang (<i>Balance</i>) model RNN terbaik skenario 1	60
Tabel 4.34. Hasil performance data seimbang (<i>Balance</i>) model RNN terbaik skenario 2	60

Tabel 4.35. Hasil performance data seimbang (<i>Balance</i>) model RNN terbaik skenario 3	61
Tabel 4.36. Hasil performance data seimbang (<i>Balance</i>) model RNN terbaik skenario 4	61
Tabel 4.37. Hasil pengujian terbaik pada data tidak seimbang (<i>Imbalance</i>)	64
Tabel 4.38. Hasil pengujian terbaik pada data seimbang (<i>Balance</i>)	65
Tabel 4.39. Hasil terbaik dari data <i>Imbalance</i> dan <i>Balance</i>	66

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyakit jantung ialah penyakit yang terus meningkat setiap tahun angka kematiannya. Data yang diperoleh dari World Health Statistics dalam World Health Organization (WHO) (2012) memperoleh hasil dari 57 juta angka kematian pada tahun 2008 dengan 48% disebabkan karena penyakit jantung [1]. Walaupun penyakit jantung merupakan penyakit yang tidak menular, namun penyakit ini merupakan jenis penyakit nomor satu mematikan di dunia. Penyakit jantung ialah terjadinya penyumbatan pada sistem pembuluh darah yang mengakibatkan ketidakseimbangan antara suplai dan kebutuhan darah [2]. Jantung merupakan organ penting pada tubuh manusia maka dari itu manusia selalu dituntut menjaga keadaan jantung dengan baik, tidak dapat dipungkiri bahwa seseorang bisa saja menderita penyakit jantung.

Tingginya faktor kematian yang disebabkan penyakit jantung dapat dicegah. Pengetahuan yang kurang dari masyarakat tentang gejala penyakit jantung, serta kurang akuratnya peralatan yang digunakan untuk mendiagnosa gejala penyakit jantung yang menyebabkan angka kematian terus bertambah [3]. *Electrocardiography* (ECG) merupakan salah satu cara untuk mendeteksi gejala terserang penyakit jantung [4]. Sinyal jantung dapat dideteksi dengan melakukan rekaman elektrokardiogram menggunakan alat *Electrocardiography* (ECG). Sinyal jantung manusia bervariasi pada setiap individu, dipengaruhi oleh usia, jenis kelamin, genetik, serta kondisi medisnya sendiri [5].

Sudah ada beberapa metode yang digunakan untuk mengklasifikasikan penyakit jantung, seperti metode *Naïve Bayes* dengan ekstraksi fitur menggunakan *Principle Component Analysis* (PCA) memberikan akurasi sebesar 77.4% [6], metode *Support Vector Machine* sebesar 87.5% dan metode *Backpropagation Neural Network* (BPNN) sebesar 85% [7]. Selain metode yang telah disebutkan ada

banyak lagi metode klasifikasi penyakit jantung yang telah dilakukan, namun tidak banyak yang menggunakan ekstraksi fitur dan menghasilkan akurasi yang tidak terlalu tinggi. Dibutuhkan metode yang baik untuk meningkatkan nilai akurasi dan mengurangi dimensi dari data yang jumlah fiturnya sangat banyak.

Metode-metode pada pendekatan *deep-learning* sangat disarankan karena telah banyak penelitian yang membuktikan bahwa nilai *performance* dan nilai akurasi menggunakan *deep-learning* menghasilkan nilai yang baik. Dalam beberapa penelitian, metode *Recurrent Neural Network* (RNN) berhasil digunakan untuk masalah klasifikasi dengan data sekuensial [8]. Oleh karena itu, *Recurrent Neural Network* (RNN) digunakan untuk melakukan klasifikasi data penyakit jantung pada penelitian ini.

Berdasarkan yang telah dipaparkan di atas, maka penelitian ini melakukan klasifikasi pada penyakit jantung menggunakan metode *Recurrent Neural Network* (RNN).

1.2 Tujuan

Penelitian ini memiliki tujuan sebagai berikut :

1. Membuat model untuk melakukan klasifikasi penyakit jantung dengan metode *Recurrent Neural Network* (RNN).
2. Mengevaluasi hasil dari model terbaik yang telah dirancang.
3. Menganalisa hasil keakurasian penelitian mendeteksi penyakit jantung menggunakan metode *Recurrent Neural Network* (RNN).

1.3 Rumusan Masalah

Rumusan masalah yang disimpulkan dari latar belakang yang telah dijelaskan di atas ialah sebagai berikut:

1. Bagaimana melakukan klasifikasi penyakit jantung pada sinyal ECG dengan metode *Recurrent Neural Network* (RNN) agar mendapatkan hasil akurasi dan *performance* yang tinggi.

2. Dataset yang pada penelitian ini menggunakan *The PTB Diagnostic ECG Database*.
3. Penelitian ini menggunakan simulasi program dengan bahasa pemrograman *Python* yang digunakan untuk mengklasifikasikan sinyal EKG.

1.4 Sistematika Penulisan

Berikut sistematika penulisan dalam penelitian ini:

BAB I PENDAHULUAN

Pada Bagian pertama bab mencantumkan latar belakang, tujuan, rumusan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada Bagian kedua bab dituliskan pengenalan mengenai penyakit jantung, *Electrocardiogram*, *PTB Diagnostic ECG Database*, *Resampling Data*, *Machine Learning*, pembahasan tentang *Artificial Neural Network* (ANN), serta sistem klasifikasi yaitu *Recurrent Neural Network* (RNN).

BAB III METODOLOGI

Pada Bagian ketiga bab menjelaskan metodologi bagaimana persiapan data, pre-processing data dan bagaimana model pengklasifikasian pada penyakit jantung dengan metode *Recurrent Neural Network* (RNN).

BAB IV HASIL DAN PEMBAHASAN

Pada Bagian keempat bab menjelaskan tentang hasil serta pembahasan dari melakukan klasifikasi penyakit jantung dengan metode *Recurrent Neural Network* (RNN).

BAB V KESIMPULAN DAN SARAN

Pada Bagian kelima berisi kesimpulan serta saran dari hasil dan pembahasan-pembahasan yang telah dilakukan pada penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Penyakit jantung adalah sebuah kondisi dimana jantung mengalami gangguan. Macam-macam bentuk gangguan yang terjadi seperti pada irama jantung, pembuluh darah jantung, katup jantung serta kelainan jantung yang terjadi sejak lahir. Jantung merupakan bagian terpenting dalam tubuh manusia, banyak kasus kematian yang diakibatkan oleh penyakit jantung. Beberapa faktor penyebab penyakit jantung itu sendiri karena tekanan darah tinggi, stress, diabetes, usia yang bertambah, kadar kolesterol tinggi, kelainan jantung sejak lahir dan sebagainya [4] [9]. Beberapa jenis penyakit jantung sebagai berikut:

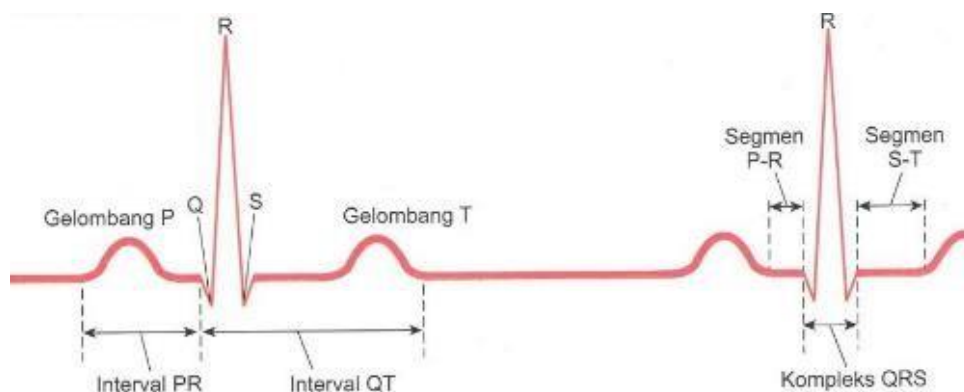
1. *Dysrhythmia*, merupakan gangguan pada sistem listrik dalam jantung yang bekerja untuk mengatur denyut jantung menyebabkan denyut jantung tidak beraturan seperti menjadi lebih lambat atau lebih cepat. Apabila jantung tidak bekerja dengan normal, jantung tidak dapat memompa darah dan mengakibatkan gangguan pada supply darah yang dapat merusak jantung dan organ penting lainnya. Gejala yang terjadi pada penderita *Dysrhythmia* seperti pusing, mual, mudah lelah, nyeri pada dada, denyut jantung tidak beraturan, sesak nafas serta dapat menyebabkan pingsan [10].
2. *Myocardial hypertrophy*, merupakan kondisi dimana sisi kiri pada dinding otot jantung (ventrikel) mengalami penebalan, yang mengalami peningkatan massa ventrikel. Ketika beban pada ventrikel kiri bertambah mengakibatkan otot ventrikel kiri jantung akan bekerja lebih keras. Gejala yang terjadi pada penderita *Myocardial hypertrophy* seperti mengalami pusing, jantung yang berdebar-debar, rasa nyeri pada dada, sesak napas serta cepat lelah [11].

2.2 Elektrokardiogram

Elektrokardiogram (EKG) merupakan sinyal yang menggambarkan kerja listrik pada jantung yang berguna untuk mendiagnosis penyakit jantung [4]. Proses ini efektif untuk mendeteksi keadaan jantung pada elektroda EKG yang dipasang pada tubuh pasien. Elektroda ditempelkan pada kulit agar dapat mendeteksi impuls listrik yang dikeluarkan jantung dan dicatat oleh Elektrokardiogram (EKG). Fungsi yang mendasar dari elektroda ialah mendeteksi ion-ion yang berjalan pada tubuh mengikuti pergerakan darah pada jantung. Transducer ini mengkonversi informasi biologis menjadi sebuah sinyal elektrik yang dapat diukur [12]. Elektrokardiogram (EKG) berisi 12 bentuk gelombang dan elektroda pada elektrokardiogram memiliki jumlah 10 buah untuk pengukuran.

2.2.1 Gelombang EKG

Aktivitas pada jantung ditampilkan berupa gelombang oleh elektrokardiogram (EKG). Gelombang EKG terdiri dari 4 tipe gelombang, yaitu gelombang P, QRS kompleks, T dan U [12]. Kompleks gelombang tersebut merupakan gabungan dari *beat* EKG. Elektrokardiogram (EKG) juga memiliki 2 jenis garis EKG yang dinamai interval dan segmen. Pada gambar 2.1 merupakan morfologi gelombang EKG.



Gambar 2. 1 Morfologi Gelombang EKG [13]

- Gelombang P, ialah hasil rekaman depolarisasi yang berada di miokardium atrium kanan dan kiri. Gelombang akan mengalami kelainan apabila atrium juga memiliki kelainan. Panjang atau durasi pada gelombang P kurang dari 0.12 detik., tinggi atau amplitudonya kurang dari 0.3 mV atau kurang dari 3 mm dan selalu positif di *lead* II dan negatif di *lead* aVR [13].
- Gelombang kompleks QRS, merupakan rekaman depolarisasi ventrikel kanan dan kiri. Gelombang QRS terdiri dari 3 gelombang yaitu [13]:
 1. Gelombang Q ialah gelombang defleksi yang negatif atau gelombang pertama yang menurun sesudah gelombang P.
 2. Gelombang R ialah gelombang defleksi yang positif atau gelombang pertama yang naik sesudah gelombang Q.
 3. Gelombang S ialah gelombang defleksi yang negatif atau menurun sesudah gelombang R.

Durasi gelombang kompleks QRS ialah durasi depolarisasi otot ventrikel yang memiliki lebar 0.06 sampai 0.12 detik [13].

- Gelombang T, merupakan gelombang dengan potensial repolarisasi di ventrikel kanan dan kiri. Gelombang T memiliki positif di *lead* I, II, V3-V6 dan negatif pada aVR [13].
- Gelombang U, merupakan gelombang yang berukuran kecil dan kadang tidak terlihat hanya muncul sewaktu-waktu saja. Gelombang U sama seperti gelombang T ialah repolarisasi ventrikel dari awal sampai akhir [13].
- Interval PR, merupakan durasi siklus atrium yang berfungsi menghitung waktu dari awal depolarisasi atrium sampai awal depolarisasi ventrikel. Durasi normal pada interval PR yaitu 0.12 sampai 0.20 detik [13].
- Interval QT, merupakan interval yang memiliki durasi depolarisasi dan repolarisasi ventrikel dari awal gelombang Q sampai akhir gelombang T [13].
- Segmen PR, segmen yang menghubungkan gelombang P dan QRS dengan garis isoelektrik. Aktivitas listrik yang digambarkan dari atrium ke ventrikel.
- Segmen ST, merupakan proses atau segmennya yang dimulai pada akhir gelombang S sampai dengan awal gelombang T [13].

2.3 PTB Diagnostic ECG Database

EKG dalam data ini diperoleh dengan menggunakan perekam prototype PTB non-komersial, dengan spesifikasinya sebagai berikut [14]:

1. 16 saluran input, (14 untuk EKG, 1 untuk respirasi, 1 untuk tegangan saluran)
2. Tegangan input: ± 16 mV, tegangan offset kompensasi hingga ± 300 mV
3. Resistansi input: 100Ω (DC)
4. Resolusi: 16 bit dengan $0.5 \mu\text{V} / \text{LSB}$ (2000 unit A/D per Mv)
5. Bandwith: 0-1 kHz (pengambilan sampel sinkron dari semua saluran)
6. Tegangan kebisingan: maks. $10 \mu\text{V}$ (pp), masing-masing $3 \mu\text{V}$ (RMS) dengan inputan dari short circuit
7. Rekaman online dari lapisan resistansi
8. Perekaman pengumpulan sinyal dalam Level kebisingan

Basis data berisi 549 rekaman dari 298 subjek (usia 17 hingga 87, rata-rata 57.2; 209 laki-laki, usia rata-rata 61.6; usia tidak dicatat untuk 1 perempuan dan 14 subjek laki-laki). Setiap subjek diwakili oleh satu hingga lima rekaman. Setiap rekaman mencakup 15 sinyal yang diukur secara simultan: 12 lead konvensional (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6) bersama dengan 3 Frank lead EKG (vx, vy, vz). Setiap sinyal diproses menjadi digital pada 1000 sampel per detiknya, dengan resolusi 16 bit pada kisaran $\pm 16,384$ mV. Atas permintaan khusus kepada kontributor database, rekaman mungkin tersedia dengan laju sampling hingga 10 KHz.

Dalam file header (.hea) dari sebagian besar rekaman EKG ini adalah ringkasan klinis yang terperinci, termasuk usia, jenis kelamin, diagnosis, dan jika berlaku data tentang riwayat medis, pengobatan dan intervensi, patologi arteri koroner, ventrikulografi, ekokardiografi, dan hemodinamik. Berikut kelas yang dimiliki pada dataset PTB diagnostic ECG database yang ditunjukkan pada tabel 2.1.

Tabel 2. 1 Dataset

Diagnostic class	Number of subjects
Myocardial infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	16
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	80

2.4 Resampling Data

Salah satu permasalahan yang sering terjadi pada *Machine Learning* yaitu ketidakseimbangan data antar kelas, data yang digunakan saat membuat model klasifikasi tidak seimbang dapat menyebabkan nilai akurasi yang rendah [15]. Pada ketidakseimbangan data memiliki beberapa metode untuk menghasilkan data agar menjadi seimbang. Pertama metode tingkat data yaitu menyeimbangkan distribusi dari data yang tidak seimbang dengan metode *undersampling* dan *over-sampling*, kedua yaitu pendekatan dengan tingkat algoritma dengan memodifikasi metode yang ada untuk menghitung nilai dari kelas minor, dan yang ketiga yaitu mengkombinasikan pendekatan algoritma dan level data [16].

Pengambilan sampel merupakan solusi untuk mengatasi ketidakseimbangan data yang tidak seimbang. Metode pengambilan sampel digunakan cenderung untuk menyesuaikan distribusi pada kelas minoritas dan kelas mayoritas dalam ketidakseimbangan pada saat pelatihan data [17]. Pendekatan *Over-sampling* lebih sering digunakan daripada metode *Under-sampling*, dikarenakan metode *Under-sampling* akan menghilangkan data dalam kelas mayoritas sehingga menyebabkan

hilangnya informasi penting dari data. Metode dasar *Over-sampling* salah satunya ialah *Random Oversampling* (ROS), yaitu metode sampling yang dilakukan dengan menyeimbangkan distribusi data melalui penerapan duplikasi data minoritas secara acak [18]. Namun, metode ini memiliki kelemahan yang menyebabkan masalah *overfitting* pada dataset. Meskipun begitu metode *Random Oversampling* (ROS) dapat meningkatkan nilai akurasi cukup baik [19].

2.5 *Machine Learning*

Machine learning merupakan pendekatan dalam kecerdasan buatan (*Artificial Intelligence*), yang digunakan untuk menirukan perilaku manusia dengan sistem yang dapat mempelajari hal baru dengan sendirinya tanpa harus terus menerus diprogram oleh manusia [20]. *Machine learning* atau pembelajaran mesin ini bertujuan untuk menyelesaikan sebuah masalah secara otomatis dengan mempelajari data. Dengan data, *Machine learning* mendapatkan wawasan, memperbaiki dan belajar hal-hal baru agar terbentuk sebuah sistem yang baik [21].

Machine learning memiliki 2 aplikasi utama yaitu, prediksi dan klasifikasi. Penerapan dua aplikasi tersebut dalam kehidupan sehari-hari, seperti pada bidang kedokteran yang mendeteksi penyakit dari gejala-gejala yang telah dikumpulkan. Atau, mendeteksi penyakit jantung dari alat rekaman *elektrokardiogram* (ECG). *Machine learning* sangat bergantung pada data, untuk mengeluarkan output dari proses kerja *Machine learning* dibutuhkan data untuk bahan training dan analisa [22].

2.6 *Artificial Neural Network* (ANN)

Artificial Neural Network (ANN) merupakan sebuah pendekatan untuk mengolah informasi yang cara kerjanya mengikuti sistem kerja otak manusia. Pada otak manusia, tiap neuron saling berhubungan dan informasi akan mengalir pada tiap-tiap neuron [23]. *Artificial Neural Network* (ANN) adalah sistem dengan keadaan yang mengikuti sesuai perintah yang masuk, strukturnya dapat

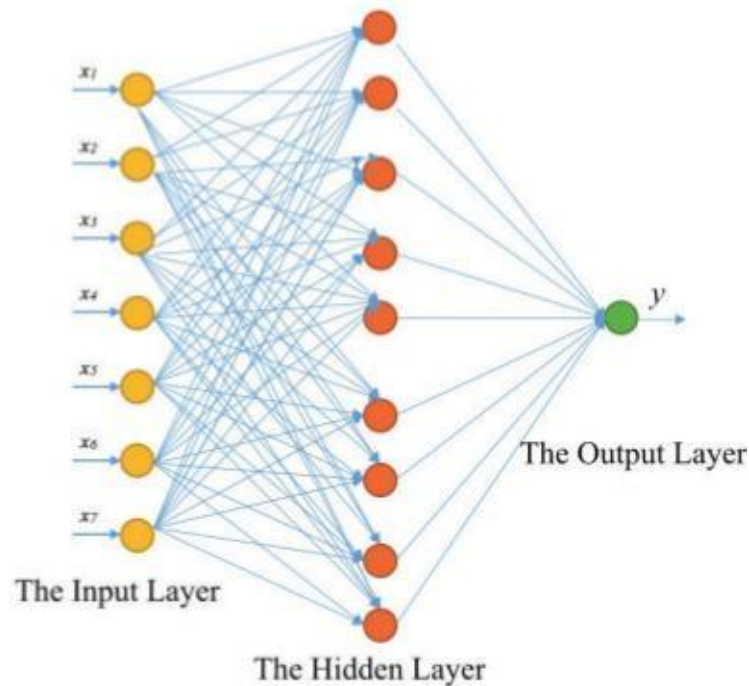
menyesuaikan berdasarkan masuknya informasi dari luar maupun dari dalam yang bergerak melalui jaringan tersebut untuk menyelesaikan masalah [24]. Metode ini ialah salah satu metode yang memiliki cara belajar yang terarah (*Supervised Learning*) dan belajar tidak terarah (*Unsupervised Learning*).

2.6.1 Konsep Dasar *Artificial Neural Network* (ANN)

Pada model *Artificial Neural Network* (ANN) memiliki elemen yang penting untuk melakukan proses komputasi dalam memecahkan masalah. Elemen penting tersebut yaitu Layer masukan, Layer tersembunyi dan Layer keluaran [23]. Pada gambar 2.2. menunjukkan model pada *Artificial Neural Network* (ANN).

1. *Input Layer*, mempunyai peran sebagai tempat informasi dari data-data yang bertindak sesuai dengan output yang diinginkan. *Input layer* memiliki beberapa neuron yang mempresentasikan parameter-parameter yang penting untuk menyelesaikan sebuah permasalahan. *Input layer* akan menyalurkan data-data kepada neuron dan dilanjutkan kepada *hidden layer* maupun *output layer*.
2. *Hidden Layer*, merupakan sebuah layer yang berada diantara layer input dan layer output. *Hidden layer* bertugas menerima data dari *Input layer* dan disalurkan ke *Output layer*.
3. *Output Layer*, merupakan layer yang menerima data dan memberikan hasil dari perhitungan input dan bobot menggunakan fungsi aktivasi. Nilai outputnya melambangkan hasil keluaran dari X menjadi nilai Y.

Berikut arsitektur dari *Artificial Neural Network* (ANN) yang ditunjukkan pada gambar 2.2.



Gambar 2. 2 Model *Arsitektur Neural Network* [23]

2.6.2 Permodelan *Artificial Neural Network* (ANN)

Arsitektur *neural network* ialah kumpulan dari susunan komponen layer-layer dan neuron pada *input layer*, *hidden layer* dan *output layer* yang terhubung dengan *weight*, *activation function* dan *learning function*. *Arsitektur neural network* memiliki 3 macam permodelan, yaitu [25]:

4. *Single layer*, merupakan permodelan *Artificial Neural Network* (ANN) yang paling sederhana. *Single layer* memiliki cara kerja yang dimulai dari node yang menjadi sumber *input layer* dan di proyeksikan pada *output layer* dari neuron. Jenis permodelan ini ialah jaringan *feedforward*. Dimana *single layer* yang berperan menjadi output dari jaringan, sedangkan inputnya tidak ikut berperan dikarenakan saat melakukan proses input proses komputasi tidak terjadi.
5. *Multi layer*, merupakan jaringan yang menampung *hidden layer* yang masuk lebih dari satu. Multi layer dapat menyelesaikan masalah lebih kompleks dibanding *single layer*. Cara kerja pada multi layer adalah input

layer akan mengirim inputannya pada jaringan, setelah itu input akan melakukan komputasi pada layer selanjutnya dan output layer berperan sebagai input dari layer selanjutnya begitu seterusnya.

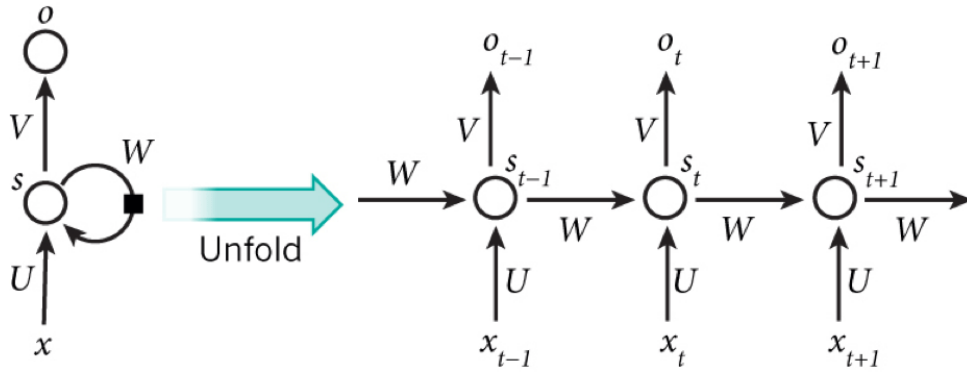
6. *Recurrent network*, merupakan arsitektur *neural network* yang dibuat khusus untuk memproses data yang berurutan (*sequential data*). Pada proses pembelajarannya *recurrent network* tidak begitu saja membuang informasi sebelumnya. *Recurrent network* mampu menyimpan memori agar pola data dapat dikenali dengan baik, sehingga dapat digunakan membuat prediksi yang baik dan akurat. *Recurrent network* akan melooping informasi dari proses sebelumnya agar informasi dari proses sebelumnya tetap tersimpan.

2.7 *Recurrent Neural Network (RNN)*

Jaringan saraf berulang merupakan pengertian dari RNN karena nilai neuron yang digunakan hidden layer sebelumnya akan diproses kembali untuk data input [26]. Proses para *Recurrent Neural Network (RNN)* akan dipanggil berulang-ulang untuk mengerjakan masukan yaitu data sekuensial. *Recurrent Neural Network (RNN)* memiliki berbagai bentuk, salah satunya yang umum digunakan ialah standar *Multi-Layer Perceptron (MLP)* yang ditambah dengan loop tambahan. Maka dari itu proses ini dapat mengeksplorasi kemampuan pemetaan non-linear yang dari MLP [27]. Mengklasifikasikan data time series dan sekuensial merupakan ciri khas dari *Recurrent Neural Network (RNN)*. Data time series sendiri merupakan data yang digabungkan menurut urutan waktu dalam rentang tertentu, sedangkan data sekuensial merupakan suatu sampel data yang diproses secara terurut dan setiap urutan berhubungan satu sama lain [28]. *Recurrent Neural Network (RNN)* memiliki fungsi aktivasi deterministik, fungsi aktivasi *Recurrent Neural Network (RNN)* ialah tan h, dengan perhitungan matematiknnya sebagai berikut [29]:

$$h_t = \tan h \left(\sum_{k=1}^t W_c^{t-k} W_{in\ k} x \right) \dots \dots \dots (2.1)$$

Berikut gambar 2.3 merupakan arsitektur kerja dari metode *Recurrent Neural Network (RNN)*.

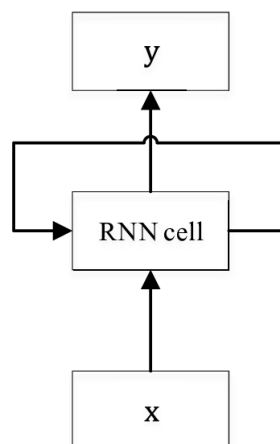


Gambar 2. 3 Arsitektur *Recurrent Neural Network* [30]

Simbol x_t pada gambar 2.3 ialah input pada tiap proses atau *time step*, simbol S_t merupakan hidden state atau memory pada tiap time step t yang berfungsi menyimpan hasil rekaman yang telah dilakukan, dan O_t merupakan output dari proses t .

2.7.1 Vanilla RNN

Vanilla RNN atau *SimpleRNN* merupakan tipe *Recurrent Neural Network* (RNN) yang sederhana, dengan melakukan proses yang sama pada tiap elemen secara berurutan dan setelah itu melakukan proses pada output yang mengacu pada komputasi sebelumnya. Namun pada prakteknya kemampuan *Vanilla RNN* sangatlah terbatas untuk mengakses informasi yang digunakan sebelumnya [32]. Berikut gambar 2.4 ialah struktur dari *Vanilla RNN*.



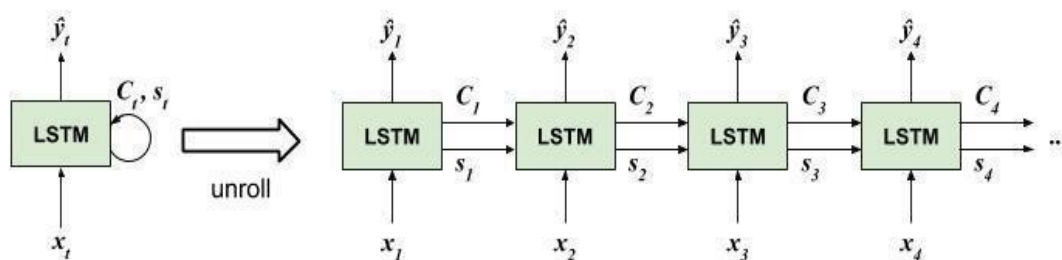
Gambar 2. 4 Struktur *Vanilla RNN*

Struktur dari *Vanilla RNN* memiliki bentuk yang melingkar, dapat dilihat pada gambar 2.4. Hal tersebut merupakan salah satu bentuk dari ANN dimana node terkoneksi dengan directional edge dengan struktur yang melingkar. Karena struktur itulah, mereka membagi parameter yang sama, tidak seperti ANN pada umumnya dengan parameter yang berbeda pada setiap layer [32].

2.7.2 Long Short Term Memory (LSTM)

Model pemrosesan lain dari *Recurrent Neural Network* (RNN) ialah *Long Short Term Memory* (LSTM) yang spesifik [30], yang mana dilakukan modifikasi pada *Recurrent Neural Network* (RNN) dengan memberi *memory cell* untuk dapat menyimpan informasi dalam waktu yang lama [33]. LSTM digunakan untuk solusi mengatasi terjadinya *vanishing gradient* pada RNN saat melakukan proses data sekuensial yang panjang.

Pada LSTM memiliki 3 gate yaitu gate masukan, forget gate dan gate keluaran. Gate masukan memiliki fungsi memberikan keputusan nilai mana yang akan masuk atau diperbarui, *forget gate* memiliki fungsi mengatur pengingat atau yang dilupakan untuk jumlah state dari kondisi sebelumnya mengikuti kebutuhan, sedangkan *output gate* yang mengatur keputusan bagian apa yang digunakan untuk konteks yang akan dihasilkan [30]. Berikut gambar 2.5 ialah arsitektur pada *Long Short Term Memory* (LSTM).



Gambar 2. 5 Arsitektur pada *Long Short Term Memory* (LSTM)

Pada gambar 2.5 LSTM memiliki pemrosesan berbeda dari arsitektur RNN. Pembedanya adalah adanya penambahan sinyal yang diperuntukan dari satu proses waktu ke proses waktu berikutnya disebut dengan konteks dan direpresentasikan dengan symbol C_t .

2.7.3 Gated Recurrent Unit (GRU)

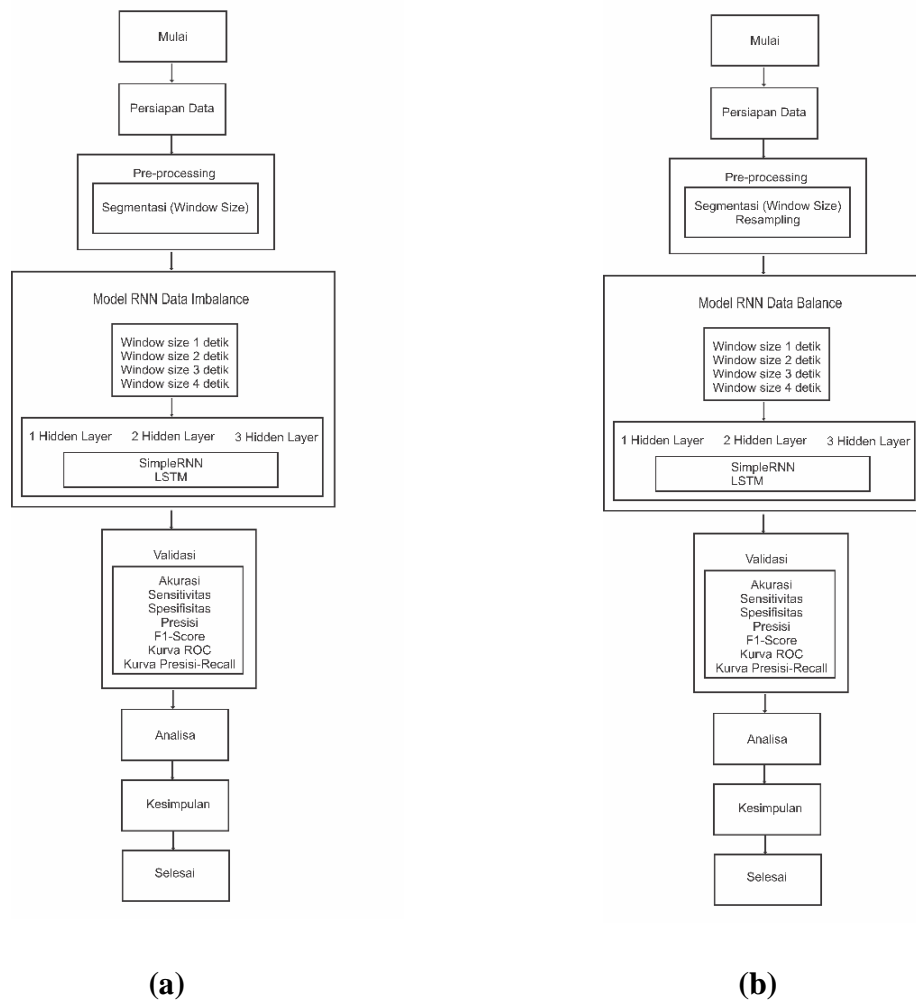
Salah satu tipe arsitektur dari RNN yang dapat mengklasifikasikan data berurutan atau time series ialah *Gated Recurrent Unit (GRU)*. *Gated Recurrent Unit (GRU)* memiliki komputasi yang sederhana dari tipe arsitektur *Recurrent Neural Network (RNN)* lainnya, GRU memiliki akurasi yang sama dan cukup efektif untuk mengatasi masalah gradien yang menghilang [31]. *Gated Recurrent Unit (GRU)* memiliki 2 gate untuk komponen pengatur alur informasi *Gated Recurrent Unit (GRU)*, ialah reset gate dan update gate. Reset gate digunakan untuk menentukan bagaimana untuk menyatukan informasi masa lalu dan input yang baru, sedangkan update gate berfungsi menentukan seberapa banyak informasi masa lalu yang didapatkan untuk tetap disimpan [30].

BAB III

METODOLOGI

3.1 Pendahuluan

Penelitian ini membahas mengenai klasifikasi penyakit jantung pada kasus menggunakan metode *Reccurent Neural Network* (RNN) dirancang dengan bahasa pemrograman *Python*. Pada penulisan ini membahas dua kasus yaitu klasifikasi pada data yang belum diresampling atau yang masih tidak seimbang (*imbalance*) dan klasifikasi pada data yang telah diresampling atau yang seimbang (*balance*). Adapun pada gambar 3.1 ialah kerangka kerja pada penelitian ini.

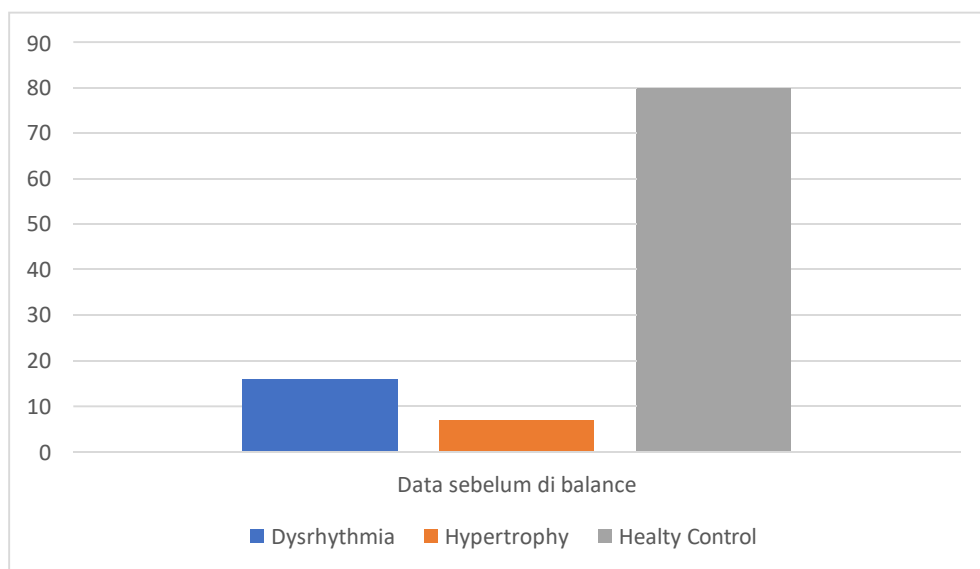


Gambar 3. 1 Kerangka kerja untuk
(a) data *Imbalance* (b) data *Balance*

Penulisan ini memiliki beberapa tahap seperti gambar 3.1 untuk membuat rancangan model klasifikasi penyakit jantung, yaitu: persiapan data, pra-pemrosesan, model *Reccurent Neural Network* (RNN), validasi, analisa penelitian, kesimpulan dan saran.

3.2 Persiapan Data

Penelitian tugas akhir ini menggunakan dataset dari *The PTB Diagnostic ECG Database - PhysioNet* dan metode klasifikasi *Reccurent Neural Network* (RNN) untuk melakukan klasifikasi dengan 3 kelas yaitu *Dysrhythmia*, *Myocardial hypertrophy*, dan *Healthy controls*. Setiap rekaman mencakup 15 sinyal yang diukur secara simultan: 12 lead konvensional (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v5, v6) bersama dengan 3 Frank lead EKG (vx, vy, vz). Dengan memiliki 103 rekaman, berikut gambar 3.2 menunjukkan kelas yang akan diklasifikasikan.



Gambar 3. 2 Pembagian Kelas

Pada gambar 3.2 untuk grafik berjumlah 16 merupakan kelas *Dysrhythmia*, untuk grafik berjumlah 7 merupakan kelas *Myocardial Hypertrophy*, dan untuk

grafik berjumlah 80 merupakan kelas *Healthy Controls*. Pada gambar 3.2 terlihat bahwa data tidak seimbang (*imbalance*), sehingga data perlu diresampling untuk menghasilkan data yang seimbang (*balance*). Dataset PTB ini belum memiliki pelabelan, maka dari itu sebelum memasuki tahap *pre-processing* dataset ini diberi label agar mempermudah proses klasifikasi. Berikut tabel 3.1 menunjukkan pembagian label pada dataset *The PTB Diagnostic ECG Database*.

Tabel 3. 1 Label dataset

Kelas	Label
<i>Healthy control</i>	0
<i>Hypertrophy</i>	1
<i>Dysrhythmia</i>	2

Pada pelabelan dataset ini dijelaskan bahwa *Dysrhythmia* dilabelkan 2, *Hypertrophy* dilabelkan sebagai 1 dan *Healthy control* dilabelkan sebagai 0.

3.3 *Pre-processing*

Pre-processing ialah sebuah proses pertama yang dilakukan dalam mengolah data input sebelum memasuki proses rancangan model untuk mengklasifikasikan penyakit jantung dengan metode *Reccurent Neural Network* (RNN), supaya memberikan kemudahan untuk klasifikasi dengan data terstruktur. Tahapan *pre-processing* pada data *The PTB Diagnostic ECG Database* yaitu *Segmentasi* dan *Resampling* data.

3.3.1 *Segmentasi*

Pada tahap pertama dalam pra-pemrosesan dilakukan tahap segmentasi, rekaman EKG akan disegmentasi berdasarkan *window size*. Penelitian ini menggunakan ritme pada sinyal EKG, dimana ritme tersebut terdiri atas beberapa

beat. Penelitian ini menggunakan 4 skenario yang digunakan untuk segmentasi sinyal EKG, sebagai berikut:

1. Skenario pertama : *Window size* 1 detik panjang rekaman waktu.
2. Skenario kedua : *Window size* 2 detik panjang rekaman waktu.
3. Skenario ketiga : *Window size* 3 detik panjang rekaman waktu.
4. Skenario keempat : *Window size* 4 detik panjang rekaman waktu.

Persamaan yang digunakan dalam menghitung nilai *window size*, yaitu:

$$\text{window size} = \text{panjang waktu rekaman} \times \text{frekuensi sampling} \quad (3.1)$$

Frekuensi sampling yang digunakan dataset ialah sebesar 1000Hz, panjang waktu rekaman ialah waktu yang digunakan untuk satu *window size*.

3.3.2 Resampling Data

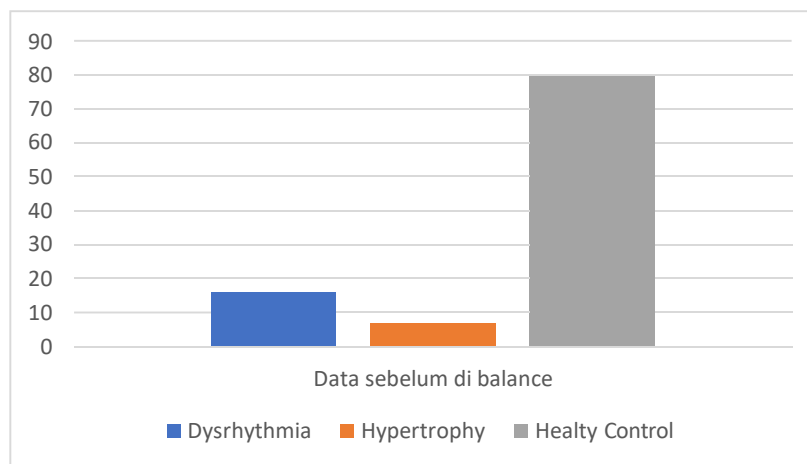
Resampling merupakan proses yang menambah sampel data yang tidak seimbang dari sampel data asli menjadi data seimbang. Penelitian ini menggunakan salah satu metode *resampling* untuk *multiclass* yaitu *Random Oversampling* (ROS). Metode *Random Oversampling* (ROS) menduplikasikan data minoritas menjadi data yang seimbang dengan data mayoritas. Data dipilih secara acak pada setiap kasus dari sampel data asli sedemikian rupa sehingga setiap sampel yang diambil memiliki jumlah kasus yang mirip dengan sampel data asli [19].

Pada penelitian setiap hasil segmentasi menghasilkan jumlah data yang berbeda-beda, berikut tabel 3.2 yang menunjukkan jumlah masing-masing data sebelum diresampling dan sesudah diresampling:

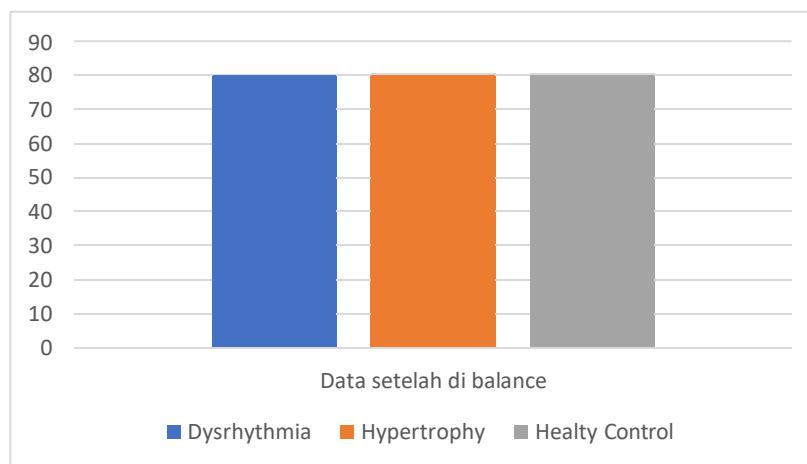
Tabel 3. 2 Jumlah data sebelum dan sesudah diresampling

Segmentasi	Data Tidak Seimbang (Imbalance)	Data Seimbang (Balance)
1 detik	2.571	6.309
2 detik	3.448	8.464
3 detik	5.180	12.698
4 detik	10.392	25.474

Pada tabel 3.2 menunjukkan jumlah untuk percobaan segmentasi 1 detik pada data tidak seimbang memiliki nilai 2.571 menjadi 6.309 setelah data seimbang, untuk percobaan segmentasi 2 detik pada data tidak seimbang memiliki nilai 3.448 menjadi 8.464 setelah data seimbang, untuk percobaan segmentasi 3 detik pada data tidak seimbang memiliki nilai 5.180 menjadi 12.698 setelah data seimbang, dan untuk percobaan segmentasi 4 detik pada data tidak seimbang memiliki nilai 10.392 menjadi 25.474 setelah data seimbang. Pada gambar 3.3 merupakan perbandingan grafik untuk data yang belum diresampling dan data yang telah diresampling.



(a)



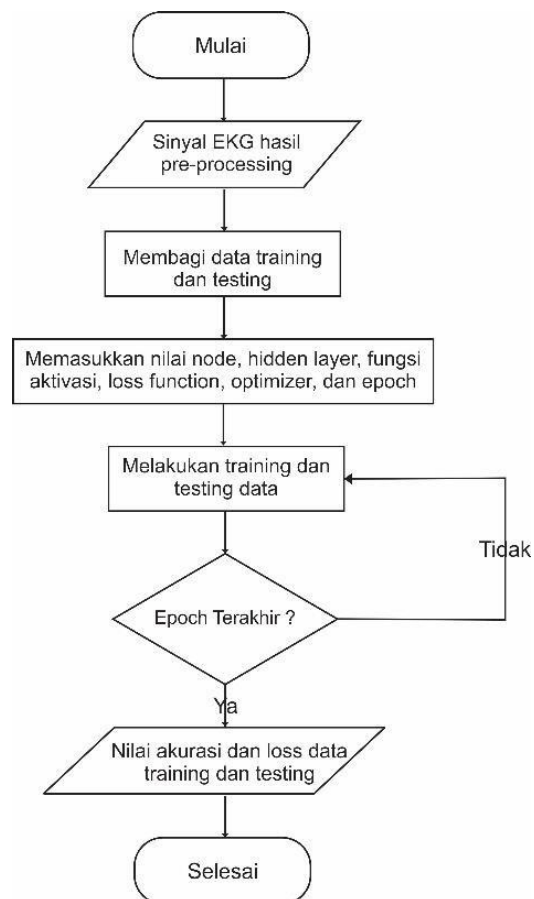
(a)

Gambar 3. 3 Perbandingan dataset

(a) sebelum di-resampling (b) setelah di-resampling

3.4 Klasifikasi *Recurrent Neural Network* (RNN)

Pada tahap ini yaitu merancang model untuk melakukan klasifikasi dengan menggunakan model *Recurrent Neural Network* (RNN). Pada model *Recurrent Neural Network* (RNN) metode yang digunakan yaitu *SimpleRNN* dan *Long short Term Memory* (LSTM). Pembagian data pada pengujian menggunakan rasio perbandingan ini yaitu 90:10 dalam persentasi (%) untuk data pelatihan dan data pengujian. Pengujian ini melakukan *tuning* dengan *batch size* sebanyak 512 iterasi. Fungsi aktivasi *softmax* dipilih untuk layer output karena akan mengembalikan peluang pada kelas masing-masing serta kelas target mendapatkan probabilitas tinggi dan menggunakan fungsi loss *categorical crossentropy*.



Gambar 3. 4 Flowchart Model *Recurrent Neural Network* (RNN)

3.5 Validasi

Validasi merupakan prediksi pada data yang telah diuji agar mengetahui apakah proses klasifikasi setelah dilatih mendapatkan hasil data yang akurat atau tidak. Proses validasi ini menggunakan matriks konfusi dengan nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*) dan FN (*False Negative*). Selanjutnya menghitung matriks evaluasi yaitu akurasi, spesifisitas, sensitivitas, presisi, *f1-score* serta kurva ROC dan kurva presisi recall.

3.5.1 Akurasi

Tahap selanjutnya setelah menghitung metriks konfusi, maka dapat menghitung nilai akurasi dengan persamaan matematisnya yang ditunjukkan pada persamaan 3.2:

$$Akurasi = \frac{\sum_{y=1}^i TP_y + \sum_{y=1}^i TN_y}{\sum_{y=1}^i TP_y + \sum_{y=1}^i TN_y + \sum_{y=1}^i FP_y + \sum_{y=1}^i FN_y} \quad (3.2)$$

3.5.2 Sensitivitas

Sensitivitas ialah pengukuran seberapa baik hasil klasifikasi dan nilai yang didapatkan pada rasio positif aktual dari deteksi yang benar, dengan persamaan matematisnya yang dapat dilihat pada persamaan 3.3:

$$Sensitivitas = \frac{\sum_{y=1}^i TP_y}{\sum_{y=1}^i TP_y + \sum_{y=1}^i FN_y} \quad (3.3)$$

3.5.3 Spesifisitas

Spesifisitas (*Recall*) ialah pengukuran seberapa baik hasil klasifikasi dan nilai yang didapatkan pada rasio positif aktual dari deteksi yang benar, dengan persamaan matematisnya yang dapat dilihat pada persamaan 3.4:

$$Spesifisitas = \frac{\sum_{y=1}^i TN_y}{\sum_{y=1}^i TN_y + \sum_{y=1}^i FP_y} \quad (3.4)$$

3.5.4 Presisi

Presisi ialah pengukuran seberapa baik hasil klasifikasi dan nilai yang didapatkan pada rasio positif aktual dari deteksi yang benar, dengan persamaan matematisnya yang dapat dilihat pada persamaan 3.5:

$$Presisi = \frac{\sum_{y=1}^i TP_y}{\sum_{y=1}^i TP_y + \sum_{y=1}^i FP_y} \quad (3.5)$$

3.5.5 F1-Score

F1-Score merupakan penentu semua hasil dari perhitungan akurasi berdasarkan pada nilai presisi dan sensitifitas. F1 dikatakan baik pada saat model hanya mendapatkan nilai false positive dan false negative yang kecil, dengan persamaan matematisnya yang dapat dilihat pada persamaan 3.6:

$$F1 \text{ Score} = \frac{2 \times \sum_{y=1}^i presisi_y \times \sum_{y=1}^i sensitifitas_y}{\sum_{y=1}^i presisi_y + \sum_{y=1}^i sensitifitas_y} \quad (3.6)$$

3.5.6 Kurva ROC

Kurva ROC (*Receiver Operating Characteristic*) merupakan sebuah bentuk yang menggambarkan prediksi klasifikasi berdasarkan hasil dari *true positive rate* (TPR) dan *false positive rate* (FPR), penilaian diambil untuk model pengelompokan yang baik apabila kurva mendekati pojok kiri atas. Dengan persamaan matematisnya yang dapat dilihat pada persamaan 3.7 dan 3.8:

$$FPR_{kelas\ ke-i} = 1 - Spesifisitas_{kelas\ ke-i} \quad (3.7)$$

Atau

$$FPR_{kelas\ ke-i} = \frac{FP_i}{TP_i + FP_i} \quad (3.8)$$

3.5.7 Kurva Presisi-Recall

Presisi-Recall adalah kurva dengan menggambarkan prediksi model classifier yang baik berdasarkan nilai sensitivitas (*recall*) dan presisi. Penilaian pada kurva presisi-recall agar dapat melihat model dengan hasil yang baik ialah kurva akan berada pada pojok kanan atas.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pendahuluan

Pada bab 4 akan membahas hasil serta analisa pengujian dari rancangan model yang telah dibuat, mengenai klasifikasi penyakit jantung pada sinyal EKG dengan metode *Recurrent Neural Network* (RNN) dengan menggunakan *The PTB Diagnostic ECG Database*.

4.2 Persiapan Dataset

Penelitian ini menggunakan dataset *The PTB Diagnostic ECG Database* dengan mengambil 3 kelas yaitu kelas *Dysrhythmia*, kelas *Myocardial hypertrophy*, dan kelas *Healthy controls*. Dataset ini tidak memiliki pelabelan, maka terlebih dahulu memberikan label pada masing-masing dataset yang ditunjukkan oleh Tabel 4.1.

Tabel 4. 1 Pembagian label pada tiap kelas dataset

Kelas	Label	Jumlah Data
<i>Healthy control</i>	0	80
<i>Hypertrophy</i>	1	7
<i>Dysrhythmia</i>	2	16
Total Data		103

Tabel 4.1 melakukan pembagian label pada masing-masing kelas yang akan diklasifikasikan, pada kelas *Healthy control* diberi label 0 dengan jumlah rekaman sebanyak 80, pada kelas *Hypertrophy* diberi label 1 dengan jumlah rekaman sebanyak 7, dan pada kelas *Dysrhythmia* diberi label 2 dengan jumlah rekaman sebanyak 16.

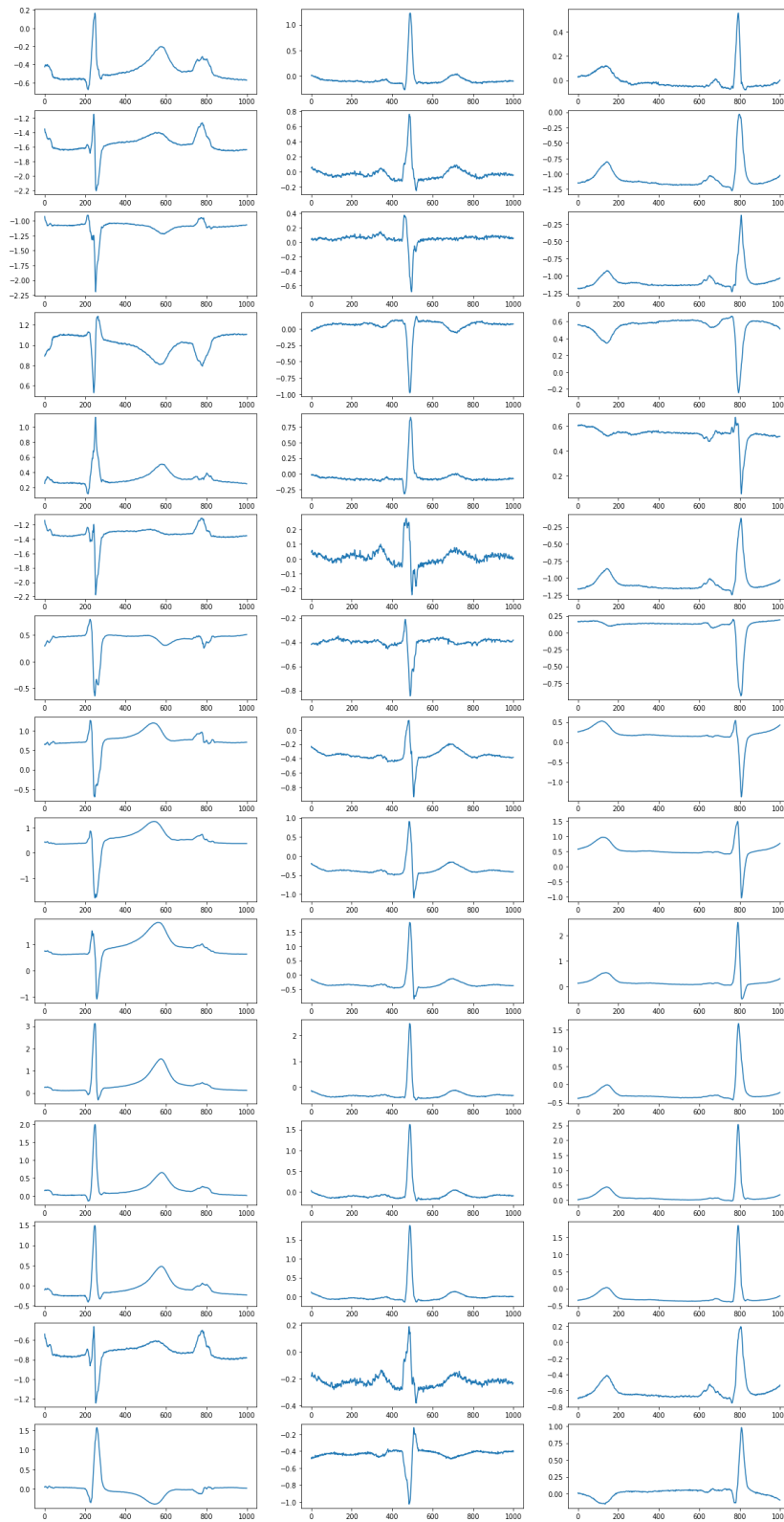
4.2.1 Hasil Pengujian Segmentasi

Tahap selanjutnya yaitu segmentasi sinyal EKG, Segmentasi pada sinyal EKG ini menggunakan *window size*. Pada pengujian ini memiliki 4 skenario untuk tiap panjang waktu perekaman, dengan frekuensi sampling yaitu 1000Hz artinya 1000 node dalam sinyal adalah 1 detik untuk skenario 1, untuk skenario ke 2 yaitu menggunakan frekuensi sampling 1000Hz yang dikalikan dengan waktu rekaman selama 2 detik, untuk skenario ke 3 yaitu menggunakan frekuensi sampling 1000Hz yang dikalikan dengan rekaman waktu selama 3 detik, dan untuk skenario ke 4 yaitu menggunakan frekuensi sampling 1000Hz yang dikalikan dengan panjang waktu perekaman 4 detik. Berikut panjang waktu perekaman pada tiap skenario :

1. Skenario pertama: rekaman waktu *window size* selama 1 detik ialah 1000.
2. Skenario kedua : rekaman waktu *window size* selama 2 detik ialah 2000.
3. Skenario ketiga : rekaman waktu *window size* selama 3 detik ialah 3000.
4. Skenario keempat: rekaman waktu *window size* selama 4 detik ialah 4000.

Pada pengujian skenario pertama menggunakan *window size* 1 detik yaitu berarti 1000 node, pada pengujian skenario 2 menggunakan *window size* 2 detik yaitu berarti 2000 node. Pada pengujian skenario 3 menggunakan *window size* 3 detik yang berarti 3000 node, dan pada pengujian skenario 4 *window size* 4 detik yang berarti 4000 node.

Pada gambar 4.1, gambar 4.2, gambar 4.3, dan gambar 4.4 dapat dilihat bentuk sinyal EKG berdasarkan masing-masing waktu perekaman *window size*. Gambar 4.1 merupakan bentuk sinyal dari hasil waktu perekaman *window size* 1 detik, gambar 4.2 merupakan bentuk sinyal dari hasil waktu perekaman *window size* 2 detik, gambar 4.3 merupakan bentuk sinyal dari hasil waktu perekaman *window size* 3 detik, dan gambar 4.4 merupakan bentuk sinyal dari hasil waktu perekaman *window size* 4 detik. Berikut gambar 4.1, gambar 4.2, gambar 4.3 dan gambar 4.4 yang menunjukkan bentuk sinyalnya.



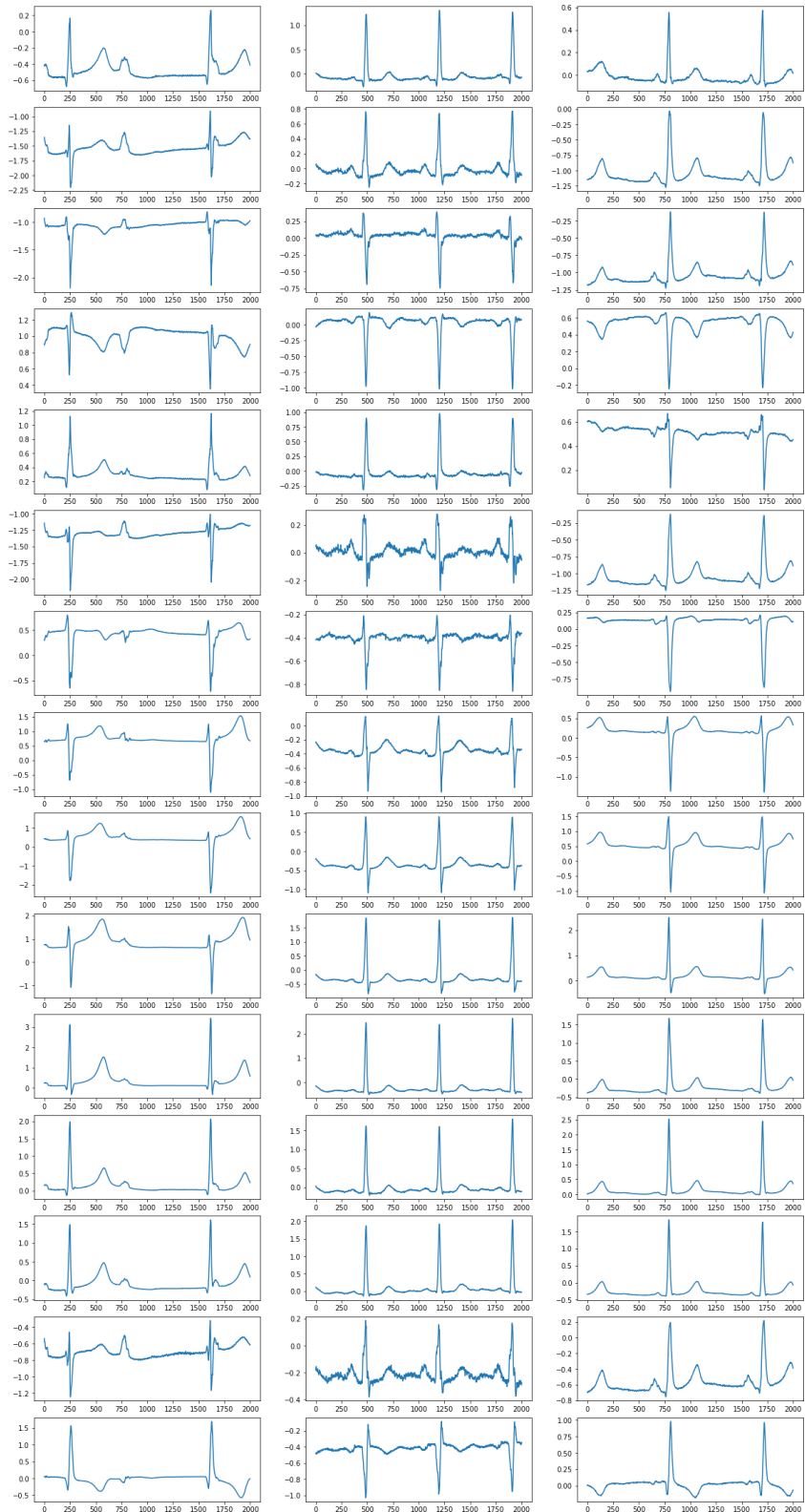
(a)

(b)

(c)

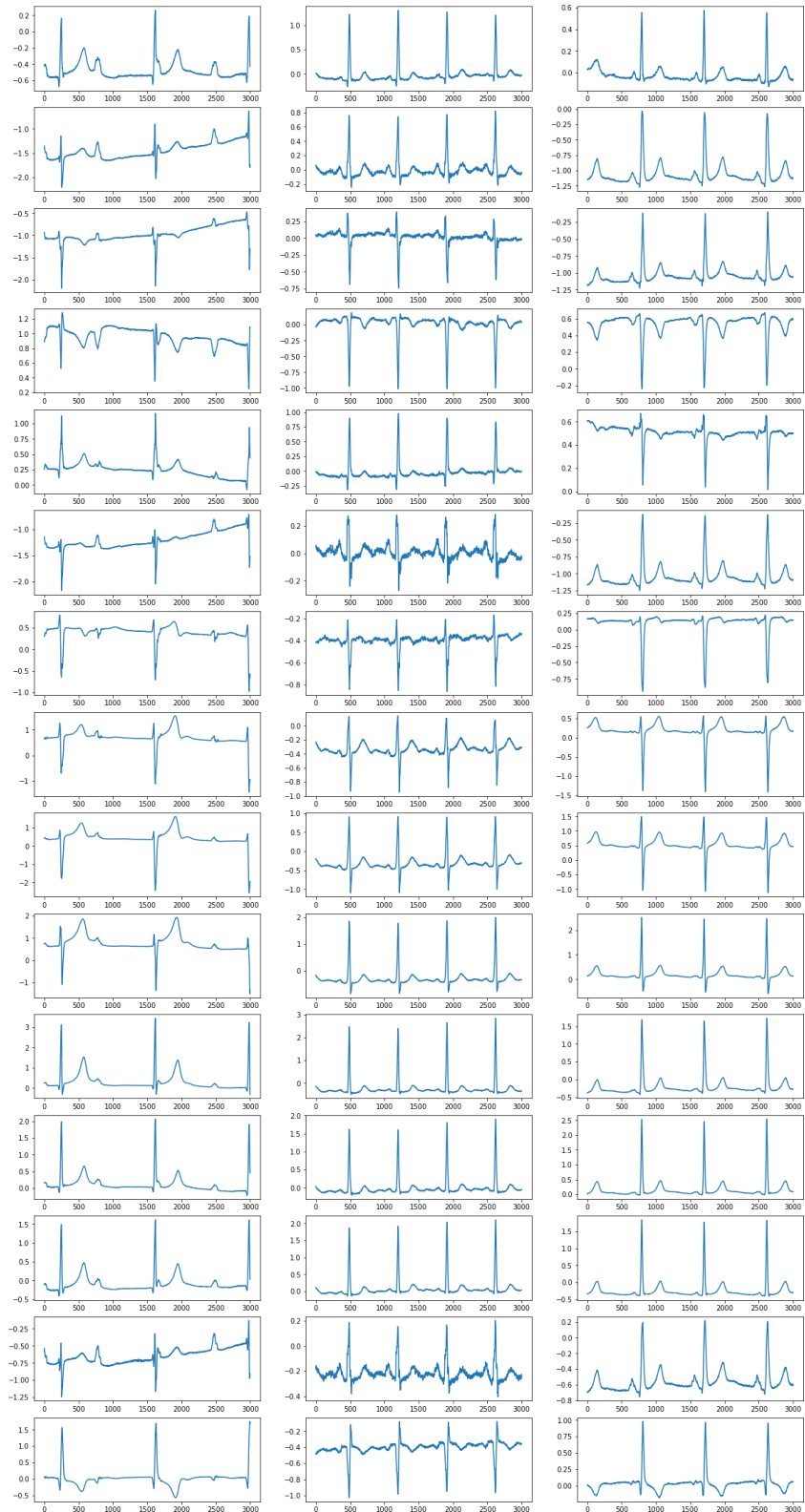
Gambar 4. 1 Sinyal EKG berdasarkan window size 1 detik

(a) *Dysrhythmia* (b) *Hypertrophy* (c) *Healthy Control*



(a) (b) (c)

Gambar 4. 2 Sinyal EKG berdasarkan *window size* 2 detik
 (a) *Dysrhythmia* (b) *Hypertrophy* (c) *Healthy Control*



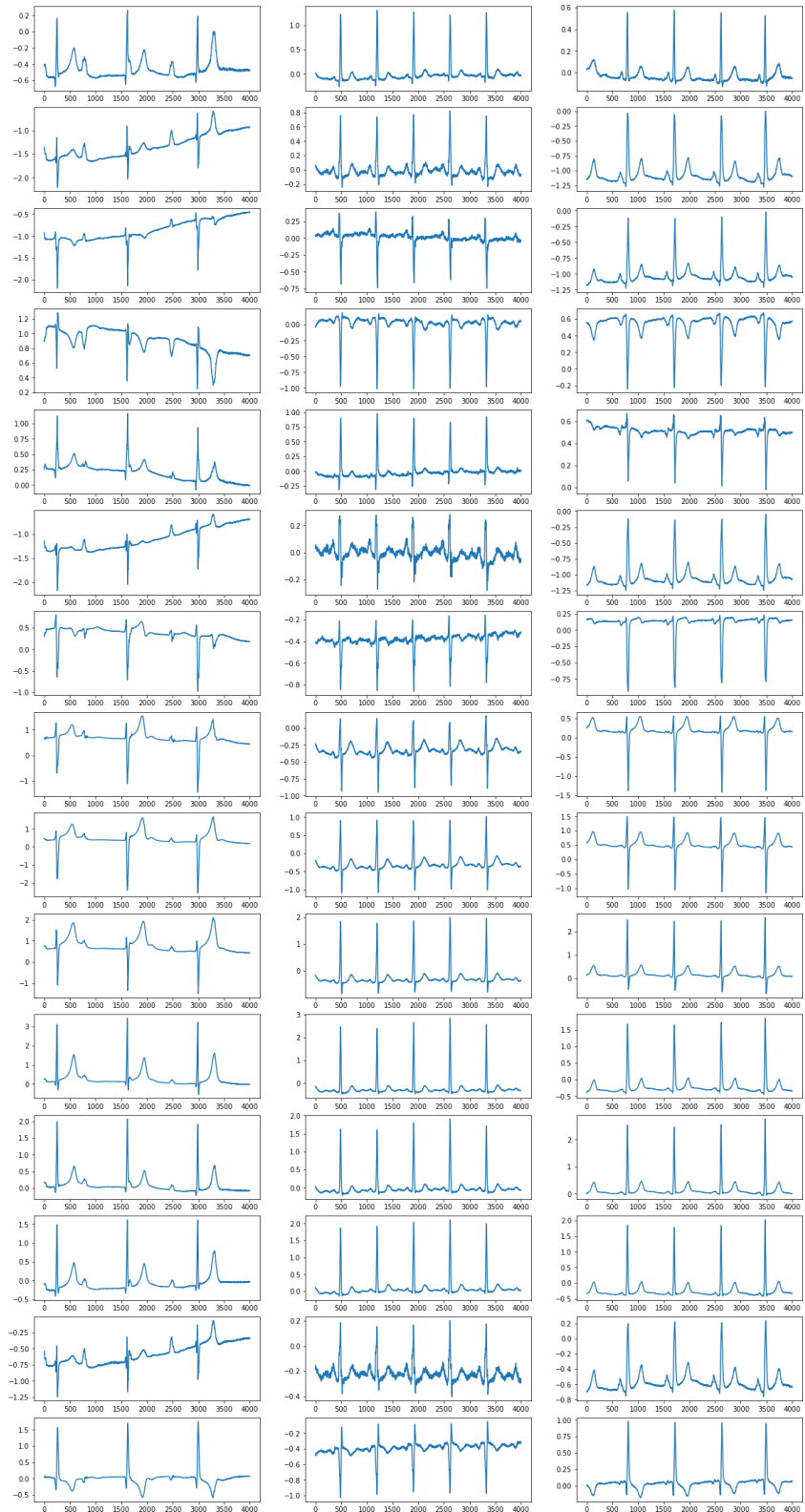
(a)

(b)

(c)

Gambar 4. 3 Sinyal EKG berdasarkan *window size* 3 detik

(a) *Dysrhythmia* (b) *Hypertrophy* (c) *Healthy Control*



(a)

(b)

(c)

Gambar 4. 4 Sinyal EKG berdasarkan *window size* 4 detik

(a) *Dysrhythmia* (b) *Hypertrophy* (c) *Healthy Control*

4.2.2 Hasil *Resampling* Data

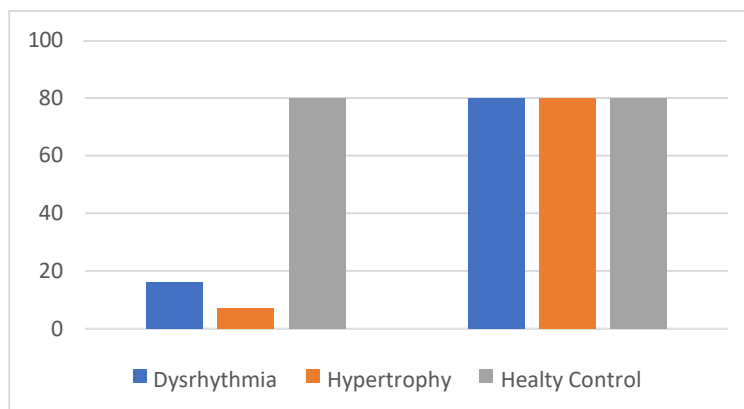
Pada dataset ini data yang digunakan masih tidak seimbang (*imbalance*) yang mengharuskan melakukan *resampling* data agar dapat mendapatkan hasil klasifikasi yang baik. *Resampling* data pengujian ini menggunakan metode *Random Oversampling* (ROS), dimana metode ini menduplikasi jumlah dataset yang kurang. Pada penelitian setiap hasil segmentasi menghasilkan jumlah data yang berbeda-beda, berikut tabel 4.2 menunjukkan jumlah masing-masing data sebelum diresampling dan sesudah diresampling:

Tabel 4. 2 Jumlah data sebelum dan sesudah diresampling

Segmentasi	Data Tidak Seimbang (Imbalance)	Data Seimbang (Balance)
1 detik	2571	6309
2 detik	3448	8464
3 detik	5180	12698
4 detik	10392	25474

Tabel 4.2 merupakan hasil dari jumlah data yang masih data asli atau belum diresampling dan jumlah data yang telah diresampling, data yang digunakan berdasarkan dari masing-masing skenario segmentasi. Untuk data segmentasi 1 detik pada data tidak seimbang memiliki data sebanyak 2571 dan data hasil resampling sebanyak 6309, data pada segmentasi 2 detik pada data tidak seimbang memiliki data sebanyak 3448 dan data hasil resampling sebanyak 8464, data pada segmentasi 3 detik pada data tidak seimbang memiliki data sebanyak 5180 dan data hasil resampling sebanyak 12698, dan data pada segmentasi 4 detik pada data tidak seimbang memiliki data sebanyak 10392 dan data hasil resampling sebanyak 25474.

Data sebelum dan sesudah diresampling pada keseluruhan skenario dapat dilihat pada gambar 4.5.



Gambar 4. 5 Grafik data sebelum dan setelah dilakukan resampling

Gambar 4.5 menunjukkan grafik data yang telah diresampling, pada gambar ditunjukkan hasil resampling dari keseluruhan rekaman. Pada rekaman resampling *Dysrhythmia* memiliki jumlah rekaman 16 yang diresampling menjadi 80, pada *Hypertrophy* memiliki jumlah rekaman 7 yang diresampling menjadi 80 dan *Healthy Control* dengan jumlah data 80.

4.3 Klasifikasi Menggunakan *Recurrent Neural Network* (RNN)

Tahap klasifikasi pengujian ini menggunakan metode RNN, dimana data yang masuk berasal dari data yang telah dilakukan pre-processing. Sebelum melakukan proses klasifikasi, data dibagi untuk melakukan proses pelatihan dan pengujian dengan perbandingan data yaitu 90:10 dalam persentase (%). Pada tahap ini terdapat 2 pengujian yaitu untuk data sebelum di-resampling dan setelah di-resampling. Pengujian tersebut dilakukan karena akan membandingkan *performance* mana yang lebih baik, data asli yang tidak seimbang (*imbalance*) atau data yang telah diresampling sehingga menjadi data yang seimbang (*balance*).

4.3.1 Hasil Pengujian Klasifikasi Menggunakan *Recurrent Neural Network* (RNN) pada Data Tidak Seimbang (*Imbalance*)

Pengujian ini menggunakan data yang belum di-resampling karena itu data yang digunakan masih tidak seimbang (*Imbalance*). Berikut hasil pengujian klasifikasi penyakit jantung dengan rasio perbandingan 90:10 dalam persentase

(%), hidden layer yang digunakan sebanyak 3 kali tuning, *batch size* 512 iterasi dan percobaan ini melakukan Early Stopping training pada model yang telah tepat dengan 1000 epoch.

4.3.1.1 Hasil Validasi Data Tidak Seimbang (*Imbalance*) pada Skenario 1

Pengujian pada skenario pertama menggunakan panjang waktu perekaman *window size* selama 1 detik yang berarti 1000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.3 hasil pengujian yang dilakukan pada data tidak seimbang (*Imbalance*) skenario 1 :

Tabel 4. 3 Hasil pengujian data tidak seimbang (*Imbalance*) pada skenario 1

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 63	SimpleRNN	99.56%	96.10%
2		Epoch 62	LSTM	99.63%	96.96%
3	2	Epoch 62	SimpleRNN	99.48%	95.41%
4		Epoch 58	LSTM	99.56%	96.70%
5	3	Epoch 59	SimpleRNN	99.50%	95.50%
6		Epoch 81	LSTM	99.81%	97.14%

Pada tabel 4.3 dilihat nilai hasil terbaik pada skenario 1 data tidak seimbang (*Imbalance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-6. Pada model menggunakan LSTM percobaan ke-6 mendapatkan akurasi training 99.81% dan akurasi testing 97.14%. Berikut gambar 4.6 yang menunjukkan gambar sinyal EKG akurasi dan loss pada skenario 1 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 6 Bentuk sinyal EKG skenario 1 pada data tidak seimbang
(a) akurasi (b) loss

Pada gambar 4.6 grafik untuk akurasi dan loss pada tiap percobaan untuk data tidak seimbang memberikan grafik yang cukup baik pada akurasi namun pada grafik loss grafik training dan testing masih belum konvergen.

4.3.1.2 Hasil Validasi Data Tidak Seimbang (*Imbalance*) pada Skenario 2

Pengujian pada skenario kedua menggunakan panjang waktu perekaman *window size* selama 2 detik yang berarti 2000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentasi (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.4 hasil pengujian yang dilakukan pada data tidak seimbang (*Imbalance*) skenario 2 :

Tabel 4. 4 Hasil pengujian data tidak seimbang (*Imbalance*) pada skenario 2

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 62	SimpleRNN	99.01%	86.80%
2		Epoch 58	LSTM	99.26%	92.01%
3	2	Epoch 60	SimpleRNN	98.89%	89.93%
4		Epoch 54	LSTM	98.82%	91.31%
5	3	Epoch 60	SimpleRNN	98.95%	89.58%
6		Epoch 71	LSTM	99.16%	90.79%

Pada tabel 4.4 dilihat nilai hasil terbaik pada skenario 2 data tidak seimbang (*Imbalance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 99.26% dan akurasi testing 92.01%. Berikut gambar 4.7 merupakan hasil sinyal EKG akurasi dan loss pada skenario 2 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 7 Bentuk sinyal EKG skenario 2 pada data tidak seimbang
(a) akurasi (b) loss

Pada gambar 4.7 grafik untuk akurasi dan loss pada tiap percobaan untuk data tidak seimbang memberikan grafik yang masih belum konvergen. Dapat dilihat pada grafik training dan testing pada tiap percobaan masing-masing belum mendekati.

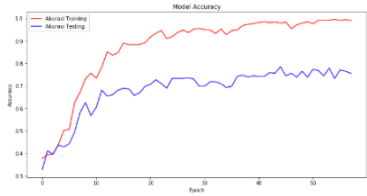
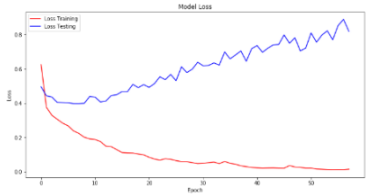
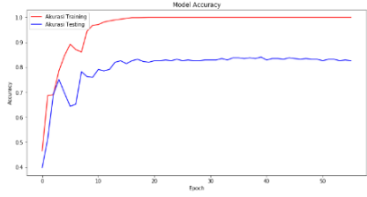
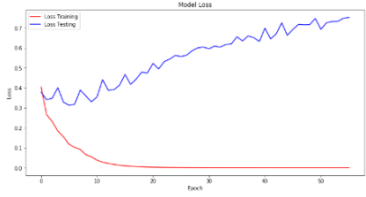
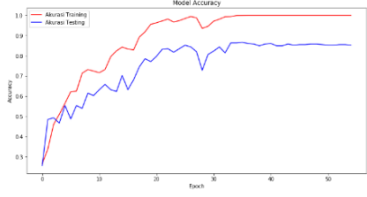
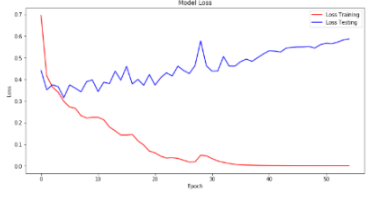
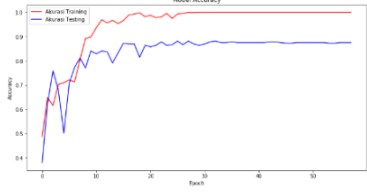
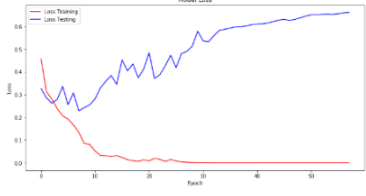
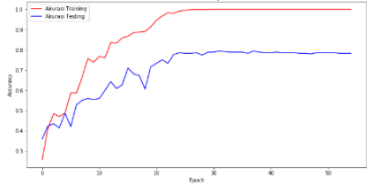
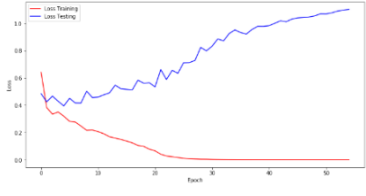
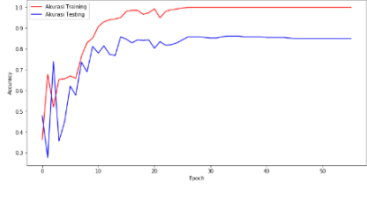
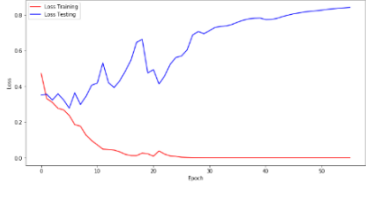
4.3.1.3 Hasil Validasi Data Tidak Seimbang (*Imbalance*) pada Skenario 3

Pengujian pada skenario ketiga menggunakan panjang waktu perekaman *window size* selama 3 detik yang berarti 3000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.5 hasil pengujian yang dilakukan pada data tidak seimbang (*Imbalace*) skenario 3 :

Tabel 4. 5 Hasil pengujian data tidak seimbang (*Imbalance*) pada skenario 3

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 58	SimpleRNN	96.66%	78.64%
2		Epoch 56	LSTM	98.25%	89.32%
3	2	Epoch 55	SimpleRNN	98.52%	86.97%
4		Epoch 58	LSTM	98.75%	86.19%
5	3	Epoch 55	SimpleRNN	97.82%	84.63%
6		Epoch 56	LSTM	98.49%	87.23%

Pada tabel 4.5 dilihat nilai hasil terbaik pada skenario 3 data tidak seimbang (*Imbalance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 98.25% dan akurasi testing 89.32%. Berikut gambar 4.8 sinyal EKG akurasi dan loss pada skenario 3 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 8 Bentuk sinyal EKG skenario 3 pada data tidak seimbang
(a) akurasi (b) loss

Pada gambar 4.8 grafik untuk akurasi dan loss pada tiap percobaan untuk data tidak seimbang memberikan grafik yang masih belum konvergen. Dapat dilihat pada grafik training dan testing pada tiap percobaan masing-masing belum mendekati.

4.3.1.4 Hasil Validasi Data Tidak Seimbang (*Imbalance*) pada Skenario 4

Pengujian pada skenario 4 menggunakan panjang waktu perekaman *window size* sebesar 4 detik yang berarti 4000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.6 hasil pengujian yang dilakukan pada data tidak seimbang (*Imbalace*) skenario 4 :

Tabel 4. 6 Hasil pengujian data tidak seimbang (*Imbalance*) pada skenario 4

Percobaan Ke-	Hidden Layer	Early Stopping (Max : 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 55	SimpleRNN	97.27%	76.57%
2		Epoch 55	LSTM	98.01%	82.86%
3	2	Epoch 53	SimpleRNN	98.01%	82.16%
4		Epoch 54	LSTM	98.21%	84.96%
5	3	Epoch 60	SimpleRNN	98.28%	85.66%
6		Epoch 57	LSTM	98.32%	84.26%

Pada tabel 4.6 dilihat nilai hasil terbaik pada skenario 4 data tidak seimbang (*Imbalance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu SimpleRNN pada percobaan ke-5. Pada model menggunakan SimpleRNN percobaan ke-5 mendapatkan akurasi training 98.28% dan akurasi testing 85.66%. Berikut gambar 4.9 menunjukkan hasil sinyal EKG akurasi dan loss pada skenario 4 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 9 Bentuk sinyal skenario 4 pada data tidak seimbang

(a) akurasi (b) loss

Pada gambar 4.9 grafik untuk akurasi dan loss pada tiap percobaan untuk data tidak seimbang memberikan grafik yang masih belum konvergen. Dapat dilihat pada grafik training dan testing pada tiap percobaan masing-masing belum saling mendekati.

Pada pengujian data tidak seimbang (*Imbalance*) ini telah didapatkan hasil terbaik dari perbandingan masing-masing skenario, ditunjukkan pada tabel 4.7.

Tabel 4. 7 Perbandingan nilai terbaik pada pengujian data *Imbalance* dari skenario pertama, skenario kedua, skenario ketiga, dan skenario keempat.

Skenario Ke-	Model Terbaik	Window Size	Hidden Layer	Early Stopping	Acc train	Acc test
1	LSTM	1000	3	Epoch 81	99.81%	97.14%
2	LSTM	2000	1	Epoch 58	99.26%	92.01%
3	LSTM	3000	1	Epoch 56	98.25%	89.32%
4	SimpleRNN	4000	3	Epoch 60	98.28%	85.66%

Pada tabel 4.7 dapat disimpulkan pengujian terbaik dari data tidak seimbang ialah skenario ke-1 dengan percobaan menggunakan model LSTM, percobaan tersebut menggunakan *window size* 1 detik, menggunakan rasio training dan testing yaitu 90:10 dan pemberhentian awal model yaitu 81 epoch. Berikut tabel *confusion matrix testing* dan *training* pada tiap kelas dari masing-masing skenario di atas.

Tabel 4. 8 Hasil *confusion matrix testing* pada data tidak seimbang (*Imbalance*) skenario 1 percobaan ke-6

m = 1.155		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control</i> (0)	938	4	9
	<i>Hypertrophy</i> (1)	5	84	3
	<i>Dysrhythmia</i> (2)	11	1	100

Pada tabel 4.8 ialah nilai *confusion matrix testing* pada masing-masing kelas untuk data tidak seimbang skenario ke-1 percobaan ke-6, nilai tersebut diperuntukkan menentukan nilai *true positive*, *true negative*, *false positive*, dan *false negative* agar dapat menghitung nilai validasi.

Tabel 4. 9 Nilai *confusion matrix testing* dari data *imbalance* untuk masing-masing kelas pada skenario pertama percobaan ke-6

Kelas	TP	TN	FP	FN
0	938	184	13	16
1	84	1038	8	5
2	100	1022	12	12

Pada tabel 4.9 merupakan nilai TP, TN, FP, dan FN dari kelas imbalance yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario 1 percobaan ke-6.

Tabel 4. 10 Hasil *confusion matrix testing* pada data tidak seimbang (*Imbalance*) skenario 2 percobaan ke-2

m = 576		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	470	4	10
	<i>Hypertrophy (1)</i>	17	26	0
	<i>Dysrhythmia (2)</i>	12	3	34

Pada tabel 4.10 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data tidak seimbang skenario kedua percobaan ke-2, nilai tersebut diperuntukkan menentukan nilai TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 11 Hasil *confusion matrix testing* dari data *imbalance* untuk masing-masing kelas pada skenario kedua percobaan ke-2

Kelas	TP	TN	FP	FN
0	470	60	14	29
1	26	504	17	7
2	34	496	15	10

Pada tabel 4.11 merupakan TP, TN, FP, dan FN dari kelas imbalance yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario kedua percobaan ke-2.

Tabel 4. 12 Hasil *confusion matrix testing* pada data tidak seimbang (*Imbalance*) skenario 3 percobaan ke-2

m = 384		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	312	6	8
	<i>Hypertrophy (1)</i>	7	14	0
	<i>Dysrhythmia (2)</i>	19	1	17

Pada tabel 4.12 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data tidak seimbang skenario ketiga percobaan ke-2, nilai tersebut digunakan untuk menentukan nilai TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 13 Nilai *confusion matrix testing* dari data *imbalance* untuk masing-masing kelas pada skenario ketiga percobaan ke-2

Kelas	TP	TN	FP	FN
0	312	31	14	26
1	14	329	7	7
2	17	326	20	8

Pada tabel 4.13 merupakan nilai TP, TN, FP, dan FN dari kelas imbalance yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario ketiga percobaan ke-2.

Tabel 4. 14 Hasil *confusion matrix testing* pada data tidak seimbang (*Imbalance*) skenario 4 percobaan ke-5

m = 286		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	9	0	18
	<i>Hypertrophy (1)</i>	3	9	11
	<i>Dysrhythmia (2)</i>	4	5	227

Pada tabel 4.14 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data tidak seimbang skenario keempat percobaan ke-5, nilai tersebut digunakan untuk menentukan nilai TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 15 Nilai *confusion matrix testing* dari data *imbalance* untuk masing-masing kelas pada skenario keempat percobaan ke-5

Kelas	TP	TN	FP	FN
0	9	236	18	7
1	9	236	14	5
2	227	18	9	29

Pada tabel 4.15 merupakan nilai TP, TN, FP, dan FN dari kelas imbalance yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario keempat percobaan ke-5.

Berdasarkan hasil confusion matrix yang didapatkan, hasil evaluasi tertinggi pada skenario pertama, skenario kedua, skenario ketiga, dan skenario keempat dapat dilihat nilai performancenya pada tabel 4.16, tabel 4.17, tabel 4.18 dan tabel 4.19.

Tabel 4. 16 Hasil performance dari data tidak seimbang (*Imbalance*) model *Recurrent Neural Network* (RNN) terbaik skenario pertama percobaan ke-6

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	97.92%	89.28%	98.84%	89.28%	89.28%
1	98.87%	94.38%	99.24%	91.30%	92.81%
0	97.48%	98.32%	93.53%	98.63%	98.47%
Rata-Rata	98.09%	93.99%	97.21%	93.07%	93.52%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 1 percobaan ke-6 pada tabel 4.16 rata-ratanya adalah akurasi sebesar 98.09%, sensitifitas sebesar 93.99%, spesifisitas sebesar 97.21%, presisi sebesar 93.07% dan F1-score sebesar 93.52%.

Tabel 4. 17 Hasil performance dari data tidak seimbang (*Imbalance*) model *Recurrent Neural Network* (RNN) terbaik skenario 2 percobaan ke-2

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	95.65%	77.27%	97.18%	69.38%	73.11%
1	95.83%	78.78%	96.86%	60.46%	68.42%
0	92.53%	94.18%	81.81%	97.10%	95.62%
Rata-Rata	94.67%	83.41%	91.95%	75.65%	79.05%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 2 percobaan ke-2 pada tabel 4.17 rata-ratanya adalah akurasi sebesar 94.67%, sensitifitas sebesar 83.41%, spesifisitas sebesar 91.95%, presisi sebesar 75.65% dan F1-score sebesar 79.05%.

Tabel 4. 18 Hasil performance dari data tidak seimbang (*Imbalance*) model *Recurrent Neural Network* (RNN) terbaik skenario 3

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	92.70%	68%	94.42%	45.94%	54.83%
1	96.35%	66.66%	98.07%	66.66%	66.66%
0	89.58%	92.30%	69.56%	95.70%	93.97%
Rata-Rata	92.88%	75.65%	87.35%	69.43%	71.82%

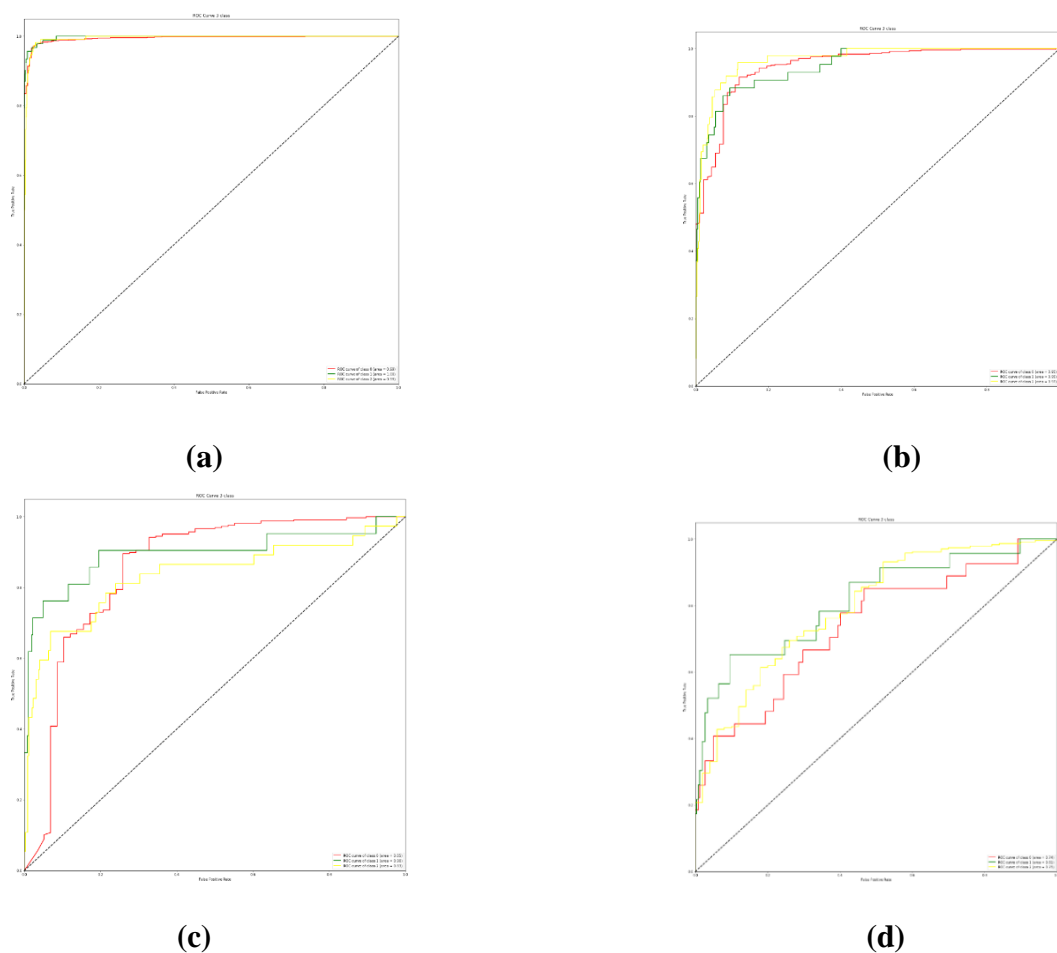
Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 3 percobaan ke-2 pada tabel 4.18 rata-ratanya adalah akurasi sebesar 92.88%, sensitifitas sebesar 75.65%, spesifisitas sebesar 87.35%, presisi sebesar 69.43% dan F1-score sebesar 71.82%.

Tabel 4. 19 Hasil performance dari data tidak seimbang (*Imbalance*) model *Recurrent Neural Network* (RNN) terbaik skenario 4

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	91.25%	56.25%	93.33%	33.33%	41.86%
1	93.35%	64.28%	94.85%	39.13%	48.64%
0	86.71%	88.67%	70%	96.18%	92.27%
Rata-Rata	90.44%	69.73%	86.06%	56.21%	60.92%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 4 percobaan ke-5 pada tabel 4.19 rata-ratanya adalah akurasi sebesar 90.44%, sensitifitas sebesar 69.73%, spesifisitas sebesar 86.06%, presisi sebesar 56.21% dan F1-score sebesar 60.92%.

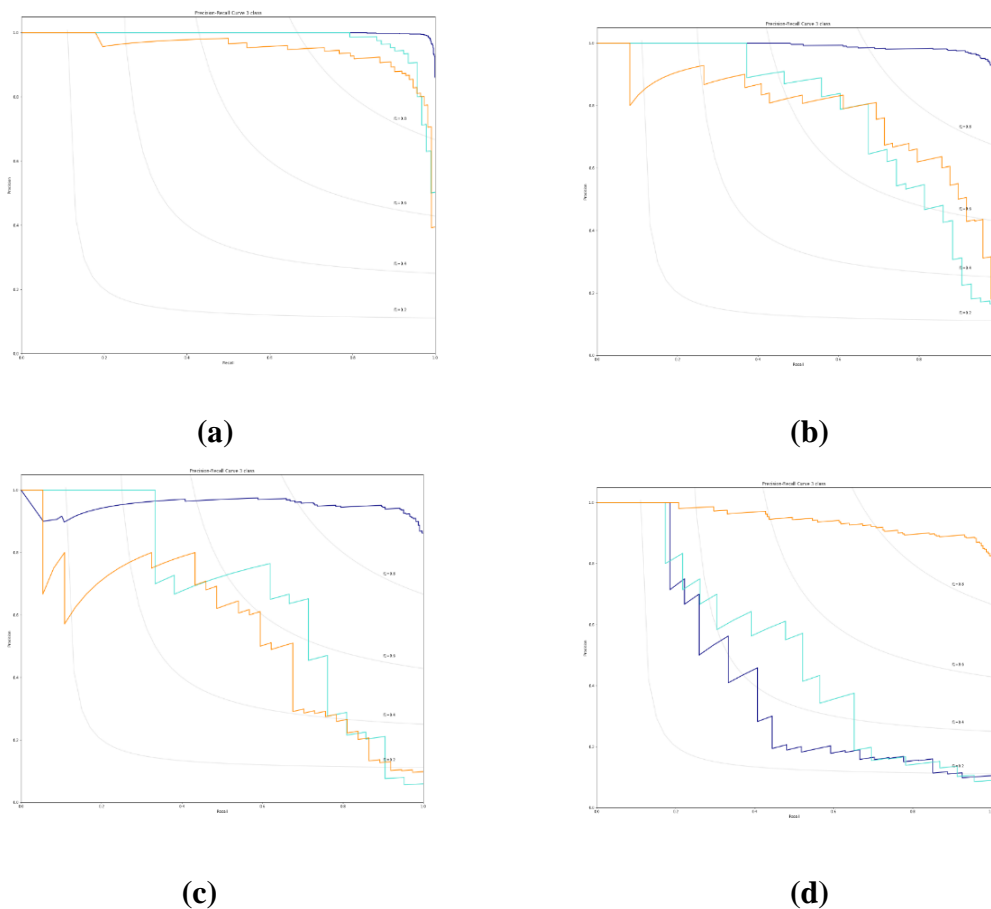
Pada tabel 4.16, tabel 4.17, tabel 4.18, dan tabel 4.19 dapat dilihat masing-masing nilai *performance* pada data tidak seimbang (*Imbalance*) menggunakan model RNN mendapatkan nilai yang sudah cukup baik walaupun ada beberapa nilai kelas yang menghasilkan nilai validasi dibawah 50%. Berikut hasil kurva roc dan presisi recall dari masing-masing skenario terbaik.



Gambar 4. 10 Hasil pengujian kurva ROC terbaik dari (a) skenario 1 (b) skenario 2 (c) skenario 3 (d) skenario 4

Kurva roc pada masing-masing skenario memperlihatkan hasil yang cukup baik walaupun masih belum cukup mendekati pojok kiri atas. Hasil kurva terbaik

dari keempat skenario ialah pada skenario 1 yang mendekati pojok kiri atas. Maka kurva tersebut telah mendapatkan model classifier yang cocok..



Gambar 4. 11 Hasil pengujian presisi-recall terbaik dari (a) skenario 1 (b) skenario 2 (c) skenario 3 (d) skenario 4

Pada kurva presisi-recall pun memperlihatkan hasil yang tidak begitu baik pada masing-masing skenario namun untuk skenario 1 sudah mendapatkan kurva dengan model pengelompokkan terbaik dikarenakan kurva telah mendekati pojok kanan atas. Untuk skenario 2, skenario 3, dan skenario 4 kurva masih belum mendekati pojok kanan atas.

4.3.2 Hasil Pengujian Klasifikasi Menggunakan *Recurrent Neural Network* (RNN) pada Data Seimbang (*Balance*)

Pengujian ini menggunakan data yang telah di-*resampling* karena itu data yang digunakan ialah data yang telah seimbang (*Balance*). Berikut hasil pengujian

klasifikasi penyakit jantung dengan rasio perbandingan 90:10 dalam persentase (%), hidden layer yang digunakan sebanyak 3 kali tunning, *batch size* 512 iterasi dan percobaan ini melakukan Early Stopping training pada model yang telah tepat dengan 1000 epoch.

4.3.2.1 Hasil Validasi Data Seimbang (*Balance*) pada Skenario 1

Pengujian pada skenario pertama menggunakan *window size* sebesar 1 detik yang berarti 1000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90%:10% serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.20 hasil pengujian yang dilakukan pada data seimbang (*Balance*) skenario 1 :

Tabel 4. 20 Hasil pengujian data seimbang (*Balance*) pada skenario 1

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 158	SimpleRNN	96.06%	92.19%
2		Epoch 499	LSTM	99.98%	99.96%
3	2	Epoch 85	SimpleRNN	96.89%	94.70%
4		Epoch 73	LSTM	99.55%	98.86%
5	3	Epoch 323	SimpleRNN	99.39%	98.30%
6		Epoch 81	LSTM	98.48%	97.24%

Pada tabel 4.20 dilihat nilai hasil terbaik pada skenario 1 data seimbang (*Balance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 99.98% dan akurasi testing 99.96%. Berikut gambar sinyal EKG akurasi dan loss pada skenario 1 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 12 Grafik (a) akurasi dan (b) loss pada tiap percobaan untuk data seimbang skenario 1

Pada gambar 4.12 grafik untuk akurasi dan loss pada tiap percobaan untuk data seimbang memberikan grafik yang baik pada akurasi dan loss. Untuk grafik terbaik ialah pada percobaan ke-2 dengan model LSTM yang berarti model classifier yang digunakan telah konvergen.

4.3.2.2 Hasil Validasi Data Seimbang (*Balance*) pada Skenario 2

Pengujian pada skenario kedua menggunakan *window size* sebesar 2 detik yang berarti 2000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.21 hasil pengujian yang dilakukan pada data seimbang (*Balance*) skenario 2 :

Tabel 4. 21 Hasil pengujian data seimbang (*Balance*) pada skenario 2

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 115	SimpleRNN	90,74%	82.84%
2		Epoch 65	LSTM	99.33%	97.73%
3	2	Epoch 135	SimpleRNN	95.58%	92.48%
4		Epoch 75	LSTM	98.65%	95.96%
5	3	Epoch 147	SimpleRNN	97.43%	94.61%
6		Epoch 62	LSTM	97.85%	96.59%

Pada tabel 4.21 dilihat nilai hasil terbaik pada skenario 2 data seimbang (*Balance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 99.33% dan akurasi testing 97.73%. Berikut gambar sinyal EKG akurasi dan loss pada skenario 2 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 13 Grafik (a) akurasi dan (b) loss pada tiap percobaan untuk data seimbang skenario 2

Pada gambar 4.13 grafik untuk akurasi dan loss pada tiap percobaan untuk data seimbang memberikan grafik yang baik pada akurasi dan loss. Untuk grafik terbaik ialah pada percobaan ke-2 dengan model LSTM yang berarti model classifier yang digunakan telah konvergen.

4.3.2.3 Hasil Validasi Data Seimbang (*Balance*) pada Skenario 3

Pengujian pada skenario ketiga menggunakan *window size* sebesar 3 detik yang berarti 3000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.22 hasil pengujian yang dilakukan pada data seimbang (*Balance*) skenario 3 :

Tabel 4. 22 Hasil pengujian data seimbang (*Balance*) pada skenario 3

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 102	SimpleRNN	90,77%	77.68%
2		Epoch 76	LSTM	99.42%	96.91%
3	2	Epoch 147	SimpleRNN	98.02%	92.66%
4		Epoch 67	LSTM	95.28%	92.13%
5	3	Epoch 155	SimpleRNN	96.60%	90.75%
6		Epoch 57	LSTM	97.92%	95.64%

Pada tabel 4.22 dilihat nilai hasil terbaik pada skenario 3 data seimbang (*Balance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 99.42% dan akurasi testing 96.91%. Berikut gambar sinyal EKG akurasi dan loss pada skenario 3 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 14 Grafik (a) akurasi dan (b) loss pada tiap percobaan untuk data seimbang skenario 3

Pada gambar 4.14 grafik untuk akurasi dan loss pada tiap percobaan untuk data seimbang memberikan grafik yang baik pada akurasi dan loss. Untuk grafik terbaik ialah pada percobaan ke-2 dengan model LSTM yang berarti model classifier yang digunakan telah konvergen.

4.3.2.4 Hasil Validasi Data Seimbang (*Balance*) pada Skenario 4

Pengujian pada skenario 4 menggunakan panjang waktu perekaman *window size* sebesar 4 detik yang berarti 4000 *node*. Percobaan yang dilakukan sebanyak 6 percobaan dan menggunakan rasio 90:10 dalam persentase (%) serta *batch size* sebesar 512 iterasi. Berikut Tabel 4.23 hasil pengujian yang dilakukan pada data seimbang (*Balance*) skenario 4 :

Tabel 4. 23 Hasil pengujian data seimbang (*Balance*) pada skenario 4

Percobaan Ke-	Hidden Layer	Early Stopping (Max: 1000)	Model Klasifikasi RNN	Acc Train	Acc test
1	1	Epoch 88	SimpleRNN	85.75%	71.22%
2		Epoch 83	LSTM	99.09%	96.15%
3	2	Epoch 136	SimpleRNN	95.86%	87.46%
4		Epoch 76	LSTM	97.89%	94.30%
5	3	Epoch 211	SimpleRNN	95.26%	89.74%
6		Epoch 63	LSTM	95.72%	93.16%

Pada tabel 4.23 dilihat nilai hasil terbaik pada skenario 4 data seimbang (*Balance*) pada tiap percobaan memiliki hasil akurasi terbaik adalah menggunakan Model klasifikasi RNN yaitu LSTM pada percobaan ke-2. Pada model menggunakan LSTM percobaan ke-2 mendapatkan akurasi training 99.09% dan akurasi testing 96.15%. Berikut gambar sinyal EKG akurasi dan loss pada skenario 4 menggunakan model klasifikasi SimpleRNN dan LSTM.

Percobaan ke-	Akurasi	Loss
1		
2		
3		
4		
5		
6		

(a)

(b)

Gambar 4. 15 Grafik (a) akurasi dan (b) loss pada tiap percobaan untuk data seimbang skenario 4

Pada gambar 4.15 grafik untuk akurasi dan loss pada tiap percobaan untuk data seimbang memberikan grafik yang baik pada akurasi dan loss. Untuk grafik terbaik ialah pada percobaan ke-2 dengan model LSTM yang berarti model classifier yang digunakan telah konvergen.

Pada pengujian data seimbang (*Balance*) ini telah didapatkan hasil terbaik dari perbandingan masing-masing skenario, dapat dilihat pada tabel 4.24.

Tabel 4. 24 Perbandingan nilai terbaik pada pengujian data *Balance* dari skenario pertama, skenario kedua, skenario ketiga, dan skenario keempat.

Skenario Ke-	Model Terbaik	Window Size	Hidden Layer	Early Stopping	Acc train	Acc test
1	LSTM	1000	1	Epoch 499	99.98%	99.96%
2	LSTM	2000	1	Epoch 65	99.33%	97.73%
3	LSTM	3000	1	Epoch 76	99.42%	96.91%
4	LSTM	4000	1	Epoch 83	99.09%	96.15%

Pada tabel 4.24 ditunjukkan bahwa pengujian terbaik dari data seimbang ialah skenario ke-1 dengan percobaan menggunakan model LSTM. Percobaan tersebut menggunakan *window size* 1 detik, menggunakan rasio training dan testing yaitu 90:10, *batch size* 512 iterasi, 1 hidden layer dan mendapatkan model terbaik yaitu pada 499 epoch. Berikut tabel *confusion matrix testing* dan *training* pada tiap kelas dari masing-masing skenario di atas.

Tabel 4. 25 Hasil *confusion matrix testing* pada data seimbang (*Balance*) skenario 1 percobaan ke-2

m = 2831		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	917	0	1
	<i>Hypertrophy (1)</i>	0	966	0
	<i>Dysrhythmia (2)</i>	0	0	947

Pada tabel 4.25 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data seimbang skenario pertama percobaan ke-2, nilai tersebut digunakan untuk menentukan TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 26 Nilai *confusion matrix testing* dari data seimbang (*Balance*) untuk masing-masing kelas pada skenario 1 percobaan ke-2

Kelas	TP	TN	FP	FN
0	917	1913	1	0
1	966	1864	0	0
2	947	1883	0	1

Pada tabel 4.26 merupakan nilai TP, TN, FP, dan FN dari kelas *Balance* yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario 1 percobaan ke-2.

Tabel 4. 27 Hasil *confusion matrix testing* pada data seimbang (*Balance*) skenario kedua percobaan ke-2

m = 1411		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	441	13	16
	<i>Hypertrophy (1)</i>	0	490	0
	<i>Dysrhythmia (2)</i>	1	2	448

Pada tabel 4.27 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data seimbang skenario kedua percobaan ke-2, nilai tersebut digunakan untuk menentukan nilai TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 28 Nilai *confusion matrix testing* dari data seimbang (*Balance*) untuk masing-masing kelas pada skenario kedua percobaan ke-2

Kelas	TP	TN	FP	FN
0	441	938	29	1
1	490	889	0	15
2	448	931	3	16

Pada tabel 4.28 merupakan nilai TP, TN, FP, dan FN dari kelas *Balance* yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario kedua percobaan ke-2.

Tabel 4. 29 Hasil *confusion matrix testing* pada data seimbang (*Balance*) skenario ketiga percobaan ke-2

m = 932		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	289	9	19
	<i>Hypertrophy (1)</i>	0	312	0
	<i>Dysrhythmia (2)</i>	1	0	311

Pada tabel 4.29 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data seimbang skenario ketiga percobaan ke-2, nilai tersebut digunakan untuk menentukan nilai TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 30 Nilai *confusion matrix testing* dari data seimbang (*Balance*) untuk masing-masing kelas pada skenario ketiga percobaan ke-2

Kelas	TP	TN	FP	FN
0	289	623	28	1
1	312	600	0	9
2	311	601	1	19

Pada tabel 4.30 merupakan TP, TN, FP, dan FN dari kelas *Balance* yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario ketiga percobaan ke-2.

Tabel 4. 31 Hasil *confusion matrix testing* pada data seimbang (*Balance*) skenario keempat percobaan ke-2

m = 702		Nilai Aktual		
		0	1	2
Nilai Prediksi	<i>Healthy control (0)</i>	221	11	11
	<i>Hypertrophy (1)</i>	0	233	0
	<i>Dysrhythmia (2)</i>	3	2	221

Pada tabel 4.31 merupakan nilai *confusion matrix testing* pada masing-masing kelas untuk data seimbang skenario keempat percobaan ke-2, nilai tersebut digunakan untuk menentukan TP, TN, FP, dan FN agar dapat menghitung nilai validasi.

Tabel 4. 32 Nilai *confusion matrix testing* dari data seimbang (*Balance*) untuk masing-masing kelas pada skenario keempat percobaan ke-2

Kelas	TP	TN	FP	FN
0	221	454	22	3
1	233	442	0	13
2	221	454	5	11

Pada tabel 4.30 merupakan nilai TP, TN, FP, dan FN dari kelas *Balance* yang ditentukan dari *confusion matrix* untuk menentukan nilai validasi pada skenario keempat percobaan ke-2.

Berdasarkan hasil *confusion matrix* yang didapatkan, hasil evaluasi tertinggi pada skenario 1, skenario 2, skenario 3, dan skenario 4 dapat dilihat nilai performancenya pada tabel 4.33, 4.34, 4.35, dan 4.36.

Tabel 4. 33 Hasil performance dari data seimbang (*Balance*) model *Recurrent Neural Network* (RNN) terbaik skenario 1

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	99.96%	99.89%	100%	100%	99.94%
1	100%	100%	100%	100%	100%
0	99.96%	100%	99.94%	99.89%	99.94%
Rata-Rata	99.97%	99.96%	99.98%	99.96%	99.96%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 1 percobaan ke-2 pada tabel 4.33 rata-ratanya adalah akurasi sebesar 99.97%, sensitifitas sebesar 99.96%, spesifisitas sebesar 99.98%, presisi sebesar 99.96% dan F1-score sebesar 99.96%.

Tabel 4. 34 Hasil performance dari data seimbang (*Balance*) model *Recurrent Neural Network* (RNN) terbaik skenario 2

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	98.65%	96.55%	99.68%	99.33%	97.92%
1	98.93%	97.02%	100%	100%	98.49%
0	97.87%	99.77%	97%	93.82%	96.71%
Rata-Rata	98.48%	97.78%	98.89%	97.72%	97.70%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 2 percobaan ke-2 pada tabel 4.34 rata-ratanya adalah akurasi sebesar 98.48%, sensitifitas sebesar 97.78%, spesifisitas sebesar 98.89%, presisi sebesar 97.72% dan F1-score sebesar 97.70%.

Tabel 4. 35 Hasil performance dari data seimbang (*Balance*) model *Recurrent Neural Network* (RNN) terbaik skenario 3

Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	97.87%	94.24%	99.83%	99.67%	96.88%
1	99.04%	97.19%	100%	100%	98.57%
0	96.91%	99.65%	95.69%	91.16%	95.22%
Rata-Rata	97.94%	97.03%	98.51%	96.94%	96.89%

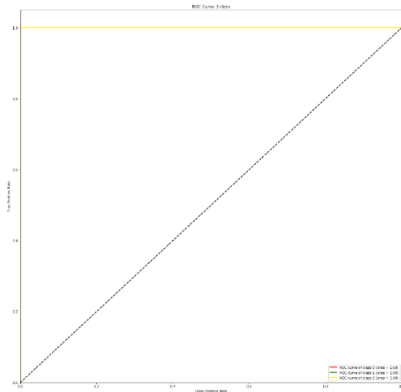
Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 3 percobaan ke-2 pada tabel 4.35 rata-ratanya adalah akurasi sebesar 97.94%, sensitifitas sebesar 97.03%, spesifisitas sebesar 98.51%, presisi sebesar 96.94% dan F1-score sebesar 96.89%.

Tabel 4. 36 Hasil performance dari data seimbang (*Balance*) model *Recurrent Neural Network* (RNN) terbaik skenario 4

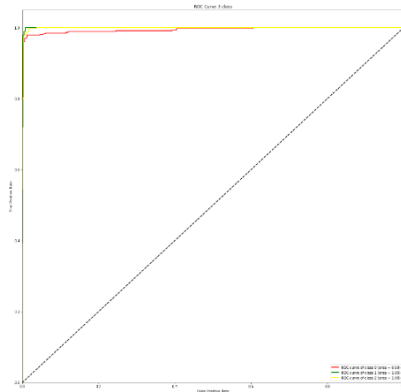
Kelas	Acc	Sensitifitas	Spesifisitas	Presisi	F1-Score
2	97.72%	95.25%	98.93%	97.78%	96.50%
1	98.14%	94.71%	100%	100%	97.28%
0	96.43%	98.66%	95.39%	90.94%	94.64%
Rata-Rata	97.43%	96.21%	98.11%	96.24%	96.14%

Nilai performance yang didapatkan dari masing-masing kelas untuk skenario 4 percobaan ke-2 pada tabel 4.36 rata-ratanya adalah akurasi sebesar 97.43%, sensitifitas sebesar 96.21%, spesifisitas sebesar 98.11%, presisi sebesar 96.24% dan F1-score sebesar 96.14%.

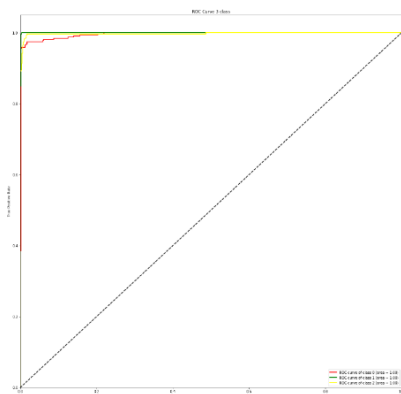
Pada tabel 4.33, tabel 4.34, tabel 4.35, dan tabel 4.36 dapat dilihat masing-masing nilai performance pada data yang telah diresampling atau data seimbang (*Balance*) menggunakan model RNN mendapatkan nilai yang lebih baik dari data tidak seimbang (*Imbalance*). Berikut gambar kurva roc dan presisi-recall pada masing-masing skenario terbaik.



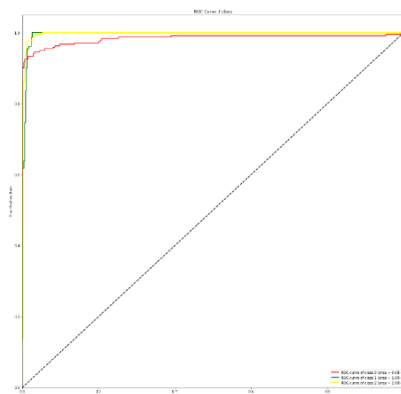
(a)



(b)



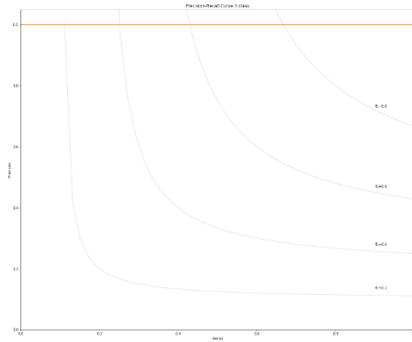
(c)



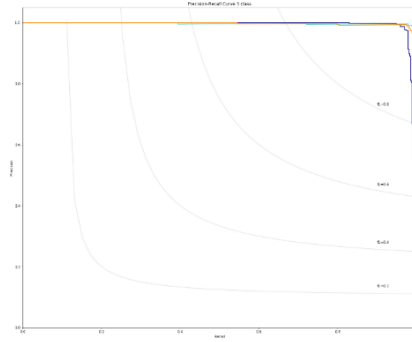
(d)

Gambar 4. 16 Hasil pengujian kurva ROC terbaik dari (a) skenario 1 (b) skenario 2 (c) skenario 3 (d) skenario 4

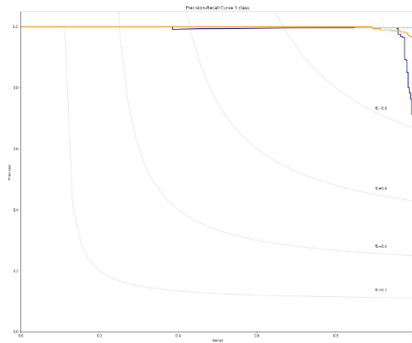
Kurva roc pada masing-masing skenario memperlihatkan hasil yang baik dengan kurva yang mendekati pojok kiri atas. Hasil kurva dari keempat skenario telah mendapatkan model classifier yang cocok.



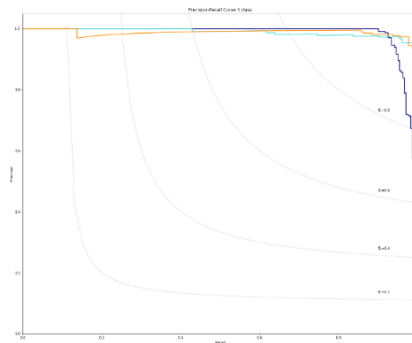
(a)



(b)



(c)



(d)

Gambar 4. 17 Hasil pengujian presisi-recall terbaik dari (a) skenario 1 (b) skenario 2 (c) skenario 3 (d) skenario 4

Pada kurva presisi-recall memperlihatkan hasil yang baik pada masing-masing skenario yang berarti kurva mendapatkan model pengelompokkan terbaik dikarenakan kurva telah mendekati pojok kanan atas.

4.4 Analisa

Analisa pada pengujian ini dapat dilihat berdasarkan tabel 4.37 dan 4.38. Pada tabel tersebut merupakan skenario dari masing-masing percobaan pada data tidak seimbang dan data seimbang.

Tabel 4. 37 Hasil pengujian terbaik pada data tidak seimbang (*Imbalance*)

Window Size	Hidden Layer	Model RNN	Epoch	Acc train	Acc test	Sen	Spe	Pre	F1	Acc
1 detik	1	SimpleRNN	63	99.56%	96.10%	93.55%	97.38%	88.06%	90.64%	97.40%
1 detik	1	LSTM	62	99.63%	96.96%	94.02%	97.38%	92.67%	93.29%	97.97%
1 detik	2	SimpleRNN	62	99.48%	95.41%	94.35%	97.66%	84.26%	88.71%	96.94%
1 detik	2	LSTM	58	99.56%	96.70%	92.54%	97.36%	91.98%	92.07%	97.80%
1 detik	3	SimpleRNN	59	99.50%	95.49%	94.80%	97.59%	85.78%	89.74%	96.99%
1 detik	3	LSTM	81	99.81%	97.14%	93.99%	97.21%	93.07%	93.52%	98.09%
2 detik	1	SimpleRNN	62	99.01%	86.80%	79.53%	87.59%	58.10%	64.60%	91.20%
2 detik	1	LSTM	58	99.26%	92.01%	83.41%	91.95%	75.65%	79.05%	94.67%
2 detik	2	SimpleRNN	60	98.89%	89.93%	87.65%	92.27%	65.00%	72.20%	93.28%
2 detik	2	LSTM	54	98.82%	91.31%	81.63%	90.87%	74.77%	77.84%	94.21%
2 detik	3	SimpleRNN	60	98.95%	89.58%	84.52%	91.88%	68.68%	74.69%	93.05%
2 detik	3	LSTM	71	99.16%	90.79%	82.46%	92.88%	73.99%	77.69%	93.86%
3 detik	1	SimpleRNN	58	96.66%	78.64%	48.25%	75.59%	50.44%	49.03%	85.76%
3 detik	1	LSTM	56	98.25%	89.32%	75.65%	87.35%	69.43%	71.82%	92.88%
3 detik	2	SimpleRNN	55	98.52%	86.97%	72.91%	85.65%	57.47%	62.78%	91.31%
3 detik	2	LSTM	58	98.75%	86.19%	69.26%	83.88%	57.28%	61.59%	90.79%
3 detik	3	SimpleRNN	55	97.82%	84.63%	71.85%	82.82%	54.15%	59.00%	89.75%
3 detik	3	LSTM	56	98.49%	87.23%	71.95%	84.82%	61.49%	65.63%	91.49%
4 detik	1	SimpleRNN	55	97.27%	76.57%	48.37%	73.86%	45.98%	46.94%	84.38%
4 detik	1	LSTM	55	98.01%	82.86%	70.29%	82.00%	53.81%	58.70%	88.57%
4 detik	2	SimpleRNN	53	98.01%	82.16%	69.06%	80.39%	50.15%	54.61%	88.11%
4 detik	2	LSTM	54	98.21%	84.96%	73.75%	85.47%	54.07%	59.28%	89.97%
4 detik	3	SimpleRNN	60	98.28%	85.66%	69.73%	86.06%	56.21%	60.92%	90.44%
4 detik	3	LSTM	57	98.32%	84.26%	68.66%	83.64%	57.57%	61.71%	89.51%

Tabel 4. 38 Hasil pengujian terbaik pada data seimbang (*Balance*)

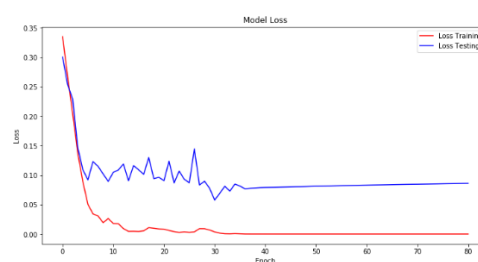
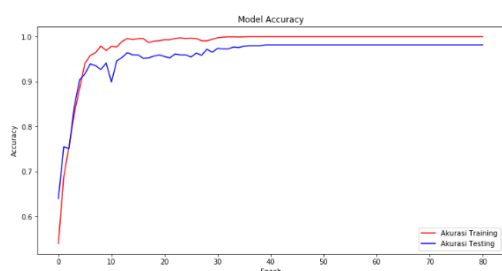
Window Size	Hidden Layer	Model RNN	Epoch	Acc train	Acc test	Sen	Spe	Pre	F1	Acc
1 detik	1	SimpleRNN	158	96.06%	92.19%	92.23%	96.28%	91.93%	91.91%	94.79%
1 detik	1	LSTM	499	99.98%	99.96%	99.96%	99.98%	99.96%	99.96%	99.97%
1 detik	2	SimpleRNN	85	96.89%	94.70%	94.64%	97.40%	94.56%	94.58%	96.46%
1 detik	2	LSTM	73	99.55%	98.86%	98.88%	99.44%	98.83%	98.85%	99.24%
1 detik	3	SimpleRNN	323	99.39%	98.30%	98.30%	99.15%	98.29%	98.29%	98.86%
1 detik	3	LSTM	81	98.48%	97.24%	97.27%	98.65%	97.18%	97.20%	98.16%
2 detik	1	SimpleRNN	115	90.74%	82.84%	83.20%	91.80%	82.86%	82.44%	88.56%
2 detik	1	LSTM	65	99.33%	97.73%	97.78%	98.89%	97.72%	97.70%	98.48%
2 detik	2	SimpleRNN	135	95.58%	92.48%	92.48%	96.29%	92.54%	92.43%	94.99%
2 detik	2	LSTM	75	98.65%	95.96%	96.07%	98.06%	95.91%	95.88%	97.30%
2 detik	3	SimpleRNN	147	97.43%	94.61%	95.02%	97.52%	94.50%	94.44%	96.40%
2 detik	3	LSTM	62	97.85%	96.59%	96.62%	98.32%	96.56%	96.56%	97.73%
3 detik	1	SimpleRNN	102	90.77%	77.68%	80.89%	90.61%	77.82%	75.78%	85.12%
3 detik	1	LSTM	76	99.42%	96.91%	97.03%	98.51%	96.94%	96.89%	97.94%
3 detik	2	SimpleRNN	147	98.02%	92.66%	93.01%	96.56%	92.75%	92.50%	95.11%
3 detik	2	LSTM	67	95.28%	92.13%	92.54%	96.21%	92.19%	92.10%	94.75%
3 detik	3	SimpleRNN	155	96.60%	90.75%	91.37%	95.72%	90.83%	90.50%	93.83%
3 detik	3	LSTM	57	97.92%	95.64%	95.78%	97.90%	95.71%	95.60%	97.09%
4 detik	1	SimpleRNN	88	85.75%	71.22%	71.75%	86.53%	71.96%	70.34%	80.81%
4 detik	1	LSTM	83	99.09%	96.15%	96.21%	98.11%	96.24%	96.14%	97.43%
4 detik	2	SimpleRNN	136	95.86%	87.46%	89.05%	94.48%	87.88%	86.96%	91.64%
4 detik	2	LSTM	79	97.89%	94.30%	94.30%	97.16%	94.44%	94.32%	96.20%
4 detik	3	SimpleRNN	211	95.26%	89.74%	89.78%	94.97%	89.90%	89.69%	93.16%
4 detik	3	LSTM	63	95.75%	93.16%	93.44%	96.71%	93.32%	93.12%	95.44%

Pada tabel 4.37 dan 4.38 dapat disimpulkan hasil terbaik dari pengujian data tidak seimbang (*Imbalance*) dan data seimbang (*Balance*). Kedua tabel tersebut menunjukkan hasil akurasi training dan testing terbaik ialah menggunakan model LSTM dikarenakan model LSTM mengklasifikasikan data lebih spesifik dan dapat mengatasi *vanishing gradient* namun pada pengujian data tidak seimbang (*Imbalance*) pada skenario 4 didapatkan model SimpleRNN merupakan hasil terbaik. Untuk data tidak seimbang (*Imbalance*) model terbaik yang didapatkan dari keseluruhan percobaan ialah pada skenario 1 dengan menggunakan 3 hidden layer, pemberhentian awal epoch ialah 81 epoch, dan model LSTM. Akurasi yang didapatkan untuk akurasi training sebesar 99.81%, akurasi testing sebesar 97.14%,

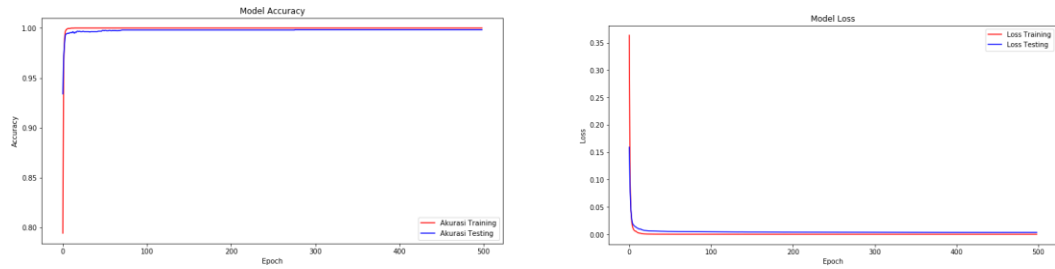
sensitivitas sebesar 93.99%, spesifisitas sebesar 97.21%, presisi sebesar 93.07%, F1-score sebesar 93.52% dan akurasi validasi sebesar 98.09%. Sedangkan untuk data seimbang (*Balance*) model terbaik yang didapatkan dari seluruh percobaan ialah pada skenario 1 dengan menggunakan menggunakan 1 hidden layer, pemberhentian awal epoch ialah 499 epoch, dan model LSTM. Akurasi yang didapatkan untuk akurasi training sebesar 99.98%, akurasi testing 99.96%, sensitivitas sebesar 99.98%, spesifisitas sebesar 99.98%, presisi sebesar 99.96% dan akurasi validasi sebesar 99.97%. Berikut rangkuman hasil terbaik pada tiap percobaan dan skenario pada data tidak seimbang (*Imbalance*) dan data seimbang (*Balance*)

Tabel 4. 39 Hasil Terbaik dari data Tidak Seimbang (*Imbalance*) dan data Seimbang (*Balance*)

Window Size	Model Terbaik	Hidden Layer	Epoch	Acc Train	Acc Test
Hasil Terbaik Data Tidak Seimbang (<i>Imbalance</i>)					
1000	LSTM	3	81	99.81%	97.14%
2000	LSTM	1	58	99.26%	92.01%
3000	LSTM	1	56	98.25%	89.32%
4000	SimpleRNN	3	60	98.28%	85.66%
Hasil Terbaik Data Seimbang (<i>Balance</i>)					
1000	LSTM	1	499	99.98%	99.96%
2000	LSTM	1	65	99.33%	97.73%
3000	LSTM	1	76	99.42%	96.91%
4000	LSTM	1	83	99.09%	96.15%



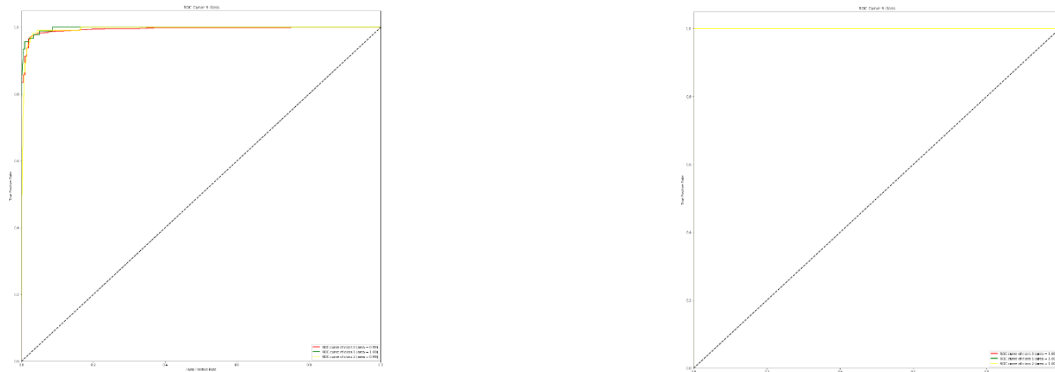
(a)



(b)

Gambar 4. 18 Grafik hasil pengujian terbaik dari (a) data tidak seimbang (*Imbalance*) dan (b) data seimbang (*Balance*)

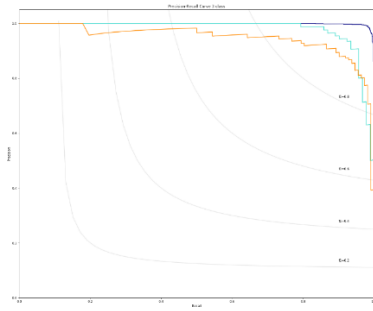
Pada gambar 4.18 dapat dilihat grafik yang didapatkan pada kedua pengujian telah mendapatkan hasil yang baik. Maka dari itu model yang dipilih pada kedua pengujian ini sudah tepat untuk mengklasifikasikan sinyal EKG pada penyakit jantung *Dysrhythmia*, *Hypertrophy*, dan *Healthy control*. Pada gambar 4.19 dan 4.20 didapatkan gambar dari kurva ROC dan Presisi-Recall dari pengujian data *Imbalance* dan data *Balance*.



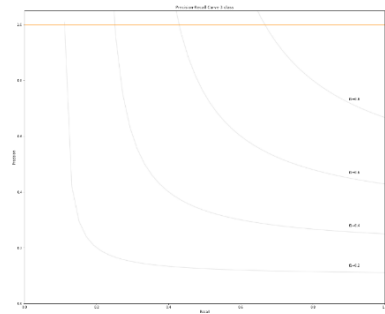
(a)

(b)

Gambar 4. 19 Kurva ROC dari pengujian (a) data *Imbalance* dan (b) data *Balance*



(a)



(b)

Gambar 4. 20 Kurva Presisi-Recall dari pengujian (a) data *Imbalance* dan (b) data *Balance*

Pada gambar 4.19 dapat dilihat bahwa kurva ROC pada data *Imbalance* dan data *Balance* sudah mendapatkan kurva yang baik karna kurva sudah mendekati pojok kiri atas yang berarti model classifier sudah cukup baik. Sedangkan pada gambar 4.20 didapatkan hasil Presisi-Recall pada data tidak seimbang (*Imbalance*) dan data seimbang (*Balance*) didapatkan kurva mendekati pojok kanan atas yang artinya model classifier yang digunakan pada kedua pengujian sudah baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian ini dapat ditarik kesimpulan:

1. Pada Pengujian ini menggunakan dua kasus yaitu pengujian data tidak seimbang dan data seimbang yang masing-masing melakukan 48 Model pengujian.
2. Skenario pada penelitian ini melakukan 4 pengujian. Skenario 1 yaitu berdasarkan jangka waktu rekaman 1 detik, skenario 2 yaitu berdasarkan jangka waktu rekaman 2 detik, skenario 3 yaitu berdasarkan jangka waktu rekaman 3 detik, dan skenario 4 yaitu berdasarkan jangka waktu rekaman 4 detik.
3. Penelitian ini menggunakan perbandingan *training* dan *testing* rasio yaitu 90 banding 10, 3 tuning hidden layer, melakukan early stopping model dan menggunakan *batch size* 512 iterasi.
4. Nilai akurasi terbaik yang didapatkan dari 48 model klasifikasi pada masing-masing kasus ialah kasus data tidak seimbang (*Imbalance*) nilai akurasi training sebesar 99.81% dan nilai akurasi testing sebesar 97.14%. Sedangkan untuk data seimbang (*Balance*) nilai akurasi training sebesar 99.98% dan nilai akurasi testing sebesar 99.96%.
5. Nilai validasi dari pengujian terbaik pada data tidak seimbang (*imbalance*) yang telah dijumlahkan mendapatkan rata-rata: nilai F1-Score sebesar 93.52%, nilai presisi sebesar 93.07%, nilai sensitivitas sebesar 93.99% , nilai spesifisitas sebesar 97.21%, dan nilai akurasi sebesar 98.09%.
6. Nilai validasi dari pengujian terbaik pada data seimbang (*balance*) yang telah dijumlahkan mendapatkan rata-rata: nilai F1-Score sebesar 99.96, nilai presisi sebesar 99.96%, nilai sensitivitas sebesar 99.96% dan nilai spesifisitas sebesar 99.98%, dan nilai akurasi sebesar 99.97%.

7. Kurva ROC dan kurva Presisi-Recall pada data tidak seimbang (*Imbalance*) dan data seimbang (*Balance*) sudah mendapatkan kurva yang baik karna pada kurva ROC sudah mendekati pojok kiri atas dan Kurva Presisi-Recall mendekati pojok kanan atas yang berarti masing-masing model classifier yang digunakan merupakan model yang tepat.

5.2 Saran

Saran untuk penelitian selanjutnya, yaitu:

1. Dapat menggunakan seluruh kelas pada dataset PTB Diagnostic ECG Database.
2. Dapat menggunakan model *Recurrent Neural Network* (RNN) yang lainnya seperti GRU dan GRU LSTM.
3. Penelitian ini dapat dilakukan dengan metode klasifikasi lain seperti Convolution Neural Network (CNN) dan lainnya.

DAFTAR PUSTAKA

- [1] B. I. Anita Agustina, Yati Afiyanti and Universitas, “PENGALAMAN PASIEN GAGAL JANTUNG KONGESTIF DALAM MELAKSANAKAN PERAWATAN MANDIRI,” vol. 1, no. 1, 2017.
- [2] Bakhtiar Rifai, “ALGORITMA NEURAL NETWORK UNTUK PREDIKSI PENYAKIT JANTUNG,” vol. IX, no. 1, pp. 1–9, 2013.
- [3] J. Pseudocode, V. Nomor, and A. P. Jantung, “ALGORITMA KLASIFIKASI DATA MINING NAÏVE BAYES BERBASIS PARTICLE SWARM,” pp. 11–14, 2014.
- [4] G. T. Ramadhan, Adiwijaya, and D. Q. Utama, “Klasifikasi Penyakit Aritmia Melalui Sinyal Elektrokardiogram (EKG) Menggunakan Metode Local Features dan Support Vector Machine,” vol. 5, no. 1, pp. 1787–1792, 2018.
- [5] D. K. Destiani *et al.*, “KLASIFIKASI SINYAL ECGAGAL JANTUNG MENGGUNAKAN WAVELET DAN JST PROPAGASI BALIK DENGAN MODIFIKASI GRADIEN KONJUGAT POLAK-RIBIERE HEART FAILURE ECG SIGNAL CLASSIFICATION USING WAVELET AND ANN BACKPROPAGATION WITH POLAK-RIBIERE CONJUGATE GRADIENT,” vol. 5, no. 1, pp. 1811–1818, 2018.
- [6] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, “Classification of Heart Disease using Artificial Neural Network Classification of Heart Disease using Artificial,” vol. 13, no. 3, 2013.
- [7] E. Olaniyi, O. K. Oyedotun, and A. Khashman, “Heart Diseases Diagnosis Using Neural Networks Arbitration,” no. November, 2015.
- [8] E. Messner, M. Zöhrer, and F. Pernkopf, “Heart Sound Segmentation - An event Detection Approach using Deep Recurrent Neural Networks,” *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1964–1974, 2018.
- [9] D. Y. ANANTA, “Klasifikasi Penyakit Gagal Jantung Kongestif

Menggunakan Artificial Neural Network (ANN) Berdasarkan Ekstraksi Fitur Multifractal Detrended Fluctuation Analysis (MFDFA) pada Variabilitas Denyut Jantung,” 2017.

- [10] S. W. Chalmers and C. R. Champion, “Pharmacotherapeutics in Cardiovascular Dysrhythmias,” *TJNP J. Nurse Pract.*, vol. 15, no. 1, p. 132–138.e1, 2019.
- [11] K. R. Chien, K. U. Knowlton, H. Zhu, and S. Chien, “Regulation of cardiac gene expression during myocardial growth and hypertrophy: molecular studies of an adaptive physiologic response,” vol. 5, no. 15, pp. 3037–3046, 2016.
- [12] A. T. Astuti, Nuryani, and A. Lestari, “Pembuatan Elektrokardiogram dan Penentuan Interval QRS secara Otomatis,” pp. 6–9, 2017.
- [13] H. Sulastomo *et al.*, “BUKU MANUAL KETERAMPILAN KLINIS INTERPRETASI PEMERIKSAAN ELEKTROKARDIOGRAFI (EKG),” pp. 1–30, 2019.
- [14] R. Bousseljot and D. K. und A. Schnabel, “Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet,” pp. 317–318, 1995.
- [15] A. Syukron and A. Subekti, “Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit,” vol. 5, no. 2, pp. 175–185, 2018.
- [16] J. Xiao, L. Xie, C. He, and X. Jiang, “Dynamic classifier ensemble model for customer classification with imbalanced class distribution,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3668–3675, 2012.
- [17] E. At, M. Aljourf, A.-M. F, and M. Shoukri, “Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method Technology & Optimization,” no. January 2016, 2018.
- [18] B. Santoso, H. Wijayanto, K. A. Notodiputro, and B. Sartono, “Synthetic

Over Sampling Methods for Handling Class Imbalanced Problems : A Review,” 2017.

- [19] S. H. Eбенуwa, M. H. D. S. Sharif, M. ALAZAB, I. (Senior Member, and A. AL-NEMRAT, “Variance Ranking Attributes Selection Techniques for Binary Classification Problem in Imbalance Data,” *IEEE Access*, vol. 7, pp. 24649–24666, 2019.
- [20] A. F. Al Musawi, “Introduction to Machine Learning,” no. February, 2018.
- [21] S. Das, A. Dey, A. Pal, and N. Roy, “Applications of Artificial Intelligence in Machine Learning : Review and Prospect,” vol. 115, no. 9, pp. 31–41, 2015.
- [22] A. Simon, M. S. Deo, S. Venkatesan, and D. R. R. Babu, “An Overview of Machine Learning and its Applications,” no. January, 2016.
- [23] A. Y. Prathama, A. Aminullah, and A. Saputra, “PENDEKATAN ANN (ARTIFICIAL NEURAL NETWORK) UNTUK PENENTUAN PROSENTASE BOBOT PEKERJAAN DAN ESTIMASI NILAI PEKERJAAN STRUKTUR PADA RUMAH SAKIT PRATAMA,” vol. 7, 2017.
- [24] M. Nielsen, “Neural Networks and Deep Learning.”
- [25] B. A. Thomas, “An introduction to neural networks for beginners.”
- [26] A. A. Rizal and S. Soraya, “Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory,” no. November 2018, 2019.
- [27] John A. Bullinaria, “Recurrent Neural Networks,” pp. 1–20, 2015.
- [28] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, “A Recurrent Latent Variable Model for Sequential Data,” pp. 1–9, 2015.
- [29] A. Darmawahyuni *et al.*, “Deep Learning with a Recurrent Network Structure in the Sequence Modeling of Imbalanced Data for ECG-Rhythm Classifier,” pp. 1–12, 2019.
- [30] S. Singh, S. K. Pandey, U. Pawar, and R. Ram, “Classification of ECG

- Arrhythmia using Recurrent Neural Networks,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1290–1297, 2018.
- [31] R. Dey and F. M. Salem, “Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks,” no. January, 2017.
- [32] J. Yang and J. Kim, “An accident diagnosis algorithm using long short-term memory,” *Nucl. Eng. Technol.*, vol. 50, no. 4, pp. 582–588, 2019.
- [33] N. K. Manaswi, *Deep Learning with Applications Using Python*. 2018.