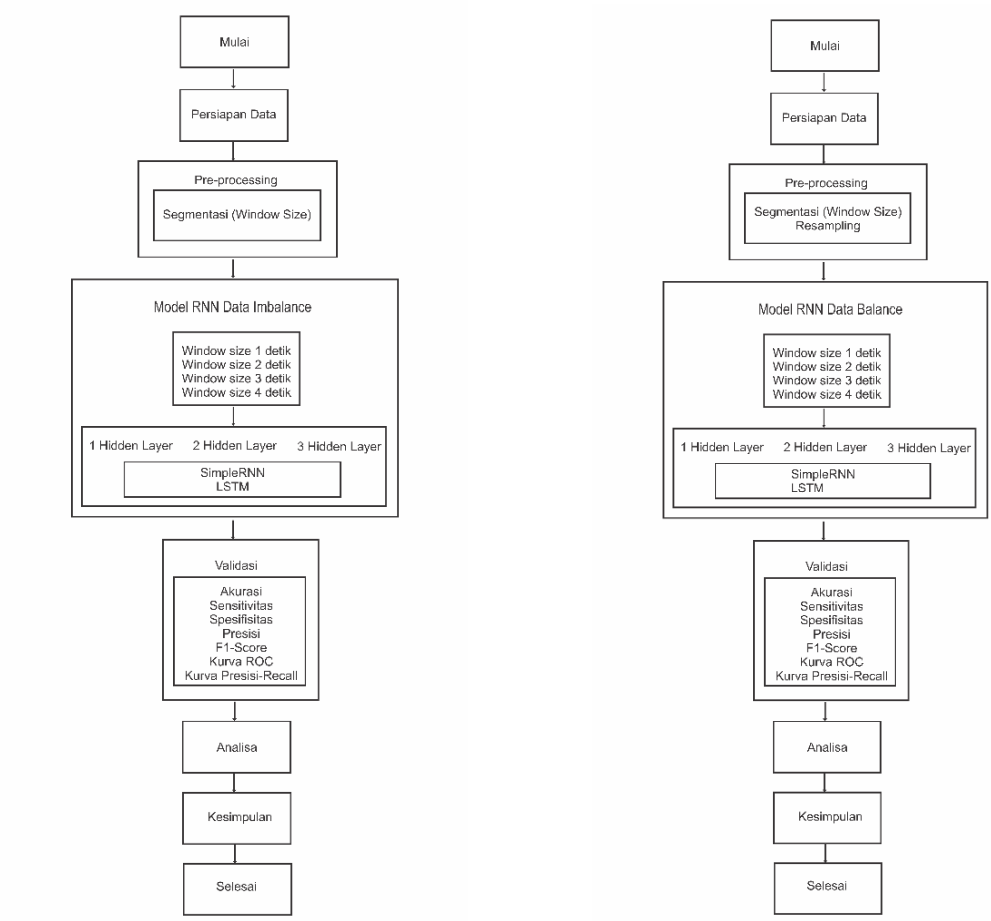
BAB III
METODOLOGI

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.



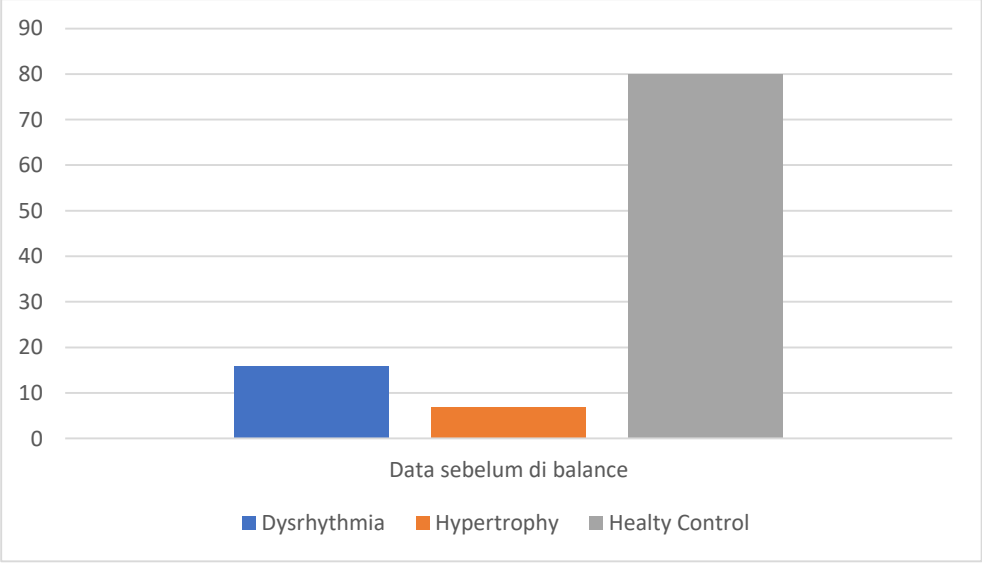
(a) **(b)**

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
Healthy control	0
Hypertrophy	1
Dysrhythmia	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 pertaman : *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:

window size = panjang waktu rekaman x frekuensi sampling (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

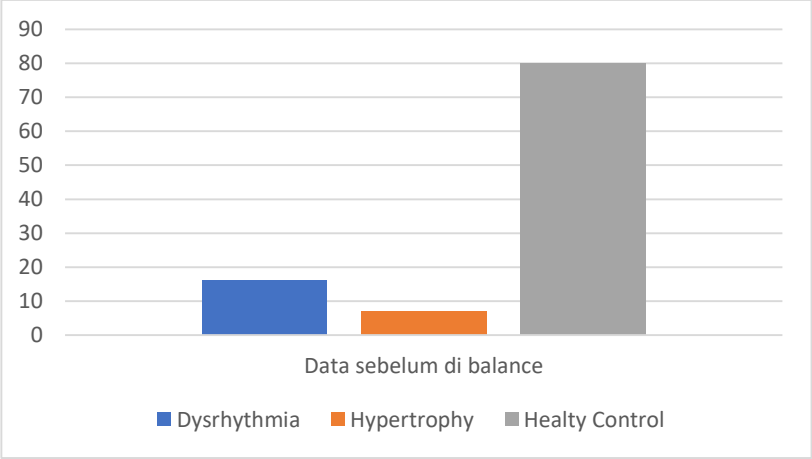
Resampling merupakan proses yang menambah sampel data yang tidak seimbang dari sampel data asli menjadi data seimbang. Penelitian ini menggunakan salah satu metode *resampling* untuk *multiclass* yaitu *Random Oversampling* (ROS). Metode *Random Oversampling* (ROS) menduplikasikan data minoritas menjadi 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 segementasi menghasilkan jumlah data yang berbeda-beda, berikut tabel 3.2 yang menunjukkan jumlah masing-masing data sebelum diresampling dan sesudah diresampling:

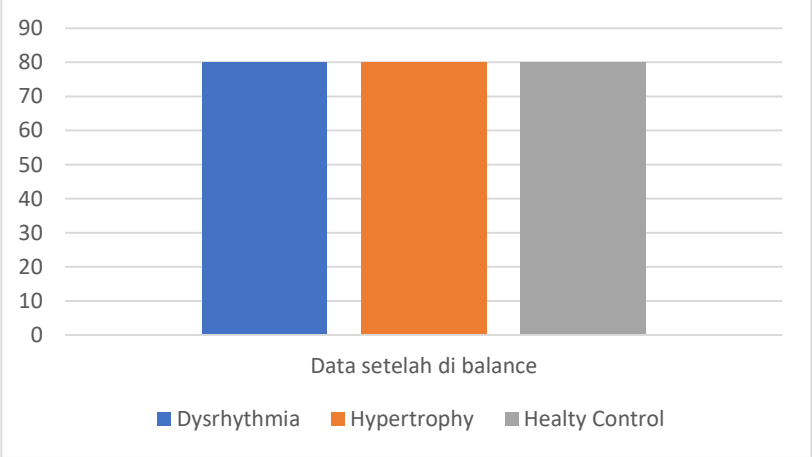
Tabel 3. 2 Jumlah data sebelum dan sesudah diresampling

Segementasi	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)



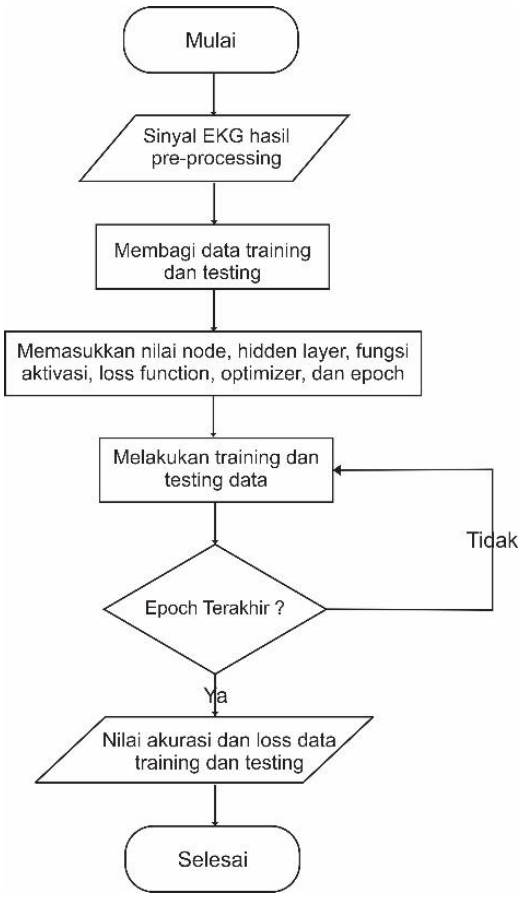
(b)

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:

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

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 pengelompokkan 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.