

PREDIKSI DETAK JANTUNG BERBASIS LSTM PADA RASPBERRY PI UNTUK PEMANTAUAN KESEHATAN PORTABEL

Ahmad Foresta Azhar Zen^{*1}, Eko Sakti Pramukantoro², Kasyful Amron³,
Viera Wardhani⁴, Putri Annisa Kamila⁵

^{1,2,3,4,5} Universitas Brawijaya, Malang

Email: ¹ahmadforesta@student.ub.ac.id, ²ekosakti@ub.ac.id, ³kasyful@ub.ac.id, ⁴viera_w.fk@ub.ac.id,
⁵putriannisa@ub.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 10 November 2023, diterima untuk diterbitkan: 08 Desember 2023)

Abstrak

Penyakit kardiovaskular atau *cardiovascular disease* (CVD) menduduki peringkat teratas penyebab kematian di dunia. Diperkirakan sekitar 17,9 juta jiwa meninggal akibat CVD pada tahun 2019, yang menyumbang sebanyak 32% dari seluruh kematian global. Penting untuk mendeteksi kelainan pada jantung sedini mungkin untuk mencegah kematian karena CVD. Peningkatan kesadaran tentang pentingnya pemantauan kesehatan diri sendiri telah mendorong perkembangan teknologi pemantauan kesehatan portabel. Dalam penelitian ini, kami mengusulkan model prediksi detak jantung berbasis *Long Short-Term Memory* (LSTM) dengan menggunakan fitur RR-Interval dan mengimplementasikan pada perangkat Raspberry Pi. Model berbasis LSTM merupakan salah satu jenis arsitektur jaringan saraf tiruan yang mampu menangani data berurutan dengan baik, sehingga sangat cocok untuk pemantauan dan prediksi detak jantung yang bersifat sekuensial. Raspberry Pi dikenal karena ukurannya yang kecil, harga yang terjangkau, kinerja yang andal, dan efisiensi komputasi yang baik. Raspberry Pi juga memungkinkan integrasi yang mudah dengan berbagai sensor, menjadikannya solusi yang cocok untuk pemantauan kesehatan yang portabel. Hasil penelitian ini menunjukkan bahwa model klasifikasi yang diusulkan memiliki kinerja yang baik dengan tingkat akurasi mencapai 96,66%. Implementasi inferensi pada Raspberry Pi juga menunjukkan performa yang baik, dengan waktu 4,82 detik untuk melakukan inferensi data sepanjang 100 detik, serta penggunaan memori sebesar 134,8MB.

Kata kunci: *prediksi detak jantung, LSTM, raspberry pi, ECG, rr-interval*

HEART BEAT PREDICTION BASED ON LSTM MODEL ON RASPBERRY PI FOR PORTABLE HEALTH

Abstract

Cardiovascular diseases (CVDs) rank as the top cause of global death. An estimated 17.9 million people succumbed to CVDs in 2019, constituting 32% of all global deaths. Detecting heart abnormalities as early as possible is crucial to prevent CVD-related fatalities. The growing awareness of the importance of self-health monitoring has driven the development of portable health monitoring technologies. In this study, we propose a Long Short-Term Memory (LSTM)-based heart beat prediction model using RR-Interval as features and implement it on the Raspberry Pi device. LSTM models are a type of artificial neural network architecture known for their ability to handle sequential data effectively, making them highly suitable for sequential heart rate monitoring and prediction. The Raspberry Pi is renowned for its compact size, affordability, reliable performance, and efficient computational capabilities. It also enables seamless integration with various sensors, making it an ideal solution for portable health monitoring. This research show that the proposed classification model performs well, achieving an accuracy rate of 96.66%. The implementation of inference on the Raspberry Pi also demonstrates good performance, with an average inference time of 4.82 seconds for processing 100 data points and a memory usage of 134.8MB.

Keywords: *heart beat prediction, LSTM, raspberry pi, ECG, rr-interval*

1. PENDAHULUAN

Penyakit kardiovaskular atau *cardiovascular disease* (CVD) menduduki peringkat teratas

penyebab kematian di dunia (World Health Organization, 2021). Diperkirakan sekitar 17,9 juta jiwa meninggal akibat CVD pada tahun 2019, yang

menyumbang sebanyak 32% dari seluruh kematian global. Penting untuk mendeteksi kelainan pada jantung sedini mungkin untuk mencegah kematian karena CVD.

Kesehatan merupakan aspek vital dalam kehidupan manusia, dan kesadaran tentang pentingnya pemantauan kesehatan diri sendiri telah menjadi fokus utama bagi banyak orang. Teknologi pemantauan kesehatan portabel telah muncul sebagai solusi yang menjanjikan untuk memberikan akses yang mudah dan terjangkau terhadap data kesehatan individu. Pemantauan detak jantung adalah salah satu parameter penting dalam diagnosis dan pemantauan kondisi kesehatan.

Dalam penelitian ini, kami fokus pada pengembangan dan implementasi model prediksi detak jantung berbasis *Long Short-Term Memory* (LSTM) pada perangkat Raspberry Pi. LSTM adalah salah satu jenis arsitektur jaringan saraf tiruan yang terkenal karena kemampuannya dalam menangani data berurutan atau sekuensial, sehingga sangat cocok untuk pemantauan dan prediksi detak jantung yang merupakan data sekuensial.

Penelitian sebelumnya (Shchetinin & Glushkova, 2022) telah mencatat kesuksesan dalam penggunaan model LSTM untuk memprediksi detak jantung dari data ECG (Elektrokardiogram) yang bersifat sekuensial. Namun, implementasi model ini pada perangkat portabel seperti Raspberry Pi adalah langkah penting untuk mewujudkan pemantauan kesehatan yang berkelanjutan dan praktis.

Raspberry Pi, sebagai komputer mini dengan ukuran dan konsumsi daya yang rendah, telah menjadi pilihan populer sebagai platform pengembangan untuk berbagai aplikasi, termasuk bidang kesehatan. Keunggulan portabilitas dan kemampuan komputasi dari Raspberry Pi menjadikannya pilihan ideal untuk diintegrasikan dengan berbagai sensor kesehatan.

Penelitian yang serupa telah dilakukan sebelumnya. Dalam penelitiannya, Ahsanuzzaman dkk. (2020) mengusulkan sebuah sistem alarm dan pemantauan ECG. Sistem tersebut berfokus pada prediksi aritmia (*atrial fibrillation*) dengan menggunakan arsitektur LSTM yang diimplementasikan ke dalam Raspberry Pi 3. Sistem tersebut menggunakan Arduino Uno sebagai *middleware* antara sensor dengan Raspberry Pi serta perangkat Android untuk menampilkan informasi ECG.

Penelitian ini memiliki tujuan untuk mengembangkan dan mengimplementasikan model prediksi detak jantung berbasis LSTM pada Raspberry Pi, serta memberikan pemahaman tentang manfaat dari hasil penelitian ini dalam meningkatkan akses dan pemantauan kesehatan secara mandiri. Dengan adanya hasil penelitian ini, diharapkan pemantauan kesehatan portabel menjadi lebih mudah diakses dan dapat membantu

meningkatkan kesadaran serta kualitas hidup masyarakat dalam menjaga kesehatan mereka.

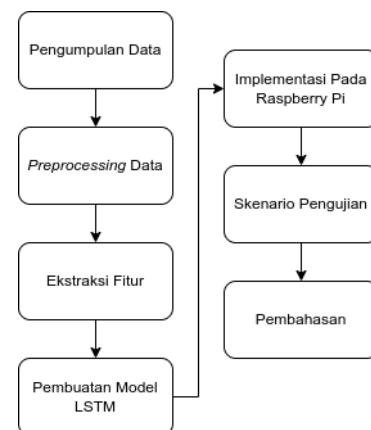
2. PENELITIAN TERKAIT

Pada penelitian sebelumnya, Pramukantoro dan Gofuku (2022) telah mengusulkan sejumlah algoritma klasifikasi detak jantung yang menggunakan fitur RR-Interval dengan 9 deskriptor menggunakan beberapa metode *machine learning* serta *deep learning*. *Machine learning* yang digunakan pada penelitian tersebut yaitu *Decision Tree*, *Gradient Boosting*, *k-Nearest Neighbors*, *Multi-layer Perceptron*, *Random Forest*, dan *Support Vector Machine*. Sedangkan model *deep learning* yang digunakan yaitu *artificial neural networks* (ANN). Penelitian tersebut mencapai akurasi 99,31% menggunakan fitur RR-Interval dengan *classifier decision tree* dan menggunakan teknik *random oversampling*. Pada penelitian tersebut juga dilakukan eksperimen implementasi inferensi secara *realtime* untuk memprediksi pasien sehat dan mendapatkan hasil yang baik.

Pada penelitian lain (Mondéjar-Guerra dkk., 2019) juga mengusulkan algoritma klasifikasi detak jantung dengan menggunakan fitur RR-Interval dengan 8 deskriptor. Penelitian tersebut menggunakan metode SVM dan mendapatkan nilai akurasi 76,2%. Penelitian tersebut juga menggunakan kombinasi RR-interval, *wavelet*, dan *higher order statistics* (HOS) dengan *classifier ensemble SVM* dan mendapat akurasi 94,5%.

3. METODE PENELITIAN

Pada penelitian ini, detak jantung diprediksi menggunakan model berbasis LSTM. Model tersebut kemudian diimplementasikan pada Raspberry Pi dan diuji menggunakan data detak jantung yang diambil langsung menggunakan sensor ECG. Metode penelitian yang digunakan pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

3.1. Pengumpulan Data

Sumber data detak jantung yang diperlukan untuk pelatihan dan pengujian diperoleh dari dataset publik *MIT-BIH arrhythmia database* (Moody & Mark, 2001). Dataset ini berisi 48 rekaman ECG yang diambil dari 47 pasien. Dari 48 rekaman tersebut, data 102, 104, 107, dan 217 tidak digunakan karena data tersebut berisi data jantung yang dipacu. Setiap rekaman terdiri dari dua kanal dengan panjang 30 menit. Pada penelitian ini hanya kanal *Modified Lead II* yang digunakan.

Dataset telah diberi label dan anotasi oleh ahli pada tiap detak jantungnya. Dari anotasi tersebut kami mengelompokkan ke dalam lima kelas seperti yang direkomendasikan oleh AAMI yaitu N, SVEB, VEB, F, dan Q (Association for the Advancement of Medical Instrumentation, 1998). Pengelompokan tersebut ditunjukkan pada tabel 1.

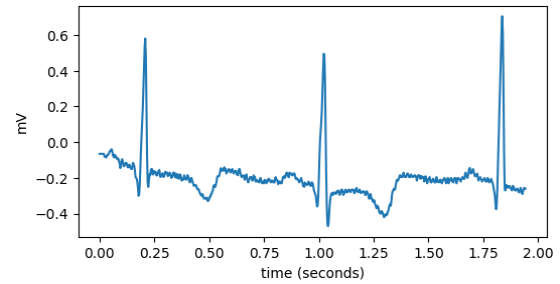
Tabel 1. Kelas Rekomendasi AAMI

AAMI	MIT-BIH
Normal (N)	N, L, R, e, j
Supraventricular Ectopic Beat (S)	A, a, J, S
Ventricular Ectopic Beat (V)	V, E
Fusion (F)	F
Unknown (Q)	/, f, Q

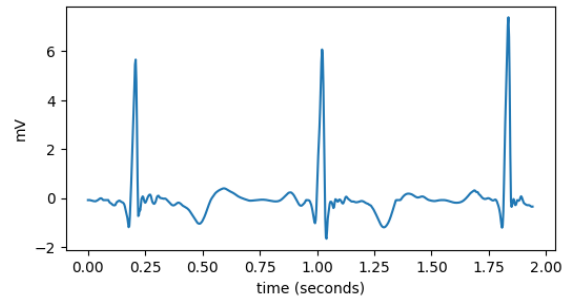
Data detak jantung kemungkinan mengandung *noise* atau *outlier*. Oleh karena itu, data perlu di-*preprocess* dengan menggunakan teknik penghapusan *noise* dan normalisasi untuk memastikan kualitas dan keandalan data yang digunakan dalam pelatihan model.

3.2. Preprocessing Data

Data detak jantung yang telah dikumpulkan akan dilakukan penghapusan *baseline*, penghapusan *noise* frekuensi tinggi, dan normalisasi. Penghapusan *baseline* dilakukan untuk menghilangkan *noise baseline wander* (BW). BW merupakan *noise* berfrekuensi rendah pada ECG yang dapat disebabkan oleh banyak hal, seperti pernapasan, elektroda yang bermuatan listrik, atau gerakan dari pasien (Lenis dkk., 2017). Setelah dilakukan penghapusan *baseline*, data dinormalisasi untuk menghindari skala yang berbeda serta memudahkan proses pelatihan model. Normalisasi dilakukan menggunakan *Z-score normalization*. Perbandingan data sebelum dan setelah dilakukan *preprocessing* ditunjukkan oleh gambar 2a dan gambar 2b. Setelah dilakukan normalisasi, data akan dibagi menjadi dua set yaitu data latihan (*training data*) sebesar 70%, dan data uji (*testing data*) sebesar 30%. Pengelompokan ini penting untuk menghindari *overfitting* dan memastikan generalisasi model yang baik.



Gambar 2a. Data Sebelum *Preprocessing*



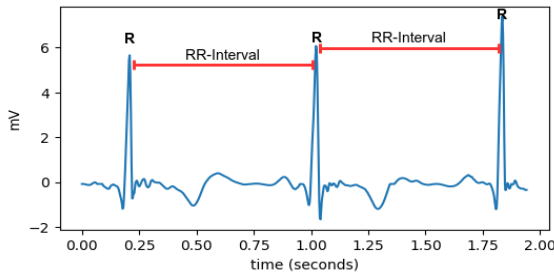
Gambar 2b. Data Setelah *Preprocessing*

Data pada dataset memiliki kelas yang tidak seimbang. Kelas N pada dataset hampir mencapai 90% dari total data. Untuk mengatasi masalah tersebut, kami juga melakukan *oversampling* untuk menyeimbangkan kelas pada dataset. Kami menggunakan teknik *Random Oversampling* (ROS), *Synthetic Minority Oversampling Technique* (SMOTE), dan *Adaptive Synthetic Sampling* (ADASYN). ROS mengulangi secara acak sampel-sampel pada kelas minoritas, sehingga jumlahnya seimbang dengan kelas mayoritas (Mohammed dkk., 2020). SMOTE membuat sampel sintetis pada kelas minoritas dengan menggunakan algoritma *k-nearest neighbor* (Chawla dkk., 2002). ADASYN merupakan pengembangan dari SMOTE. ADASYN membuat sampel sintetis pada kelas minoritas berdasarkan seberapa sulit kelas itu dipelajari untuk menghasilkan lebih banyak sampel sintetis pada kelas minoritas yang sulit dipelajari dibandingkan dengan yang lebih mudah dipelajari (He dkk., 2008).

3.3. Ekstraksi Fitur

Pada penelitian ini kami menggunakan RR-Interval sebagai fitur yang digunakan. RR-Interval merupakan jarak antara dua titik R pada sinyal ECG. Gambar 3 menunjukkan visualisasi dari RR-Interval. RR-Interval digunakan sebagai fitur karena pada penelitian sebelumnya (Pramukantoro & Gofuku, 2022) telah mengusulkan klasifikasi detak jantung menggunakan fitur RR-Interval dan mendapatkan hasil yang baik. Selain itu, sensor yang digunakan, yaitu Polar H10, dapat mengirimkan data RR-Interval setiap detik sehingga membuka peluang untuk pengembangan prediksi secara *real-time* pada penelitian mendatang.

Dari data RR-Interval, kami melakukan ekstraksi menjadi 9 fitur mengikuti pendekatan pada penelitian terkait, yaitu RR0, RR-1, RR+1, RR0/avgRR, tRR0, RR-1/avgRR, RR-1/RR0, RR+1/avgRR, dan RR+1/RR0 (Pramukantoro & Gofuku, 2022). Deskripsi tiap fitur dapat dilihat pada tabel 2. Ekstraksi fitur dilakukan dengan menggunakan jendela sepanjang 42 data. Rata-rata RR-Interval (avgRR) yang digunakan yaitu rata-rata dari 42 data termasuk RR-0.



Gambar 3. RR-Interval

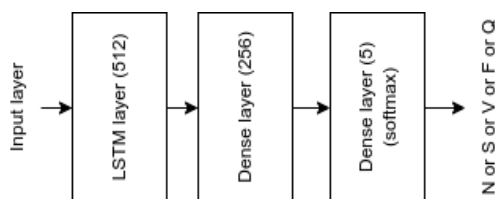
Tabel 2. Deskripsi RR-Interval (Pramukantoro & Gofuku, 2022)

Fitur	Deskripsi
RR0	Nilai RRi saat ini
RR-1	Nilai RRi sebelumnya
RR+1	Nilai RRi selanjutnya
RR0/avgRR	Nilai RRi saat ini dibagi dengan rata-rata 42 RRi sebelumnya
tRR0	(RRi saat ini - rata-rata RR) / stddevRR
RR-1/avgRR	Nilai RRi sebelumnya / rata-rata RRi
RR-1/RR0	Nilai RRi sebelumnya / RRi saat ini
RR+1/avgRR	Nilai RRi selanjutnya / rata-rata RRi
RR+1/RR0	Nilai RRi selanjutnya / RRi saat ini

3.4. Pembuatan Model LSTM

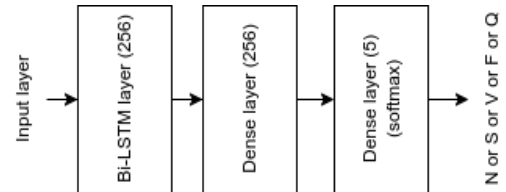
Terdapat tiga model yang kami gunakan dalam penelitian ini. Model yang kami gunakan yaitu model *Long Short-Term Memory* (LSTM), *Bidirectional LSTM* (Bi-LSTM), serta *LSTM Fully Convolutional Network* (LSTM-FCN).

LSTM merupakan jenis *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi masalah hilangnya informasi lampau yang terjadi pada RNN tradisional (Hochreiter & Schmidhuber, 1997). LSTM mampu mengingat informasi yang disimpan dalam jangka panjang. Model LSTM yang digunakan pada penelitian ini memiliki arsitektur seperti pada gambar 4. Model terdiri dari satu buah LSTM layer dan dua buah *dense layer*. Model LSTM dilatih dengan menggunakan dua set *hyperparameter* yang berbeda. Model pertama menggunakan 512 unit LSTM dan 256 unit *dense*. Model kedua menggunakan 256 unit LSTM dan 128 unit *dense*.



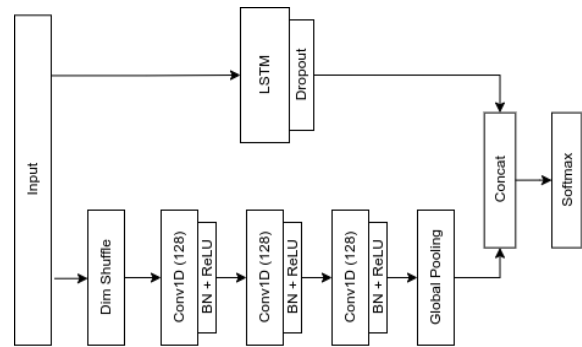
Gambar 4. Arsitektur LSTM

Bi-LSTM merupakan *Recurrent Neural Network* (RNN) yang dapat dilatih secara bersamaan secara dua arah (maju dan mundur) dengan *hidden layer* terpisah (Yu dkk., 2019). Bi-LSTM menggabungkan LSTM dengan *bidirectional RNN* (BRNN) untuk mengatasi keterbatasan RNN konvensional yang hanya dapat memanfaatkan konteks sebelumnya. Arsitektur model Bi-LSTM yang digunakan dalam penelitian ini dapat dilihat pada gambar 5.



Gambar 5. Arsitektur Bi-LSTM

Model LSTM-FCN merupakan model yang menggabungkan *Long Short-Term Memory* (LSTM) dengan *Fully Convolutional Network* (FCN) (Karim dkk., 2018). Model terdiri dari satu *layer LSTM* dan tiga *layer convolution* satu dimensi yang digabungkan. Sebelum LSTM layer terdapat *dimension shuffle* yang akan mengacak input yang dilakukan untuk menghindari *overfitting*. Selain itu kami memodifikasi model LSTM-FCN dengan menukar posisi *dimension shuffle block* yang sebelumnya berada pada sebelum blok LSTM ke sebelum blok *convolution* dengan tujuan mempertahankan dimensi waktu. Arsitektur model LSTM-FCN yang telah dimodifikasi dapat dilihat pada gambar 6.



Gambar 6. Arsitektur LSTM-FCN Modifikasi

Seluruh model dilatih menggunakan *framework* Tensorflow dengan optimasi Adam, ukuran *batch* 256, dan jumlah *epoch* 50. Model dilatih dengan menggunakan data latih yang telah diproses. Model dilatih menggunakan server DGX A100. Pada proses pelatihan juga dilakukan optimasi dan *tuning* parameter untuk mencapai tingkat akurasi yang optimal.

3.5. Implementasi pada Raspberry Pi

Model LSTM yang telah dilatih akan diintegrasikan ke dalam perangkat Raspberry Pi. Model diimplementasikan dengan menggunakan bahasa pemrograman Python menggunakan *framework* Flask.

Sebelum diimplementasikan, model terlebih dahulu dilakukan optimasi agar dapat berjalan dengan baik pada perangkat portabel seperti Raspberry Pi. Model dikonversi menjadi model TensorFlow Lite. Model TensorFlow Lite memiliki ukuran serta dependensi yang lebih kecil daripada model tensorflow serta optimal untuk dijalankan pada perangkat dengan spesifikasi rendah.

3.6. Skenario Pengujian

Model yang telah diimplementasikan pada Raspberry Pi akan diuji dengan menggunakan data detak jantung yang diperoleh dari sensor ECG Polar H10. Performa model pada Raspberry Pi akan dievaluasi berdasarkan tingkat akurasi prediksi dan efisiensi komputasi. Pengujian ini diperlukan untuk memastikan bahwa model yang diusulkan dapat diandalkan, efisien, dan mampu memberikan prediksi detak jantung dengan cepat dalam situasi pemantauan kesehatan portabel yang sesungguhnya.

Model Raspberry Pi yang digunakan untuk pengujian yaitu Raspberry Pi 3B. Perangkat ini memiliki spesifikasi prosesor ARMv8 *quad-core* 1,2GHz, dengan RAM 1GB LPDDR2. Perangkat ini juga mendukung koneksi *ethernet* hingga 100Mbps serta 2.4Ghz WiFi yang memberikan dukungan yang baik untuk konektivitas perangkat. Pengujian dilakukan dengan melakukan inferensi menggunakan data detak jantung yang memiliki panjang 100 detik kemudian direkam waktu prediksi, penggunaan CPU, serta penggunaan memori. Pengujian dilakukan 5 kali untuk setiap *classifier* dan diambil nilai rata-rata dari hasil tersebut.

3.7. Pembahasan

Model akan diuji menggunakan dataset pengujian yang telah dipisahkan dari dataset pelatihan. Pengujian ini bertujuan untuk mengevaluasi akurasi dan kinerja model LSTM pada

perangkat Raspberry Pi. Hasil pengujian akan dianalisis secara mendalam dan dibandingkan dengan metode prediksi detak jantung lain yang telah ada. Hasil dari pengujian ini akan menjadi subjek utama pembahasan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Training Model

Setelah melakukan pelatihan dan pengujian model prediksi detak jantung berbasis LSTM menggunakan data uji, kami memperoleh hasil yang ditunjukkan pada tabel 3. Seluruh model yang telah kami latih berhasil mencapai tingkat akurasi yang tinggi dalam memprediksi detak jantung. Seluruh model mencapai akurasi lebih dari 96% dengan akurasi tertinggi 96,66% yang diraih oleh model LSTM(256) dan LSTM-FCN modifikasi. Hasil ini menunjukkan bahwa model yang dilatih memiliki kualitas yang baik dalam melakukan prediksi.

Tabel 4 dan tabel 5 menunjukkan *confusion matrix* pada evaluasi model LSTM-FCN dan model LSTM(256). *Confusion matrix* tersebut menunjukkan bahwa kedua model mampu mengenali kelas N dengan cukup baik. Namun, terdapat sejumlah misklasifikasi yang signifikan pada kelas lainnya, terutama pada kelas F dan Q.

Evaluasi hasil pada dataset yang telah mengalami *oversampling* ditunjukkan oleh tabel 6. Hasil ini menunjukkan bahwa, dalam penelitian ini, penggunaan teknik *oversampling* justru mengakibatkan penurunan kinerja model. Tabel 7 menunjukkan *confusion matrix* dari model LSTM(256) yang dilatih menggunakan data yang telah dilakukan *oversampling*. *Confusion matrix* tersebut menunjukkan bahwa model yang telah dilatih dengan data yang mengalami *oversampling* cenderung memiliki keterbatasan dalam mengenali kelas N dibandingkan pada model yang dilatih dengan data yang tidak dilakukan *oversampling*.

Tabel 8 menunjukkan komparasi *classifier* terbaik yang kami usulkan, yaitu LSTM-FCN Modifikasi, dengan *classifier* dari penelitian-penelitian lain yang terkait. Dari tabel tersebut menunjukkan *classifier* yang kami usulkan memiliki performa yang kompetitif jika dibandingkan dengan sejumlah penelitian sebelumnya.

Tabel 3. Hasil Evaluasi Model Menggunakan Data Uji

Feature	Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
RR-I	LSTM(512)	96,49	96,24	96,49	96,27
	LSTM(256)	96,66	96,49	96,66	96,38
	Bi-LSTM	96,50	96,38	96,50	96,29
	LSTM-FCN	96,34	96,10	96,34	96,01
	Modified LSTM-FCN	96,66	96,49	96,66	96,44

Tabel 4. Confusion Matrix Model Modified LSTM-FCN

		Classifier				
		n	s	v	f	q
Reference	N	25666	21	222	11	0
	S	95	641	72	0	0

V	311	48	1591	4	0
F	174	0	7	55	0
Q	1	1	0	0	0

Tabel 5. Confusion Matrix Model LSTM(256)

		Classifier				
--	--	------------	--	--	--	--

Reference

	n	s	v	f	q		F	184	1	5	46	0
N	25710	29	175	6	0		Q	1	0	1	0	0
S	85	654	69	0	0							
V	345	64	1543	2	0							

Tabel 6. Hasil Evaluasi Menggunakan Teknik Oversampling

Oversampling method	Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
SMOTE	LSTM(256)	92,25	95,53	92,25	93,65
	Modified LSTM-FCN	88,34	95,01	88,34	91,01
ROS	LSTM(256)	92,93	95,62	92,93	92,93
	Modified LSTM-FCN	90,07	95,69	90,07	92,39
ADASYN	LSTM(256)	92,25	95,53	92,25	93,65
	Modified LSTM-FCN	87,47	94,33	87,47	90,05

Tabel 7. Confusion Matrix Model LSTM(256) ROS

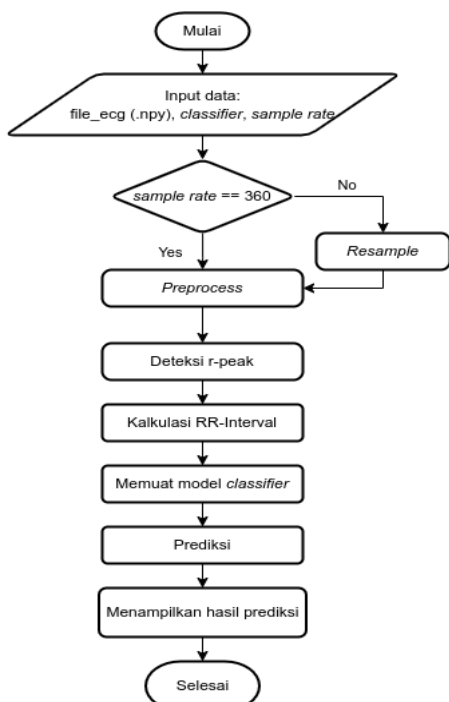
		Classifier					
		n	s	v	f	q	
Reference	N	24424	125	503	859	9	
	S	58	668	82	0	0	
	V	204	69	1642	39	0	
	F	68	1	26	140	1	
	Q	1	0	1	0	0	

Tabel 8. Komparasi dengan Penelitian Lain

Works	Classifier	Feature	Classes	Accuracy(%)
(Essa & Xie, 2021)	CNN-LSTM + RRHOS-LSTM	Morfologi, RR-Interval, HOS	4: N, S, V, F	95,81
(Mondéjar-Guerra dkk., 2019)	Ensemble SVM	RR-interval, HOS, Wavelet, Morfologi	4: N, S, V, F	94,5
(Pandey, Janghel & Vani, 2020)	Ensemble SVM	Wavelet, R-R interval, HOS, Morfologi	4: N, S, V, F	94,4
(Pramukantoro & Gofuku, 2022)	Decision Tree (ROS)	RR-Interval	5: N, S, V, F, Q	99,31
Penelitian kami	Modified LSTM-FCN	RR-Interval	5: N, S, V, F, Q	96,66

4.2. Implementasi Inferensi

Inferensi diimplementasikan ke dalam bentuk aplikasi web. Aplikasi dibangun menggunakan *framework* Flask. Diagram alir dari program implementasi dapat dilihat pada gambar 7.

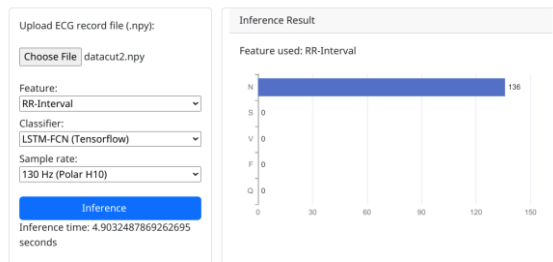


Gambar 7. Diagram Alir Program Inferensi

Program akan menerima input data ECG dengan format (.npz), pilihan *classifier*, serta sample rate dari data ECG yang digunakan. Data ECG akan dilakukan *resampling* untuk menyesuaikan *sample rate* dari data agar sesuai dengan *sample rate* yang digunakan pada tahap *training* yaitu 360hz. Data yang telah dilakukan *resampling* akan di-*preprocess* sama seperti pada tahap *preprocess* data *training*. Data yang telah di-*preprocess* akan dilakukan deteksi titik puncak R pada ECG. Algoritma Pan-Tompkins (Pan & Tompkins, 1985) digunakan untuk menentukan titik puncak R pada ECG. Setelah diketahui titik R dari ECG, dilakukan kalkulasi fitur-fitur RR-Interval seperti yang dilakukan pada tahap ekstraksi fitur. Setelah mendapatkan fitur, model *classifier* dimuat dan digunakan untuk melakukan prediksi terhadap fitur. Hasil prediksi kemudian ditampilkan pada halaman web berupa visualisasi diagram batang.

Tampilan antarmuka web ditunjukkan pada gambar 8. Pengguna perlu mengunggah data ECG dan memilih *classifier* yang akan digunakan serta *sample rate* dari data ECG yang dikirim. Setelah proses inferensi selesai, web akan menampilkan hasil berupa jumlah detak jantung yang terdeteksi pada setiap kelasnya.

ECG LSTM Inference



Gambar 8. Tampilan Antarmuka Web Inferensi

4.3. Hasil Pengujian

Hasil inferensi pada perangkat Raspberry Pi 3B ditunjukkan pada tabel 9. Inferensi dilakukan pada data ECG sepanjang 100 detik. Hasil pengujian mengenai performa model dapat dilihat lebih detail pada tabel 10. Seluruh model dapat berjalan dengan waktu kurang dari 17 detik dengan waktu tersingkat pada model LSTM-FCN modifikasi dengan waktu inferensi 4,82 detik. Waktu inferensi yang singkat tersebut menunjukkan bahwa model memiliki performa yang baik pada perangkat Raspberry Pi 3B dan bahkan memungkinkan deteksi secara *realtime* terutama pada model LSTM-FCN modifikasi.

Tabel 9. Hasil Inferensi 100 Detik ECG

Model	N	S	V	F	Q
LSTM (512)	136	0	0	0	0
LSTM (256)	136	0	0	0	0
Bi-LSTM	136	0	0	0	0
Modified LSTM-FCN	136	0	0	0	0

Tabel 10. Hasil Pengujian pada Raspberry Pi 3B

Model	Waktu Inferensi (detik)	Penggunaan Memori (MB)	Penggunaan CPU (%)
LSTM(512)	16,56	137,2	100
LSTM(256)	7,26	134,4	100
Bi-LSTM	9,47	134	100
Modified LSTM-FCN	4,82	134,8	100

Pada penggunaan memori, seluruh model dapat berjalan dengan puncak penggunaan memori kurang dari 138MB, dengan penggunaan memori terendah pada model Bi-LSTM yaitu 134MB. Penggunaan memori ini menunjukkan bahwa model mampu dijalankan dengan baik pada perangkat Raspberry Pi 3B yang memiliki memori 1GB.

Pada penggunaan CPU, seluruh model memiliki puncak penggunaan CPU yang sama yaitu 100%. Dari total 4 *core* CPU yang ada pada Raspberry Pi 3B, program hanya menggunakan 1 *core* CPU dengan penggunaan 100%. Hal ini terjadi karena program berjalan secara *single processing* dan tidak dioptimasi untuk dapat berjalan menggunakan beberapa *core* CPU.

5. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan dan menguji model prediksi detak jantung berbasis LSTM pada Raspberry Pi untuk pemantauan kesehatan portabel. Hasil yang dicapai menunjukkan akurasi yang tinggi, waktu respons yang cepat, dan efisiensi komputasi yang baik. Penggunaan perangkat portabel seperti Raspberry Pi untuk pemantauan kesehatan memiliki potensi untuk meningkatkan akses dan kesadaran kesehatan bagi masyarakat. Diharapkan hasil penelitian ini akan memberikan kontribusi positif dalam perkembangan teknologi kesehatan dan pemantauan kesehatan portabel di masa depan.

6. UCAPAN TERIMA KASIH

Kami berterima kasih atas dukungan finansial yang diberikan oleh LPPM Universitas Brawijaya untuk proyek penelitian ini dengan nomor kontrak 612.86/UN10.C20/2023. Pendanaan dari mereka sangat penting dalam memungkinkan kami untuk melakukan penelitian ini dan memajukan pemahaman kami tentang prediksi detak jantung otomatis. Kami juga mengucapkan terima kasih kepada para dosen, staf, dan sumber daya di FILKOM dan Artificial Intelligence Center, di Universitas Brawijaya yang telah memainkan peran penting dalam keberhasilan penyelesaian penelitian ini.

DAFTAR PUSTAKA

- AHSANUZZAMAN, S.M., AHMED, T. dan RAHMAN, MD.A., 2020. Low Cost, Portable ECG Monitoring and Alarming System Based on Deep Learning. In: *2020 IEEE Region 10 Symposium (TENSYPMP)*. 2020 IEEE Region 10 Symposium (TENSYPMP). pp.316–319. <https://doi.org/10.1109/TENSYPMP50017.2020.9231005>.
- ASSOCIATION FOR THE ADVANCEMENT OF MEDICAL INSTRUMENTATION, 1998. Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms. *ANSI/AAMI EC57*.
- CHAWLA, N.V., BOWYER, K.W., HALL, L.O. dan KEGELMEYER, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, pp.321–357.
- ESSA, E. dan XIE, X., 2021. An Ensemble of Deep Learning-Based Multi-Model for ECG Heartbeats Arrhythmia Classification. *IEEE Access*, 9, pp.103452–103464. <https://doi.org/10.1109/ACCESS.2021.3098986>.
- HE, H., BAI, Y., GARCIA, E.A. dan LI, S., 2008. ADASYN: Adaptive synthetic sampling

- approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. pp.1322–1328.
<https://doi.org/10.1109/IJCNN.2008.4633969>.
- HOCHREITER, S. dan SCHMIDHUBER, J., 1997. Long short-term memory. *Neural computation*, 9(8), pp.1735–1780.
- KARIM, F., MAJUMDAR, S., DARABI, H. dan CHEN, S., 2018. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access*, 6, pp.1662–1669.
<https://doi.org/10.1109/ACCESS.2017.2779939>.
- LENIS, G., PILIA, N., LOEWE, A., SCHULZE, W.H.W. dan DÖSSEL, O., 2017. Comparison of Baseline Wander Removal Techniques considering the Preservation of ST Changes in the Ischemic ECG: A Simulation Study. *Computational and Mathematical Methods in Medicine*, 2017, p.9295029.
<https://doi.org/10.1155/2017/9295029>.
- MOHAMMED, R., RAWASHDEH, J. dan ABDULLAH, M., 2020. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. In: *2020 11th International Conference on Information and Communication Systems (ICICS)*. pp.243–248.
<https://doi.org/10.1109/ICICS49469.2020.239556>.
- MONDÉJAR-GUERRA, V., NOVO, J., ROUCO, J., PENEDO, M.G. dan ORTEGA, M., 2019. Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers. *Biomedical Signal Processing and Control*, 47, pp.41–48.
<https://doi.org/10.1016/j.bspc.2018.08.007>.
- MOODY, G.B. dan MARK, R.G., 2001. The impact of the MIT-BIH arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3), pp.45–50.
- PAN, J. dan TOMPKINS, W.J., 1985. A real-time QRS detection algorithm. *IEEE transactions on bio-medical engineering*, 32(3), pp.230–236.
<https://doi.org/10.1109/TBME.1985.325532>.
- PANDEY, S.K., JANGHEL, R.R. dan VANI, V., 2020. Patient Specific Machine Learning Models for ECG Signal Classification. *Procedia Computer Science*, 167, pp.2181–2190.
<https://doi.org/10.1016/j.procs.2020.03.269>.
- PRAMUKANTORO, E.S. dan GOFUKU, A., 2022. A Heartbeat Classifier for Continuous Prediction Using a Wearable Device. *Sensors*, 22(14), p.5080.
<https://doi.org/10.3390/s22145080>.
- SHCHETININ, E. dan GLUSHKOVA, A., 2022. Arrhythmia detection using resampling and deep learning methods on unbalanced data. *Computer Optics*, 46(6), pp.980–987.
<https://doi.org/10.18287/2412-6179-CO-1112>.
- WORLD HEALTH ORGANIZATION, 2021. *Cardiovascular Diseases (CVDs)*. [online] Tersedia di: <<https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-cvds>> [Diakses 21 September 2023].
- YU, Y., SI, X., HU, C. dan ZHANG, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7), pp.1235–1270.