

IMPLEMENTASI SISTEM INFERENSI KLASIFIKASI DETAK JANTUNG BERBASIS STREAM PROCESSING PADA JETSON NANO

Muhammad Rifqi Ar - Raffly¹, Eko Sakti Pramukantoro², Kasyful Amron³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ muhammadrifqi15@student.ub.ac.id, ² eko.sakti@ub.ac.id, ³ kasyful@ub.ac.id

Abstrak

Penyakit kardiovaskular merupakan penyebab kematian tertinggi di dunia, mencapai 31% dari total kematian, dengan 17,9 juta korban jiwa pada tahun 2019. Deteksi dini menjadi krusial untuk mencegah penyakit ini. Teknologi seperti otomasi deteksi detak jantung dengan perangkat wearable dan machine learning dapat mendukung upaya tersebut. Namun, sistem inferensi sebelumnya menggunakan batch processing yang menyebabkan penundaan signifikan. Penelitian ini mengembangkan sistem inferensi klasifikasi detak jantung berbasis stream processing, memungkinkan data diproses secara langsung dan kontinu. Sistem ini menggunakan model LSTM 512, 256, dan FCN dengan fitur RR-Interval, morphology, dan wavelet. Dikembangkan sebagai aplikasi web menggunakan framework Flask, sistem ini menghubungkan sensor Polar H10 melalui Web Bluetooth API di JavaScript. Frontend web menangani koneksi sensor, pengiriman data ECG ke backend, dan visualisasi hasil inferensi dalam bentuk grafik. Backend melakukan inferensi pada perangkat Jetson Nano. Hasil pengujian menunjukkan model LSTM 256 wavelet unggul dalam waktu respons, LSTM 512 morphology terbaik dalam throughput, dan LSTM 256 dengan fitur RR-Interval atau wavelet paling efisien dalam penggunaan sumber daya.

Kata kunci: Penyakit kardiovaskular, deteksi dini, stream processing, model LSTM, Polar H10, Jetson Nano

Abstract

Cardiovascular diseases are the leading cause of death globally, accounting for 31% of all deaths, with 17.9 million fatalities in 2019. Early detection is crucial to prevent these diseases. Technologies such as automated heart rate detection with wearable devices and machine learning support this effort. However, previous inference systems utilized batch processing, causing significant delays. This study develops a heart rate classification inference system based on stream processing, enabling continuous and real-time data processing. The system employs LSTM 512, 256, and FCN models with RR-Interval, morphology, and wavelet features. Developed as a web application using the Flask framework, it connects to the Polar H10 sensor via the Web Bluetooth API in JavaScript. The web frontend handles sensor connections, ECG data transmission to the backend, and visualization of inference results through charts. The backend performs inference on the Jetson Nano edge device. Testing results indicate that the LSTM 256 wavelet model excels in response time, the LSTM 512 morphology model achieves the highest throughput, and the LSTM 256 model with RR-Interval or wavelet features is the most resource-efficient.

Keywords: Cardiovascular disease, early detection, Stream Processing, LSTM models, Polar H10, Jetson Nano

1. PENDAHULUAN

Penyakit jantung atau yang biasa disebut sebagai penyakit kardiovaskular merupakan salah satu penyakit dengan penyumbang kematian terbanyak yaitu sekitar 31% (WHO, 2024). Penyakit ini memakan hingga 17,9 juta jiwa setiap tahunnya (WHO, 2024). Deteksi dini menjadi penting untuk dapat melakukan pencegahan. Teknologi kesehatan seperti

otomasi deteksi detak jantung dengan perangkat wearable dan machine learning dapat dimanfaatkan untuk deteksi dini.

Salah satu perangkat wearable yang dapat digunakan untuk mendapatkan data ECG adalah Polar H10. Polar H10 merupakan wearable device yang dapat digunakan untuk menggantikan penggunaan holter monitor (Gilgen-Ammann, Schweizer and Wyss, 2019). Polar H10 dapat disambungkan dengan bluetooth. Namun perangkat ini saja tidak

cukup. Diperlukan teknologi yang dapat melakukan klasifikasi data ECG secara otomatis.

Penggunaan deep learning menjadi salah satu pendekatan untuk klasifikasi data ECG. Saat ini terdapat model deep learning yang telah dikembangkan oleh Zen et al.. Model tersebut menggunakan Long Short Term Memory. Model yang telah dikembangkan Zen et al.. memiliki akurasi hingga 96%. Model yang dikembangkan oleh Zen et al. juga memiliki waktu inferensi 12 ms, menunjukkan bahwa model ini dapat dimanfaatkan pada data ECG secara stream.

Inferensi data ECG secara stream processing menjadi sangat penting dalam konteks kebutuhan real-time. Stream processing memungkinkan data ECG yang diterima secara kontinyu untuk langsung dilakukan inferensi. Hal ini memberikan informasi secara langsung kepada tenaga kesehatan sehingga dapat mempercepat pengambilan keputusan dan penanganan. Dalam sistem stream processing, data ECG tidak perlu menunggu hingga selesai direkam sepenuhnya sebelum dapat dianalisis, sehingga sangat relevan untuk aplikasi kesehatan yang membutuhkan respons cepat.

Pada penelitian ini dilakukan implementasi sistem inferensi ECG berbasis stream processing yang terhubung dengan sensor Polar H10 melalui koneksi bluetooth. Data ECG yang dikirim sensor akan diterima oleh web frontend, dan dikirim ke backend untuk diinferensi. Hasil dari inferensi akan dikirim kembali untuk ditampilkan dalam bentuk chart.

Penelitian ini menggunakan perangkat Jetson Orin Nano karena memiliki performa yang unggul. Jetson Orin Nano dirancang khusus untuk mendukung beban kerja AI dan machine learning dengan GPU berbasis arsitektur NVIDIA Ampere, yang mampu mencapai performa komputasi hingga 40 TOPS (tera operations per second). Perangkat ini juga memiliki kapasitas memori yang lebih besar dan bandwidth yang lebih tinggi, sehingga mendukung eksekusi inferensi model deep learning secara efisien dan cepat. Dengan kemampuan tersebut, Jetson Orin Nano diharapkan dapat memenuhi kebutuhan sistem inferensi deteksi aritmia secara real-time dengan performa tinggi dan efisiensi daya yang optimal.

2. KAJIAN PUSTAKA

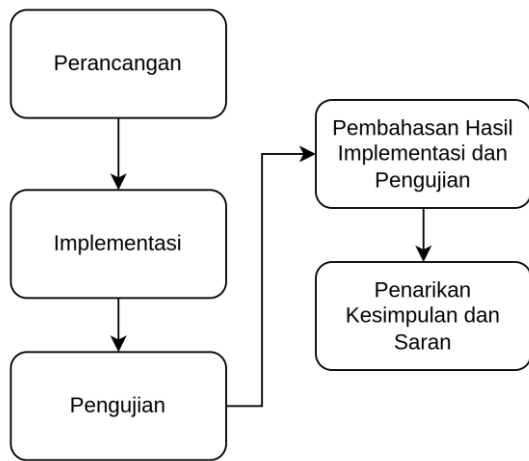
Terdapat beberapa penelitian sebelumnya yang dijadikan sebagai landasan pustaka. Penelitian pertama dilakukan oleh Zen et al.

dengan judul Prediksi Detak Jantung Berbasis LSTM pada Raspberry Pi. Masalah yang diangkat dari penelitian ini adalah masalah penyakit jantung yang menjadi penyakit dengan penyumbang kematian terbanyak secara global. Diperlukan teknologi pemantauan kesehatan portabel. Zen et al. pada penelitian ini mengusulkan model klasifikasi data ECG. Model yang diusulkan menggunakan Long Short Term memory. Model dikembangkan antara lain LSTM 512, LSTM 256, dan LSTM FCN. Model tersebut dikembangkan dengan dataset MiT BIH Arrhythmia. Model dilatih dengan hyperparameter tuning dengan 50 epoch. Model tersebut memiliki akurasi sekitar 96%. Waktu yang diperlukan model tersebut untuk melakukan inferensi adalah sekitar 12 ms, hal ini menjadikan model tersebut dapat digunakan pada sistem inferensi secara stream processing.

Penelitian lain hadir dengan judul ECG-based real-time arrhythmia monitoring using quantized deep neural networks. Penelitian ini dibawakan oleh Demelo Ribeiro e al.. Berangkat dari masalah yang sama, penelitian ini bertujuan untuk membantu tenaga kesehatan dalam melakukan interpretasi data ECG. Model yang digunakan pada penelitian ini menggunakan CNN satu dimensi yang terkuantisasi. Sistem yang dibangun akan menampilkan hasil dari inferensi dalam bentuk chart. Menampilkan hasil inferensi dalam bentuk chart dapat memudahkan tenaga kesehatan dalam melakukan interpretasi data ECG. Menampilkan hasil inferensi dalam bentuk chart menjadi inspirasi penulis dalam membuat sistem inferensi klasifikasi data ECG dengan stream processing.

3. METODE PENELITIAN

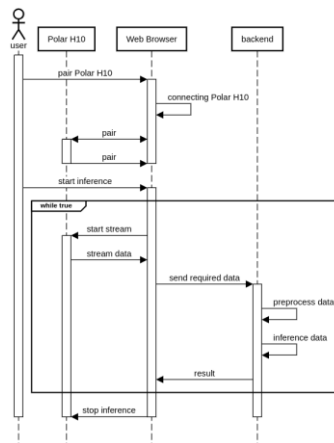
Penelitian ini akan melakukan implementasi sistem inferensi klasifikasi data ECG dengan stream processing. Tahapan yang dilakukan dalam penelitian ini dapat dilihat pada gambar 1



Gambar 1. Metode Penelitian

4. PERANCANGAN

Penulis melakukan implementasi sistem inferensi berbasis stream processing dengan model yang telah dibuat oleh Zen et al.. Sistem memiliki beberapa fitur. Fitur pertama adalah sistem dapat disambungkan dengan sensor Polar H10 melalui bluetooth. Nantinya data ECG akan diteruskan ke backend sesuai dengan fitur dan model yang dipilih oleh pengguna. Backend akan menerima dan melakukan inferensi. Fitur kedua adalah hasil dari inferensi akan ditampilkan dalam bentuk chart secara langsung.



Gambar 2. Sequence Diagram Umum Sistem



Gambar 3 Komponen Sistem Inferensi

Sistem yang dibangun oleh penulis memiliki 3 komponen utama. Komponen tersebut terdiri dari Sensor, Web-Frontend, dan Backend. Sensor disini merupakan komponen

sistem yang berfungsi untuk mengambil data ECG. Sensor juga akan dihubungkan ke browser Google Chrome dengan menggunakan bluetooth low energy. Sensor akan membaca aktivitas elektrik jantung. Data ECG akan dikirim melalui bluetooth dan akan diterima oleh Web-Frontend.

Web-Frontend merupakan komponen yang akan banyak berinteraksi dengan user. Web-Frontend nantinya akan digunakan sebagai penghubung sensor. Web-Frontend akan mendapatkan data ECG melalui Web Bluetooth API yang disediakan oleh browser Google Chrome. Web-Frontend juga akan meneruskan data ECG ke backend untuk dilakukan inferensi sesuai dengan fitur dan model yang dipilih oleh pengguna. Web-Frontend juga akan menampilkan hasil inferensi yang dikirim oleh backend. Hasil inferensi nantinya akan ditampilkan dalam bentuk chart. Chart yang digunakan untuk menampilkan hasil inferensi adalah chart yang ada pada sistem yang telah dikembangkan zen et al..

Backend merupakan komponen dari sistem yang digunakan dalam melakukan inferensi. Inferensi akan dilakukan dengan model yang telah dikembangkan oleh Zen et al.. Inferensi akan dilakukan dengan tensorflow GPU. Inferensi dilakukan berdasarkan fitur yang dipilih oleh pengguna. Fitur yang dapat dipilih oleh pengguna antara lain RR-Interval, Morphology, dan Wavelet. Ketiganya memiliki kebutuhan informasi data ECG yang berbeda-beda. Fitur rr-Interval menggunakan membutuhkan 44 rr-Interval. Sementara fitur morphology dan wavelet membutuhkan data ECG per window atau per PQRST.

5. IMPLEMENTASI



Gambar 4 Hasil Implementasi Sistem

Gambar diatas merupakan hasil implementasi dari perancangan sistem. Hasil dari implementasi diatas menunjukkan bahwa sistem dapat disambungkan dengan sensor Polar H10. Data yang diambil dari Polar H10 kemudian diolah untuk diambil informasi berupa RR-Interval atau 180 data ECG. Informasi tersebut dikirim ke backend untuk dilakukan inferensi. Hasil inferensi akan ditampilkan oleh web-frontend dalam bentuk grafik seperti pada gambar 4.

6. PENGUJIAN

Pengujian merupakan tahapan selanjutnya setelah dilakukan implementasi sistem. Pengujian ini dilakukan untuk melakukan analisa terhadap pengaruh penggunaan model. Pengujian yang dilakukan adalah pengujian kinerja dan penggunaan resource. Pengujian mengenai kinerja meliputi waktu respons dan throughput. Pengujian mengenai penggunaan resource meliputi penggunaan CPU, Memory, GPU dan waktu inferensi. Sementara pengujian kinerja meliputi throughput dan waktu respons. Skenario dari pengujian adalah pengguna memakai sensor Polar H10 sesuai dengan gambar 5. Pengujian akan dilakukan selama 10 menit. Semua model dari setiap fitur akan dilakukan pengujian. Pengujian akan dilakukan di jetson nano.



Gambar 5 Hasil Implementasi Sistem

7. PEMBAHASAN HASIL PENGUJIAN

Tabel 1. Penggunaan Resources saat pengujian

	waktu inferensi (s)	GPU (%)	CPU (%)	Memory (MB)	GPU Memory (MB)
AVG	0,2	1,08	42,38	268,5	201,9
MAX	5,91	98,9	84,32	330,3	353,8
MAX	0,11	0	16,7	217,6	97,6

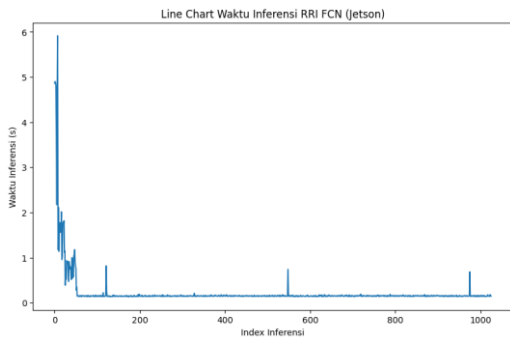
Tabel 1 diatas menunjukkan penggunaan resource dari seluruh model ketika menjalankan skenario pengujian. Waktu inferensi disaat melakukan skenario pengujian memiliki rata – rata 0.2 detik. Waktu inferensi terbesar yang didapatkan yaitu sekitar 5 detik pada model RRI FCN. Waktu inferensi paling kecil yang didapatkan adalah 0,11 detik pada model Wavelet 256. Penggunaan GPU untuk semua model yang diuji dari setiap fitur memiliki rata – rata 1,08 %. Untuk penggunaan GPU terbesar didapati pada model Morphology 512, untuk terkecilnya adalah 0%. Penggunaan CPU untuk semua model yang diuji memiliki rata 42,38%, dengan adanya spike penggunaan CPU terbesar pada fitur Wavelet model LSTM 256 pada 84.3% dan penggunaan CPU terkecil ada pada RRI 256 pada 16,7%. Seluruh model yang diuji memiliki rata – rata penggunaan memory sebesar 268,5 MB. Model dengan penggunaan memory terkecil ada pada LSTM FCN pada fitur RR – Interval yang sebesar 330,3 MB, sementara penggunaan memory terkecil didapati pada model LSTM FCN pada fitur Morphology dengan 217,6 MB. Penggunaan GPU memory pada seluruh model memiliki rata – rata 201,9 MB. Penggunaan GPU memory terbesar ada pada 353,8 MB pada model LSTM 512 untuk fitur RR – interval dan Morphology, sementara penggunaan GPU memory yang terkecil didapatkan pada model LSTM 256 untuk fitur RR – Interval dan Wavelet.

Tabel 2. Max GPU saat pengujian

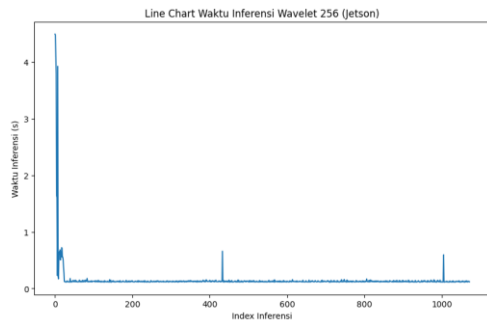
	Max GPU (%)
AVG	54.78
MAX	98.9
MAX	13

Penggunaan Maximum GPU dari semua model memiliki rata – rata 54,78%. Nilai maksimum dari semua maksimum GPU yang dimiliki setiap model adalah 98,9% dan nilai minimumnya ada pada 13%. Semua model mengalami spike yang banyak terjadi pada banyak titik. Namun diantara banyaknya spike, penggunaan GPU justru banyak yang 0. Hal ini dikarenakan spike tetap terjadi namun dalam waktu yang sangat singkat sehingga jetson stats tidak dapat mencatat spike tersebut.

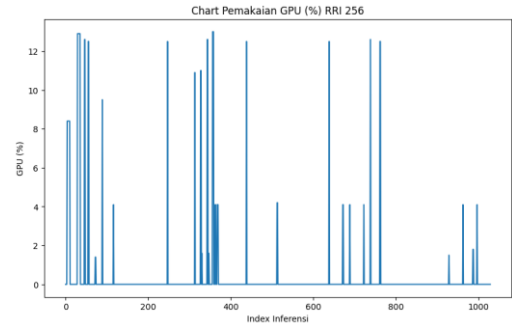
Penggunaan Maximum GPU dari semua model memiliki rata – rata 54,78%. Nilai maksimum dari semua maksimum GPU yang dimiliki setiap model adalah 98,9% dan nilai minimumnya ada pada 13%. Semua model mengalami spike yang banyak terjadi pada banyak titik. Namun diantara banyaknya spike, penggunaan GPU justru banyak yang 0. Hal ini dikarenakan spike tetap terjadi namun dalam waktu yang sangat singkat sehingga jetson stats tidak dapat mencatat spike tersebut.



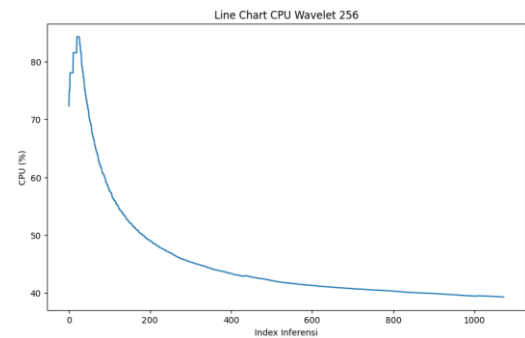
Gambar 6 Line Chart Waktu Inferensi untuk model LSTM FCN fitur RR – Interval



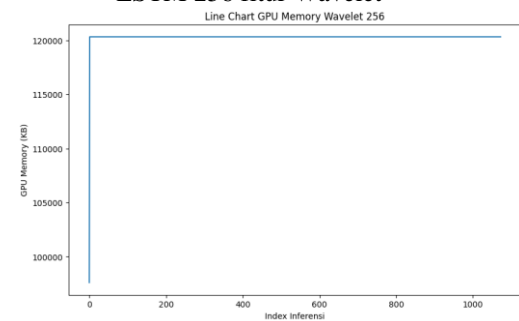
Gambar 7 Line Chart Waktu Inferensi Model LSTM 256 fitur Wavelet



Gambar 8 Chart Pemakaian GPU Model LSTM 512 Fitur Morphology



Gambar 9 Grafik Penggunaan CPU pada model LSTM 256 fitur Wavelet



Gambar 10 Penggunaan CPU pada model LSTM 256 fitur RR – interval

Bisa dilihat pada beberapa gambar di, bahwa terdapat kenaikan di awal inferensi untuk semua model pada semua fitur. Hal ini dikarenakan di awal inferensi terdapat load model namun paket telah dikirim. Hal ini menjadikan terdapat banyak sekali paket yang menumpuk di awal. Setelah model berhasil di load, maka paket yang menumpuk tadi akan dilakukan inferensi. Inferensi paket yang banyak waktu di awal ini meningkatkan penggunaan resource. Waktu inferensi juga akan mengalami lonjakan hingga 5 detik. Waktu inferensi yang meningkat juga akan mempengaruhi waktu respons.

Tabel 3. Hasil Pengujian Throughput

	Throughput	Inferensi Berhasil	Inferensi Gagal
AVG	1,76	1058.6	3.33
MAX	1,79	1077	5
MAX	1,71	1027	2

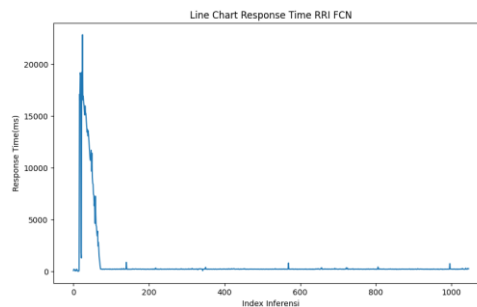
Throughput dihitung dengan menghitung request yang berhasil dibagi dengan waktu pengambilan data yaitu 600 detik. Rata - Rata throughput dari semua model adalah 1,76 request berhasil per detik. Throughput terbesar ada pada Model LSTM 512 pada fitur morphology dengan nilai throughput 1,795 request berhasil per detik. Nilai throughput terkecil ada pada model LSTM FCN pada fitur RR - Interval dengan nilai 1,71 request berhasil per detik. Hasil dari throughput menunjukkan bahwa diperlukan optimasi lebih lanjut dalam meminimalisir waktu respons. Penyebab dari error atau inferensi gagal adalah, division by zero atau ada nilai ECG yang isinya 0, TypeError karena data ECG yang diterima adalah NaN, yang terakhir adalah Jumlah RR – Interval yang terkadang lebih dari yang seharusnya. Pada saat awal inferensi, sistem akan melakukan load model terlebih dahulu, RR – Interval akan ditambahkan hingga menjadi 44 RR – Interval, kemudian paket akan tetap dikirim. Setelah dikirim, kode web frontend di set untuk menunggu respons terlebih dahulu baru melakukan pop indeks pertama dan semua indeks akan di shift ke kiri untuk menyisakan indeks terakhir agar dapat diisi dengan data RR – Interval terbaru. Namun respons masih belum datang karena menunggu load model, alhasil RR – Interval akan terus menambahkan indeks baru hingga mencapai 48 RR – Interval, tentunya akan menyebabkan error perhitungan pada backend.

Tabel 4. Hasil Pengujian Waktu Respon

	Rata - rata waktu respons (ms)	Waktu Respons (ms)
AVG	516,17	
MAX	752,6	22856,08
MAX	309,15	0

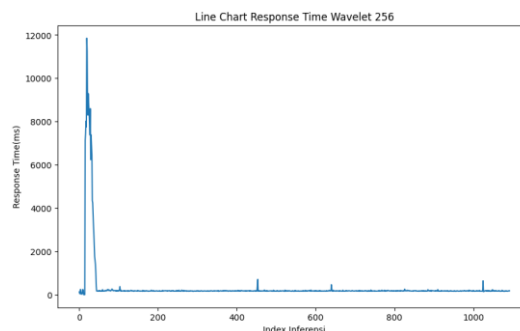
Pada tabel 4 diatas merupakan hasil dari pengujian kinerja dari sistem. Hal yang diuji seputar rata – rata waktu respons, throughput, waktu respons maksimum dan waktu respons minimum. Untuk rata – rata waktu respons semua model pada setiap fitur memiliki rata – rata 516 ms. Untuk rata – rata waktu respons

tertinggi ada pada model LSTM FCN untuk fitur RR – Interval yaitu pada 752,6 ms. Sementara rata – rata waktu respons paling rendah ada pada model LSTM 256 untuk fitur wavelet pada 309,1 ms. Waktu respons paling banyak didapati pada model LSTM FCN pada fitur RR – Interval dengan waktu 22,8 detik dan waktu respons paling rendah ada pada Model LSTM 256 untuk fitur Morphology dengan waktu yaitu pada 142,3 ms.



Gambar 11 Chart Waktu Respons Model LSTM FCN fitur RR – Interval

Gambar 11 menunjukkan waktu respons dengan menggunakan model LSTM FCN fitur RR – Interval. Pada awal inferensi terlihat waktu respons dari sistem melonjak hingga mencapai 22,8 detik. Informasi ini juga menunjukkan bahwa model LSTM FCN fitur RR – Interval membutuhkan waktu yang lama untuk melakukan load model. Setelah lonjakan, waktu inferensi menjadi stabil dibawah 1 detik. Namun perlu diketahui juga bahwa rata – rata waktu respons ketika menggunakan model ini cukup tinggi yaitu pada 752,7 ms.



Gambar 12 Chart Waktu Respons Model LSTM FCN fitur RR – Interval

Gambar 12 menunjukkan waktu respons yang didapat dari menjalankan model LSTM 256. Terdapat lonjakan di awal hingga menyentuh 12 detik, hal ini terjadi karena adanya load model di awal inferensi. Waktu respons cenderung stabil setelah terjadinya

penurunan waktu respons. Waktu respons setelah terjadinya penurunan beradJudul Menggunakan Font Times New Roman 14a di bawah 1 detik, hal ini menjadikan rata – rata waktu respons model LSTM 256 fitur wavelet yang terkecil yaitu pada 309,1 ms.

8. KESIMPULAN

Penelitian ini berhasil melakukan implementasi sistem inferensi klasifikasi detak jantung berbasis stream processing pada jetson nano. Berikut merupakan kesimpulan yang dapat diambil dari penelitian ini.

Sistem Inferensi Klasifikasi Detak Jantung Berbasis Stream Processing pada Jetson Nano dikembangkan sebagai layanan web yang dapat diakses menggunakan browser google chrome. Sistem dikembangkan dengan framework flask dan menggunakan model deep learning LSTM 512, 256 dan FCN yang telah dikembangkan Zen et al.. Fitur yang dapat digunakan pada inferensi diantaranya adalah RR – Interval, Morphology dan Wavelet. Sensor yang digunakan untuk mendapatkan data ECG adalah Polar H10 yang disambungkan dengan bluetooth. Sistem dapat melakukan inferensi secara stream processing atau kontinyu. Inferensi dengan model LSTM dilakukan pada perangkat tepi Jetson Nano.

Penggunaan resources yang dilakukan pada setiap model dalam waktu 10 menit, didapati bahwa waktu inferensi semua model memiliki rata – rata 0,2 detik. Model dengan waktu inferensi paling cepat adalah model LSTM 256 fitur Wavelet yaitu dengan rata – rata 0,15 detik, sementara waktu inferensi paling lama didapati pada model LSTM FCN fitur RR – Interval yaitu hingga 5,9 detik. Penggunaan GPU semua model yang diuji memiliki rata – rata 1,08% untuk semua model.

Model dengan penggunaan GPU paling besar ada pada Morphology 512 yaitu model LSTM 512 fitur Morphology yaitu maksimum pemakaian GPU 98,9%, Sementara model dengan penggunaan GPU paling kecil adalah model LSTM 256 fitur RR – Interval dengan maksimum penggunaan GPU ada pada 13 %.

Penggunaan CPU pada semua model yang diuji dengan 42,38%. Penggunaan CPU terbesar ada pada model LSTM 256 fitur Wavelet yaitu sekitar 84,3%, sementara penggunaan CPU terkecil ada pada model LSTM 256 fitur RR – Interval yaitu pada 16,8%.

Penggunaan memory pada semua model memiliki rata – rata 268,5 MB. Penggunaan

Memory terbesar didapati pada model LSTM FCN pada fitur RR – Interval yaitu pada 330,3 MB. Penggunaan memory terkecil ada pada model LSTM FCN fitur morphology yaitu pada 217,6 MB.

Penggunaan GPU memory memiliki rata – rata 201,9 MB untuk semua model yang diuji. Penggunaan GPU memory paling banyak didapati pada model LSTM 512 untuk fitur RR – Interval dan fitur Morphology. Penggunaan GPU memory terkecil didapati pada model LSTM 256 untuk fitur RR – Interval dan fitur Wavelet yaitu 97,6 MB.

Hasil Pengujian kinerja didapati throughput didapati bahwa rata – rata throughput yang didapat untuk semua model yang diuji adalah 1,76 request berhasil per detik. Throughput terbesar didapatkan oleh model LSTM 512 fitur Morphology dengan 1,795 request berhasil per detik. Throughput terkecil didapatkan pada model LSTM FCN fitur RR – interval dengan 1,711 request berhasil per detik. Jumlah Inferensi berhasil dari semua model memiliki rata – rata 1058,6 inferensi yang berhasil. Model dengan inferensi berhasil terbanyak ada pada model LSTM 512 fitur Morphology dengan jumlah inferensi yang berhasil sebanyak 1077. Model dengan jumlah inferensi berhasil paling sedikit adalah model LSTM FCN fitur RR – Interval yaitu dengan 1027 inferensi yang berhasil. Rata – rata inferensi gagal adalah 3,33 dari semua model. Inferensi yang gagal paling banyak ada pada model LSTM FCN fitur RR – Interval yaitu 5 inferensi yang gagal.

Berdasarkan informasi dari berbagai hasil dan metriks pengujian, didapatkan beberapa informasi yang dapat diambil. Jika sistem membutuhkan waktu response yang sangat singkat, maka model LSTM 256 fitur wavelet dapat menjadi pilihan. Jika membutuhkan throughput yang besar maka dapat menggunakan model LSTM 512 fitur morphology. Jika membutuhkan penggunaan resources yang efisien dapat memilih antara model LSTM 256 fitur RR – Interval atau wavelet.

9. DAFTAR PUSTAKA

- WHO. 2024. Cardiovascular diseases. [online] Tersedia di: <<https://www.who.int/health-topics/cardiovascular-diseases>> [Diakses 16 December 2024].
- Pramukantoro, E.S. and Gofuku, A., 2022. A Heartbeat Classifier for Continuous

- Prediction Using a Wearable Device. *Sensors*, 22(14), p.5080. <https://doi.org/10.3390/s22145080>. Lilia na, D. Y. & Mahmudy, W. F. 2006. Penerapan Algoritma Genetika pada Otomatisasi Penjadwalan Kuliah. *Laporan Penelitian DPP/SPP*. FMIPA Universitas Brawijaya, Malang.
- Zen, A., Pramukantoro, E., Amron, K., Wardhani, V. and Kamila, P., 2023. Prediksi Detak Jantung Berbasis LSTM pada Raspberry Pi untuk Pemantauan Kesehatan Portabel. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 10, pp.1555–1562. <https://doi.org/10.25126/jtiik.1078015>.
- Phanden, R. K., Jain, A. & Verma, R. 2013. An approach for integration of process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 26(4), 284-302.
- Gilgen-Ammann, R., Schweizer, T. and Wyss, T., 2019. RR interval signal quality of a heart rate monitor and an ECG Holter at rest and during exercise. *European Journal of Applied Physiology*, 119(7), pp.1525–1532. <https://doi.org/10.1007/s00421-019-04142-5>.
- Tala, F. Z. 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *Ph.D. Thesis*. Universiteit van Amsterdam.
- Polar, 2024. H10 Panduan pengguna | Sensor Denyut Jantung Polar H10. [online] Available at: <https://support.polar.com/e_manuals/h10-heart-rate-sensor/polar-h10-user-manual-bahasa-indonesia/introduction.htm> [Accessed 17 December 2024].
- Heuer, A.J., 2017. Wilkins' Clinical Assessment in Respiratory Care - E-Book: Wilkins' Clinical Assessment in Respiratory Care - E-Book. Elsevier Health Sciences.
- Kannathal, N., Acharya, U.R., Ng, E.Y.K., Krishnan, S.M., Min, L.C. and Laxminarayan, S., 2006. Cardiac health diagnosis using data fusion of cardiovascular and haemodynamic signals. *Computer Methods and Programs in Biomedicine*, 82(2), pp.87–96. <https://doi.org/10.1016/j.cmpb.2006.01.009>.
- Ilbeigipour, S., Albadvi, A. and Akhondzadeh Noughabi, E., 2021. Real-Time Heart Arrhythmia Detection Using Apache Spark Structured Streaming. *Journal of Healthcare Engineering*, 2021(1), p.6624829. <https://doi.org/10.1155/2021/6624829>.
- NVIDIA, N., 2024. NVIDIA Jetson Nano. [online] NVIDIA. Available at: <<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/>> [Accessed 17 December 2024].