

Komputasi Paralel untuk Mendeteksi Gelombang QRS Complex Menggunakan YOLO Deep Learning

Muhammad Achsan Hujjatul Islam, Muhtadin

Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember Surabaya, Indonesia

Email: achsan16@mhs.its.ac.id, muhtadin@ee.its.ac.id

Abstrak

Penyakit jantung merupakan salah satu penyebab kematian terbesar di dunia. Berdasarkan Riset Kesehatan Dasar (Riskesdas) Kementerian Kesehatan, pada tahun 2018 prevalansi penyakit jantung di Indonesia mencapai 1.5% dari jumlah seluruh penduduk. Salah satu penyebab munculnya penyakit jantung dapat diketahui melalui kondisi ritme jantung. *Aritmia* merupakan suatu kelainan ritme jantung yang tidak beraturan, terlalu cepat atau terlalu lambat. Kelainan *aritmia* dapat dideteksi menggunakan Elektrokardiogram (EKG). Deteksi *aritmia* dengan menggunakan teknologi pada bidang *machine learning* sangat diperlukan, sehingga dapat mendeteksi dan dilakukan terapi/pengobatan *aritmia* sedini mungkin untuk mengurangi resiko. Saat ini telah berkembang cara pembacaan dalam jangka panjang dikenal *Long Term EKG*. Pembacaan ini menghasilkan banyak data, yang tentunya membutuhkan waktu yang lama dalam mengolahnya. Berdasarkan hal tersebut, maka dimanfaatkannya teknologi komputasi paralel yaitu pemrosesan komputasi secara bersamaan dengan memaksimalkan sumber daya perangkat yang dimiliki sehingga dapat mempersingkat waktu yang dibutuhkan.

Kata kunci: Aritmia, EKG, Komputasi Paralel

Diterima Redaksi: 6-Nov-2024 Selesai Revisi: 15-Nov-2024 Diterbitkan Online: 15-Des-2024
DOI: <https://doi.org/10.59378/jcenim.v2i3.62>

I. Pendahuluan

Jantung merupakan salah satu organ manusia yang berfungsi dalam sistem peredaran darah dengan membawa unsur-unsur penting seperti oksigen, nutrisi, dll. Jika jantung mengalami kelainan, maka organ tubuh lainpun akan ikut terdampak dan mengalami gangguan juga yang pada akhirnya dapat mengakibatkan kematian bagi seseorang. Penyakit jantung merupakan salah satu penyebab kematian terbesar di dunia, menurut badan KemKes Indonesia salah satunya adalah *aritmia*.

Aritmia merupakan suatu kelainan dimana irama detak jantung tidak beraturan, terlalu cepat, ataupun terlalu lambat. Aktivitas jantung dapat direkam dengan menggunakan alat yang disebut Elektrokardiogram (EKG) dan akan memberikan keluaran sinyal EKG yaitu data sinyal rekaman aktivitas kecil listrik jantung berupa kurva tegangan fungsi waktu yang terdiri dari berbagai puncak. Pada sinyal EKG, dalam setiap satu detakan jantung terdiri atas 1 gelombang P, 1 kompleks QRS dengan R sebagai titik puncak, dan 1 gelombang T. Data gelombang-gelombang tersebut digunakan sebagai parameter dalam mendeteksi kondisi jantung tergolong *aritmia* atau tidak, serta untuk mengklasifikasikan jenis-jenis *aritmia*.

Berkembangnya teknologi saat ini, karakteristik QRS pada data EKG dapat dianalisis secara otomatis dengan menggunakan algoritma komputer. Teknologi komputasi paralel dapat memaksimalkan algoritma/sistem tersebut sehingga waktu yang dibutuhkan menjadi lebih singkat. Sumber daya perangkat yang menunjang yang dibangun dengan permodelan arsitektur komputasi *parallel* dapat meningkatkan efisiensi waktu pada proses komputasi data EKG karena proses bisa dikerjakan secara bersamaan dalam satu waktu.

Komputasi paralel adalah metode komputasi untuk menyelesaikan permasalahan komputasi dengan membagi beban komputasi ke dalam beberapa bagian kecil sub proses komputasi. Sub komputasi tersebut dijalankan bersamaan pada prosesor yang berbeda secara independen dan saling berinteraksi satu sama lain [1]. Komputasi paralel menggunakan 2 atau lebih *processor* untuk meningkatkan performa komputasi dalam mengolah data besar/banyak [2].

II. Penelitian Terkait

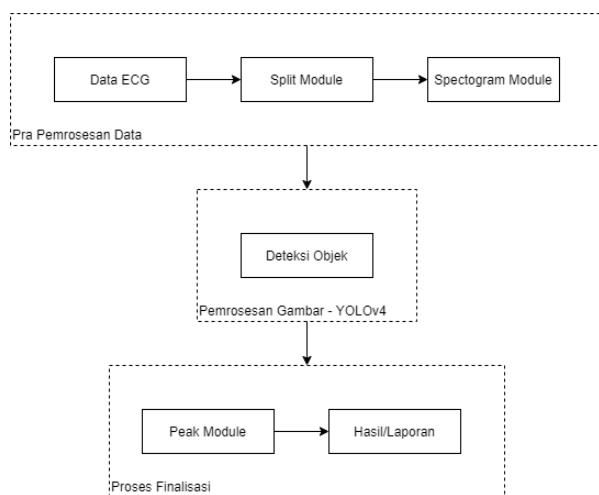
Beberapa penelitian sudah dilakukan dalam pengolahan sinyal EKG dan juga implementasi komputasi paralel di dalamnya. Penelitian pertama berjudul “Detection and Analysis of ECG Based on parallel Computing Technology” oleh Shen Qi dan Wang Yishan. Penelitian ini menjelaskan model paralel untuk deteksi dan analisis elektrokardiogram (EKG) dengan menggunakan arsitektur *Message Passing Interface* (MPI) yaitu dari Cluster IBM dengan 15 node. Dalam memodelkan algoritma paralel, penulis menganalisa algoritma serial dari deteksi dan analisis EKG yang selanjutnya setiap algoritma serial tersebut diperlakukan secara paralel untuk digunakan pada arsitektur paralel menggunakan MPI [3]. Algoritma yang digunakan terdiri dari 5 modul yaitu *Input Module*, *Filtering Module*, *Detecting Module*, *Analysis Module* dan *Output Module*. Pada tiga bagian dari lima module dapat dikerjakan oleh sub node secara *parallel*.

Penelitian kedua dilakukan oleh Poorna Chandra Suraj B. N dkk berjudul “Paralel Processing Architecture for ECG Signal Analysis”, penelitian ini merancang salah satu platform yang fleksibel untuk menganalisa EKG yang fleksibel, platform ini dibangun untuk kebutuhan komputasi yang lebih cepat, kebutuhan sistem secara *real-time*, dan persyaratan lainnya. Platform ini dirancang menggunakan arsitektur *pipeline* dengan adanya prosesor tambahan yang bertugas melakukan compute the thresholds, selection of the thresholds, tuning parameters, decision algorithms dll [4].

Penelitian terakhir berjudul “Automatic Detection of Arrhythmias Using a YOLO-Based Network with Long-Duration ECG Signals” oleh Won Hee Hwang dkk. Penelitian ini berkaitan dengan salah satu metode *machine learning* dalam mendeteksi objek yaitu YOLO, hasil model klasifikasi *aritmia* berbasis YOLO dapat mendeteksi masing-masing detak jantung dan mengklasifikasikan lima jenis *aritmia* dalam segmen EKG 10-detik tanpa langkah *beat extraction* dengan menggabungkan langkah *beat extraction*, *feature extraction*, dan klasifikasi menjadi satu module. Penulis mengusulkan model YOLO 1D yang dapat mendeteksi *aritmia*, model tersebut dapat mendeteksi lebih dari satu jenis *aritmia* dalam urutan yang panjang. Model YOLO dimodifikasi menjadi model YOLO 1D dan mengganti kotak pembatas dengan jendela pembatas dan pada akhirnya model hanya memprediksi satu koordinat untuk membuat jendela pembatas [5].

III. Desain dan Implementasi

Penelitian ini bertujuan untuk mendapatkan titik puncak-R pada data rekaman jantung dari seorang pasien yang dilaporkan berupa laporan aktivitas jantung dengan memberikan waktu-waktu kejadian titik puncak-R tersebut. Sistem ini dimulai dari input berupa data ECG, sebagai percobaan menggunakan data dari MIT-BIH *Arrhythmia Database*. Kemudian dari data tersebut dilakukan preprocessing data meliputi split data, mengubah data menjadi bentuk *spectrogram*, selanjutnya dataset di-training untuk didapatkan weight filenya. Setelah mendapatkan hasil model training -weight file-, model tersebut digunakan untuk mendeteksi objek PQRST sinyal ECG. Dan pada proses akhir menjalankan proses pencarian puncak R pada grafik PQRST dan memberikan hasil laporan waktu kejadian. Alur kerja dari sistem ini dijelaskan melalui blok diagram pada Gambar 1.



Gambar 1: Blok Diagram Utama Sistem

Terdapat 3 proses utama yang membangun tujuan dari penelitian ini, yaitu preprocessing data menge-nai pengelompokan kondisi mata, kemudian deteksi objek dengan menggunakan YOLOv4, tahap terakhir adalah pencarian nilai puncak dari hasil prediksi deteksi objek dan menyusunnya dalam bentuk laporan waktu.

A. Pra-pemrosesan Data

Pada tahapan ini terbagi lagi menjadi 2 sub proses di dalamnya yaitu Split Module yang bertujuan untuk memecah data aktivitas jantung yaitu data EKG menjadi data-data kecil dan Spectogram Module yang melakukan transformasi data teks menjadi bentuk data gambar.

1. Split Module

Selama proses Split Module ini, data yang digunakan tidak dirubah baik menambahkan/mengurangi nilai pada data rekaman aktivitas jantung yang ada, namun hanya membagi-bagi datanya menjadi data yang lebih kecil dan menyimpannya ke dalam file baru. Pemecahan data dilakukan berdasarkan waktu dengan rentang 1 detik sehingga setiap data nya membawa 360 data aktivitas jantung.

Jenis pemotongan data selama 1 detik ini terbagi menjadi 2 yaitu, jenis pertama adalah data hasil pemotongan dari pemotongan murni detik datum waktu dan jenis kedua diperoleh dari penggabungan setengah bagian terakhir data detik sebelumnya ditambahkan dengan setengah bagian pertama dari data detik setelahnya sehingga didapatkan data dengan panjang 1 detik pula.

2. Spectrogram Module

Hasil rekaman aktivitas jantung yang dihasilkan tidak sepenuhnya bagus, terdapat noise pada sinyal EKG sehingga diperlukan perlakuan khusus dalam melakukan pengolahan sinyal EKG tersebut. Penelitian ini menggunakan pendekatan STFT (*Short-Time Fourier Transform*) yang masih merupakan pengembangan dari FFT (*Fast Fourier Transform*) yaitu salah satu algoritma yang digunakan untuk pengolahan sinyal. Pendekatan melalui STFT ini diharapkan bisa mengoptimalkan nilai akurasi yang dihasilkan oleh sistem namun ternyata juga akan memberikan efek samping yaitu kebutuhan konsumsi waktu yang besar.

Metode yang terjadi pada STFT adalah membagi sinyal waktu yang lebih lama menjadi segmen yang lebih pendek dengan panjang yang sama dan kemudian menghitung transformasi Fourier secara terpisah pada setiap segmen yang lebih pendek [6]. Hal ini sudah diterapkan pada penelitian ini yaitu dengan memecah sinyal EKG dalam bentuk satuan detik yang telah dihasilkan oleh proses Split Module dan selanjutnya spektrum tersebut diplot ke dalam bentuk *spektrogram* yang merupakan representasi visual dari spektrum frekuensi sinyal terhadap waktu yang direpresentasikan dalam bentuk berupa gambar grafik.

B. Deteksi Objek

Proses deteksi yang digunakan dalam penelitian ini menggunakan algoritma YOLO dan terbagi menjadi 2 bagian proses yaitu Training YOLOv4 yang bertujuan untuk mendapatkan file *weight*/model *Machine Learning* dan Testing YOLOv4 yang digunakan untuk menghasilkan hasil prediksi deteksi objeknya.

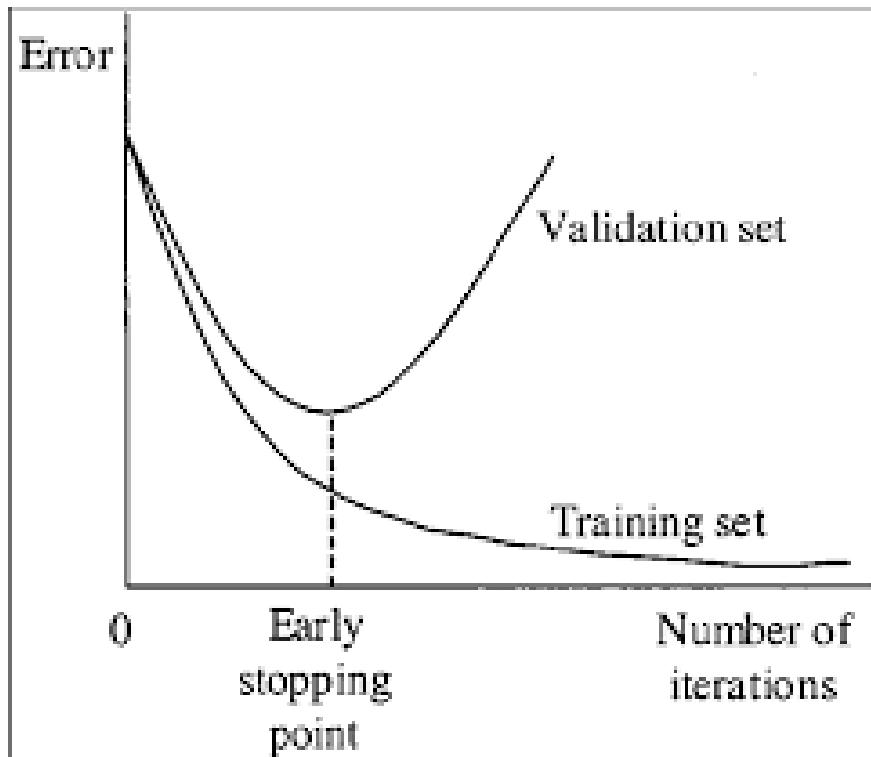
YOLO mendekripsi objek dengan menggunakan *unified model* yaitu single convolutional network yang memprediksi beberapa kotak pembatas (*bounding boxes*) serta probabilitas dari kotak-kotak tersebut secara bersamaan. Pertama-tama, YOLO membagi citra input ke dalam grid S x S. Jika pusat dari sebuah objek jatuh di dalam salah satu sel grid, maka sel grid itu bertanggung jawab untuk mendekripsi objek tersebut. Setiap sel grid memprediksi kotak pembatas dan *confidence score* dari tiap kotak pembatas tersebut [7].

1. Training YOLOv4

Penelitian ini membagi dataset menjadi 2 bagian yaitu Training Set dengan komposisi sebesar 80% dan Testing Set sebesar 20%. Selanjutnya dataset tersebut akan melalui tahapan labelling data yaitu pemberian label objek peak pada *spektrogram*. Setiap *spektrogram* pada dataset akan ditandai dengan kotak pembatas yang dinamakan *bounding box* untuk dijadikan sebagai pembelajaran model ketika proses training YOLOv4 dan akan menghasilkan keluaran model training berupa *weight-file*.

Sesuai dengan dokumentasi yang dituliskan oleh Alexey AB, jumlah iterasi selama training adalah kelipatan 2000 sesuai dengan jumlah kelas yang digunakan namun tidak boleh kurang dari jumlah dataset training dan minimal sebanyak 6000 iterasi [8]. Sistem dalam tugas akhir ini kelas yang ingin dicari hanyalah 1 kelas yaitu objek peak sehingga konfigurasi untuk jumlah iterasi pada training YOLO sebesar 6000 iterasi.

Dalam dokumentasi juga dijelaskan bahwa adanya proses pemberhentian secara paksa. Pemberhentian iterasi dilakukan untuk mencegah *overfitting* terhadap model yang telah dijelaskan pada dokumentasi dan Gambar 2 adalah grafik ilustrasi dari perlakuan ini.



Gambar 2: Grafik Early Stop [8]

Tahapan training tidak harus dilakukan setiap kali menjalankan sistem, namun sebagai gantinya harus mempunyai file *weight* yang diperoleh dari training sebelumnya. Setelah file *weight* didapatkan maka setiap kali deteksi objek bisa langsung melakukan tahapan Testing YOLOv4 dengan file *weight* tersebut. File *weight* juga perlu di-update sesuai kebutuhan dengan melihat nilai evaluasi dari model *Machine Learning* atau dalam hal ini adalah file *weight* yang dihasilkan tersebut.

2. Testing YOLOv4

Selanjutnya adalah tahapan testing dengan menggunakan model pembelajaran dari hasil keluaran pada tahapan training yaitu berupa *weight file*. File ini digunakan sebagai acuan saat melakukan deteksi objek dan juga menjadi faktor penentu dalam bagus tidaknya model dalam melakukan deteksi objek.

C. Laporan Data

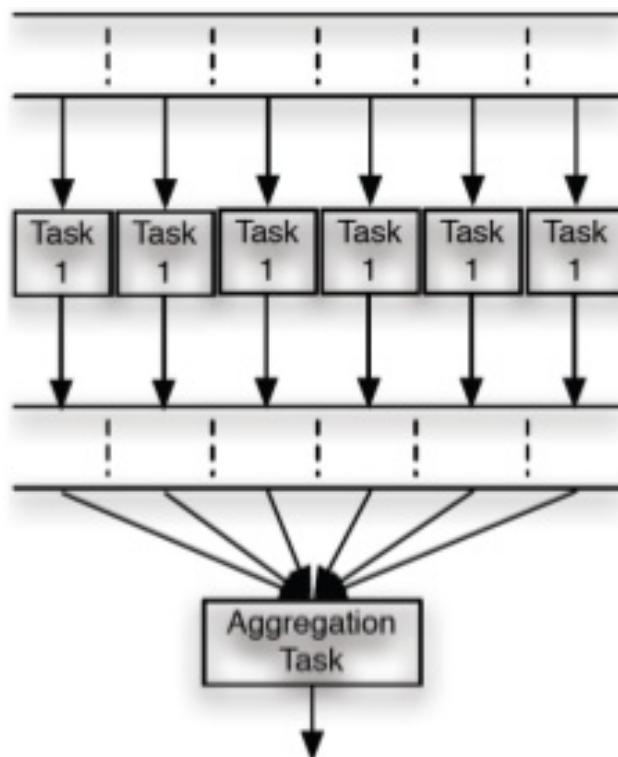
Laporan data ini merupakan hasil akhir dari sistem yang berisikan waktu-waktu yang menjadi titik puncak dari aktivitas jantung. Pada bagian pelaporan terdapat Peak Module yang bertugas menentukan nilai peak pada titik samplenya dan hasilnya dibentuk ke dalam laporan berupa file sesuai dengan data EKG nya.

File keluaran dari tahapan deteksi objek menggunakan YOLOv4 tidak hanya berisikan nilai deteksi objek saja melainkan juga memberikan keterangan tambahan lainnya termasuk yang tidak diperlukan, sehingga diperlukan filter untuk mendapatkan data yang diinginkan.

D. Model Paralel

Komputasi paralel adalah metode komputasi untuk menyelesaikan permasalahan komputasi dengan membagi beban komputasi ke dalam beberapa bagian kecil sub proses komputasi. Sub komputasi tersebut dijalankan bersamaan pada prosesor yang berbeda secara independen dan saling berinteraksi satu sama lain [1]. Komputasi paralel menggunakan 2 atau lebih *processor* untuk meningkatkan performa komputasi dalam mengolah data besar/banyak [2].

Komputasi paralel digunakan untuk mempercepat waktu eksekusi program. Semua algoritma pada rangkaian sistem ini bisa dijalankan bersamaan secara paralel dengan model paralel yang diterapkan menggunakan sistem *Pool*. Dalam menjalankan paralelisasi, *Pool* terbagi menjadi 2 bagian utama yaitu *Pool Manager* yang bertugas sebagai pengontrol data dengan menyalurkan data untuk diserahkan kepada *Pool Worker* dan *Pool Worker* yang bertugas sebagai pengeksekusi program dengan data yang telah diberikan oleh *Pool Manager*.



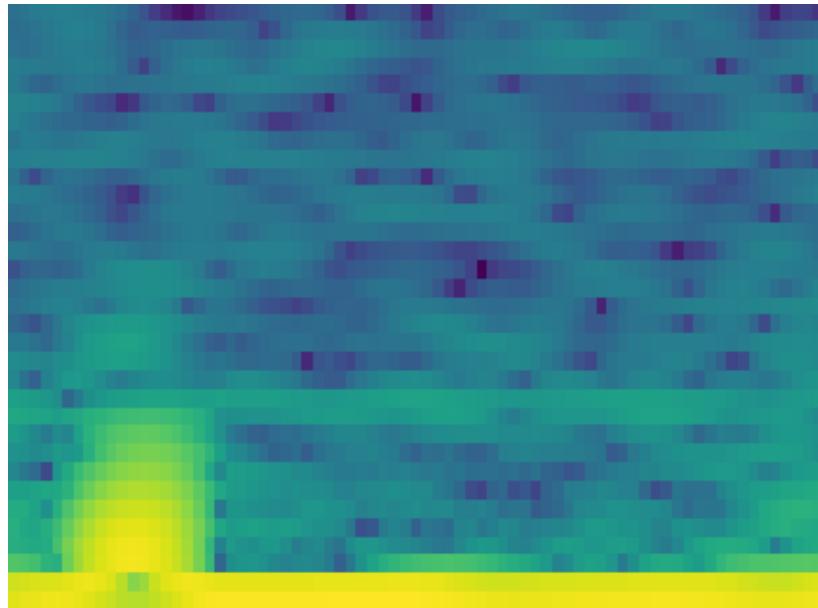
Gambar 3: Ilustrasi Data Parallelism [9]

Sebelum menjalankan proses secara paralel menggunakan *Pool* diperlukan pembagian data dan dikelompokkan ke dalam sebuah wadah yang dinamakan *task queue*. *Task Queue* yang didapatkan oleh *Pool* merupakan daftar data yang akan diproses yaitu banyaknya data masukan di setiap proses eksekusinya dan setiap prosesornya akan menjalankan tugas eksekusi program yang sama, sehingga bisa dikatakan bahwa metode paralel yang diterapkan dalam penelitian ini menggunakan *data parallelism*. Gambar 3 merupakan ilustrasi dari metode *data parallelism* dalam penerapan paralel model.

IV. Pengujian dan Analisis

A. Pra-pemrosesan Data

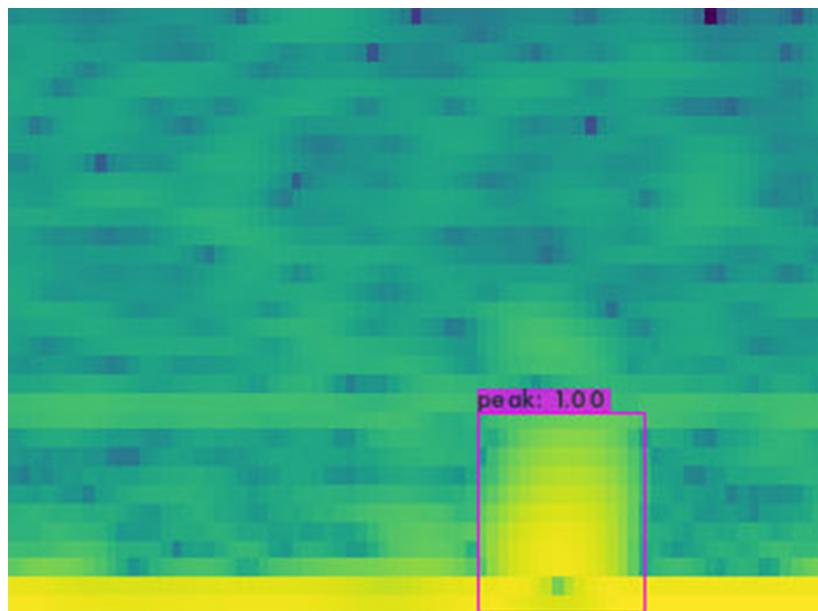
Data masukan akan melalui tahapan split module dan menghasilkan file-file baru yang membawa data rekaman selama 1 detik. Selanjutnya setiap data teks hasil split module akan diubah ke bentuk data gambar berupa *spektrogram* seperti pada Gambar 4.



Gambar 4: Bentuk Spektrogram dari data ECG-100 detik ke 0 sampai detik ke 1

B. Deteksi Objek

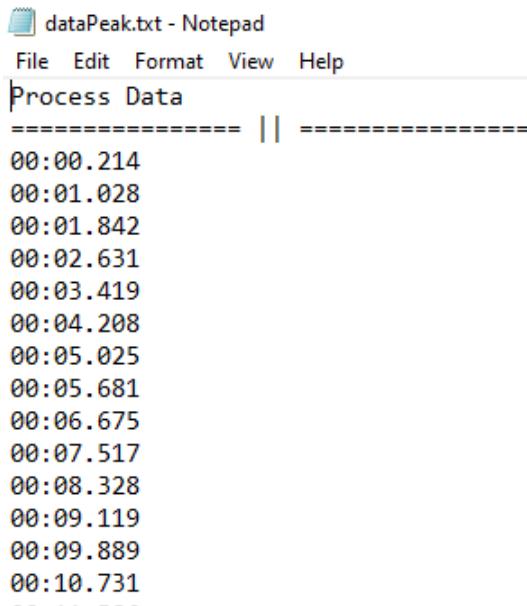
Weight-file hasil training YOLOv4 akan dimuat dan digunakan dalam melakukan deteksi objek. Algoritma YOLO akan mendeteksi objek dengan memberikan *bounding box* dan menuliskan *confidence score* dari prediksi *bounding box* tersebut. Pada Gambar 5 ditunjukkan bahwa YOLO dapat mendeteksi objek-peak dengan nilai keyakinan sebesar 100%.



Gambar 5: Contoh hasil deteksi objek-peak menggunakan YOLOv4

C. Hasil Laporan

Tahapan akhir pada sistem ini yaitu melakukan pelaporan aktivitas jantung yang berisikan waktu kejadian saat mencapai peak-R pada sinyal EKG seperti yang ditunjukkan pada Gambar 6.



```

dataPeak.txt - Notepad
File Edit Format View Help
Process Data
=====
00:00.214
00:01.028
00:01.842
00:02.631
00:03.419
00:04.208
00:05.025
00:05.681
00:06.675
00:07.517
00:08.328
00:09.119
00:09.889
00:10.731
=====
    
```

Gambar 6: Hasil laporan waktu setelah melakukan proses Peak Module

D. Implementasi Paralel

Pengujian dilakukan beberapa kali dengan data masukan file ecg yang bervariasi baik dari sisi kuantitas data rekaman yang ada pada setiap filenya maupun dari jumlah file yang dijalankan dan tentunya setiap pengujian tersebut dilakukan secara sekuensial maupun paralel untuk mengukur implementasi paralel yang diterapkan pada sistem dan diperoleh waktu eksekusi oleh program yang ditunjukkan pada Tabel 1.

Tabel 1: Hasil perolehan waktu eksekusi selama pengujian dengan menggunakan file ECG 30 menit

Sequence Paralel	Split	Spectogram	YOLO	Peak	Total
Sekuensial	00m 03d	04m 57d	30m 43d	00m 03d	35m 46d
2 Core	00m 03d	03m 38d	29m 06d	00m 03d	32m 50d
3 Core	00m 03d	02m 08d	30m 02d	00m 01d	32m 14d
4 Core	00m 03d	01m 53d	30m 26d	00m 02d	32m 24d

E. Evaluasi Paralel

1. Hasil perbandingan sekuensial dengan paralel

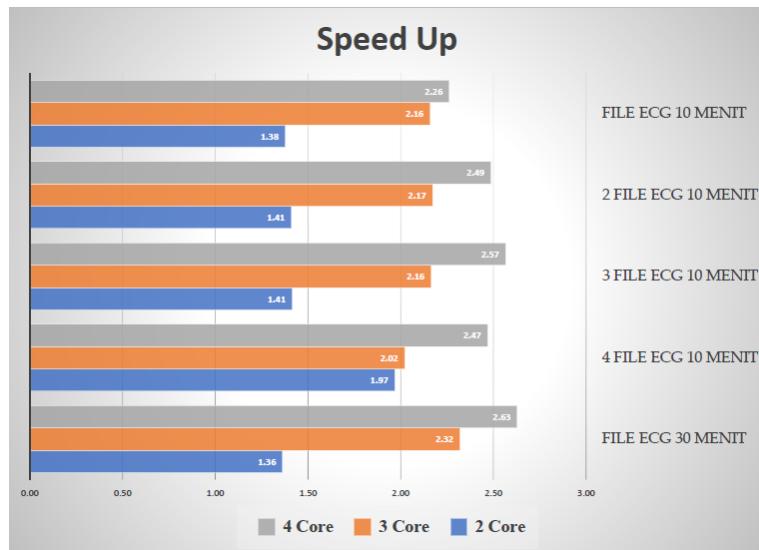
Implementasi paralel dievaluasi dan salah satunya menghasilkan nilai perbandingan eksekusi secara sekuensial dengan paralel menggunakan 4 prosesor yang ditunjukkan pada Tabel 2. Pada tabel tersebut diketahui bahwa nilai speed up paling besar terjadi pada pengujian ke-5 yaitu bernilai 2,63 dengan nilai efisiensi sebesar 0,66.

Tabel 2: Evaluasi paralel model pada paralel yang menggunakan 4 inti prosesor

Node	Sekuensial	4 Core	Sp	Ep
File 10 Menit	95	42	2,26	0,57
2 File 10 Menit	189	76	2,49	0,62
3 File 10 Menit	290	113	2,57	0,64
4 File 10 Menit	368	149	2,47	0,62
File 30 Menit	297	113	2,63	0,66

2. Grafik Speed Up

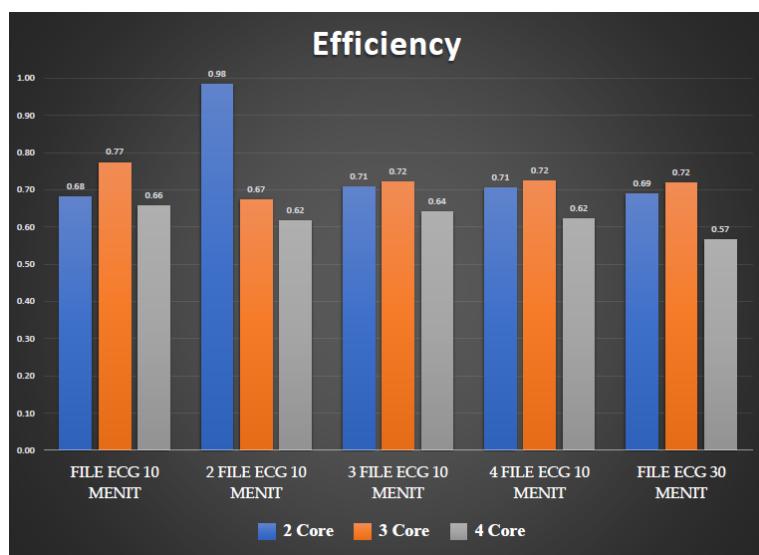
Grafik pada Gambar 7 merupakan grafik nilai speed up evaluasi dari paralel model yang menggunakan 2 inti, 3 inti, dan 4 inti prosesor. Terlihat pada grafik tersebut bahwa eksekusi paralel menggunakan 4 inti prosesor selalu mendapatkan nilai speed up yang paling tinggi, hal ini membuktikan bahwa peningkatan jumlah prosesor pada paralel model berbanding terbalik dengan waktu eksekusi yang dibutuhkan. Jadi, semakin banyak inti prosesor yang bekerja dalam mengeksekusi sistem maka semakin cepat waktu eksekusi yang dibutuhkan.



Gambar 7: Grafik Speed Up pada model paralel

3. Grafik Efficiency

Grafik pada Gambar 8 merupakan grafik nilai efficiency evaluasi dari paralel model yang menggunakan 2 inti, 3 inti, dan 4 inti prosesor. Terlihat pada grafik tersebut bahwa nilai efficiency paralel model tertinggi sebesar 0,98 yang terjadi saat melakukan eksekusi program paralel menggunakan 2 inti prosesor dengan masukan 2 file EKG selama 10 menit.



Gambar 8: Grafik Efficiency pada model paralel

V. Kesimpulan

Berdasarkan pengujian yang telah dilakukan terhadap implementasi sistem yang dibuat pada penelitian ini, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem berjalan baik dan berhasil menghasilkan laporan waktu kejadian aktivitas jantung namun akurasi masih buruk pada jenis tertentu.
2. Komputasi secara paralel telah berhasil memberikan efektivitas dalam penggunaan daya resource dan efisiensi waktu yang dibutuhkan lebih cepat dalam mengeksekusi program.
3. Hasil evaluasi dari paralel model memberikan nilai speed up tertinggi sebesar 2.63 yang terjadi pada paralel dengan menggunakan 4 inti prosesor dan nilai efficiency tertinggi sebesar 0.98 yang terjadi ketika menggunakan paralel dengan 2 inti prosesor.

Daftar Pustaka

- [1] I. Wibawa, I. Giriantari, and M. Sudarma, "Komputasi paralel menggunakan model message passing pada sim rs (sistem informasi manajemen rumah sakit)," *Majalah Ilmiah Teknologi Elektro*, vol. 17, p. 439, December 2018.
- [2] M. Susminkanti and W. Dewaynta, "Komputasi paralel eigenvalue dalam penyelesaian difusi multi-group menggunakan metode householder deflasi dan divide conquer," in *Lokakarya Komputasi dalam Sains dan Teknologi Nuklir*, 2012, pp. 341–352.
- [3] S. Qi and W. Yishan, "Detection and analysis of ecg based on parallel computing technology," in *2009 International Forum on Computer Science-Technology and Applications*, vol. 1, 2009, pp. 106–109.
- [4] N. P. V. Hegde and A. Thakur, "Parallel processing architecture for ecg signal analysis," *International Journal of Machine Learning and Computing*, pp. 291–293, January 2013.
- [5] W. H. Hwang, C. H. Jeong, D. H. Hwang, and Y. C. Jo, "Automatic detection of arrhythmias using a yolo-based network with long-duration ecg signals," *Engineering Proceedings*, vol. 2, no. 1, 2020, available: <https://www.mdpi.com/2673-4591/2/1/84>.
- [6] N. M. Alfarisi, "You only look once (yolo) algoritma deep learning object detection terbaik," Online, last accessed on 2021-02-19. Available: <https://haiqilmuhamadalfarisi.medium.com/you-only-look-once-yolo-algoritma-deep-learning-object-detection-terbaik-49ed81de9e9>.
- [7] A. AB, "darknet," 2020, Online, last accessed on 2020-02-06. Available: <https://github.com/AlexeyAB/darknet>.
- [8] P. Tröger, "Openmpi - parallel programming concepts - week 6," p. 69, June 2014.