

# Analisis Performa Klaster *Single Board Computer* dalam Implementasi *Singular Value Decomposition*

Syahrul Azka, Muhaza Liebenlito, dan Taufik Edy Sutanto



Volume 13, Issue 3, Pages 286–292, Dec. 2025

Diterima 11 Juli 2025, Direvisi 12 Oktober 2025, Disetujui 19 Oktober 2025, Diterbitkan 1 Desember 2025

To Cite this Article : S. Azka, M. Liebenlito, dan T. E. Sutanto, "Analisis Performa Klaster *Single Board Computer* dalam Implementasi *Singular Value Decomposition*", *Euler J. Ilm. Mat. Sains dan Teknol.*, vol. 13, no. 3, pp. 286–292, 2025, <https://doi.org/10.37905/euler.v13i3.33367>

© 2025 by author(s)

## JOURNAL INFO • EULER : JURNAL ILMIAH MATEMATIKA, SAINS DAN TEKNOLOGI



	Homepage	:	<a href="http://ejournal.ung.ac.id/index.php/euler/index">http://ejournal.ung.ac.id/index.php/euler/index</a>
	Journal Abbreviation	:	Euler J. Ilm. Mat. Sains dan Teknol.
	Frequency	:	Three times a year
	Publication Language	:	English (preferable), Indonesia
	DOI	:	<a href="https://doi.org/10.37905/euler">https://doi.org/10.37905/euler</a>
	Online ISSN	:	2776-3706
	License	:	Creative Commons Attribution-NonCommercial 4.0 International License
	Publisher	:	Department of Mathematics, Universitas Negeri Gorontalo
	Country	:	Indonesia
	OAI Address	:	<a href="http://ejournal.ung.ac.id/index.php/euler/oai">http://ejournal.ung.ac.id/index.php/euler/oai</a>
	Google Scholar ID	:	QF_r-gAAAAJ
	Email	:	<a href="mailto:euler@ung.ac.id">euler@ung.ac.id</a>

## JAMBURA JOURNAL • FIND OUR OTHER JOURNALS



Jambura Journal of Biomathematics



Jambura Journal of Mathematics



Jambura Journal of Mathematics Education



Jambura Journal of Probability and Statistics

# Analisis Performa Kluster *Single Board Computer* dalam Implementasi *Singular Value Decomposition*

Syahrul Azka<sup>1,\*</sup>, Muhaza Liebenlito<sup>1</sup>, dan Taufik Edy Sutanto<sup>1</sup><sup>1</sup>Program Studi Matematika, UIN Syarif Hidayatullah, Tangerang Selatan 15412, Indonesia

## ARTICLE HISTORY

Diterima 11 Juli 2025  
Direvisi 12 Oktober 2025  
Disetujui 19 Oktober 2025  
Diterbitkan 1 Desember 2025

## KATA KUNCI

Komputasi Berkelanjutan  
Komputasi Paralel  
Matematika Komputasi  
*Single Board Computer*  
*Singular Value Decomposition*

## KEYWORDS

Computational Mathematics  
Parallel Computing  
*Single Board Computer*  
Sustainable Computing  
*Singular Value Decomposition*

**ABSTRAK.** Penelitian ini bertujuan untuk mengevaluasi kinerja operasi *Singular Value Decomposition* berbasis *divide-and-conquer* pada dua jenis kluster komputasi, yaitu kluster PC berbasis Intel Core i5-12400 dan kluster *Single Board Computer* berbasis Allwinner H618. Evaluasi dilakukan terhadap tiga metrik utama: waktu eksekusi, *speedup*, dan konsumsi energi. Pengujian dilakukan pada tiga ukuran matriks ( $2.160 \times 2.160$ ,  $3.240 \times 3.240$ , dan  $5.400 \times 5.400$ ) dengan variasi jumlah inti prosesor dari 1 hingga 12 core. Konsumsi energi diukur menggunakan wattmeter dengan merekam daya puncak selama eksekusi. Hasil menunjukkan bahwa kluster PC menghasilkan waktu eksekusi tercepat, namun memiliki keterbatasan dalam skalabilitas paralel dengan *speedup* maksimum  $10,31 \times$  dan konsumsi energi  $2,07$  Wh pada pengujian matriks  $5.400 \times 5.400$  dengan dua belas core. Sebaliknya, kluster SBC menunjukkan efisiensi paralel yang jauh lebih tinggi dengan *speedup*  $117,75 \times$  dan konsumsi energi hanya  $0,23$  Wh pada konfigurasi yang sama. Temuan ini menunjukkan bahwa kluster SBC memiliki potensi sebagai solusi komputasi numerik paralel yang hemat energi, ekonomis, dan relevan untuk pengembangan infrastruktur komputasi berkelanjutan di lingkungan pendidikan tinggi, selaras dengan tujuan Sustainable Development Goals 7.

**ABSTRACT.** This study aims to evaluate the performance of *Singular Value Decomposition* operations based on the *divide-and-conquer* method on two computing cluster architectures: an Intel Core i5-12400-based PC cluster and an Allwinner H618-based Orange Pi Zero 3 *Single Board Computer* cluster. The evaluation focuses on three key metrics: execution time, *speedup*, and energy consumption. Experiments were conducted on three matrix sizes ( $2160 \times 2160$ ,  $3240 \times 3240$ , and  $5400 \times 5400$ ) with processor cores ranging from 1 to 12. Energy consumption was measured using a wattmeter by recording peak power during execution. The results show that the PC cluster achieves faster execution times but exhibits limited parallel scalability, reaching a maximum *speedup* of  $10.31 \times$  and energy consumption of  $2.07$  Wh for the  $5400 \times 5400$  matrix with 12 cores. In contrast, the SBC cluster demonstrates significantly higher parallel efficiency, achieving a *speedup* of  $117.75 \times$  with energy consumption of only  $0.23$  Wh under the same configuration. These findings indicate that the SBC cluster offers a promising energy-efficient, cost-effective solution for parallel numerical computing, particularly for sustainable computing infrastructure in higher education, in alignment with the Sustainable Development Goals 7.



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License. **Editorial of EULER:** Department of Mathematics, Universitas Negeri Gorontalo, Jln. Prof. Dr. Ing. B. J. Habibie, Bone Bolango 96554, Indonesia.

## 1. Pendahuluan

Kecerdasan artifisial (*Artificial Intelligence/AI*) telah menjadi pendorong utama dalam berbagai inovasi teknologi modern. Namun, di balik kemajuan pesat tersebut, tersimpan dampak lingkungan yang signifikan akibat tingginya kebutuhan energi selama proses pelatihan model. Sebagai contoh, pelatihan model AI berskala besar seperti GPT-3 dapat mengonsumsi hingga  $1.287$  MWh listrik dan menghasilkan sekitar  $552$  ton emisi  $CO_2$  atau setara dengan emisi tahunan dari  $123$  mobil berbahan bakar bensin [1]. Fakta ini menunjukkan bahwa pertumbuhan AI modern turut berkontribusi terhadap peningkatan jejak karbon global, terutama melalui kebutuhan komputasi berkinerja tinggi [2].

Sektor Teknologi Informasi dan Komunikasi (TIK)—yang mencakup pusat data dan infrastruktur jaringan—telah menyum-

bang sekitar  $1,4\%$  dari total emisi gas rumah kaca global pada tahun 2020 [3]. Proyeksi menunjukkan bahwa konsumsi energi pusat data global akan meningkat lebih dari tiga kali lipat antara tahun 2020 hingga 2035, dengan kontribusi terbesar berasal dari server yang dioptimalkan untuk AI (*AI-optimised servers*) [4]. Lonjakan ini sejalan dengan meningkatnya kebutuhan daya komputasi untuk mendukung beban kerja AI, yang diperkirakan mencapai  $85\text{--}134$  TWh pada tahun 2027 [5]. Kondisi tersebut menegaskan adanya kebutuhan mendesak untuk menyeimbangkan pertumbuhan teknologi dengan prinsip keberlanjutan lingkungan. Upaya ini sejalan dengan tujuan *Sustainable Development Goals* (SDG) 7, yang menekankan pentingnya menjamin akses terhadap energi yang bersih, terjangkau, andal, dan berkelanjutan bagi semua pihak [6].

Untuk menjawab tantangan tersebut, berkembang berbagai pendekatan yang berfokus pada efisiensi energi dalam sistem

\*Penulis Korespondensi.

komputasi modern. Salah satunya adalah pemanfaatan perangkat hemat daya seperti kluster *Single Board Computer* (SBC). Dalam beberapa tahun terakhir, SBC telah banyak digunakan dalam bidang ilmu komputer dan teknik, terutama di lingkungan pendidikan tinggi. Karakteristiknya yang ringkas, terjangkau, serta fleksibel menjadikan SBC cocok untuk mendukung pembelajaran berbasis proyek, termasuk dalam topik robotika, *Internet of Things* (IoT), komputasi paralel, dan pemrosesan numerik [7]. Dengan demikian, kluster SBC berpotensi menjadi solusi efisien dan ekonomis dalam pengujian serta pengembangan algoritma komputasi dan AI pada skala kecil hingga menengah.

Sejumlah penelitian telah mengevaluasi peningkatan performa kluster SBC seiring dengan kemajuan perangkat kerasnya. Salah satunya [8] menunjukkan bahwa kluster Raspberry Pi 4 Model B memiliki performa komputasi yang mampu menyamai sistem PC berprosesor Intel Core i5-12490F generasi ke-12 ketika diskalakan hingga 13 node. Penelitian lain [9] membandingkan performa Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+, dan Odroid C2, dan menemukan bahwa kluster 16-node mampu mencapai performa hingga 60 GFLOPS. Selain itu, penelitian [10] dan [11] membuktikan bahwa pendekatan paralel berbasis SBC dapat meningkatkan efisiensi pemrosesan numerik dan pengolahan citra secara signifikan, sekaligus memperkuat potensi SBC sebagai platform komputasi ramah energi.

Meskipun penelitian-penelitian sebelumnya menunjukkan potensi kluster SBC dalam komputasi paralel, masih terdapat kekosongan dalam eksplorasi implementasi algoritma dekomposisi matriks pada perangkat ini. Salah satu metode yang paling fundamental dan luas penggunaannya adalah Singular Value Decomposition (SVD). SVD memainkan peran penting dalam berbagai aplikasi dunia nyata, mulai dari sistem rekomendasi seperti Netflix untuk analisis preferensi pengguna [12], hingga pengolahan citra digital seperti watermarking dan kompresi gambar [13]. Selain itu, dalam data science dan machine learning, SVD digunakan secara luas untuk dimensionality reduction dan Principal Component Analysis (PCA) karena kemampuannya mengekstraksi informasi penting dari data berdimensi tinggi serta mengurangi noise [14]. Dengan fleksibilitasnya dalam menguraikan struktur data kompleks dan stabilitas numeriknya yang tinggi, SVD menjadi pilihan ideal untuk mengevaluasi kemampuan komputasi numerik pada sistem paralel seperti kluster SBC.

Liu et al. [15] mengembangkan algoritma *divide-and-conquer* yang dioptimalkan untuk sistem komputasi heterogen berbasis CPU-GPU dalam menyelesaikan operasi SVD. Dengan membagi masalah utama menjadi submasalah yang lebih kecil dan menjalankannya secara paralel, hasil eksperimen menunjukkan peningkatan performa hingga 33 kali lipat dibandingkan pustaka komputasi konvensional seperti LAPACK dan MKL. Pendekatan ini memperkuat relevansi SVD sebagai tolok ukur komputasi numerik intensif dalam sistem paralel modern.

Berbeda dengan penelitian-penelitian sebelumnya yang berfokus pada peningkatan performa atau skalabilitas sistem, penelitian ini menambahkan dimensi penting berupa pengukuran konsumsi energi sebagai bagian dari evaluasi menyeluruh. Dengan demikian, studi ini tidak hanya mengevaluasi kecepatan eksekusi, tetapi juga menilai efisiensi energi dari dua jenis kluster yang mewakili pendekatan konvensional dan alternatif—yakni kluster PC berbasis prosesor Intel Core i5 dan kluster SBC ber-

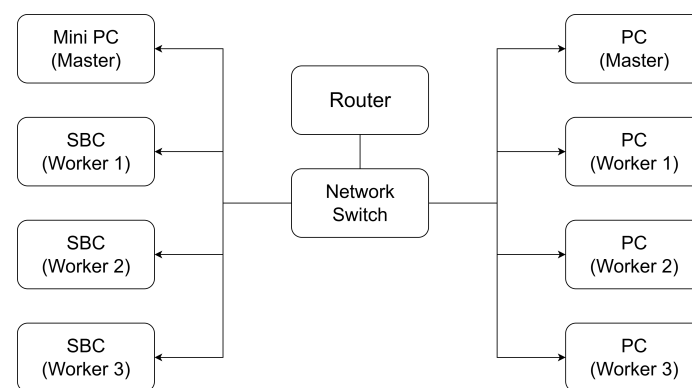
basis prosesor Cortex-A53 Allwinner H618. Pendekatan ini selaras dengan semangat SDG 7, yang menargetkan akses terhadap energi bersih dan terjangkau.

Melalui pengukuran langsung terhadap konsumsi daya dan performa waktu eksekusi, penelitian ini memproyeksikan kluster SBC sebagai solusi komputasi yang tidak hanya terjangkau secara ekonomi, tetapi juga lebih ramah lingkungan dibandingkan sistem konvensional. Dengan daya komputasi yang cukup untuk mendukung beban kerja paralel menengah, kluster SBC berpotensi besar untuk diimplementasikan dalam konteks pembelajaran komputasi dan pemrosesan numerik di perguruan tinggi. Hasil perbandingan performa dan konsumsi energi yang diperoleh diharapkan dapat menjadi landasan empiris bagi pengembangan sistem komputasi yang efisien, hemat daya, dan berkelanjutan.

## 2. Metode

### 2.1. Rancangan dan Prosedur Eksperimen

Penelitian ini menggunakan konsep komputasi paralel yang diterapkan pada kluster SBC dan kluster PC. Komputasi paralel memungkinkan suatu program untuk dibagi menjadi sejumlah proses yang lebih kecil, yang kemudian didistribusikan ke berbagai *node* dalam kluster untuk dijalankan secara simultan [16]. Tujuan utamanya adalah untuk meningkatkan kecepatan komputasi dan efisiensi penggunaan sumber daya, terutama saat menangani data besar atau perhitungan kompleks seperti kecerdasan artifisial, atau simulasi ilmiah. Dalam penelitian ini, eksperimen dilakukan dengan ilustrasi kluster sebagai berikut:

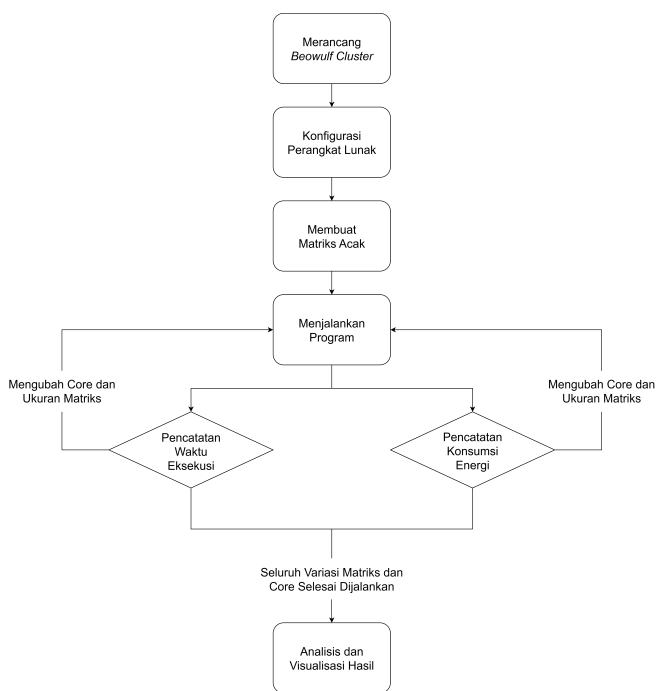


Gambar 1. Topologi kluster SBC dan kluster PC pada penelitian ini

Pada Gambar 1 menunjukkan arsitektur jaringan kluster yang digunakan dalam penelitian ini, yang terdiri dari dua kluster berbeda. Masing-masing kluster memiliki satu *master node* yang bertugas mengkoordinasikan proses, serta tiga *worker node* yang menjalankan tugas-tugas komputasi secara paralel. Seluruh *node* terhubung melalui sebuah jaringan *Local Area Network* (LAN) melalui *network switch* yang dikoneksikan ke *router*, memungkinkan komunikasi antar *node* berjalan secara efisien dalam satu jaringan lokal. Arsitektur ini dirancang untuk membandingkan performa antara dua jenis kluster dalam skenario komputasi paralel.

Untuk memberikan gambaran yang lebih jelas bagaimana eksperimen ini dijalankan, diagram alir proses akan disajikan pada Gambar 2.

Dalam penelitian ini, data yang digunakan berupa matriks persegi yang berisi bilangan acak kontinu berdistribusi *uniform*



Gambar 2. Prosedur eksperimen dalam penelitian ini

pada rentang 0 sampai 1. Matriks tersebut dihasilkan menggunakan fungsi dari pustaka NumPy. Untuk mengkaji dampak ukuran matriks terhadap performa komputasi dan efisiensi energi, eksperimen dilakukan menggunakan tiga skenario ukuran matriks, yaitu:

- Kecil :  $2160 \times 2160$ ,
- Sedang :  $3240 \times 3240$ , dan
- Besar :  $5400 \times 5400$ .

Setiap percobaan dilakukan pada variasi jumlah core yang berbeda, yaitu 1, 3, 6, 9, dan 12, dengan tujuan untuk mengevaluasi pengaruh skala perhitungan terhadap waktu eksekusi, speedup, dan konsumsi daya. Waktu dihitung dalam satuan detik menggunakan pustaka time di Python, sementara konsumsi daya diperoleh melalui pengukuran langsung menggunakan wattmeter digital selama proses eksekusi berlangsung.

### 2.2. Spesifikasi Perangkat Keras

Tabel 1 menyajikan rincian spesifikasi perangkat keras yang digunakan dalam pelaksanaan penelitian ini. Sebagai bagian dari analisis efisiensi energi, konsumsi daya pada masing-masing kluster diukur secara manual menggunakan wattmeter digital. Pengukuran dilakukan selama proses eksekusi program dengan variasi jumlah core dan ukuran matriks, sehingga data konsumsi daya dapat terekam secara langsung dan akurat sesuai kondisi aktual setiap kluster.

### 2.3. Singular Value Decomposition

Singular Value Decomposition adalah salah satu teknik dekomposisi matriks yang fundamental dalam aljabar linear dan banyak digunakan dalam berbagai aplikasi seperti pemrosesan sinyal, kompresi data, pengenalan pola, dan pembelajaran mesin. Secara matematis, SVD mendekomposisi sebuah matriks  $A$  berukuran  $m \times n$  menjadi tiga komponen, yaitu  $U$ ,  $V$ , dan  $\Sigma$ . Bentuk

Tabel 1. Spesifikasi perangkat keras Kluster SBC dan kluster PC

Komponen	Kluster SBC	Kluster PC
Arsitektur CPU	ARMv8	x86_64
Prosesor	Cortex-A53 Allwinner H618	Intel Core i5-12400
Clock-Speed	1.5 GHz	2.50 GHz
Jumlah Node	3	3
Jumlah Core yang Digunakan (per node)	4	4
RAM (per node)	4 GB	8 GB
Sistem Operasi	Ubuntu 22.04.4 LTS	Ubuntu 24.04.1 LTS
Jaringan	Ethernet 1 Gbps	Ethernet 100 Mbps

umum dari SVD bisa dilihat pada pers. (1) berikut:

$$A = U \Sigma V^T, \tag{1}$$

di mana  $U$  dan  $V$  adalah matriks ortogonal, dan  $\Sigma$  adalah matriks diagonal yang berisi nilai-nilai singular non-negatif [17].

Dalam penelitian ini, digunakan pendekatan algoritma divide-and-conquer untuk mengimplementasikan SVD secara manual, tanpa menggunakan fungsi built-in dari pustaka eksternal seperti LAPACK. Algoritma divide-and-conquer bekerja dengan membagi matriks besar menjadi sub-masalah yang lebih kecil secara rekursif (Divide), menyelesaikannya secara lokal (Conquer), dan menggabungkan kembali hasilnya menjadi solusi global (Combine) [9]. Keunggulan utama pendekatan ini adalah struktur algoritmiknya yang sangat cocok untuk paralelisasi, karena sub-masalah pada tahap Conquer dapat dieksekusi secara independen.

Dari sisi kompleksitas waktu, implementasi divide-and-conquer SVD memiliki kompleksitas waktu  $O(N^3)$  untuk menghitung seluruh nilai dan vektor singular dari matriks persegi berukuran  $N \times N$ . Namun, jika hanya nilai singular yang dihitung, kompleksitasnya dapat ditekan menjadi  $O(N^2)$ . Penelitian Ming Gu dan Eisenstat juga menunjukkan bahwa dengan menggunakan optimasi tambahan seperti Fast Multipole Method, kompleksitas tersebut dapat dikurangi hingga  $O(N^2)$  atau bahkan  $O(N \log^3 N)$  untuk kasus tertentu [18].

Implementasi algoritma ini diparalelkan dengan membagi sub-masalah ke beberapa proses yang berjalan secara independen di berbagai core dalam kluster. Paralelisasi dilakukan pada tahap Conquer dan Combine, di mana setiap sub-masalah dikerjakan oleh proses terpisah yang kemudian hasilnya digabungkan kembali. Komunikasi data antar proses diatur melalui mekanisme distribusi dan pengumpulan hasil, yang memungkinkan pembagian dan penggabungan informasi antar proses. Pendekatan ini bertujuan untuk mengevaluasi performa komputasi paralel dalam konteks efisiensi waktu eksekusi dan konsumsi energi, tanpa memfokuskan pada optimasi algoritmik dari SVD itu sendiri.

### 2.4. Implementasi MPI

Message Passing Interface (MPI) merupakan standar komunikasi yang umum digunakan dalam sistem komputasi paralel terdistribusi, khususnya pada arsitektur kluster. MPI memungkinkan banyak proses yang berjalan secara paralel di berbagai node untuk saling berkomunikasi dan bertukar data melalui mekanisme message-passing, tanpa memerlukan memori bersama.

Dalam penelitian ini, MPI digunakan untuk melakukan paralelisasi algoritma divide-and-conquer SVD berdasarkan metode

Gu dan Eisenstat [18]. Implementasi terdiri dari tiga fase, yaitu reduksi bidiagonal menggunakan transformasi Householder, divide-and-conquer rekursif dengan pembagian matriks menjadi submatriks  $B_1$  dan  $B_2$ , dan penggabungan hasil melalui penyelesaian *secular equation*. Skema komunikasi menggunakan pola *scatter* untuk distribusi submasalah ke *worker node* dan *gather* untuk pengumpulan hasil parsial. *Master node* melakukan koordinasi transformasi bidiagonal dan penyelesaian *secular equation*, sementara *worker node* menangani komputasi submatriks secara rekursif. Strategi ini mendukung evaluasi performa dan efisiensi energi antara kluster SBC dan kluster PC. Ilustrasi implementasi SVD pada penelitian ini dapat dilihat pada Gambar 3.

patan waktu yang signifikan, dengan *speedup* mencapai hingga 24 kali lipat dibandingkan dengan eksekusi sekuensial. Hal ini menunjukkan bahwa metrik *speedup* dapat memberikan gambaran yang jelas mengenai dampak paralelisasi terhadap waktu eksekusi dalam sistem kluster.

Selain itu, konsumsi energi diukur menggunakan wattmeter untuk mengetahui energi yang digunakan selama proses komputasi. Energi total dihitung dengan rumus:

$$\text{Energi}(Wh) = \frac{P \times t}{3600}, \tag{3}$$

di mana  $P$  adalah daya tertinggi (dalam watt) yang terbaca pada wattmeter, dan  $t$  adalah waktu eksekusi (dalam detik). Pengukuran konsumsi energi ini dilakukan untuk menilai seberapa besar energi yang dibutuhkan oleh sistem dalam menyelesaikan tugas SVD pada jumlah *core* tertentu. Dengan membandingkan nilai konsumsi energi antar kluster, dapat diketahui sistem mana yang lebih efisien dalam hal penggunaan daya saat menjalankan komputasi numerik berskala besar.

### 2.6. Analisis Statistik dan Keterbatasan Eksperimen

Analisis statistik dilakukan untuk memastikan perbedaan hasil antar kluster dan jumlah *core* memiliki signifikansi yang dapat dipertanggungjawabkan. Uji  $t$  dua sampel independen digunakan untuk menguji pengaruh jenis kluster (PC dan SBC) terhadap konsumsi energi, dengan taraf signifikansi 5%. Selanjutnya, analisis varians (ANOVA) digunakan untuk menguji:

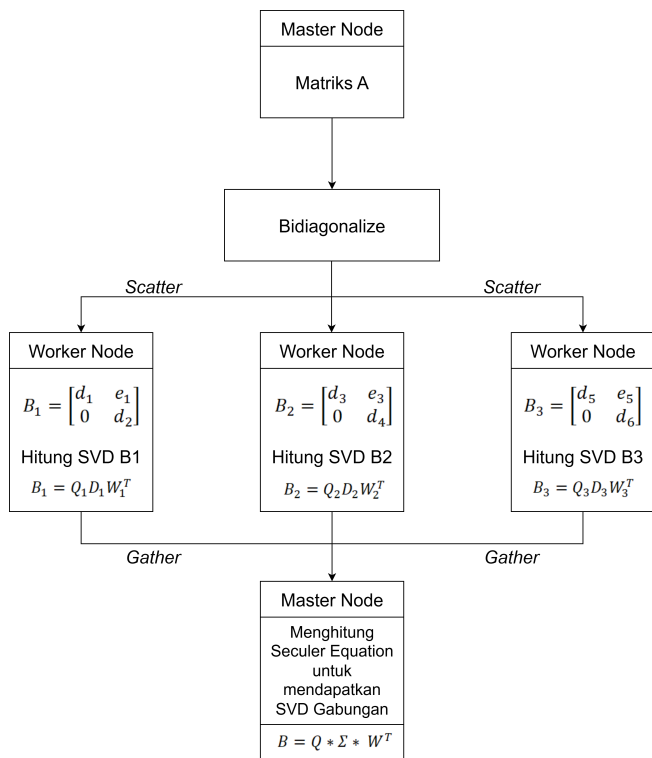
- pengaruh jenis kluster terhadap waktu eksekusi,
- pengaruh jenis kluster dan jumlah *core* terhadap konsumsi energi, dan
- pengaruh jumlah *core* terhadap *speedup*.

Sebelum pengujian dilakukan, data diuji asumsi normalitas (Shapiro–Wilk) dan homogenitas varians (Levene). Jika hasil signifikan ( $p < 0,05$ ), maka disimpulkan terdapat perbedaan yang bermakna antar kelompok.

### 3. Hasil dan Pembahasan

Bagian ini menyajikan hasil pengujian kinerja kluster SBC Orange Pi Zero 3 dalam menjalankan operasi SVD dengan pendekatan *divide-and-conquer*. Evaluasi dilakukan berdasarkan tiga metrik utama, yakni waktu eksekusi, *speedup*, dan konsumsi energi, terhadap tiga ukuran matriks berbeda. Seluruh eksperimen dijalankan dengan variasi jumlah inti prosesor (1 hingga 12 *core*) guna mengamati dampak skalabilitas terhadap performa sistem. Untuk keperluan perbandingan, eksperimen serupa juga dilakukan pada kluster PC berbasis prosesor Intel Core i5-12400. Visualisasi hasil disajikan untuk masing-masing kluster, sehingga memungkinkan analisis komparatif terkait efisiensi komputasi dan konsumsi daya dari kedua sistem.

Secara keseluruhan, pada Gambar 4 kluster PC menunjukkan performa waktu eksekusi yang jauh lebih cepat dibandingkan kluster SBC pada Gambar 5 untuk seluruh ukuran matriks. Peningkatan jumlah *core* yang digunakan menyebabkan penurunan signifikan dalam waktu eksekusi. Sebagai contoh, pada matriks berukuran besar, waktu eksekusi kluster PC menurun dari 312,67 detik (satu *core*) menjadi 30,34 detik (dua belas *core*), mencerminkan penurunan sebesar sekitar 90,30%. Sementara itu, pada kluster SBC, waktu eksekusi untuk ukuran matriks yang sama



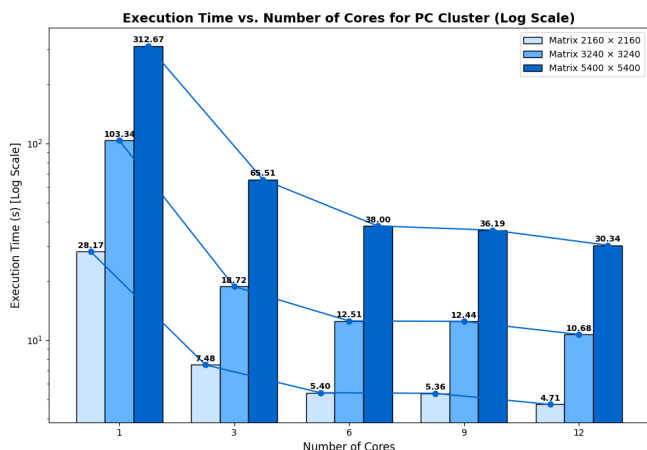
Gambar 3. Ilustrasi distribusi tugas pada implementasi SVD menggunakan MPI dengan skenario satu master node dan tiga worker node

### 2.5. Evaluasi Performa dan Konsumsi Energi

Evaluasi performa dalam penelitian ini dilakukan dengan mengukur waktu eksekusi program pada berbagai konfigurasi *core*. Waktu eksekusi dihitung dengan mencatat durasi yang diperlukan untuk menyelesaikan tugas SVD pada kluster yang digunakan menggunakan pustaka *time* di Python. Dari hasil pengukuran waktu eksekusi, metrik *speedup* digunakan untuk mengukur peningkatan performa ketika program dijalankan secara paralel dibandingkan dengan eksekusi secara serial. *Speedup* dihitung sebagai rasio antara waktu eksekusi pada satu *core* ( $T_{serial}$ ) terhadap waktu eksekusi pada beberapa *core* ( $T_{parallel}$ ), yang dirumuskan sebagai berikut:

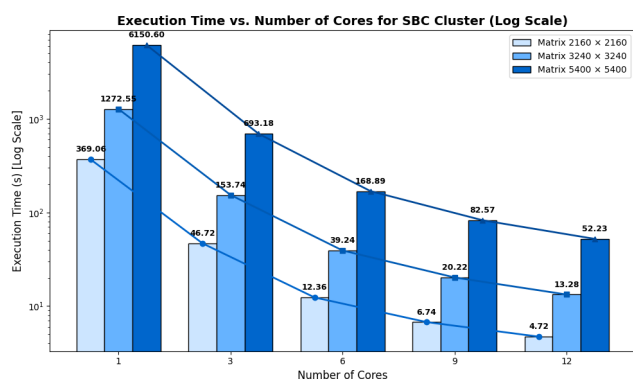
$$\text{Speedup} = \frac{T_{serial}}{T_{parallel}}. \tag{2}$$

Berdasarkan penelitian oleh Dubey dan Kagdi [10], peningkatan jumlah *core* pada kluster Raspberry Pi menunjukkan perce-



**Gambar 4.** Waktu eksekusi untuk operasi SVD di kluster PC dengan jumlah core dan tiga ukuran matriks yang berbeda

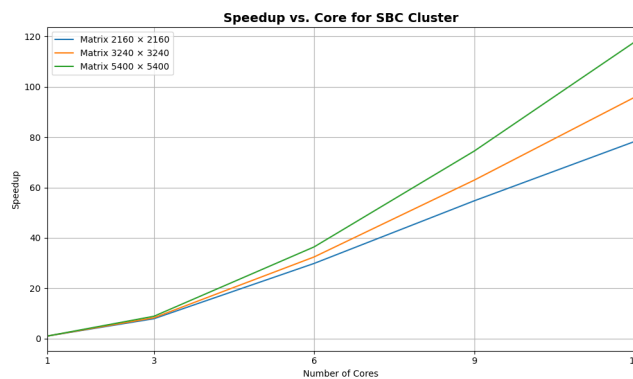
menurun jauh lebih drastis dari 6150,60 detik (satu core) menjadi 52,23 detik (dua belas core), menunjukkan efisiensi paralelisme yang sangat baik dengan penurunan sekitar 99,10%.



**Gambar 5.** Waktu eksekusi untuk operasi SVD di kluster SBC dengan jumlah core dan tiga ukuran matriks yang berbeda

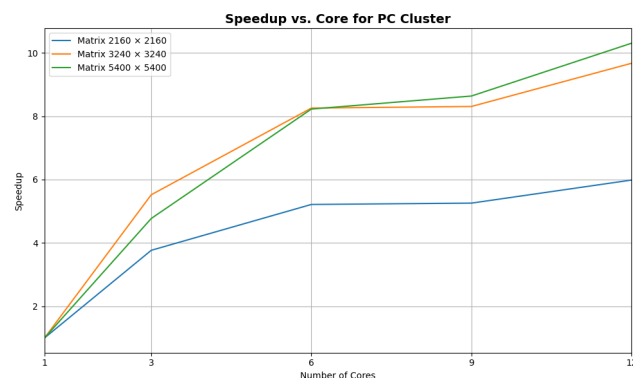
Tren grafik memperlihatkan bahwa pada kluster SBC, peningkatan jumlah core memberikan dampak signifikan terutama dari 1 hingga 3 core, kemudian terus menurun meskipun melandai pada 9 hingga 12 core—mengindikasikan titik saturasi performa di mana penambahan core tidak lagi menghasilkan pengurangan waktu yang proporsional. Kluster SBC meskipun berbasis perangkat hemat daya, tetap menunjukkan skalabilitas tinggi. Sebaliknya, kluster PC memiliki performa awal yang sangat baik namun dengan penurunan efisiensi paralel yang relatif lebih terbatas, ditunjukkan dengan pelandaian penurunan waktu eksekusi setelah penggunaan lebih dari enam core. Hal ini menunjukkan bahwa pada titik tertentu, overhead komunikasi dan sinkronisasi antar thread mulai mendominasi proses, membatasi peningkatan performa paralel.

Pada kluster SBC, peningkatan jumlah core secara konsisten menghasilkan peningkatan speedup yang hampir linier untuk seluruh ukuran matriks yang diuji seperti yang dapat dilihat pada Gambar 6. Untuk matriks berukuran  $5.400 \times 5.400$ , speedup mencapai sekitar 117 kali saat menggunakan dua belas core di-



**Gambar 6.** Speedup untuk operasi SVD di SBC kluster dengan jumlah core dan tiga ukuran matriks yang berbeda

bandingkan dengan satu core. Begitu pula untuk matriks ukuran  $3.240 \times 3.240$  dan  $2.160 \times 2.160$ , masing-masing menunjukkan speedup sebesar 95 kali dan 78 kali lebih besar. Pola peningkatan yang hampir proporsional ini mengindikasikan bahwa kluster SBC mampu menskalakan paralelisme secara sangat efisien, bahkan hingga jumlah core yang tinggi. Hal ini menunjukkan bahwa pembagian beban kerja dan komunikasi antar node dalam kluster SBC dapat dilakukan secara optimal tanpa menyebabkan bottleneck yang berarti.



**Gambar 7.** Speedup untuk operasi SVD di PC kluster dengan jumlah core dan tiga ukuran matriks yang berbeda

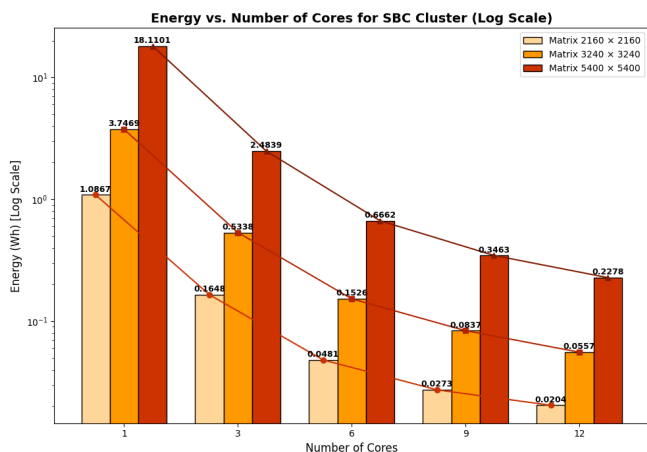
Sebaliknya, pada Gambar 7 terlihat bahwa peningkatan speedup pada kluster PC terbatas dan menunjukkan gejala saturasi setelah enam core. Untuk matriks berukuran  $5.400 \times 5.400$ , speedup maksimum yang dicapai dengan dua belas core hanya sekitar 10,3, sementara untuk matriks berukuran  $3.240 \times 3.240$  dan  $2.160 \times 2.160$  masing-masing mencapai 9, 6 dan 5, 9. Hal ini menunjukkan bahwa meskipun performa per core dari PC lebih tinggi dibandingkan SBC, skalabilitas paralelnya tidak sebaik SBC. Penurunan efisiensi ini kemungkinan besar disebabkan oleh meningkatnya overhead komunikasi dan keterbatasan arsitektur prosesor multi-core ketika jumlah core bertambah.

Kluster SBC, meskipun memiliki performa individu yang lebih rendah, menunjukkan efisiensi paralel yang lebih stabil dan konsisten. Sementara itu, kluster PC unggul pada jumlah core yang sedikit, tetapi menghadapi keterbatasan dalam peningkatan performa ketika jumlah core diperbanyak.

Selain waktu eksekusi, konsumsi energi menjadi indikator

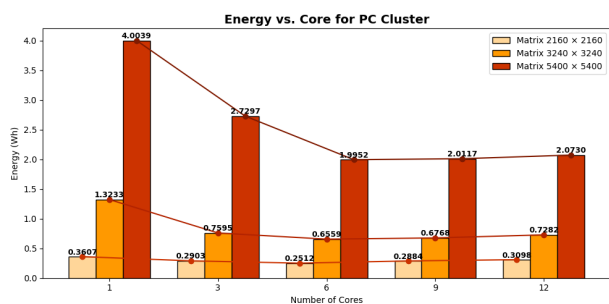
Tabel 2. Hasil uji beda statistik

No	Uji Statistik	Variabel yang Diuji	Nilai F / t	df	p - value	Keterangan
1	t-test	Jenis Kluster terhadap Konsumsi Daya (Watt)	$t = 8.406$	—	0.0000	Signifikan
2	ANOVA 1	Jenis Kluster terhadap Waktu Eksekusi (Paralel)	$F = 9.897$	(1,148)	0.0020	Signifikan
3	ANOVA 2	Jumlah Core terhadap Konsumsi Daya (Watt)	$F = 6.213$	(4,145)	0.0001	Signifikan
4	ANOVA 3	Jumlah Core terhadap Speedup	$F = 21.243$	(4,145)	0.0000	Signifikan



Gambar 8. Konsumsi energi untuk operasi SVD di SBC kluster dengan jumlah core dan tiga ukuran matriks yang berbeda

penting dalam menilai seberapa hemat daya suatu sistem komputasi saat menjalankan tugas-tugas paralel. Berdasarkan visualisasi hasil eksperimen, kluster SBC yang terlihat pada Gambar 8 menunjukkan konsumsi energi yang lebih baik dibandingkan kluster PC, khususnya dalam memproses matriks berukuran besar.



Gambar 9. Konsumsi energi untuk operasi SVD di PC kluster dengan jumlah core dan tiga ukuran matriks yang berbeda

Pada penggunaan satu core, kluster SBC mengonsumsi energi sebesar 18,1101 Wh untuk matriks berukuran  $5.400 \times 5.400$ , jauh lebih tinggi dibandingkan kluster PC yang hanya membutuhkan 4,0039 Wh untuk ukuran matriks yang sama. Namun, saat jumlah core ditingkatkan hingga dua belas, konsumsi energi pada kluster SBC turun drastis menjadi 0,2278 Wh, atau mengalami penurunan sekitar 98,7% dari konsumsi awal. Sedangkan, pada kluster PC penurunannya tidak sebesar yang terjadi pada kluster SBC seperti yang terlihat pada Gambar 9. Dari 4,0039 Wh (satu core) menjadi 2,0730 Wh (dua belas core), terjadi penurunan konsumsi energi sekitar 48,20%.

Meskipun kluster PC memiliki performa waktu eksekusi

yang lebih tinggi, hasil ini menunjukkan bahwa kluster SBC jauh lebih efisien dalam penggunaan energi ketika jumlah core diperbanyak. Arsitektur SBC mampu memanfaatkan paralelisme secara optimal tanpa meningkatkan konsumsi daya secara signifikan, sehingga lebih sesuai untuk komputasi hemat energi pada beban kerja besar seperti operasi SVD pada matriks berukuran  $5.400 \times 5.400$ .

Selain itu, uji statistik digunakan untuk mendukung klaim adanya perbedaan performa dan efisiensi sistem. Uji  $t$  seperti yang ditunjukkan pada Tabel 2 menunjukkan perbedaan signifikan pada konsumsi daya antara kluster PC dan SBC ( $p < 0.001$ ), dengan kluster SBC terbukti lebih hemat energi. Hasil ANOVA mengindikasikan bahwa jenis kluster berpengaruh signifikan terhadap waktu eksekusi paralel ( $p = 0.002$ ), sedangkan jumlah core memberikan pengaruh nyata terhadap konsumsi daya ( $p = 0.0001$ ) dan nilai speedup ( $p < 0.0001$ ). Temuan ini menegaskan bahwa perbedaan arsitektur perangkat keras dan skala paralelisme secara langsung memengaruhi keseimbangan antara performa komputasi dan efisiensi energi.

Selain hasil yang telah disajikan, terdapat beberapa faktor yang berpotensi mempengaruhi performa sistem dan perlu menjadi perhatian dalam pengembangan selanjutnya. Salah satunya adalah overhead komunikasi antar node yang muncul akibat proses pertukaran data selama eksekusi paralel. Overhead ini dapat meningkatkan waktu total komputasi, terutama ketika jumlah node bertambah. Selain itu, mekanisme sinkronisasi dan mutual exclusion antar proses juga dapat menyebabkan latensi tambahan apabila tidak dikelola secara efisien, terutama pada tahap pembagian dan penggabungan hasil komputasi.

Dari sisi skalabilitas, penelitian ini terbatas pada konfigurasi tiga node, sehingga belum sepenuhnya merepresentasikan perilaku sistem dalam skala yang lebih besar. Penelitian sebelumnya [9] telah menunjukkan bahwa kluster SBC masih dapat diskalakan hingga 16 node dengan peningkatan performa yang signifikan, meskipun efisiensinya cenderung menurun seiring bertambahnya jumlah node akibat meningkatnya overhead komunikasi. Oleh karena itu, penelitian lanjutan disarankan untuk mengeksplorasi skalabilitas ke atas serta mengkaji dampak komunikasi antar node terhadap efisiensi sistem dalam konfigurasi kluster berukuran besar.

#### 4. Kesimpulan

Penelitian ini mengkaji evaluasi sistem komputasi paralel dengan memasukkan efisiensi energi sebagai metrik kunci di samping performa. Melalui perbandingan antara kluster PC berbasis Intel Core i5 dan kluster Single Board Computer (SBC) berbasis Cortex-A53, studi ini menemukan bahwa meskipun PC lebih cepat, kluster SBC secara dramatis lebih hemat energi, dengan penghematan daya mencapai 98% dan speedup hingga  $117 \times$  pada konfigurasi dua belas core. Dengan keunggulan tersebut, kluster SBC berpotensi besar untuk diimplementasikan dalam kon-

teks pembelajaran komputasi dan pemrosesan numerik di perguruan tinggi. Temuan ini menyoroti potensi SBC sebagai solusi komputasi paralel yang ekonomis dan ramah lingkungan, sejalan dengan *Sustainable Development Goals* (SDG) 7, serta menegaskan pentingnya integrasi metrik efisiensi energi dalam penilaian sistem di lingkungan akademik.

Meskipun hasilnya signifikan, penelitian ini memiliki keterbatasan seperti jumlah *node* yang sedikit, performa prosesor ARM yang lebih rendah dari x86, dan pengukuran energi berbasis *peak power*. Oleh karena itu, studi lanjutan disarankan untuk menggunakan *real-time power logging* untuk profil energi yang lebih akurat, memperluas jumlah *node* untuk menguji skalabilitas, serta menguji berbagai algoritma paralel. Sebagai langkah strategis ke depan, integrasi kluster SBC dengan sumber energi terbarukan seperti panel surya diusulkan untuk memperkuat penerapan konsep *green computing* secara praktis di dunia riset dan pendidikan.

**Kontribusi Penulis.** Syahrul Azka: Mendesain dan menyusun infrastruktur kluster komputer, melakukan eksperimen, penulisan awal. Muhaza Liebenlito: Editor, supervisi, validasi. Taufik Edy Sutanto: Editor, supervisi, validasi. Semua penulis telah membaca dan menyetujui versi manuskrip yang diterbitkan.

**Ucapan Terima Kasih.** Para penulis mengucapkan terima kasih kepada editor dan reviewer yang telah berkontribusi dalam penelitian ini dan dalam penyusunan manuskrip. Kami sangat menghargai editor dan reviewer atas masukan serta dukungannya dalam menyempurnakan karya ini.

**Pembiayaan.** Penelitian ini tidak menerima pendanaan eksternal.

**Konflik Kepentingan.** Para penulis menyatakan tidak ada konflik kepentingan yang terkait dengan artikel ini.

**Ketersediaan Data.** Tidak tersedia.

## Referensi

- [1] Y. I. Alzoubi and A. Mishra, "Green artificial intelligence initiatives: Potentials and challenges," *Journal of Cleaner Production*, vol. 468, 8 2024, doi: 10.1016/j.jclepro.2024.143090.
- [2] J. Cowsls, A. Tsamados, M. Taddeo, and L. Floridi, "The ai gambit: leveraging artificial intelligence to combat climate change—opportunities, challenges, and recommendations," *AI and Society*, vol. 38, pp. 283–307, 2 2023, doi: 10.1007/s00146-021-01294-x.
- [3] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig, and D. Rolnick, "Aligning artificial intelligence with climate change mitigation," *Nature Climate Change*, vol. 12, pp. 518–527, 6 2022, doi: 10.1038/s41558-022-01377-7.
- [4] T. Spencer and S. Singh, "Energy and ai: World energy outlook special report," International Energy Agency, Tech. Rep., 2025, [Online]. Available: [www.iea.org/terms](http://www.iea.org/terms) [Accessed: 2-June-2025].
- [5] P. Li, J. Yang, M. A. Islam, and S. Ren, "Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models," *Communications of the ACM*, 4 2023, doi: 10.48550/arXiv.2304.03271.
- [6] U. N. D. of Economic and S. Affairs, "The 17 goals | sustainable development," [Online]. Available: <https://sdgs.un.org/goals/goal7> [Accessed: 5-May-2025].
- [7] J. Ariza and H. Baez, "Understanding the role of single-board computers in engineering and computer science education: A systematic literature review," pp. 304–329, 1 2021, doi: 10.1002/cae.22439.
- [8] P. Liu, X. Cao, and Y. Jia, "Performance evaluation and analysis of scalable raspberry pi 4 model b clusters," 6 2024, doi: 10.21203/rs.3.rs-4460804/v1.
- [9] P. J. Basford, S. J. Johnston, C. S. Perkins, T. Garnock-Jones, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, J. Singer, and S. J. Cox, "Performance analysis of single board computer clusters," *Future Generation Computer Systems*, vol. 102, pp. 278–291, 1 2020, doi: 10.1016/j.future.2019.07.040.
- [10] P. Dubey and A. Kagdi, "Run time analysis of matrix multiplication using raspberry pi cluster supercomputer," Tech. Rep., 2020, doi: 10.13140/RG.2.2.22065.20324/1.
- [11] A. N. Fauzie, S. P. Sakti, and Rahmadwati, "Parallel implementation of gaussian filter image processing on a cluster of single board computer," *Jurnal EEECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)*, vol. 17, pp. 82–88, 12 2023, doi: 10.21776/jeeccis.v17i3.1672.
- [12] T. Widiyaningtyas, M. I. Ardiansyah, and T. B. Adji, "Recommendation Algorithm Using SVD and Weight Point Rank (SVD-WPR)," *Big Data and Cognitive Computing*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040121.
- [13] L. Wang and H. Ji, "A Watermarking Optimization Method Based on Matrix Decomposition and DWT for Multi-Size Images," *Electronics (Switzerland)*, vol. 11, no. 13, 2022, doi: 10.3390/electronics11132027.
- [14] A. Falini, "A review on the selection criteria for the truncated SVD in Data Science applications," *Journal of Computational Mathematics and Data Science*, vol. 5, no. July, p. 100064, 2022, doi: 10.1016/j.jcmds.2022.100064.
- [15] D. Liu, R. Li, D. J. Lilja, and W. Xiao, "A divide-and-conquer approach for solving singular value decomposition on a heterogeneous system," in *Proceedings of the ACM International Conference on Computing Frontiers*. ACM, 2013, pp. 1–10, doi: 10.1145/2482767.2482811.
- [16] L. C. Moleong, A. M. Rumagit, and B. A. Sugiarso, "Implementasi cluster computing untuk render animasi," *e-Jurnal Teknik Elektro dan Komputer*, vol. 2, pp. 1–8, 2013, doi: 10.35793/jtek.v2i3.2109.
- [17] H. Li, T. Liu, X. Wu, and Q. Chen, "A bearing fault diagnosis method based on enhanced singular value decomposition," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 3220–3230, 5 2021, doi: 10.1109/TII.2020.3001376.
- [18] M. Gu and S. C. Eisenstat, "A divide-and-conquer algorithm for the bidiagonal svd," *SIAM Journal on Matrix Analysis and Applications*, vol. 16, pp. 79–92, 12 1992, doi: 10.1137/S0895479893255124.