

Comparison of RStudio and Python Performance on Green Computing-Based Big Data Analytics

Salman¹, Heni Sulastr², Muhammad Al Husaini³

^{1,3}Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya, Indonesia

²Program Studi Sistem Informasi, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya, Indonesia

Email: 207006010@student.unsil.ac.id¹, henisulastr@unsil.ac.id², alhusaini@unsil.ac.id³

Received 30 Maret 2026 | Revised 5 April 2026 | Accepted 10 April 2026

ABSTRAK

Perkembangan data berskala menengah mendorong kebutuhan platform analitik yang tidak hanya cepat, tetapi hemat sumber daya. Penelitian ini bertujuan membandingkan performa R (melalui RStudio) dan Python terhadap Big Data Analytics berbasis Green Computing pada data skala menengah. Melalui metode komparatif pada dataset “AI4I 2020 Predictive Maintenance”, penelitian ini mengukur waktu eksekusi, utilisasi CPU, penggunaan memori, dan konsumsi energi. Hasil pengujian menunjukkan RStudio mencatat waktu eksekusi 19.916,67 ms, utilisasi CPU 0,08%, penggunaan memori 75,43 MB, dan konsumsi energi 440,95 J, sedangkan Python mencatat 270.447,73 ms, utilisasi CPU 1,14%, penggunaan memori 689,67 MB, dan konsumsi energi 7.178,24 J. Temuan menunjukkan bahwa perbedaan performa bersifat kontekstual terhadap karakteristik workload dan desain pipeline analitik, serta menegaskan adanya trade-off antara efisiensi komputasi dan fleksibilitas ekosistem dalam kerangka Green Computing.

Kata kunci: green computing, big data analytics, benchmarking, rstudio, python

ABSTRACT

The growth of medium scale data is driving the need for analytics platforms that are not only fast, but also resource-efficient. This research aims to compare the performance of R (via RStudio) and Python on Green Computing based Big Data Analytics on medium scale data. Through comparative methods on the “AI4I 2020 Predictive Maintenance” dataset, this research measures execution time, CPU utilization, memory usage and energy consumption. Test results show that RStudio recorded an execution time of 19,916.67 ms, CPU utilization 0.08%, memory usage 75.43 MB, and energy consumption 440.95 J, while Python recorded 270,447.73 ms, CPU utilization 1.14%, memory usage 689.67 MB, and energy consumption 7,178.24 J. The findings show that the performance differences are contextual to the characteristics. workload and analytical pipeline design, and emphasizes the existence of a trade-off between computing efficiency and ecosystem flexibility in the Green Computing framework.

Keywords: green computing, big data analytics, benchmarking, rstudio, python

1. PENDAHULUAN

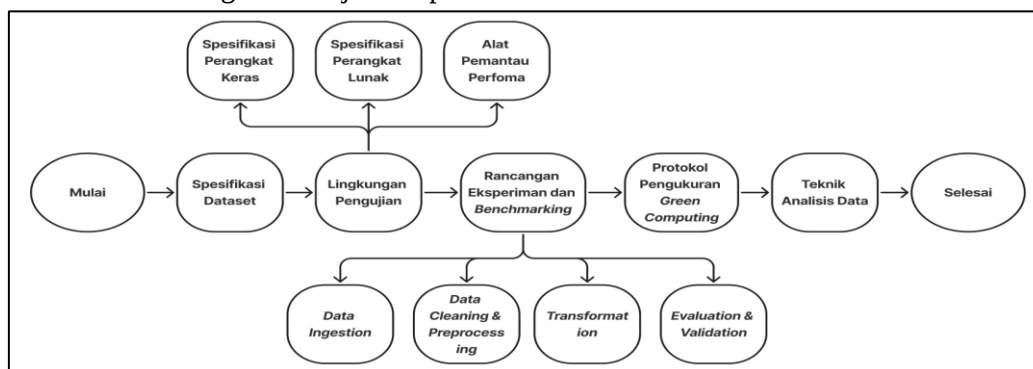
Fenomena *big data* tidak hanya berkaitan dengan penyimpanan, tetapi juga kebutuhan pengolahan data yang efektif agar data dapat diubah menjadi informasi bernilai untuk mendukung pengambilan keputusan, terutama melalui *pipeline* analitik yang terukur. Dalam ekosistem sains data, *R* (melalui *RStudio*) dan *Python* merupakan dua platform yang paling luas digunakan. *R* (melalui *RStudio*) dikenal kuat pada analisis statistik, sedangkan *Python* menonjol pada fleksibilitas ekosistem *machine learning* dan integrasi sistem [1], [2]. Seiring meningkatnya intensitas analitik, kebutuhan komputasi juga meningkat dan berdampak pada konsumsi energi. Jika tidak dikelola dengan baik, beban komputasi dapat memperbesar jejak energi digital. Oleh karena itu, penerapan *Green Computing* penting untuk memastikan proses analitik tidak hanya berorientasi pada performa, tetapi juga mempertimbangkan aspek keberlanjutan melalui pengukuran pemanfaatan sumber daya dan energi [3], [4].

Tinjauan penelitian terdahulu menunjukkan bahwa studi komparatif antara *R* (melalui *RStudio*) dan *Python* masih beragam dari sisi rancangan eksperimen dan metrik yang dilaporkan, serta sebagian belum mengintegrasikan evaluasi *Green Computing* secara komprehensif. Batko dan Ślęzak [5] menyoroti penerapan *Big Data Analytics* pada bidang kesehatan, namun tidak berfokus pada perbandingan lintas platform dari sisi metrik sumber daya. Maheshwari [6] melaporkan perbedaan waktu eksekusi beberapa bahasa pemrograman akan tetapi, pada ruang lingkup studi tersebut evaluasi masih dominan pada aspek waktu tanpa melibatkan rangkaian metrik sumber daya yang lebih lengkap. Nahrstedt, dkk. [7] mengevaluasi konsumsi energi dan performa pada pustaka pengolahan data di *Python* (misalnya *pandas* dan *polars*), namun kajiannya berada pada konteks pustaka dalam ekosistem *Python* dan belum diarahkan sebagai komparasi lintas platform pada *pipeline* analitik yang terintegrasi.

Berdasarkan celah tersebut, penelitian ini melakukan studi komparatif untuk membandingkan performa *R* (melalui *RStudio*) dan *Python* pada *big data analitik* berbasis *Green Computing*. Performa pada penelitian ini diukur menggunakan empat metrik utama, yaitu waktu eksekusi, utilisasi *CPU*, penggunaan memori, dan konsumsi energi, pada dataset “*AI4I 2020 Predictive Maintenance*”. Dengan parameter evaluasi yang terukur, penelitian ini diharapkan memberikan bukti empiris mengenai performa komputasi kedua platform serta menjadi pertimbangan bagi praktisi dalam memilih platform analitik sesuai kebutuhan implementasi dan prinsip keberlanjutan teknologi.

2. METODOLOGI

Penelitian ini menggunakan pendekatan komparatif melalui benchmarking untuk membandingkan performa *R* (melalui *RStudio*) dan *Python* dalam konteks *Big Data Analytics* berbasis *Green Computing*. Alur penelitian secara ringkas ditunjukkan pada Gambar 1.



Gambar 1. Diagram alur penelitian perbandingan performa *RStudio* dan *Python*, mencakup tahapan spesifikasi dataset, persiapan lingkungan pengujian, rancangan benchmarking, protokol *Green Computing*, hingga analisis dan evaluasi data.

2.1 Dataset

Penelitian menggunakan dataset “*AI4I 2020 Predictive Maintenance*” sebagai data uji. Dataset ini berisi 10.000 observasi dengan 14 atribut, mencakup fitur proses dan label kegagalan mesin. Dataset digunakan sebagai input yang sama pada kedua platform untuk menjaga kesetaraan data uji.

2.2 Lingkungan Pengujian

Seluruh pengujian dijalankan pada satu *testbed* fisik yang sama secara *bare-metal* (tanpa virtualisasi) untuk meminimalkan *overhead* eksternal. Spesifikasi perangkat keras utama menggunakan prosesor Intel Core i5-8250U dengan arsitektur 4 core fisik dan 8 thread logika (base frequency 1,60 GHz), RAM 8 GB DDR4, dan penyimpanan SSD 256 GB pada sistem operasi Windows 11 Home. Untuk menjaga kondisi environment yang stabil dan validitas pengukuran, skenario pengujian diisolasi dengan ketentuan: perangkat terhubung secara konstan ke sumber daya listrik dengan mode daya *best performance*, menonaktifkan pembaruan otomatis OS, serta menutup seluruh aplikasi dan *service* latar belakang yang tidak relevan agar tidak terjadi anomali pada utilisasi *core/thread CPU*.

2.3 Rancangan Eksperimen dan Benchmarking

Beban kerja analitik dirancang dalam empat tahapan yang dipetakan secara fungsional setara pada kedua platform, yaitu *data ingestion*, *data cleaning and preprocessing*, *transformation*, serta *evaluation and validation*. Pemetaan operasi dilakukan untuk memastikan struktur data, tujuan operasi, dan keluaran antara berada pada kondisi pengujian yang sebanding. Untuk memastikan metodologis dan kesetaraan *workload*, Tabel 1 menyajikan detail operasi *benchmarking* pada setiap tahapan *pipeline* analitik yang diimplikasikan pada kedua platform.

Tabel 1. Operasi Benchmarking pada Pipeline Analitik

Tahapan	Operasi Utama	R (RStudio)	Python	Parameter/Standar
Data Ingestion	Membaca CSV ke <i>dataframe</i>	<code>data.table::fread()</code>	<code>pandas.read_csv()</code>	Input dataset sama
Data Ingestion	Penyeragaman nama kolom	<code>make.names()</code>	<code>df.rename()</code>	Konsisten antar platform
Data Cleaning & Preprocessing	Menghapus kolom identitas/non-fitur	<code>dplyr::select(-Product.ID, -UDI)</code>	<code>df.drop(['Product ID', 'UDI'], axis=1)</code>	Kolom yang dihapus disetarakan
Data Cleaning & Preprocessing	Menetapkan target	<code>y <- data\$Machine.failure</code>	<code>y = df['Machine failure']</code>	Target: <i>Machine failure</i>
Data Cleaning & Preprocessing	Menghapus label turunan dari fitur	<code>dplyr::select(-TWF, -HDF, -PWF, -OSF, -RNF)</code>	<code>df.drop(['TWF', 'HD F', 'PWF', 'OSF', 'RN F'], axis=1)</code>	Mencegah <i>label leakage</i>
Transformation	Encoding Type → numerik	<code>factor(Type, levels=c('L', 'M', 'H')) → numerik</code>	<code>replace({'L':0, 'M':1, 'H':2})</code>	Mapping: L=0, M=1, H=2
Evaluation & Validation	Train-test split 70:30 (stratified)	<code>set.seed(42); caret::createDataPartition(y, p=0.7, times=1, list=FALSE)</code>	<code>train_test_split(test_size=0.3, stratify=y, random_state=42)</code>	Seed: 42
Evaluation & Validation	Oversampling data latih (SVM-SMOTE)	<code>set.seed(42); smotefamily::SVM.SMOTE(X_train, y_train)</code>	<code>oversample = SVMSMOTE(random_state = 42)</code>	Seed: 42
Evaluation & Validation	Model klasifikasi ekuivalen	<code>glm(Machine.failure ~ ., family=binomial, data=train)</code>	<code>LogisticRegression(solver='lbfgs', max_iter=1000, random_state=42)</code>	Model: <i>Logistic Regression</i> Seed: 42

Penelitian ini menggunakan pendekatan *functional equivalence*, di mana operasi pada kedua platform disetarakan berdasarkan tujuan dan keluaran akhir, bukan berdasarkan implementasi teknis yang identik baris per baris. Perbedaan sintaks dan mekanisme internal dipertahankan sebagai bagian dari karakteristik ekosistem masing-masing platform, sehingga perbandingan performa mencerminkan kekuatan dan kelemahan alami dari *R* dan *Python* dalam konteks penggunaan nyata.

2.4 Protokol Pengukuran Green Computing

Parameter yang diukur meliputi waktu eksekusi, utilisasi *CPU*, penggunaan memori, dan konsumsi energi. Waktu eksekusi dicatat pada level skenario menggunakan fungsi pencatat waktu pada *R* `Sys.time()/proc.time()`, *Python* `time.time()/timeit.default_timer()`, kemudian dilaporkan dalam satuan milidetik (ms). Utilisasi *CPU* dan penggunaan memori dicatat pada level proses menggunakan sampling periodik setiap 0,5 detik. Pada *R*, utilisasi *CPU* proses dihitung dari selisih waktu *CPU* kumulatif proses antar-sampling `ps::ps_cpu_times()`, yang dinormalisasi terhadap interval sampling dan jumlah core sedangkan pada *Python*, utilisasi *CPU* proses dibaca melalui `psutil.Process().cpu_percent()` pada interval yang sama. Penggunaan memori proses pada *R* dicatat menggunakan `ps::ps_memory_info()` sedangkan pada *Python* menggunakan `psutil.Process().memory_info().rss` dan nilai memori disajikan dalam satuan *megabyte* (MB). Konsumsi energi diukur menggunakan alat pemantau *Intel Power Gadget* (IPG) versi 3.6 yang membaca register perangkat keras secara langsung melalui *Model-Specific Registers* (MSR). Pengukuran difokuskan pada metrik *Processor Energy* (Level Paket *CPU*), yang mencakup total daya yang ditarik oleh seluruh *core/thread* aktif, *memory controller*, dan *integrated graphics* selama proses analitik berjalan.

IPG dikonfigurasi untuk mencatat data (*logging*) dengan resolusi pengambilan sampel setiap 100 milidetik. Karena IPG mencatat akumulasi energi dalam satuan *joule*, konsumsi energi spesifik untuk setiap skenario pengujian (E_{run}) dihitung menggunakan rumus selisih metrik kumulatif energi pada *timestamp* akhir (E_{akhir}) dan *timestamp* awal (E_{awal}) saat *pipeline* dieksekusi, sesuai dengan persamaan (1) matematik sebagai berikut:

$$E_{run} = E_{akhir} - E_{awal} \quad (1)$$

Keterangan:

E_{run} = Total konsumsi energi spesifik selama satu skenario pengujian berjalan.

E_{akhir} = Nilai metrik kumulatif energi pada *timestamp* akhir eksekusi.

E_{awal} = Nilai metrik kumulatif energi pada *timestamp* awal eksekusi.

Melalui Persamaan (1) tersebut, energi latar belakang OS yang tidak terkait dengan eksekusi *RStudio* maupun *Python* dapat dieliminasi secara efektif.

2.5 Teknik Analisis Data

Untuk memastikan validitas hubungan antara data mentah yang direkam dan hasil analisis akhir, seluruh *log* data metrik (file CSV) yang dihasilkan oleh sistem monitor dan IPG selama skenario berjalan diekstraksi secara sekuensial. Data utilisasi *CPU* (dalam persentase) dan penggunaan memori (dalam *megabyte*) yang direkam setiap interval 0,5 detik dianalisis menggunakan statistik deskriptif untuk mendapatkan nilai rata-rata (*mean*), puncak (*peak*), dan persentil ke-95 (*P95*) sepanjang durasi eksekusi. Sementara itu, data konsumsi energi (dalam *joule*) dihitung berdasarkan selisih kumulatif menggunakan rumus yang telah ditetapkan. Setiap *pipeline* diuji sebanyak tiga kali replikasi independen untuk mengontrol variansi data, dan nilai yang disajikan pada hasil akhir merupakan nilai rata-rata dari ketiga

iterasi tersebut. Seluruh hasil rekapitulasi ini disajikan dalam bentuk tabel dan grafik untuk memudahkan perbandingan performa antara *R* (melalui *RStudio*) dan *Python* pada setiap parameter. Hasil yang diperoleh dianalisis secara komparatif dengan penelitian terdahulu yang relevan, dengan memperhatikan perbedaan beban kerja (*workload*) dan desain eksperimen yang dapat memengaruhi hasil pengukuran.

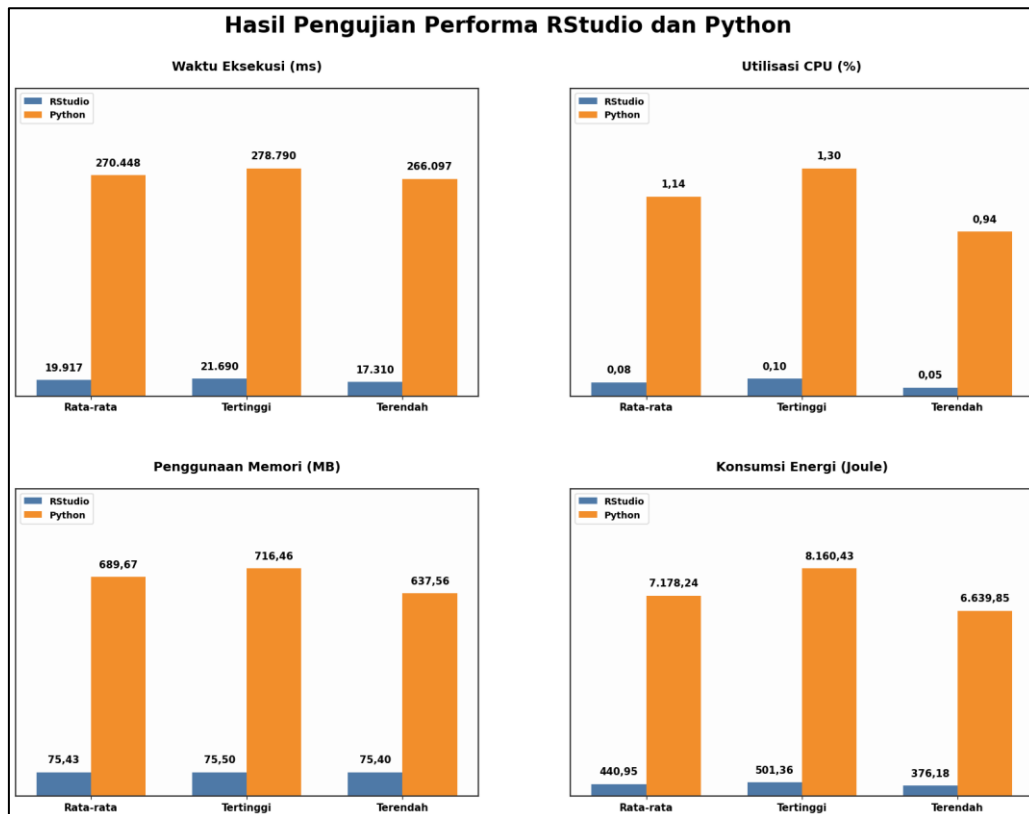
3. HASIL DAN PEMBAHASAN

Penelitian dilakukan pada dataset “*AI4I 2020 Predictive Maintenance*”, dengan mengukur empat parameter *Green Computing*, yaitu waktu eksekusi, utilisasi *CPU*, penggunaan memori, dan konsumsi energi. Setiap skenario pengujian dijalankan sebanyak tiga kali untuk meningkatkan reliabilitas pengukuran, kemudian diringkas menggunakan nilai rata-rata. Ringkasan hasil pengukuran disajikan pada Tabel 2 sebagai perbandingan performa pada kedua platform.

Tabel 2. Hasil Rata-rata Pengujian Performa *RStudio* dan *Python*

Parameter	<i>RStudio</i> (Rata-rata)	<i>Python</i> (Rata-rata)
Waktu Eksekusi (ms)	19.916,67 ms	270.447,73 ms
Utilisasi <i>CPU</i> (%)	0,08 %	1,14 %
Penggunaan Memori (MB)	75,43 MB	689,67 MB
Konsumsi Energi (J)	440,95 J	7.178,24 J

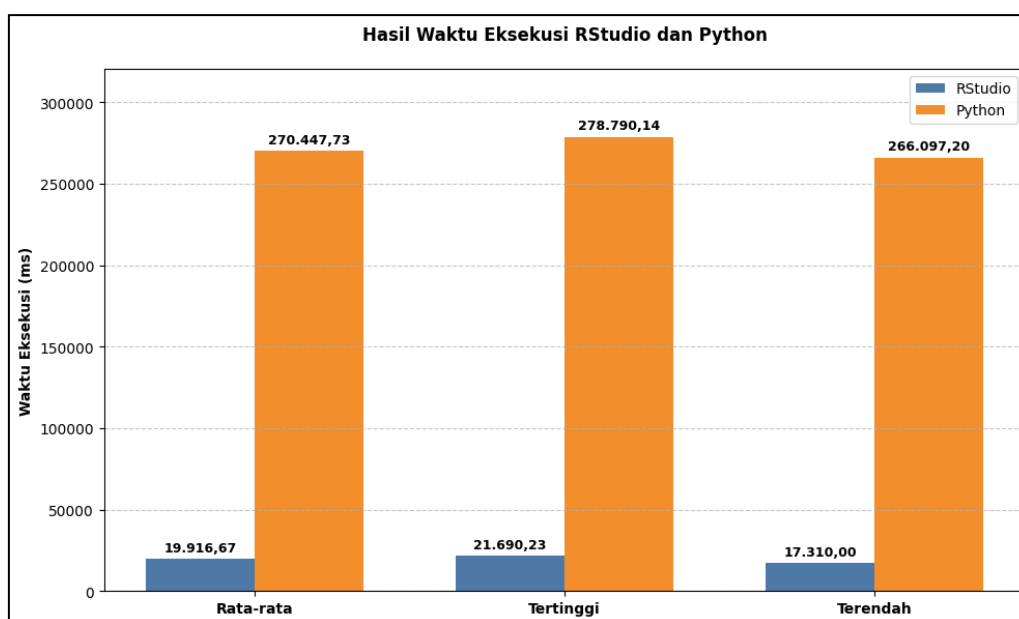
Visualisasi perbedaan performa ditampilkan pada Gambar 2 sebagai gambaran awal sebelum interpretasi per metrik. Secara umum, nilai pada *RStudio* cenderung lebih rendah dibanding *Python* pada keempat parameter yang diukur.



Gambar 2. Perbandingan performa *RStudio* dan *Python* berdasarkan nilai rata-rata, tertinggi, dan terendah pada empat parameter pengujian (waktu eksekusi, utilisasi *CPU*, memori, dan konsumsi energi)

RStudio mencatat waktu eksekusi rata-rata 19.916,67 ms, sedangkan Python 270.447,73 ms. Selisih ini menunjukkan perbedaan efisiensi *end-to-end* pada alur kerja analitik yang diuji. Dalam interpretasi *Green Computing*, durasi eksekusi berkaitan langsung dengan lamanya pemanfaatan sumber daya komputasi, sehingga proses yang lebih lama berpotensi meningkatkan akumulasi konsumsi energi, terutama pada beban kerja yang dominan *CPU*.

Perbedaan waktu eksekusi yang signifikan terjadi karena RStudio memanfaatkan operasi tervektorisasi yang sangat teroptimasi untuk manipulasi data tabular, meminimalkan *loop* eksplisit yang lambat. Sebaliknya, Python mengalami *overhead* waktu akibat proses interpretasi dan langkah *preprocessing* tambahan yang membebani durasi eksekusi secara keseluruhan. Perbandingan waktu eksekusi ditampilkan pada Gambar 3.

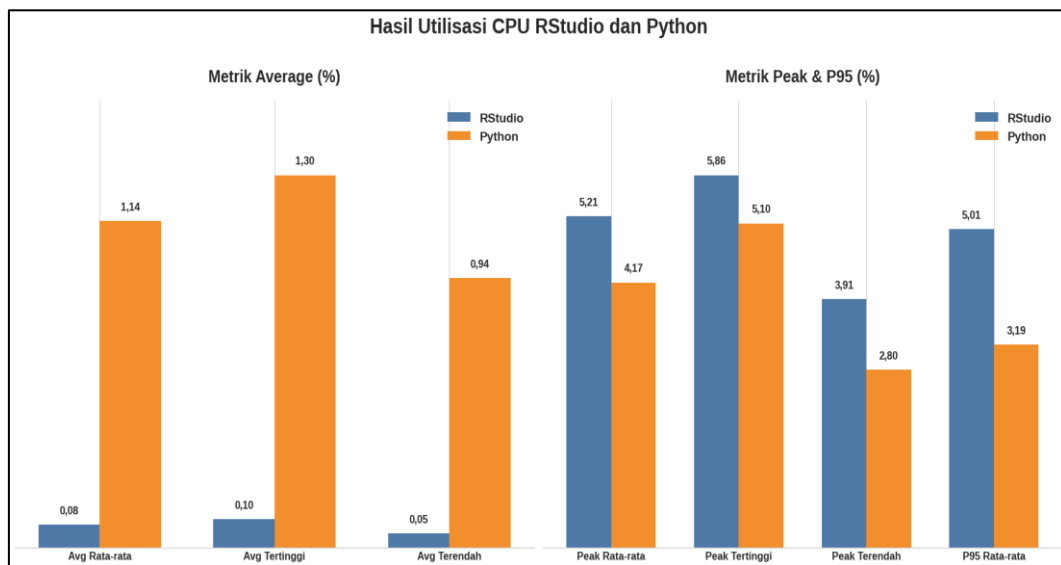


Gambar 3. Perbandingan waktu eksekusi antara RStudio dan Python dalam satuan milidetik (ms) selama tahapan *pipeline* analitik

Hasil pengukuran menunjukkan utilisasi *CPU* rata-rata pada RStudio sebesar 0,08 %, sedangkan Python sebesar 1,14 %. Perbedaan nilai rata-rata mengindikasikan variasi intensitas beban prosesor selama *pipeline* berjalan. Metrik *CPU* pada penelitian ini dipahami sebagai indikator intensitas yang melengkapi metrik waktu eksekusi dan konsumsi energi, sehingga interpretasinya perlu dilihat bersama durasi proses yang terjadi.

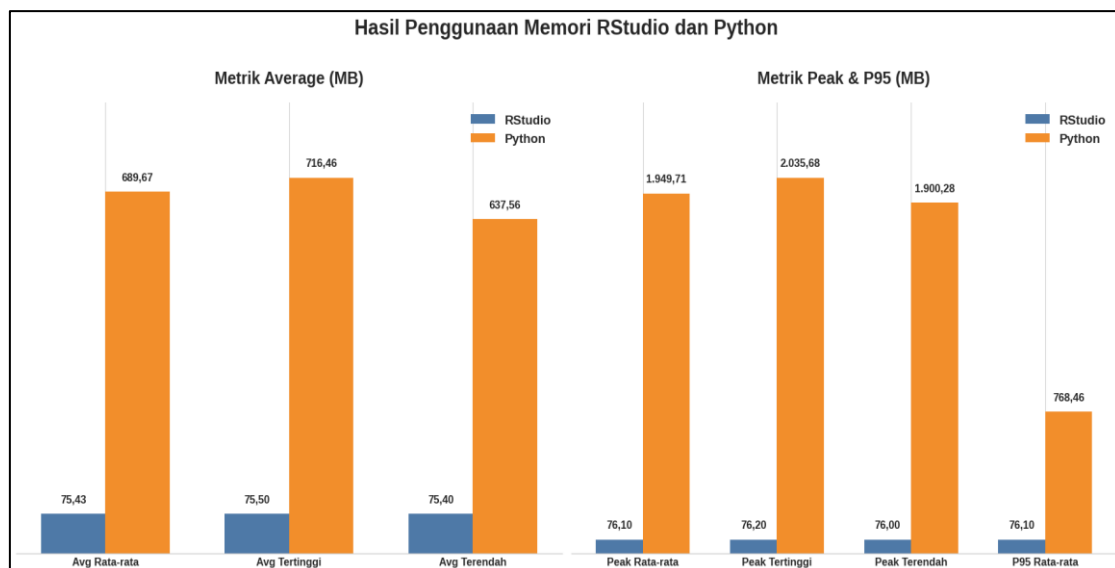
Tingginya utilisasi *CPU* pada Python dibandingkan RStudio mengindikasikan intensitas komputasi yang lebih besar. Hal ini disebabkan oleh mekanisme internal pustaka *pandas* yang sering kali melibatkan pembuatan objek sementara selama proses transformasi data, yang memaksa prosesor bekerja lebih keras dibandingkan mekanisme *in-place* yang lebih efisien pada R. Perbandingan utilisasi *CPU* ditunjukkan pada Gambar 4.

Penggunaan memori proses pada RStudio sebesar 75,43 MB, sedangkan Python 689,67 MB. Perbedaan ini mencerminkan kebutuhan RAM pada level proses selama *pipeline* analitik berlangsung, yang dapat dipengaruhi oleh representasi *dataframe*, pembentukan objek antara, serta strategi alokasi memori pada pustaka/engine yang digunakan. Dalam kerangka *Green Computing*, kebutuhan memori yang tinggi berpotensi meningkatkan *overhead* sistem dan menurunkan performa komputasi pada perangkat dengan sumber daya terbatas.



Gambar 4. Perbandingan tingkat utilisasi CPU antara RStudio dan Python dalam persentase (%) berdasarkan metrik rata-rata, puncak (peak), dan persentil ke-95 (P95)

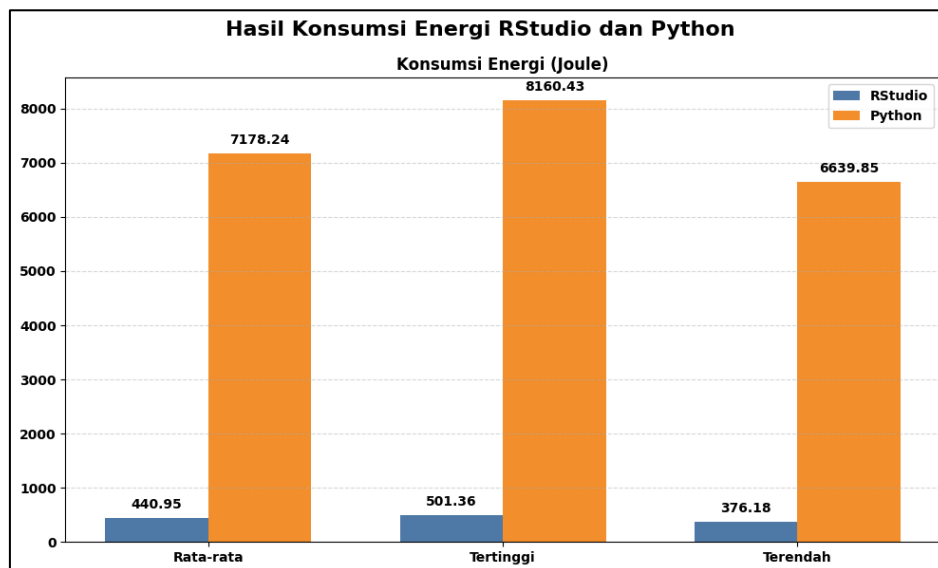
Lonjakan penggunaan memori pada *Python* mencerminkan *overhead* dari struktur data *pandas* yang memuat seluruh *dataframe* ke dalam RAM beserta metadata tambahan. Berbeda dengan *R* yang memiliki manajemen memori efisien untuk vektor dan matriks, *Python* membutuhkan alokasi memori yang jauh lebih besar untuk menangani volume data yang sama, terutama saat melakukan operasi penyalinan data selama tahap *cleaning*. Perbandingan penggunaan memori dapat dilihat pada Gambar 5.



Gambar 5. Perbandingan tingkat penggunaan memori proses antara RStudio dan Python dalam satuan megabyte (MB) berdasarkan metrik rata-rata, puncak (peak), dan persentil ke-95 (P95)

Konsumsi energi pada *RStudio* menunjukkan rata-rata 440,95 J, sedangkan *Python* 7.178,24 J. Secara umum, energi kumulatif dipengaruhi oleh kombinasi durasi eksekusi dan intensitas beban komputasi platform yang membutuhkan durasi lebih panjang dan mempertahankan beban *CPU* lebih tinggi cenderung menghasilkan konsumsi energi lebih besar.

Konsumsi energi merupakan fungsi dari durasi waktu dan intensitas beban *CPU*. Karena *Python* membutuhkan waktu eksekusi yang jauh lebih lama dengan beban *CPU* yang lebih tinggi, akumulasi energi yang dikonsumsi menjadi sangat besar. Ini membuktikan prinsip *Green Computing* bahwa efisiensi kode dan kecepatan eksekusi adalah kunci utama dalam penghematan energi. Perbandingan konsumsi energi dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan total konsumsi energi antara *RStudio* dan *Python* dalam satuan *joule* (J) berdasarkan nilai rata-rata, tertinggi, dan terendah

Secara keseluruhan, hasil menunjukkan pola konsisten bahwa metrik durasi eksekusi serta pemanfaatan sumber daya (*CPU*, memori, dan energi) pada *Python* lebih besar dibanding *RStudio* pada skenario uji penelitian ini. Temuan ini perlu dipahami sebagai bukti empiris pada rancangan *pipeline* yang diuji, sehingga tidak diarahkan sebagai klaim universal bahwa satu platform selalu unggul pada seluruh kondisi. Temuan penelitian ini juga dibandingkan dengan beberapa kajian sebelumnya untuk menilai konsistensi pola hasil pada konteks yang berbeda. Ringkasan hubungan antar penelitian disajikan pada Tabel 3.

Tabel 3. Perbandingan Hasil dengan Penelitian Sebelumnya

Penelitian	Platform/Library	Waktu Eksekusi (ms)	Utilisasi CPU (%)	Penggunaan Memori (MB)	Konsumsi Energi (J)
[8]	RStudio vs Python	41,8 – 1,0	—	—	—
[7]	Python (polars vs pandas)	81.840 – 169.690	1.147,00 - 282,60	3,14% - 3,42%	1.382,10 – 2.179,80
[9]	RStudio vs Python	83,06 – 10,13	—	460 – 315	—
[10]	RStudio vs Python	2.037 – 5.591	—	123,583 - 125,461	—
[11]	RStudio vs Python	47,6 – 7.800	—	—	46,34 - 671,54
[Penelitian ini, 2025]	RStudio vs Python	19.916,67 – 270.447,73	0,08 - 1,14	76,10 - 689,67	440,95 - 7.178,24

Tabel 2 menunjukkan bahwa temuan komparatif pada penelitian terdahulu tidak selalu memberikan arah hasil yang seragam. Variasi tersebut umumnya dipengaruhi oleh perbedaan rancangan uji, metrik yang dilaporkan, jenis operasi yang dominan, serta konfigurasi pustaka/engine dan lingkungan pengujian. Oleh karena itu, temuan penelitian ini diposisikan sebagai bukti empiris pada skenario uji yang diterapkan, bukan sebagai klaim universal keunggulan satu platform pada seluruh kondisi.

Berdasarkan analisis performa dan karakteristik ekosistem, pemilihan platform analitik tidak dapat dilakukan secara tunggal, melainkan melibatkan kompromi (*trade-off*) antara efisiensi sumber daya dan fleksibilitas pengembangan. Ringkasan karakteristik performa dan implikasi pemilihan platform berdasarkan temuan penelitian ini dirangkum pada Tabel 4.

Tabel 4. Ringkasan Performa *Green Computing* dan Analisis *Trade-Off*

Aspek	<i>R</i> (melalui <i>RStudio</i>)	<i>Python</i>	Implikasi Pemilihan
Performa dan efisiensi pada skenario uji	Cenderung lebih efisien pada waktu dan energi	Kebutuhan sumber daya lebih tinggi pada sebagian metrik	Relevan untuk perangkat terbatas dan target efisiensi energi
Pemanfaatan memori proses	Lebih rendah dan relatif stabil	Berpotensi mengalami lonjakan penggunaan memori	Memerlukan pengendalian/memonitor memori pada <i>pipeline</i> kompleks
Fleksibilitas ekosistem	Kuat untuk analisis statistik dan pengolahan data tabular	Ekosistem luas	Strategi untuk <i>pipeline end-to-end</i> dan integrasi sistem
Orientasi keputusan	Efisiensi sumber daya	Fleksibilitas implementasi	Ditentukan oleh prioritas proyek dan karakteristik <i>workload</i>

4. KESIMPULAN

Penelitian ini membandingkan performa penggunaan sumber daya komputasi antara *R* (melalui *RStudio*) dan *Python* pada *big data* analitik berbasis *Green Computing* melalui empat metrik, yaitu waktu eksekusi, utilisasi *CPU*, penggunaan memori proses, dan konsumsi energi. Pada skenario uji penelitian ini, *RStudio* mencatat waktu eksekusi 19.916,67 ms, utilisasi *CPU* 0,08 %, memori 75,43 MB, dan energi 440,95 J, sedangkan *Python* mencatat 270.447,73 ms, utilisasi *CPU* 1,14 %, memori 689,67 MB, dan energi 7.178,24 J.

Temuan tersebut menunjukkan bahwa perbedaan performa tidak hanya dipengaruhi oleh platform, tetapi karakteristik *workload* dan desain *pipeline* analitik yang diterapkan. Dalam kerangka *Green Computing*, evaluasi performa perlu mempertimbangkan kebutuhan sumber daya dan energi pada beban kerja yang diuji, serta *trade-off* terhadap kebutuhan integrasi ekosistem. Dengan demikian, pemilihan platform sebaiknya ditentukan oleh konteks implementasi dan rancangan *pipeline*. Hasil penelitian ini tidak dimaksudkan sebagai klaim universal, melainkan bukti empiris pada skenario pengujian yang digunakan.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Jurusan Informatika Fakultas Teknik Universitas Siliwangi atas dukungan dalam penelitian ini, serta kepada tim dosen pembimbing yang telah memberikan arahan selama proses penelitian dan penulisan artikel ilmiah ini.

DAFTAR PUSTAKA

- [1] S. Ahmed, M. Wardat, H. Bagheri, B. D. Cruz, dan H. Rajan, "Characterizing Bugs in Python and R Data Analytics Programs," Jun 2023, [Daring]. Tersedia pada: <http://arxiv.org/abs/2306.08632>
- [2] B. A. Rick J. Scavetta, *Python and R for the Modern Data Scientist*, vol. 103, no. Book Review 2. Foundation for Open Access Statistic, 2022. doi: 10.18637/jss.v103.b02.

- [3] M. Sneha, A. Arya, dan P. Agarwal, “Big Data Analysis and Machine Learning for Green Computing: Concepts and Applications,” dalam *Big Data Analysis for Green Computing: Concepts and Applications*, CRC Press, 2021, hlm. 91–111. doi: 10.1201/9781003032328-7.
- [4] S. G. Paul dkk., “A Comprehensive Review of Green Computing: Past, Present, and Future Research,” 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2023.3304332.
- [5] K. Batko dan A. Ślęzak, “The use of Big Data Analytics in healthcare,” *J. Big Data*, vol. 9, no. 1, Des 2022, doi: 10.1186/s40537-021-00553-4.
- [6] B. R. Maheshwari, “Comparative Analysis of Program Execution Time Required by Python, R and Julia Compiler,” *Indian Journal of Computer Science and Technology*, 2024, doi: 10.59256/indjst.20240301002.
- [7] F. Nahrstedt, M. Karmouche, K. Bargiel, P. Banijamali, A. Nalini Pradeep Kumar, dan I. Malavolta, “An Empirical Study on the Energy Usage and Performance of Pandas and Polars Data Analysis Python Libraries,” dalam *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun 2024, hlm. 58–68. doi: 10.1145/3661167.3661203.
- [8] A. H. Gandomi, F. Chen, dan L. Abualigah, “Big Data Analytics Using Artificial Intelligence,” *Electronics 2023, Vol. 12, Page 957*, vol. 12, no. 4, hlm. 957, Feb 2023, doi: 10.3390/ELECTRONICS12040957.
- [9] L. Sakka dkk., “Simple (yet Efficient) Function Authoring for Vectorized Engines,” vol. 17, no. 12, hlm. 2150–8097, 2024, doi: 10.14778/3685800.3685836.
- [10] P. Pedreira dkk., “Velox: Meta’s Unified Execution Engine”, doi: 10.14778/3554821.3554829.
- [11] Y. Zhou dkk., “On the Opportunities of Green Computing: A Survey,” Nov 2023, [Daring]. Tersedia pada: <http://arxiv.org/abs/2311.00447>
- [12] P. Chavan, A. Kadam, dan K. Mistry, “Optimizing Software Development: A Comparative Study of Python and R for Enhanced Efficiency and Quality,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 10, hlm. 1969–1974, Okt 2023, doi: 10.22214/ijraset.2023.56354.
- [13] R. Schneider, F. Machado, C. Trois, G. Descovi, V. Maran, dan A. Machado, “Speeding Up the Simulation Animals Diseases Spread: A Study Case on R and Python Performance in PDSA-RS Platform,” dalam *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, Science and Technology Publications, Lda, 2024, hlm. 651–658. doi: 10.5220/0012556200003690.
- [14] S. H. S. Herho dan S. N. Kaban, “Quantitative Performance Analysis of Spring-Mass-Damper Control Systems: A Comparative Implementation in Python and R,” *Original Article Indonesian Journal of Applied Mathematics*, vol. 05, no. 01, hlm. 10–26, 2025, doi: 10.35472/indoja.
- [15] R. Chattaraj, S. Chimalakonda, V. S. Sharma, dan V. Kaulgud, “Who Wins the Race? (R Vs Python) - An Exploratory Study on Energy Consumption of Machine Learning Algorithms,” Agu 2025, [Daring]. Tersedia pada: <http://arxiv.org/abs/2508.17344>