Comparative Analysis of Web Browser Performance and Security: A Study of Google Chrome and Mozilla Firefox

Valentino angga[a], Rui Cahya[b], Davin Yapiter[c], Achmad Ridwan[d]
[a,b,c,d]*Universitas Prima Indonesia*
Corresponding Author:
[a]*Valentinoangga222@gmail.com*

ABSTRACT
This study presents a comparative analysis of the performance and security of Google Chrome (v147.0.7727.138) and Mozilla Firefox (v150.0.1) through empirical testing on an AMD Ryzen 5 4600H device with 16 GB DDR4 RAM running Windows 11 64-bit. Each test scenario was repeated five times, with results analyzed using standard deviation and 95% margin of error. Speedometer 3.1 and JetStream 2.0 were used as benchmark instruments. Results show Chrome outperforms Firefox on Speedometer 3.1 (19.72 ± 0.243 vs. 15.53 ± 0.143; 26.95% difference) and JetStream 2.0 (252.553 ± 2.339 vs. 168.590 ± 2.717; 49.80% difference). Firefox demonstrates superior resource efficiency in RAM (2,451.9 MB vs. 2,533.5 MB) and CPU usage (17.2% vs. 19.6%). From a security standpoint, Firefox shows a stronger commitment to user privacy through its Enhanced Tracking Protection. Browser selection ultimately depends on whether users prioritize computational performance or resource efficiency and privacy.

Keywords : Web Browser, Google Chrome, Mozilla Firefox, Performance, Comparative Analysis

INTRODUCTION

Web browsers have become one of the most essential software applications in modern digital life. Serving as the primary interface between users and internet services, browsers function not merely as web page viewers but also as platforms for executing complex web-based applications. In this context, browser performance and security represent two critical dimensions that directly affect user productivity, data security, and overall experience (Google Inc., 2026; Mozilla Foundation, 2026). Google Chrome, launched by Google in 2008, has dominated the global browser market with a share exceeding 60%. Its Chromium-based multi-process architecture enables individual tab isolation, preventing a crash in one tab from affecting others (Google Inc., 2026). Mozilla Firefox, as the oldest actively developed open-source browser, offers an alternative emphasizing user privacy, code transparency, and resource efficiency through its Quantum engine (Mozilla Foundation, 2026). The comparison between these two browsers is particularly relevant given their fundamentally different development philosophies. Chrome is developed by a corporation whose business model relies on user data, whereas Firefox is developed by the Mozilla Foundation — a non-profit organization placing privacy and internet freedom at the core of its mission (Madhusudhan & Surashe, 2022). These differences manifest in technical decisions ranging from third-party cookie handling to built-in security feature

implementation. Previous studies have examined specific aspects of browser comparisons, yet comprehensive investigations based on primary experimental data with statistical validation remain limited (Herwanto, 2025; Mohamed & Ismail, 2022). The purpose of this study is to address that gap through direct empirical testing supplemented by statistical analysis, providing data-driven insights for both end users and system developers. The specific research objectives are: (1) to evaluate the performance of both browsers through statistically validated benchmark testing; (2) to analyze system resource consumption efficiency; (3) to compare security and privacy mechanisms; and (4) to provide evidence-based recommendations for browser selection.

LITERATURE REVIEW

Modern Browser Architecture

Browser architecture has evolved from the single-process model to a more secure and reliable multi-process model. Google Chrome implements a process-per-site-instance architecture in which each tab, extension, and browser component runs within a separate operating system process, ensuring that a crash in one tab does not propagate to the entire browser while providing additional security isolation through sandboxing (Google Inc., 2026). Mozilla Firefox adopts the Electrolysis (e10s) model, dividing processes into a main browser component and several content processes, enabling more efficient workload distribution and resulting in lower RAM consumption (Mozilla Foundation, 2026; Herwanto, 2025).

From a rendering engine perspective, Chrome uses Blink while Firefox uses Gecko. This fundamental difference influences how web pages are processed, including support for the latest web standards and rendering performance optimization (Ollila et al., 2022). Ollila et al. (2022) further demonstrated that differences in rendering engine implementation affect web application performance across browsers differently depending on the JavaScript framework used.

Browser Performance and Benchmarks

Heričko et al. (2021) emphasize the importance of representative benchmark methodologies, noting that test results can vary significantly depending on network conditions, hardware specifications, and the type of content accessed. The researchers identified that no single benchmark tool can comprehensively represent all real-world performance scenarios, advocating for multi-tool measurement approaches — an approach adopted in this study.

Speedometer 3.1 measures web application responsiveness through simulations of real user interactions across multiple JavaScript frameworks (WebKit, 2026a), while JetStream 2.0 evaluates JavaScript engine performance through comprehensive tests including WebAssembly operations and memory management scenarios (WebKit, 2026b). Mohamed and Ismail (2022) found that browser performance can differ substantially between desktop and mobile platforms, underscoring the importance of specifying the testing environment. Van Riet et al. (2023) further affirm that web performance optimization is multidimensional and cannot be reduced to a single metric.

Browser Security and Privacy

Madhusudhan and Surashe (2022) conducted a comprehensive review identifying that both Chrome and Firefox implement equivalent core security features, yet significant philosophical differences exist in their approach to user data handling and privacy. Aslan et al. (2023) highlight that browsers represent a significant attack vector in the modern cybersecurity ecosystem, where mechanisms such as Content Security Policy (CSP), HTTP Strict Transport Security (HSTS), and sandboxing constitute critical defense layers.

Shahid et al. (2022) identified that effective sandboxing, automatic security updates, and protection against common cyberattacks are essential security parameters for modern browsers. Pan and Ruiz-Martinez (2023) specifically examined the evolution of tracking protection in Chrome, finding that while Google has committed to improving privacy through its Privacy Sandbox initiative, the actual implementation remains a subject of ongoing debate among privacy researchers. Firefox, by contrast, has taken a more aggressive approach through its Enhanced Tracking Protection (ETP), which by default blocks cross-site trackers, third-party cookies in strict mode, and fingerprinters — a mechanism Aslan et al. (2023) identify as increasingly critical given the rise of fingerprinting as an alternative to cookie-based tracking.

METHODS

Research Design

This study employs a mixed-method approach integrating direct experimental testing (primary quantitative data) with a systematic literature review. The quantitative component follows a controlled experimental design, where browser performance is measured under identical conditions to ensure internal validity and replicability (American Psychological Association, 2020). The literature review component adheres to systematic review guidelines, focusing on peer-reviewed publications from the past 10 years retrieved from IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Google Scholar.

Testing Environment and Sampling

Performance testing was conducted on a single workstation to eliminate inter-device variability. The device specifications are presented in Table 1. To ensure consistency, the following protocol was strictly enforced: (1) all non-essential applications were closed prior to each test session; (2) internet connectivity was maintained stable throughout; (3) each benchmark scenario was repeated five times per browser — a repetition count selected in accordance with the statistical recommendation for small-sample comparative studies, in which five or more repeated measurements are considered sufficient to estimate the population mean with acceptable precision when supported by standard error and 95% confidence interval analysis (Heričko et al., 2021; Mohamed & Ismail, 2022); and (4) the arithmetic mean of the five repetitions was used as the final reported value.

Table 1. Testing Device Specifications

Component	Specification
Processor	AMD Ryzen 5 4600H with Radeon Graphics
RAM	DDR4 16 GB
Storage	SSD 512 GB
Operating System	Windows 11 64-bit
Google Chrome Version	147.0.7727.138 (Official Build) (64-bit)
Mozilla Firefox Version	150.0.1 (64-bit)

Source: Authors' testing setup (2026)

Data Collection

Resource consumption data (RAM and CPU usage) were measured using Windows Task Manager under identical five-tab browsing conditions. Specifically, after opening five tabs with equivalent web content on each browser and allowing a 60-second stabilization period for background processes to settle, RAM and CPU values were recorded once per session. This single-point measurement was repeated across five independent sessions per browser, and the values reported in Table 4 represent the average of these five observations. This protocol was designed to ensure that resource readings reflected stable, steady-state conditions rather than transient startup activity, thereby supporting measurement replicability.

Statistical Analysis

To enhance result validity, statistical analysis was applied to the five-repetition benchmark data. The following statistics were computed: arithmetic mean ($\bar{x}$), sample standard deviation (s), standard error of the mean ($SE = s/\sqrt{n}$), and 95% confidence margin of error ($MoE = 1.96 \times SE$). The coefficient of variation ($CV = s/\bar{x} \times 100\%$) served as the primary reliability indicator, with $CV < 5\%$ denoting acceptably consistent results not significantly influenced by systemic disturbances (Heričko et al., 2021).

Security Evaluation Framework

Security evaluation was conducted through a structured comparative analysis framework adapted from Madhusudhan and Surashe (2022) and Shahid et al. (2022), encompassing six dimensions: (1) process sandbox and isolation mechanisms; (2) automatic security update policies; (3) phishing and malware protection systems; (4) Content Security Policy (CSP) implementation; (5) cross-site cookie and tracking management; and (6) user data privacy policies and transparency.

RESULTS

This section presents the empirical results obtained from the benchmark testing and security evaluation of Google Chrome and Mozilla Firefox. All quantitative data were collected under controlled conditions across five repeated trials. Statistical indicators — including mean ($\bar{x}$), standard deviation (s), 95% margin of error (MoE), and coefficient of variation (CV) — are

reported for each benchmark instrument to substantiate result reliability. Tables and figures are presented as detached elements to facilitate clear visual inspection of findings.

Speedometer 3.1 Results

Table 2 and Figure 1 present the complete Speedometer 3.1 results for both browsers across all five trials. Speedometer 3.1 simulates real user interactions with web applications built on popular JavaScript frameworks, where a higher score indicates greater UI responsiveness.

Table 2. Speedometer 3.1 Test Results

Trial	Google Chrome	Mozilla Firefox
1	19.30	15.30
2	19.60	15.50
3	19.90	15.75
4	20.00	15.60
5	19.80	15.52
Mean (x̄)	**19.72**	**15.53**
Std. Deviation (s)	**0.277**	**0.164**
Margin of Error (95%)	**±0.243**	**±0.143**
Coeff. of Variation (CV)	**1.41%**	**1.05%**

Source: Authors' direct testing (2026)

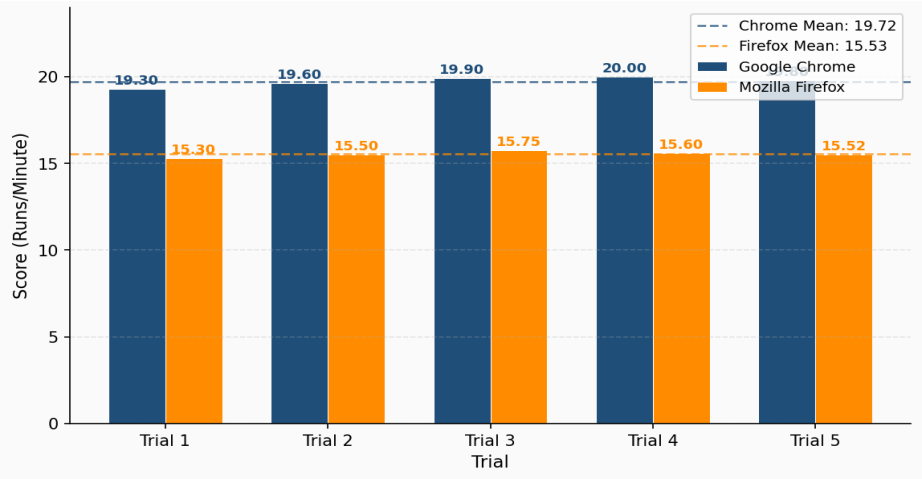


Figure 1. Speedometer 3.1 Per-Trial Score Comparison

Google Chrome consistently achieved higher scores in every trial, yielding a mean of 19.72 compared to Firefox's 15.53 — a performance difference of 26.95%. The low CV values (Chrome: 1.41%; Firefox: 1.05%), both well below the 5% threshold, confirm that results are consistent and reliable across repetitions. The non-overlapping 95% confidence intervals (Chrome: 19.72 ± 0.243; Firefox: 15.53 ± 0.143) provide an indication that the observed difference is unlikely to be explained by measurement variance alone; however, non-overlapping confidence intervals are not equivalent to a formal test of statistical significance. No inferential statistical test (e.g., independent-samples t-test or Mann-Whitney U test) was conducted in this study due to the small and fixed sample size (n = 5 per browser), which limits the statistical power of such tests. Accordingly, the performance difference between the two browsers is reported as a descriptively substantial and consistent gap based on the observed measurements, rather than a formally statistically significant finding. Future studies with larger sample sizes are encouraged to apply formal inferential tests to validate this pattern.

JetStream 2.0 Results

Table 3 and Figure 2 present the JetStream 2.0 results, which evaluate JavaScript engine performance under computationally intensive workloads including WebAssembly operations and advanced memory management scenarios.

Table 3. JetStream 2.0 Test Results

Trial	Google Chrome	Mozilla Firefox
1	256.375	173.686
2	252.620	167.940
3	253.248	167.908
4	249.060	165.209
5	251.463	168.208
Mean ($\bar{x}$)	**252.553**	**168.590**
Std. Deviation (s)	**2.669**	**3.100**
Margin of Error (95%)	**±2.339**	**±2.717**
Coeff. of Variation (CV)	**1.06%**	**1.84%**

Source: Authors' direct testing (2026)

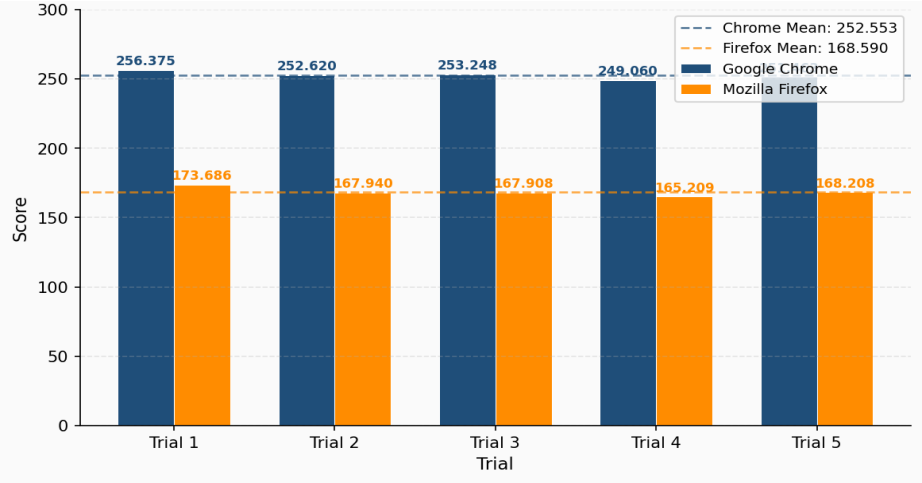


Figure 2. JetStream 2.0 Per-Trial Score Comparison

Chrome's computational advantage is considerably more pronounced on JetStream 2.0, achieving a mean of 252.553 versus Firefox's 168.590 — a 49.80% difference amounting to 83.963 points. CV values of 1.06% and 1.84% confirm high result reliability across trials. The non-overlapping 95% confidence intervals (Chrome: 252.553 ± 2.339; Firefox: 168.590 ± 2.717) further indicate that the observed gap is unlikely to be attributable to measurement variance. As with Speedometer 3.1, no formal inferential test was conducted; the difference is therefore characterized as a descriptively large and consistent performance gap under the conditions tested.

System Resource Consumption

Table 4 and Figure 3 report RAM and CPU consumption recorded under five simultaneously active browser tabs with equivalent web content on both browsers.

Table 4. System Resource Consumption Results

Parameter	Google Chrome	Mozilla Firefox
RAM Usage (5 active tabs)	2,533.5 MB	2,451.9 MB
CPU Usage (average)	19.6%	17.2%
RAM Difference	81.6 MB	Firefox saves 3.22%
CPU Difference	2.4 percentage points	Firefox lower by 12.24%

Source: Authors' direct testing (2026)

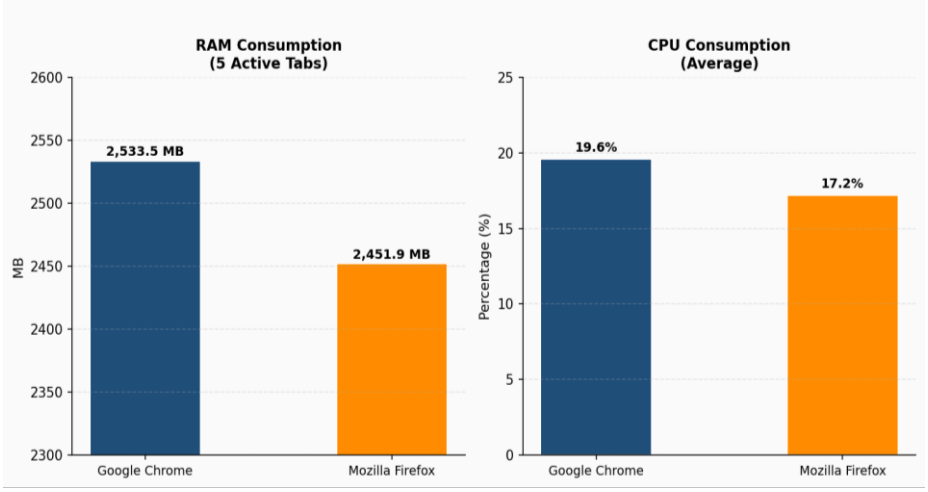


Figure 3. System Resource Consumption Comparison

Firefox consumed 2,451.9 MB of RAM across five active tabs, saving 81.6 MB (3.22%) relative to Chrome's 2,533.5 MB. Firefox also recorded lower average CPU utilization at 17.2% compared to Chrome's 19.6% — a difference of 2.4 percentage points (12.24%). These findings demonstrate Firefox's consistent advantage in system resource efficiency, attributable to its content process sharing architecture (Mozilla Foundation, 2026).

Security Feature Comparison

Table 5 presents the structured security feature comparison evaluated across the six-dimension framework. Figure 4 illustrates the overall testing and evaluation workflow applied throughout this study, from environment setup through to final comparative analysis.

Table 5. Browser Security Features Comparison

Security Feature	Google Chrome	Mozilla Firefox
Process Sandboxing	Multi-level sandbox	Content sandbox
Automatic Updates	Yes (silent)	Yes (with notification)
Anti-Phishing Protection	Google Safe Browsing	Google Safe Browsing
Third-Party Cookie Blocking	Incognito mode only	Default (strict mode)
Enhanced Tracking Protection	Limited	Full (ETP) ✓
DNS over HTTPS	Yes (optional)	Yes (default in some regions)
Content Security Policy	Yes	Yes
HTTPS-Only Mode	HTTPS-First (optional)	HTTPS-Only (optional)
Source Code Transparency	Chromium (partial)	Fully open-source ✓

Security Feature	Google Chrome	Mozilla Firefox
Default Telemetry Collection	Active (can be disabled)	Minimal (can be disabled)

Source: Adapted from Madhusudhan & Surashe (2022); Pan & Ruiz-Martinez (2023); Shahid et al. (2022); Aslan et al. (2023)

Both browsers implement equivalent core security mechanisms, including process sandboxing, automatic updates, and anti-phishing protection through Google Safe Browsing. However, Firefox demonstrates a more proactive default privacy posture: its Enhanced Tracking Protection (ETP) is active by default, third-party cookies are blocked in strict mode without user configuration, and telemetry data collection is minimal by default. Chrome, while offering comparable technical security infrastructure, defaults to a more data-permissive configuration (Madhusudhan & Surashe, 2022; Pan & Ruiz-Martinez, 2023).

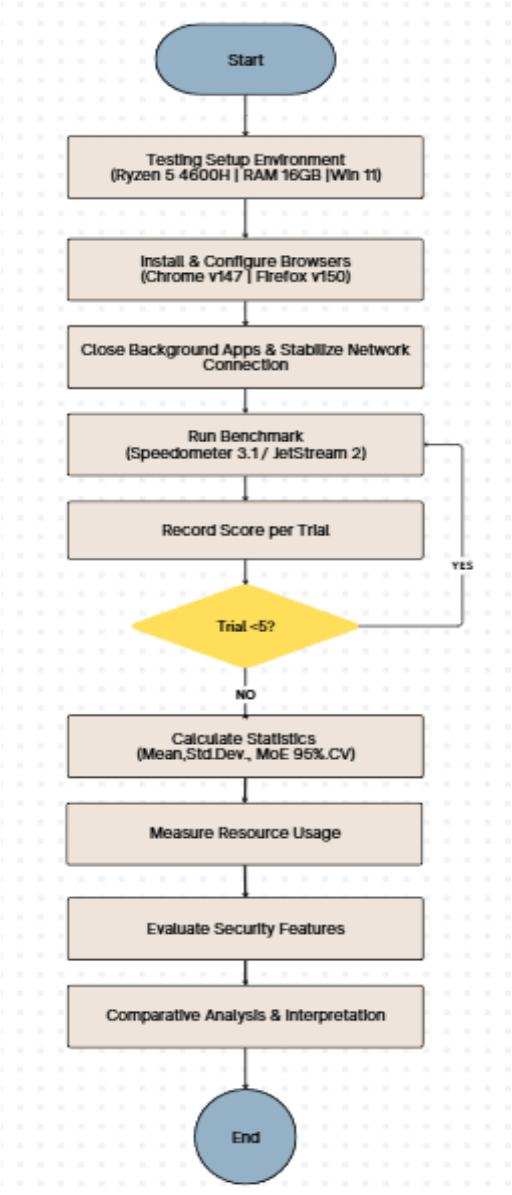


Figure 4. Research Testing Flowchart

Figure 4 summarizes the complete end-to-end testing procedure applied in this study. The flowchart captures the sequential steps from environment setup, browser installation, background application closure, and iterative benchmark execution (with the five-repetition decision loop explicitly shown), through to statistical computation, resource measurement, security evaluation, and final comparative analysis compilation.

Overall Comparative Summary

Table 6 consolidates all evaluated dimensions into a single comparative summary, enabling direct comparison across performance, resource efficiency, and security categories.

Table 6. Overall Comparative Summary: Google Chrome vs. Mozilla Firefox

Dimension	Google Chrome	Mozilla Firefox
Speedometer 3.1 ($\bar{x}$ ± MoE)	19.72 ± 0.243 ✓	15.53 ± 0.143
JetStream 2.0 ($\bar{x}$ ± MoE)	252.553 ± 2.339 ✓	168.590 ± 2.717
RAM Consumption (5 tabs)	2,533.5 MB	2,451.9 MB ✓
CPU Consumption	19.6%	17.2% ✓
Privacy & Anti-Tracking (default)	Limited	Full (ETP) ✓
Source Code Transparency	Partial (Chromium)	Full ✓
Ecosystem & Integration	✓ Google (extensive)	Independent & open

Source: Authors' research analysis (2026)

As summarized in Table 6, Chrome holds a clear advantage in both performance benchmarks, while Firefox leads in RAM efficiency, CPU efficiency, privacy protection, and source code transparency. These contrasting strengths underscore that no single browser is comprehensively superior across all evaluated dimensions, and that selection should be guided by the user's dominant priority.

DISCUSSION

The primary purpose of this study was to comparatively evaluate the performance and security of Google Chrome and Mozilla Firefox through direct empirical measurement. The findings confirm that neither browser is universally superior — each demonstrates distinct advantages aligned with different user priorities.

Chrome's consistent and descriptively large computational performance advantage — 26.95% on Speedometer 3.1 and 49.80% on JetStream 2.0 — is consistent with findings reported by Herwanto (2025) and Mohamed and Ismail (2022), and can be attributed to the aggressive just-in-time (JIT) compilation optimizations of the V8 engine and the Blink rendering architecture. This mechanism is extensively documented in the Chromium project's technical architecture documentation (Google Inc., 2026) and corroborated by prior research on trace-

based JIT compilation strategies, which established that aggressive JIT approaches yield substantially higher throughput in JavaScript-intensive workloads (Gal et al., 2009). These results carry practical implications particularly for web developers, application testers, and users who rely heavily on JavaScript-intensive web applications, for whom Chrome's computational advantage may translate into perceptibly improved responsiveness in daily use. Ollila et al. (2022) similarly noted that Chrome's rendering advantages are most pronounced in computationally demanding framework-based applications.

Firefox's advantages in resource efficiency — 3.22% less RAM and 12.24% lower CPU usage — though seemingly modest in absolute terms, carry important practical implications for long-term multi-tab usage and for devices with constrained hardware resources. Van Riet et al. (2023) affirm that resource efficiency is a critical and often undervalued dimension of browser performance in real-world usage scenarios. The privacy dimension further amplifies Firefox's value proposition: its Enhanced Tracking Protection, which by default blocks cross-site trackers and fingerprinters, represents a meaningfully higher baseline privacy standard than Chrome's default configuration. Pan and Ruiz-Martinez (2023) note that Chrome's ongoing Privacy Sandbox transition, while promising, has not yet delivered comparable default protection.

An unexpected finding relates to CPU usage, where Firefox — despite its lower JavaScript benchmark scores — consumed less CPU than Chrome during idle multi-tab browsing. This suggests that Chrome's performance advantage in isolated computation tasks comes at the cost of higher background resource consumption, a trade-off that may not be desirable for all usage scenarios. This finding diverges somewhat from the assumption that higher computational performance necessarily implies better overall resource efficiency (Van Riet et al., 2023).

From a security standpoint, both browsers meet baseline enterprise security standards. However, Firefox's open-source model provides a transparency advantage that Chrome's partial open-source approach (Chromium) does not fully replicate, as noted by Madhusudhan and Surashe (2022). For organizations and individuals for whom auditability and minimal data collection are priorities, this distinction carries practical significance.

The managerial and practical implication of these findings is that browser selection should be treated as a context-dependent decision rather than a universal recommendation. Organizations deploying browsers at scale should consider their dominant workload type, hardware constraints, and privacy policy requirements when standardizing on a browser.

CONCLUSION

This study provided an empirical comparison of Google Chrome v147.0.7727.138 and Mozilla Firefox v150.0.1 across performance, resource efficiency, and security dimensions, supported by descriptive statistical analysis including mean, standard deviation, and 95% confidence intervals. Chrome demonstrated consistent and descriptively substantial computational superiority — 26.95% on Speedometer 3.1 and 49.80% on JetStream 2.0 — driven by its optimized V8 engine. As no formal inferential statistical tests were conducted, these differences are characterized as descriptively large and consistent performance gaps rather than formally statistically significant findings. Firefox demonstrated superior system

resource efficiency, consuming 3.22% less RAM and 12.24% less CPU, alongside a stronger default commitment to user privacy through Enhanced Tracking Protection. Both browsers implement equivalent core security mechanisms. Browser selection should be guided by user-specific priorities: Chrome for performance-first and Google ecosystem-integrated contexts; Firefox for privacy-first and resource-constrained contexts. Future research should extend this investigation to mobile platforms, multiple operating systems, varying network conditions, and battery consumption analysis, and should incorporate formal inferential statistical tests to strengthen the generalizability of comparative conclusions.

LIMITATION

This study is subject to several limitations that should be considered when interpreting its findings. First, testing was conducted on a single device (AMD Ryzen 5 4600H, 16 GB RAM, Windows 11 64-bit), which limits the generalizability of results to other hardware configurations and operating systems. The influence of hardware on browser performance is well-documented (Mohamed & Ismail, 2022), and results on devices with different processor architectures or memory capacities may differ. Second, network conditions were held constant and were not systematically varied; since network speed and latency directly affect page loading performance, the findings may not fully reflect real-world performance under diverse connectivity scenarios. Third, the absence of mobile platform testing means the results cannot be directly applied to Chrome and Firefox on smartphones or tablets, where resource constraints and rendering pipelines differ substantially. Fourth, RAM and CPU measurements were recorded as single-point values rather than continuous time-series data, limiting the depth of statistical analysis for these parameters relative to the five-repetition benchmark tests. Fifth, this study does not include battery power consumption measurement, an increasingly important factor given the prevalence of laptop and mobile device usage. These limitations collectively mean that the findings should be interpreted as indicative rather than definitive, and replication across diverse hardware, software, and network environments is recommended.

REFERENCES

Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics*, *12*(6), 1333. https://doi.org/10.3390/electronics12061333

Google Inc. (n.d.). *The Chromium Projects: Multi-Process Architecture*. Chromium Project. Retrieved April 29, 2026, from https://www.chromium.org

Heričko, T., Šumak, B., & Brdnik, S. (2021). Towards Representative Web Performance Measurements with Google Lighthouse. *Proceedings of the 2021 7th Student Computer Science Research Conference (StuCoSReC)*, 39–42. https://doi.org/10.18690/978-961-286-516-0.9

Herwanto, A. (2025). Analisis Perbandingan Kinerja Browser: Studi Kasus Google Chrome, Mozilla Firefox. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(3), 3545–3549. https://doi.org/10.31004/riggs.v4i3.2512

Madhusudhan, R., & Surashe, S. V. (2022). Privacy and Security Comparison of Web Browsers: A Review. In L. Barolli, F. Hussain, & T. Enokido (Eds.), *Advanced Information Networking and Applications* (Vol. 451, pp. 459–470). Springer International Publishing. https://doi.org/10.1007/978-3-030-99619-2_44

Mohamed, A., & Ismail, I. (2022). A Performance Comparative on Most Popular Internet Web Browsers. *Procedia Computer Science*, *215*, 589–597. https://doi.org/10.1016/j.procs.2022.12.061

Mozilla Foundation. (2026, April 8). *Firefox Multi-Process Architecture Documentation*. MDN Web Docs. https://developer.mozilla.org/en-US/

Ollila, R., Mäkitalo, N., & Mikkonen, T. (2022). Modern Web Frameworks: A Comparison of Rendering Performance. *Journal of Web Engineering*. https://doi.org/10.13052/jwe1540-9589.21311

Pan, R., & Ruiz-Martínez, A. (2023). Evolution of web tracking protection in Chrome. *Journal of Information Security and Applications*, *79*, 103643. https://doi.org/10.1016/j.jisa.2023.103643

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences*, *12*(8), 4077. https://doi.org/10.3390/app12084077

Van Riet, J., Malavolta, I., & Ghaleb, T. A. (2023). Optimize along the way: An industrial case study on web performance. *Journal of Systems and Software*, *198*, 111593. https://doi.org/10.1016/j.jss.2022.111593

WebKit. (n.d.-a). *JetStream 2.0*. BrowserBench. Retrieved April 30, 2026, from https://browserbench.org/JetStream2.0/

WebKit. (n.d.-b). *Speedometer 3.1*. BrowserBench. Retrieved April 30, 2026, from https://browserbench.org/Speedometer3.1/