

ANALISIS KINERJA SISTEM BERKAS LINUX PADA LINGKUNGAN DESKTOP DENGAN SUMBER DAYA TERBATAS

Raphael Jameson, Jusia Amanda Ginting
Teknik Informatika, Universitas Bunda Mulia
Jl. Lodan Raya Ancol, Jakarta Utara, Indonesia
s32220017@student.ubm.ac.id

ABSTRAK

Linux dikenal sebagai sistem operasi yang stabil, namun kinerjanya pada lingkungan *desktop* dengan sumber daya terbatas sering kali terhambat oleh overhead perangkat lunak. Inefisiensi sistem berkas pada perangkat kelas pemula menjadi hambatan utama yang menyebabkan latensi tinggi. Penelitian ini bertujuan menganalisis dan membandingkan kinerja sistem berkas Ext4, XFS, dan Btrfs guna menentukan opsi paling optimal untuk perangkat terbatas. Metodologi melibatkan benchmarking komprehensif (waktu *boot*, respons aplikasi, latensi I/O, *throughput*, dan simulasi alur kerja pengembang) pada perangkat keras AMD A4-5100 dengan RAM 8GB. Hasil menunjukkan XFS unggul dalam *throughput*, sementara Btrfs tercepat dalam waktu *boot*. Namun, Ext4 memberikan performa paling seimbang, khususnya pada efisiensi CPU dan waktu kompilasi. Implementasi melalui remastering Ubuntu 24.04 berbasis Ext4 dengan teknologi ZRAM dan lingkungan XFCE berhasil menurunkan waktu *boot* hingga 48% dan meningkatkan responsivitas aplikasi secara signifikan.

Kata kunci : Analisis Kinerja, Sistem Berkas, Linux, Ext4, XFS, Btrfs.

1. PENDAHULUAN

Sistem operasi Linux telah menjadi solusi preferensial yang diadopsi secara luas dalam berbagai sektor teknologi, mulai dari infrastruktur server berskala masif hingga pemanfaatan harian pada komputasi personal. Keberhasilan ini sebagian besar diatribusikan pada karakteristik *open source* serta fleksibilitas konfigurasi yang komprehensif [1]. Linux diakui secara global sebagai sistem yang memiliki stabilitas dan reliabilitas tinggi dalam menopang arsitektur jaringan yang kompleks, yang pada gilirannya mengaksentuasi pertumbuhan adopsi secara berkelanjutan di sektor industri teknologi informasi [2]. Relevansi fenomena tersebut semakin signifikan dalam konteks transformasi infrastruktur teknologi informasi dan komunikasi yang berkembang pesat sebagai konsekuensi dari arus globalisasi yang kian intensif [3], [4]. Perkembangan ini memberikan implikasi mendalam yang meluas ke berbagai lapisan masyarakat, menempatkan teknologi bukan sekadar sebagai instrumen pelengkap, melainkan sebagai determinan kritis dalam keberlangsungan operasional organisasi [5]. Dalam struktur ekosistem komputasi tersebut, sistem berkas memegang peranan vital sebagai perantara esensial antara instruksi sistem operasi dengan perangkat keras penyimpanan fisik. Efisiensi sistem berkas merupakan variabel fundamental yang menentukan performansi berbagai skenario aplikasi, mulai dari manajemen data hingga responsivitas antarmuka pengguna [6].

Kendatipun literatur mengenai analisis performansi sistem berkas pada infrastruktur komputasi tingkat tinggi telah terdokumentasi secara komprehensif [7], masih terdapat diskontinuitas literatur yang mengevaluasi perilaku sistem berkas pada perangkat keras dengan spesifikasi terbatas [8]. Realitas di lapangan menunjukkan bahwa sebagian

besar perangkat yang digunakan di seluruh dunia masih bersifat kelas pemula (*entry-level*) [9]. Performa operasional pada kategori perangkat ini sangat dipengaruhi oleh kendala sumber daya yang restriktif, di mana latensi interaksi antara memori utama dan media penyimpanan sering kali menjadi faktor penghambat utama yang memicu degradasi performa sistem secara signifikan, termasuk munculnya gejala latensi responsif dan ketidakefisienan dalam pemuatan data [10]. Ketidakpastian mengenai efektivitas sistem berkas standar Linux pada konfigurasi sumber daya terbatas menghadirkan tantangan bagi upaya optimalisasi siklus pakai perangkat keras lama melalui pendekatan perangkat lunak yang strategis.

Penelitian ini diartikulasikan secara spesifik untuk melakukan tinjauan komparatif mendalam terhadap tiga sistem berkas utama dalam ekosistem Linux, yaitu Ext4, XFS, dan Btrfs, dengan lokus implementasi pada perangkat desktop kategori *entry-level*. Fokus utama penelitian ini adalah mengidentifikasi konfigurasi penyimpanan yang paling optimal melalui pengukuran responsivitas sistem yang mencakup metrik inisialisasi sistem (*booting*), efisiensi kompilasi program yang intensif, serta efektivitas pengelolaan metadata [11]. Melalui dekonstruksi karakteristik teknis masing-masing sistem berkas di bawah tekanan sumber daya terbatas, penelitian ini diharapkan dapat merumuskan landasan teoretis serta memberikan rekomendasi manajerial bagi pengguna maupun pengembang distribusi Linux. Upaya ini bertujuan untuk meningkatkan efisiensi operasional dan utilitas perangkat keras dengan keterbatasan fisik, sekaligus menjadi langkah strategis dalam mendukung aspek keberlanjutan (*sustainability*) melalui reduksi limbah elektronik.

Kerangka metodologi pemecahan masalah dalam penelitian ini dimulai dengan pelaksanaan prosedur

benchmarking yang terstruktur menggunakan instrumen pengukuran standar industri, antara lain *systemd-analyze* untuk fase inisiasi, *fio* untuk analisis *input/output* mentah, serta simulasi alur kerja pengembang guna memperoleh profil data performa yang akurat dan reliabel. Data kuantitatif yang diperoleh berfungsi sebagai basis pertimbangan empiris untuk menentukan sistem berkas terbaik yang akan diadopsi dalam tahap implementasi nyata melalui teknik *remastering* sistem operasi Ubuntu 24.04. Proses optimisasi dilakukan secara integratif melalui modifikasi parameter kernel, seperti mekanisme *data writeback*, pemanfaatan teknologi ZRAM sebagai solusi kompresi memori virtual guna memitigasi keterbatasan RAM fisik, serta adopsi lingkungan desktop XFCE yang dirancang untuk meminimalkan beban sumber daya grafis. Implementasi langkah-langkah tersebut bertujuan untuk mereduksi *overhead* perangkat lunak, sehingga menghasilkan sistem operasi yang lebih responsif, stabil, dan efisien pada perangkat dengan spesifikasi terbatas.

2. TINJAUAN PUSTAKA

2.1. Lingkungan Desktop dan Sistem Operasi

Lingkungan *desktop* merujuk pada ekosistem perangkat lunak yang dirancang untuk pengguna akhir pada komputer pribadi, dicirikan oleh antarmuka pengguna grafis yang mempermudah interaksi. Sistem Operasi (OS) sendiri bertindak sebagai jembatan esensial antara pengguna dan perangkat keras, mengelola sumber daya seperti prosesor dan memori [12], [13]. *Linux*, sebagai sistem operasi mirip Unix, telah berkembang pesat dan diterapkan mulai dari server, pengembangan perangkat lunak, hingga Internet of Things [14], [15]. Inti dari *Linux* adalah kernel yang mengelola perangkat keras dan menjaga integritas sistem [16], yang dapat dikonfigurasi secara fleksibel dengan memuat modul saat *runtime* [17].

2.2. Sistem Berkas Linux

Sistem berkas menghadapi tantangan dalam mengimbangi kemajuan kapasitas SSD modern, di mana hambatan utama sering bergeser ke tumpukan I/O perangkat lunak [18]. Pendekatan baru seperti pengindeksan dalam penyimpanan diperlukan untuk mencegah degradasi performa [19]. Dalam penelitian ini, tiga sistem berkas utama dievaluasi, yaitu Ext4, XFS, dan Btrfs. Ext4 merupakan sistem berkas standar untuk sebagian besar distribusi *Linux* yang dibangun dari fondasi Ext3 dengan memperkenalkan *extent* untuk meningkatkan kinerja pada berkas besar. Stabilitasnya menjadikan Ext4 pilihan populer untuk komputasi performa tinggi [20], [21] serta dinilai lebih tangguh terhadap kegagalan, meskipun proses pemulihannya bisa memakan waktu lebih lama karena kompleksitas pemeriksaan [22]. Ext4 memiliki kinerja yang baik bahkan dalam kondisi tervirtualisasi [23] dan lebih efektif dibandingkan NTFS pada media SSD [24].

Selanjutnya, XFS adalah sistem berkas *journaling* berkinerja tinggi yang dirancang untuk lingkungan dengan beban I/O intensif [25]. XFS menggunakan manajemen ruang berbasis *allocation groups* dan mendukung sistem berkas berukuran sangat besar hingga ukuran exbibytes [26], [27]. Fitur buffer sebelum penulisan pada XFS meminimalkan inefisiensi, menjadikannya cocok untuk server basis data [28]. Sementara itu, Btrfs hadir sebagai sistem berkas modern dengan fitur *Copy-on-Write* (CoW), *snapshot*, dan *subvolumes* yang memungkinkan manajemen partisi fleksibel [6]. Meskipun menawarkan fitur integritas data yang kuat, mekanisme CoW pada Btrfs dapat menyebabkan *Overhead* pada perangkat dengan sumber daya terbatas [29].

Beberapa studi sebelumnya telah melakukan evaluasi mendalam terhadap kinerja sistem di berbagai platform perangkat keras. Analisis kinerja Linux pada *Mini PC* menunjukkan bahwa konfigurasi kernel yang tepat sangat memengaruhi performa CPU serta menghasilkan efisiensi latensi memori yang baik [8]. Dalam konteks penggunaan sistem berkas pada lingkungan terbatas lainnya, penelitian terdahulu membuktikan bahwa Ext4 tetap mempertahankan performa yang stabil dan sangat baik meskipun dioperasikan dalam kondisi tervirtualisasi [23].

3. METODE PENELITIAN

Penelitian ini dilaksanakan dengan menggunakan pendekatan *Benchmarking* yang terstruktur dan sistematis untuk mengevaluasi kinerja sistem berkas dalam kondisi yang mensimulasikan penggunaan dunia nyata secara akurat. Pendekatan ini dipilih karena memungkinkan pengukuran kuantitatif terhadap metrik kualitas yang relevan, seperti responsivitas dan stabilitas sistem di bawah tekanan. Lingkungan pengujian dirancang secara cermat di atas perangkat keras yang merepresentasikan kategori "sumber daya terbatas" atau kelas pemula yang umum digunakan oleh masyarakat luas. Perangkat keras yang digunakan meliputi prosesor AMD A4-5100 APU dengan kecepatan 1.55GHz dan 4 inti pemrosesan, yang sering menjadi hambatan utama dalam komputasi modern. Dukungan memori RAM sebesar 8 GB DDR3 dan media penyimpanan SSD V-GEN berkapasitas 256GB digunakan untuk memastikan bahwa hasil pengujian relevan dengan standar minimum perangkat saat ini. Basis sistem operasi yang digunakan adalah Ubuntu 24.04.3 LTS dengan kernel versi 6.x, yang merupakan distribusi populer dan relevan.

Untuk menjamin validitas dan reliabilitas data, rangkaian pengujian dilakukan pada tiga instalasi sistem operasi yang identik, di mana satu-satunya variabel pembeda adalah jenis sistem berkas root yang digunakan: Ext4, XFS, dan Btrfs. Setiap skenario pengujian diulang sebanyak lima kali untuk meminimalisir anomali data dan mendapatkan nilai rata-rata yang konsisten. Prosedur ini dirancang untuk mengungkap karakteristik performa masing-masing

sistem berkas dalam menangani berbagai jenis beban kerja, mulai dari inisialisasi sistem hingga simulasi tugas pengembangan perangkat lunak yang kompleks.

Berikut adalah rincian langkah-langkah pengujian yang dilakukan dalam penelitian ini:

a. Pengukuran Waktu *Boot*

Pengujian ini bertujuan untuk menilai efisiensi sistem berkas dalam menangani proses saat inisialisasi sistem operasi. Alat *systemd-analyze* digunakan untuk mengukur total waktu *boot*

b. Pengukuran Waktu Respons Aplikasi

Skenario ini dirancang untuk mengevaluasi pengalaman pengguna secara langsung. Pengujian mengukur durasi waktu peluncuran aplikasi produktivitas umum, yaitu Firefox, LibreOffice, dan Gedit. Pengujian dilakukan dalam dua kondisi berbeda: kondisi *Idle* (tanpa beban) dan kondisi di bawah tekanan beban kerja tinggi. Beban kerja tinggi disimulasikan menggunakan alat *stress-ng* untuk membebani CPU dan memori, sehingga menguji stabilitas dan manajemen antrean I/O sistem berkas saat sumber daya sistem sedang diperebutkan.

c. Pengukuran Latensi I/O

Langkah ini menggunakan alat uji standar industri, Flexible I/O Tester (*fio*), untuk mengukur waktu respons (latensi) sistem berkas dalam menangani permintaan baca dan tulis. Pengujian mencakup pola akses sekuensial dan acak pada berbagai ukuran blok data. Latensi yang rendah sangat krusial untuk responsivitas sistem yang terasa "cepat" oleh pengguna, terutama saat menangani banyak berkas kecil.

d. Pengukuran *Throughput* Data

Melengkapi analisis latensi, pengujian ini juga menggunakan *fio* namun difokuskan untuk mengukur bandwidth maksimal atau kecepatan transfer data (dalam MB/s) yang dapat ditangani oleh sistem berkas. Skenario ini melibatkan transfer data besar (1MB blok) untuk mensimulasikan penyalinan berkas media, serta transfer data kecil (4KB blok) untuk mensimulasikan lalu lintas data sistem operasi. Konfigurasi pengujian diatur sebagai *microbenchmark* untuk meminimalkan *Overhead* CPU dan mengisolasi performa murni dari subsistem penyimpanan.

e. Simulasi Alur Kerja Pengembang

Sebagai representasi beban kerja tingkat lanjut, skenario ini mensimulasikan aktivitas pengembang perangkat lunak dengan melakukan kompilasi kode sumber proyek Busybox. Proses ini melibatkan manipulasi ribuan berkas kecil secara intensif, yang sangat membebani manajemen metadata sistem berkas. Tahapan pengujian mencakup proses *Git clone* (penyalinan repositori), konfigurasi proyek, kompilasi penuh (*Full Build*), kompilasi inkremental (*Incremental Build*), hingga pembersihan direktori (*Clean*).

f. Implementasi Optimisasi Sistem

Tahap akhir dari metodologi ini melibatkan penerapan hasil analisis kinerja untuk merancang ulang (*remastering*) sistem operasi Ubuntu 24.04, dengan menggunakan sistem berkas yang terbukti paling seimbang dan optimal sebagai fondasi utama. Proses optimisasi dilakukan secara menyeluruh dengan tujuan memaksimalkan efisiensi sumber daya pada perangkat keras terbatas. Langkah-langkah strategis yang diambil meliputi penyetelan mendalam pada parameter kernel dan konfigurasi *mounting* sistem berkas untuk mengurangi operasi tulis yang tidak esensial serta meningkatkan responsivitas I/O. Selain itu, manajemen memori ditingkatkan melalui implementasi teknologi kompresi memori virtual, yang bertujuan mengurangi ketergantungan sistem pada media penyimpanan fisik yang lambat saat memori penuh. Pada lapisan antarmuka pengguna, lingkungan *desktop* standar digantikan dengan alternatif yang jauh lebih ringan dan efisien, disertai dengan penghapusan paket perangkat lunak yang membebani sistem. Seluruh konfigurasi yang telah dioptimasi ini kemudian dibekukan dan dikemas kembali menjadi citra sistem (*image*) yang siap didistribusikan untuk memvalidasi peningkatan kinerja yang dicapai dibandingkan instalasi standar.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Waktu *Boot*

Tabel 1. Rata-Rata Waktu *Boot*

Sistem Berkas	Waktu Total (s)
Btrfs	50.19
Ext4	50.62
XFS	52.92

Berdasarkan data pada Tabel 1, secara keseluruhan, sistem berkas btrfs menunjukkan kinerja *boot* tercepat dengan catatan waktu 50.187 detik. Hasil ini menempatkannya di posisi terdepan, diikuti oleh ext4 yang mencatatkan waktu 50.618 detik, mengindikasikan selisih waktu yang sangat tipis, yaitu sekitar 0.431 detik. Sementara itu, sistem berkas xfs mencatatkan waktu *boot* terlama dengan 52.921 detik, berbeda sekitar 2.734 detik dari btrfs. Dengan demikian, btrfs dapat diidentifikasi sebagai sistem berkas yang paling efisien dalam konteks waktu *boot* total dalam pengujian ini.

4.2. Rata-Rata Respons Aplikasi

Tabel 2. Rata-Rata Waktu Respons Aplikasi

Aplikasi	Kondisi	Ext4 (ms)	XFS (ms)	Btrfs (ms)
LibreOffice	Tanpa Beban	3251	3585	3141
LibreOffice	Dengan Beban	7832	8164	7254
Firefox	Tanpa Beban	6537	6632	6663

Aplikasi	Kondisi	Ext4 (ms)	XFS (ms)	Btrfs (ms)
Firefox	Dengan Beban	16843	15860	17358
Gedit	Tanpa Beban	964	983	1015
Gedit	Dengan Beban	2323	2382	2296

Pada pengujian respons aplikasi yang ditampilkan pada Tabel 2, dalam kondisi tanpa beban (*Idle*), perbedaan kinerja antar sistem berkas relatif tipis. Ext4 dan XFS bersaing ketat, misalnya pada aplikasi Firefox dengan waktu luncur masing-masing 6.5 detik dan 6.6 detik. Namun, ketika sistem diberi beban kerja berat, waktu respons melambat pada semua sistem berkas. Ext4 menunjukkan stabilitas yang lebih baik pada aplikasi ringan seperti Gedit dengan waktu 2.3 detik, sedikit lebih cepat dibandingkan XFS (2.4 detik). Di sisi lain, Btrfs mencatatkan waktu tercepat untuk aplikasi berat seperti LibreOffice di bawah tekanan (7.25 detik). Secara keseluruhan, Ext4 menawarkan profil kinerja yang konsisten di berbagai kondisi beban, menjadikannya pilihan yang stabil untuk penggunaan harian.

4.3. Analisis Latensi I/O

Tabel 3. Rata-Rata Latensi I/O

Kategori Berkas	Operasi	Ext4 (µs)	XFS (µs)	Btrfs (µs)
Berkas Kecil (4KB)	Tulis Sekuensial	21	18	24
Berkas Kecil (4KB)	Baca Acak	129	133	201
Berkas Kecil (4KB)	Tulis Acak	24	22	32
Berkas Besar (1MB)	Tulis Sekuensial	2088	798	3559
Berkas Besar (1MB)	Baca Acak	4075	4628	4873
Berkas Besar (1MB)	Baca Sekuensial	2019	2023	2074

Hasil pengukuran latensi I/O pada Tabel 3 memperlihatkan karakteristik responsivitas yang beragam pada setiap sistem berkas. Pada operasi berkas kecil (4KB), XFS mencatatkan kinerja penulisan terbaik dengan latensi tulis sekuensial 18 µs dan tulis acak 22 µs, sedikit lebih cepat dibandingkan Ext4 (21 µs dan 24 µs). Namun, pada operasi pembacaan acak berkas kecil yang sangat sering terjadi dalam penggunaan *desktop*, Ext4 memimpin dengan latensi 129 µs, mengungguli XFS (133 µs) dan Btrfs (201 µs).

Untuk berkas besar (1MB), XFS mendominasi secara signifikan dalam operasi penulisan sekuensial dengan latensi sangat rendah yaitu 798 µs, jauh lebih efisien dibandingkan Ext4 (2088 µs) dan Btrfs (3559 µs). Namun, Ext4 menunjukkan keunggulan yang konsisten pada operasi pembacaan berkas besar. Ext4 mencatatkan latensi pembacaan sekuensial tercepat

sebesar 2019 µs dan pembacaan acak tercepat sebesar 4075 µs, mengungguli XFS dan Btrfs di kedua kategori pembacaan tersebut. Hal ini mengindikasikan bahwa Ext4 sangat andal dalam menyajikan data, baik yang diakses secara urut maupun acak.

4.4. Analisis Throughput Data

Tabel 4. Rata-Rata Throughput Data

Operasi	Ext4 (MB/s)	XFS (MB/s)	Btrfs (MB/s)
Baca Sekuensial (1MB)	421.5	478.1	369.9
Tulis Sekuensial (1MB)	255.5	400.3	199.3
Baca Acak (4KB)	56.5	61.2	57.5
Tulis Acak (4KB)	75.9	46.6	41.2

Analisis *Throughput* pada Tabel 4 menunjukkan kapasitas transfer data maksimal dari masing-masing sistem berkas. XFS mendominasi dalam operasi tulis sekuensial file besar dengan kecepatan 400.3 MB/s, mengungguli Ext4 yang mencapai 255.5 MB/s. Hal ini mengukuhkan posisi XFS sebagai pilihan ideal untuk manipulasi data besar. Sebaliknya, Ext4 menunjukkan keunggulan pada *Throughput* tulis acak (*random write*) untuk file kecil, mencapai 75.9 MB/s, jauh melampaui XFS (46.6 MB/s) dan Btrfs (41.2 MB/s). Keunggulan pada pola tulis acak ini relevan untuk beban kerja sistem operasi yang sering melakukan pembaruan kecil pada log dan berkas konfigurasi.

4.5. Analisis Simulasi Alur Kerja Pengembang

Tabel 5. Kinerja Alur Kerja Pengembang

Tahapan	Ext4	XFS	Btrfs
<i>Git clone</i> (s)	0.79	0.79	0.88
<i>Configure</i> (s)	15.98	16.12	15.79
<i>Full Build</i> (s)	111.49	121.65	112.53
<i>Incremental Build</i> (s)	28.40	28.00	27.65
<i>Clean</i> (s)	2.18	2.12	2.12
Utilisasi CPU <i>Full Build</i> (%)	317.4	292.8	314.2

Dalam simulasi alur kerja pengembang seperti yang ditunjukkan pada Tabel 5, Ext4 membuktikan keandalannya pada tugas komputasi berat. Tahap *Full Build*, yang merupakan proses paling intensif, diselesaikan dalam waktu 111.49 detik oleh Ext4, lebih cepat dibandingkan Btrfs (112.53 detik) dan XFS (121.65 detik). Lebih lanjut, Ext4 mencatatkan utilisasi CPU tertinggi sebesar 317.4%, yang menunjukkan bahwa sistem berkas ini mampu melayani permintaan data secepat prosesor membutuhkannya, meminimalkan waktu tunggu. Sebaliknya, XFS tertinggal dalam skenario ini, mengindikasikan adanya hambatan dalam menangani ribuan operasi metadata kecil yang khas pada proses kompilasi.

4.6. Hasil Implementasi Optimisasi

Berdasarkan analisis komprehensif di atas, Ext4 menunjukkan keseimbangan terbaik antara kecepatan

boot, konsistensi aplikasi, keunggulan tulis acak, dan efisiensi kompilasi sehingga sistem berkas ini dipilih sebagai basis untuk tahap *remastering*.

Tabel 6. Perbandingan Performa Sebelum dan Sesudah Optimisasi

Metrik	Ubuntu Ext4 Standar (<i>Idle</i>)	Ubuntu Ext4 Standar (<i>Stress</i>)	Ubuntu Ext4 Teroptimasi (<i>Idle</i>)	Ubuntu Ext4 Teroptimasi (<i>Stress</i>)
Waktu <i>Boot</i>	49.61 s	-	25.83 s	-
Waktu Kompilasi (<i>Full Build</i>)	125.83 s	-	60.35 s	-
Waktu Kompilasi (<i>Inc. Build</i>)	10.13 s	-	9.15 s	-
Respons LibreOffice	4.21 s	8.54 s	1.60 s	3.30 s
Respons Firefox	5.94 s	13.03 s	4.17 s	15.09 s
Respons Gedit	0.94 s	2.30 s	0.98 s	0.98 s
Penggunaan RAM	1154 MB	6612 MB	847 MB	5607 MB
Penggunaan Swap	0 MB	3175 MB	0 MB	1571 MB
Penggunaan CPU	2.00%	96.81%	0.21%	95.17%
Latensi I/O	1.34 ms	10.47 ms	1.34 ms	1.50 ms
Suhu CPU	27.60 °C	32.10 °C	26.90 °C	31.75 °C

Hasil pengujian dari implementasi optimisasi menunjukkan peningkatan performa yang signifikan. Waktu *boot* berhasil dipangkas dari 49.61 detik menjadi 25.83 detik, sebuah penghematan waktu sekitar 48% yang sangat terasa bagi pengguna. Peningkatan responsivitas juga terlihat pada peluncuran LibreOffice, di mana waktu peluncuran saat *Idle* turun dari 4.21 detik menjadi 1.60 detik. Penggunaan RAM saat *Idle* berkurang drastis dari 1154 MB menjadi 847 MB, memberikan ruang lebih lega bagi aplikasi pengguna. Yang paling krusial, di bawah kondisi stres, latensi I/O turun signifikan dari 10.47 ms menjadi 1.50 ms, membuktikan bahwa kombinasi Ext4 dengan parameter *writeback* dan manajemen memori ZRAM sangat efektif menjaga responsivitas sistem dalam keadaan sumber daya terbatas.

5. KESIMPULAN DAN SARAN

Penelitian ini menyimpulkan bahwa pada lingkungan *desktop* dengan sumber daya terbatas, tidak ada satu sistem berkas yang unggul secara mutlak di seluruh kategori pengujian, namun terdapat spesialisasi yang jelas di mana XFS dominan dalam *Throughput* data besar dan Btrfs efisien dalam waktu *boot*. Meskipun demikian, Ext4 terbukti sebagai pilihan paling rasional dan seimbang untuk sistem operasi utama berkat efisiensi metadata dan penggunaan CPU yang optimal, serta kinerja tulis acak yang superior, yang ketika dikombinasikan dengan teknik *remastering* (optimisasi kernel, ZRAM, lingkungan *desktop* ringan) mampu menurunkan waktu *boot* hingga 48% dan mereduksi latensi I/O secara masif. Berdasarkan temuan ini, penelitian selanjutnya disarankan untuk melakukan pengujian jangka panjang guna memantau degradasi kinerja akibat fragmentasi, menganalisis efisiensi energi pada perangkat lawas, serta mengeksplorasi sistem berkas F2FS (*Flash-Friendly File System*) dan optimisasi kernel lanjut seperti modifikasi I/O scheduler untuk

melihat batas maksimal performa pada perangkat keras terbatas.

DAFTAR PUSTAKA

- [1] D. O. Nasrullah, A. W. Hertaliando, R. Tullah, and F. Ferawati, "Evaluasi Perbandingan Sistem Operasi Linux untuk Pemanfaatan Client-Server: Studi Kasus Ubuntu dan Debian," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no. 1, pp. 242–248, 2024, doi: 10.57152/malcom.v5i1.1702.
- [2] M. Boras, J. Balen, and K. Vdovjak, "Performance Evaluation of Linux Operating Systems," *Proc. 2020 Int. Conf. Smart Syst. Technol. SST 2020*, vol. 23, no. 2, pp. 115–120, 2020, doi: 10.1109/SST49455.2020.9264055.
- [3] Y. M. Geasela and N. Legowo, "Designing Information System Architecture Based on Education 4.0 Case Study: Senior High School Institutions of Indonesia," *J. Comput. Sci.*, vol. 18, no. 7, pp. 622–637, 2022, doi: 10.3844/JCSSP.2022.622.637.
- [4] J. Saputra and J. A. Ginting, "Penyembunyian Data Menggunakan Metode Overwriting Metadata," *JIKO (Jurnal Inform. dan Komputer)*, vol. 7, no. 2, p. 296, 2023, doi: 10.26798/jiko.v7i2.692.
- [5] A. A. Sukmandhani, Y. M. Gaesela, M. Mery, E. Edgardo, K. Kevin, and N. Fernando, "Testing dan Evaluasi Performa Website E-Commerce," *JBASE - J. Bus. Audit Inf. Syst.*, vol. 6, no. 2, pp. 34–45, 2023, doi: 10.30813/jbase.v6i2.4673.
- [6] V. Dakic, M. Kovac, and I. Videc, "High-Performance Computing Storage Performance and Design Patterns—Btrfs and ZFS Performance for Different Use Cases," *Computers*, vol. 13, no. 6, p. 139, 2024, doi: <https://doi.org/10.3390/computers13060139>.
- [7] S. W. Yoo *et al.*, "DJFS: Directory-Granularity Filesystem Journaling for CMM-H SSDs," *Proc.*

- 23rd USENIX Conf. File Storage Technol. (FAST 2025), 2025.
- [8] H. Fryonanda, A. Febrian Kasmar, M. Ibrahim Nasution, E. Rosman, and W. Nissa, "Analisis Kinerja Linux Pada Mini Pc: Kinerja Prosesor, Memori Dan Disk Drive," no. September, pp. 1–13, 2023.
- [9] Y. Jiao *et al.*, "BetrFS: A Compleat File System for Commodity SSDs," *EuroSys 2022 - Proc. 17th Eur. Conf. Comput. Syst.*, pp. 610–627, 2022, doi: 10.1145/3492321.3519571.
- [10] W. B. Mingardi and G. M. D. Vieira, "Characterizing Synchronous Writes in Stable Memory Devices," 2020, doi: 10.5753/wperformance.2019.6458.
- [11] Zacky Rafian Fawwauzy, Taufik Eka Albani, and Elkin Rilvani, "Perbandingan Teknik pada Struktur Sistem File Windows & Linux," *Repeater Publ. Tek. Inform. dan Jar.*, vol. 3, no. 1, pp. 69–79, 2025, doi: 10.62951/repeater.v3i1.343.
- [12] I. Yunianto and K. Adhiyarta, "Jurnal Review: Perbandingan Sistem Operasi Linux Dengan Sistem Operasi Windows," *JUPITER J. Comput. Inf. Technol.*, vol. 1, no. 1, pp. 1–7, 2020, doi: 10.53990/jupiter.v1i1.3.
- [13] H. R. Sanjaya *et al.*, "Sistem Operasi Pada Komputer Yang Paling Banyak Digunakan," vol. 2, pp. 9–17, 2024.
- [14] A. Putri Amanda Nasution, A. Zahra Iswandy Lubis, A. Izmi, H. Atiqah Tampubolon, and I. Gunawan, "Pengaruh Sistem Operasi Terhadap Sistem Informasi," *J. Inov. Artif. Intell. Komputasional Nusantara*, vol. 2, no. 1, pp. 42–45, 2025.
- [15] H. I. Amrullah, A. Nur, F. Amarta, and E. Rilvani, "Fleksibilitas dan Kesederhanaan Arsitektur Sistem Operasi Linux," no. 2020, 2025.
- [16] X. Tan, M. Zhou, and B. Fitzgerald, "Scaling open source communities: An empirical study of the linux kernel," *Proc. - Int. Conf. Softw. Eng.*, pp. 1222–1234, 2020, doi: 10.1145/3377811.3380920.
- [17] E. Kuiter, C. Sundermann, T. Thüm, T. Hess, S. Krieter, and G. Saake, "How Configurable is the Linux Kernel? Analyzing Two Decades of Feature-Model History," *ACM Trans. Softw. Eng. Methodol.*, 2025, doi: 10.1145/3729423.
- [18] Y. Zhan *et al.*, "Rethinking the Request-to-IO Transformation Process of File Systems for Full Utilization of High-Bandwidth SSDs," *Proc. 23rd USENIX Conf. File Storage Technol. (FAST 2025)*, 2025.
- [19] J. Koo *et al.*, "Modernizing file system through in-storage indexing," *Proc. 15th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2021*, pp. 75–92, 2021.
- [20] B. Đorđević, K. Janjić, and N. Kraljević, "File System Performance Comparison With the Hypervisors Esxi and Xen," pp. 10–17, 2024, doi: 10.15308/sinteza-2024-10-17.
- [21] J. Jeong, B. Kim, and J. Cho, "A Digital Forensic Process for Ext4 File System in the Flash Memory of IoT Devices," *J. KIISE*, vol. 48, no. 8, pp. 865–870, 2021, doi: 10.5626/jok.2021.48.8.865.
- [22] W. Aji Purnama, Y. Servanda, and dan Agus Wijayanto, "Analisis Efektifitas Recovery Toolkit Untuk Media Penyimpanan Menggunakan Pendekatan Criteria-Based Evaluation Analysis of the Effectiveness of the Recovery Toolkit for Storage Media Using a Criteria-Based Evaluation Approach," vol. 8, no. 1, pp. 35–48, 2025, [Online]. Available: <http://journal.ubm.ac.id/index.php/jbase>
- [23] S. Farizy and T. Gunawan, "Analisa Performa Sistem Berkas Ext4 Pada Kondisi Tervirtualisasi," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 30, no. 1, pp. 1–6, 2020, doi: 10.37277/stch.v30i1.726.
- [24] B. P. Sterniczuk, "Comparison of EXT4 and NTFS filesystem performance P orównanie wydajności systemu plików EXT4 i NTFS," vol. 25, no. June, pp. 297–300, 2022.
- [25] D. Kim, K. Min, J. Oh, and Y. Won, "ScaleXFS: Getting scalability of XFS back on the ring," *Proc. 20th USENIX Conf. File Storage Technol. FAST 2022*, pp. 329–343, 2022.
- [26] K. Min, D. Kim, S. Lim, and Y. Won, "Polling Based Per-core Workqueue Management in XFS Journaling," *Proc. - 2022 IEEE 11th Non-Volatile Mem. Syst. Appl. Symp. NVMSA 2022*, pp. 39–44, 2022, doi: 10.1109/NVMSA56066.2022.00016.
- [27] N. N. Panteleev, S. S. Panov, and A. V. Matveev, "Сравнение характеристик и возможностей современных файловых систем Linux: ext4, xfs, btrfs," *Компьютерные и информационные науки*, vol. 74, no. 11, pp. 125–140, 2022.
- [28] K. E. V. Bilibio and J. S. Silva, "Avaliação de Desempenho de Sistemas de Arquivos Linux: Um Estudo Comparativo com Benchmarks," pp. 121–131, 2025, doi: 10.5753/ersimt.2025.8895.
- [29] S. Shaikh, "Billion-files File Systems (BfFS): A Comparison," 2024, [Online]. Available: <http://arxiv.org/abs/2408.01805>