LAPORAN SISTEM OPERASI

"ANALISIS PERBANDINGAN PERFORMA WINDOWS SUBSYSTEM FOR LINUX (WSL) UBUNTU DAN LINUX UBUNTU PADA VIRTUAL MACHINE MENGGUNAKAN SYSBENCH"

Dosen Pengampu: Ferdi Chahyadi, S.Kom., M.Cs

Disusun Oleh Kelompok:

Nabiilah Rifa Musyifa (2401020162)

Atsilah Halimatus Sa'diyah (2401020142)

Nurfaizah Rasikha (2401020156)

Shalsabyla Finta Azalea (2401020167)

Munfarida (2401020166)

Program Studi Teknik Informatika
Fakultas Teknik dan Teknologi Kemaritiman
Universitas Maritim Raja Ali Haji

2025/2026

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah Subhanahu wa Ta'ala atas rahmat dan karunia-Nya, sehingga laporan proyek penelitian **"Benchmarking Performa Sistem Operasi: Analisis Komparatif Linux vs Windows (WSL)"** dapat diselesaikan dengan baik.

Laporan ini disusun sebagai bagian dari tugas mata kuliah **Sistem Operasi**. Penelitian dilakukan untuk membandingkan performa antara sistem operasi Linux (Ubuntu) dan Windows melalui WSL dengan fokus pada tiga aspek utama: **CPU, RAM, dan Disk I/O** menggunakan alat benchmark **Sysbench**.

Kami menyampaikan terima kasih kepada:

1. Bapak/Ibu Dosen pengampu mata kuliah Sistem Operasi atas bimbingan dan ilmu yang diberikan.
2. Seluruh anggota Kelompok 5 atas kerja sama dan kontribusi penuh selama pengerjaan proyek.
3. Pihak-pihak lain yang turut mendukung terselesaikannya penelitian ini.

Kami menyadari sepenuhnya bahwa laporan ini masih jauh dari kata sempurna. Terdapat berbagai keterbatasan dan kekurangan baik dalam metodologi penelitian, analisis data, maupun penyajian hasil. Oleh karena itu, kami membuka diri untuk menerima segala bentuk kritik dan saran yang konstruktif guna penyempurnaan laporan ini di masa yang akan datang.

Semoga laporan penelitian ini dapat bermanfaat tidak hanya bagi kami sebagai penulis, tetapi juga bagi pembaca pada umumnya, khususnya bagi mereka yang tertarik untuk mendalami studi tentang sistem operasi dan evaluasi performa sistem komputasi.

Tanjungpinang, 18 Desember 2024

Penulis

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komputer dan sistem operasi mendorong kebutuhan akan sistem yang mampu mengelola sumber daya perangkat keras secara efisien. Sistem operasi memiliki peran penting dalam mengatur penggunaan CPU, memori, dan media penyimpanan agar kinerja sistem dapat berjalan secara optimal. Perbedaan mekanisme pengelolaan sumber daya pada suatu sistem operasi dapat menyebabkan perbedaan performa, meskipun dijalankan pada perangkat keras yang sama.

Saat ini, penggunaan sistem operasi Linux tidak hanya terbatas pada instalasi langsung pada perangkat keras, tetapi juga dijalankan melalui berbagai lingkungan virtualisasi. Salah satu implementasi yang banyak digunakan adalah Windows Subsystem for Linux (WSL), yang memungkinkan lingkungan Linux berjalan di dalam sistem operasi Windows. Selain itu, Linux juga dapat dijalankan melalui teknologi Virtual Machine yang menyediakan lingkungan terisolasi di atas sistem operasi host.

Perbedaan cara kerja antara WSL dan Virtual Machine menyebabkan adanya perbedaan dalam pengelolaan sumber daya sistem. Pada WSL, lingkungan Linux terintegrasi dengan sistem operasi Windows sehingga manajemen CPU, memori, dan sistem file dipengaruhi oleh kebijakan Windows sebagai host. Sementara itu, Linux pada Virtual Machine dijalankan sebagai sistem operasi tamu dengan alokasi sumber daya yang diatur melalui hypervisor. Kondisi ini berpotensi menimbulkan perbedaan performa sistem pada kedua lingkungan tersebut.

Berdasarkan kondisi tersebut, diperlukan analisis performa untuk mengetahui bagaimana perbedaan lingkungan WSL dan Virtual Machine memengaruhi kinerja Linux, khususnya pada aspek CPU, memori, dan disk. Pengujian performa dilakukan menggunakan tools benchmarking agar hasil yang diperoleh bersifat terukur dan dapat dibandingkan secara objektif.

Dalam penelitian ini, perbandingan dilakukan menggunakan Windows Subsystem for Linux (WSL) dan Linux Ubuntu yang dijalankan pada Virtual Machine sebagai representasi lingkungan pengujian sistem operasi Windows dan Linux. Pendekatan ini digunakan untuk menganalisis

performa CPU, RAM, dan disk pada kedua lingkungan tersebut tanpa mengklaim pengujian dilakukan pada sistem operasi native secara langsung.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana perbandingan performa CPU antara Ubuntu pada Windows Subsystem for Linux (WSL) dan Ubuntu pada Virtual Machine?
2. Bagaimana perbandingan performa memori antara Ubuntu pada Windows Subsystem for Linux (WSL) dan Ubuntu pada Virtual Machine?
3. Bagaimana perbandingan performa disk antara Ubuntu pada Windows Subsystem for Linux (WSL) dan Ubuntu pada Virtual Machine?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menganalisis performa CPU pada Ubuntu yang dijalankan di lingkungan Windows Subsystem for Linux (WSL) dan Virtual Machine.
2. Menganalisis performa memori pada Ubuntu yang dijalankan di lingkungan Windows Subsystem for Linux (WSL) dan Virtual Machine.
3. Menganalisis performa disk pada Ubuntu yang dijalankan di lingkungan Windows Subsystem for Linux (WSL) dan Virtual Machine.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan pemahaman mengenai perbedaan karakteristik performa Linux pada lingkungan WSL dan Virtual Machine.

2. Menjadi referensi bagi mahasiswa atau pengguna yang ingin memilih lingkungan Linux yang sesuai dengan kebutuhan pengembangan dan pengujian.
3. Menambah wawasan dalam bidang sistem operasi, khususnya terkait pengaruh virtualisasi terhadap performa sistem.

BAB 2
LANDASAN TEORI

2.1 Sistem Operasi

Sistem operasi merupakan perangkat lunak sistem yang berfungsi sebagai penghubung antara pengguna, aplikasi, dan perangkat keras komputer. Sistem operasi menyediakan antarmuka bagi pengguna serta bertanggung jawab dalam mengelola dan mengoordinasikan seluruh sumber daya sistem agar dapat digunakan secara optimal.

Dalam konteks manajemen sumber daya, sistem operasi memiliki peran penting dalam pengaturan penggunaan CPU, memori utama (RAM), serta perangkat penyimpanan. Pengelolaan ini dilakukan melalui mekanisme penjadwalan proses, manajemen memori, dan pengendalian input/output. Efektivitas mekanisme tersebut sangat berpengaruh terhadap performa sistem secara keseluruhan.

Selain itu, sistem operasi juga bertugas menjaga stabilitas dan keamanan sistem dengan mengatur isolasi antar proses serta mengelola hak akses pengguna. Perbedaan arsitektur dan mekanisme kerja sistem operasi dapat menyebabkan perbedaan performa meskipun dijalankan pada perangkat keras yang sama. Oleh karena itu, pemahaman mengenai peran sistem operasi menjadi dasar penting dalam melakukan analisis perbandingan performa.

2.2 Windows Subsystem for Linux (WSL)

Windows Subsystem for Linux (WSL) adalah fitur pada sistem operasi Windows yang memungkinkan pengguna menjalankan lingkungan Linux secara langsung di dalam Windows tanpa perlu melakukan dual boot atau instalasi sistem operasi Linux secara terpisah. WSL dirancang untuk memberikan kompatibilitas perintah dan aplikasi Linux dengan tingkat integrasi yang tinggi terhadap ekosistem Windows.

Pada WSL, sistem Linux dijalankan di atas lingkungan Windows sehingga pengelolaan sumber daya seperti CPU, RAM, dan sistem file tetap berada di bawah kendali sistem operasi Windows. Alokasi sumber daya untuk lingkungan Linux dilakukan secara dinamis mengikuti

kebijakan manajemen sumber daya Windows. Kondisi ini menyebabkan performa Linux pada WSL tidak sepenuhnya terlepas dari pengaruh sistem operasi host.

Selain aspek manajemen sumber daya, WSL juga memiliki integrasi sistem file yang erat dengan Windows. Akses terhadap file Linux dan Windows dapat dilakukan secara langsung, sehingga memudahkan proses pengembangan dan pengujian aplikasi lintas platform. Namun, integrasi ini juga berpotensi memengaruhi performa operasi input/output, terutama pada pengujian disk.

2.3 Linux Ubuntu pada Virtual Machine

Linux Ubuntu pada Virtual Machine merujuk pada sistem operasi Linux Ubuntu yang dijalankan sebagai sistem operasi tamu (guest operating system) di atas perangkat lunak virtualisasi (hypervisor) dengan sistem operasi host berupa Windows. Pendekatan ini memungkinkan Linux dijalankan secara terisolasi dalam sebuah lingkungan virtual yang menyerupai komputer fisik.

Pada lingkungan Virtual Machine, Linux Ubuntu menggunakan kernel Linux sepenuhnya untuk mengelola proses, memori, dan sistem file. Namun, akses terhadap perangkat keras fisik tidak dilakukan secara langsung, melainkan melalui lapisan virtualisasi yang disediakan oleh hypervisor. Lapisan ini bertugas menerjemahkan permintaan dari sistem operasi tamu ke perangkat keras fisik.

Penggunaan virtualisasi menyebabkan adanya overhead sistem yang dapat memengaruhi performa, terutama pada pengujian yang melibatkan penggunaan CPU dan disk secara intensif. Meskipun demikian, lingkungan Virtual Machine cenderung memberikan performa yang lebih stabil dan terisolasi dibandingkan WSL, karena alokasi sumber daya dapat diatur secara eksplisit.

2.4 Virtual Machine dan Virtualisasi

Virtual Machine merupakan teknologi yang memungkinkan satu perangkat keras fisik menjalankan beberapa sistem operasi secara bersamaan. Teknologi ini bekerja dengan

memanfaatkan hypervisor sebagai lapisan pengelola yang mengatur alokasi sumber daya fisik seperti CPU, memori, dan penyimpanan ke masing-masing sistem operasi tamu.

Virtualisasi memberikan fleksibilitas tinggi dalam pengujian dan pengembangan sistem karena memungkinkan simulasi berbagai lingkungan tanpa memerlukan perangkat keras tambahan. Dalam konteks penelitian ini, virtualisasi digunakan untuk menyediakan lingkungan Linux yang terisolasi sehingga perbandingan performa dapat dilakukan secara terkontrol.

Namun, penggunaan virtualisasi juga membawa konsekuensi berupa overhead sistem. Overhead ini muncul akibat adanya lapisan tambahan antara sistem operasi dan perangkat keras, sehingga perlu diperhatikan dalam analisis hasil pengujian performa.

2.5 Sysbench

Sysbench merupakan perangkat lunak benchmarking berbasis command-line yang digunakan untuk menguji dan mengevaluasi performa sistem komputer secara kuantitatif. Sysbench banyak digunakan dalam pengujian sistem karena bersifat ringan, fleksibel, serta mampu menghasilkan metrik performa yang konsisten.

Sysbench menyediakan berbagai jenis pengujian yang mencakup komponen utama sistem, seperti CPU, memori, dan disk. Pengujian CPU bertujuan untuk mengukur kemampuan prosesor dalam menangani operasi komputasi, sedangkan pengujian memori digunakan untuk mengevaluasi kecepatan akses dan transfer data pada RAM. Pengujian disk dilakukan untuk mengukur performa operasi input/output pada media penyimpanan.

Dalam penelitian ini, Sysbench digunakan karena mampu menjalankan stress test secara berulang dengan parameter yang sama, sehingga hasil pengujian dapat dibandingkan secara objektif. Pengujian dilakukan beberapa kali untuk meminimalkan pengaruh fluktuasi sistem dan memperoleh hasil yang lebih representatif.

Bab ini menjadi dasar teoritis untuk memahami perbedaan lingkungan pengujian yang digunakan serta menjadi acuan dalam menganalisis hasil pengujian performa yang dibahas pada bab selanjutnya.

BAB 3
METODOLGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode eksperimen dengan tujuan untuk menganalisis dan membandingkan performa sistem pada dua lingkungan sistem operasi yang berbeda, yaitu Windows Subsystem for Linux (WSL) dan Linux Ubuntu yang dijalankan pada Virtual Machine. Metode eksperimen dipilih karena memungkinkan pengujian dilakukan secara langsung dengan memberikan beban kerja tertentu pada sistem, kemudian mengamati hasil yang dihasilkan.

Pendekatan eksperimen dinilai sesuai dengan tujuan penelitian karena fokus utama proyek ini adalah mengukur performa CPU, RAM, dan disk secara objektif berdasarkan data hasil pengujian. Dengan metode ini, kondisi pengujian dapat dikendalikan sehingga perbedaan hasil yang diperoleh lebih mencerminkan perbedaan karakteristik lingkungan sistem operasi yang diuji.

Pengujian dilakukan menggunakan tools benchmarking sysbench untuk menghasilkan data performa yang bersifat kuantitatif. Data tersebut selanjutnya dianalisis untuk memberikan gambaran mengenai perbedaan performa pada masing-masing platform.

3.2 Objek Penelitian

Objek penelitian dalam proyek ini adalah performa sistem pada dua platform sistem operasi, yaitu Windows Subsystem for Linux (WSL) dan Linux Ubuntu yang dijalankan pada Virtual Machine. Kedua platform dipilih karena mewakili dua pendekatan berbeda dalam menjalankan lingkungan Linux pada komputer berbasis Windows.

Pada platform WSL, Linux Ubuntu dijalankan di dalam sistem operasi Windows dengan integrasi langsung terhadap sistem host. Sementara itu, pada platform Virtual Machine, Linux Ubuntu dijalankan dalam lingkungan virtual yang terpisah dari sistem operasi host, dengan alokasi sumber daya tersendiri.

Objek penelitian difokuskan pada pengukuran performa CPU, memori (RAM), dan disk. Kedua platform diuji menggunakan skenario dan parameter pengujian yang sama agar hasil yang diperoleh dapat dibandingkan secara adil dan konsisten.

Penggunaan WSL dan Virtual Machine dalam penelitian ini bersifat representatif, di mana WSL dijalankan dan dikelola oleh sistem operasi Windows, sedangkan Linux Ubuntu pada Virtual Machine dijalankan dalam lingkungan virtualisasi. Pendekatan ini digunakan untuk menggambarkan perbedaan karakteristik performa kedua lingkungan sistem operasi berdasarkan hasil pengujian benchmarking.

3.3 Perangkat dan Lingkungan Pengujian

Pengujian dilakukan pada satu perangkat keras yang sama untuk meminimalkan perbedaan hasil akibat variasi spesifikasi hardware. Dengan menggunakan perangkat keras yang sama, perbedaan performa yang muncul diharapkan berasal dari perbedaan lingkungan sistem operasi, bukan dari perbedaan perangkat keras.

Spesifikasi perangkat keras meliputi prosesor, kapasitas memori utama (RAM), dan media penyimpanan yang sama. Sistem operasi Windows digunakan sebagai host system, kemudian WSL dengan distribusi Ubuntu dijalankan di atasnya. Selain itu, Linux Ubuntu juga dijalankan pada sebuah Virtual Machine dengan konfigurasi sumber daya yang disesuaikan agar sebanding.

Lingkungan pengujian diatur agar kondisi sistem relatif stabil selama proses benchmarking berlangsung, sehingga hasil pengujian tidak dipengaruhi oleh aktivitas sistem lain.

3.4 Alat dan Instrumen Penelitian

Alat utama yang digunakan dalam penelitian ini adalah sysbench. Sysbench merupakan tools benchmarking yang digunakan untuk menguji performa sistem, khususnya pada aspek CPU, memori (RAM), dan disk I/O. Tools ini banyak digunakan dalam pengujian performa sistem karena mampu menghasilkan metrik yang konsisten dan mudah dianalisis.

Sysbench dipilih karena dapat dijalankan pada lingkungan Linux, termasuk pada WSL dan Virtual Machine, serta mendukung berbagai jenis pengujian yang sesuai dengan tujuan penelitian. Penggunaan tools yang sama pada kedua platform bertujuan untuk memastikan bahwa perbedaan hasil pengujian benar-benar disebabkan oleh perbedaan lingkungan sistem operasi.

Parameter pengujian disesuaikan agar mencerminkan beban kerja yang seragam pada kedua platform.

3.5 Prosedur Pengujian

Prosedur pengujian dilakukan secara bertahap dan sistematis untuk memastikan hasil yang diperoleh dapat dipercaya. Sebelum pengujian dimulai, lingkungan sistem pada kedua platform dipersiapkan dengan konfigurasi yang serupa.

Tahapan pengujian meliputi:

1. Menyiapkan lingkungan WSL dan Virtual Machine dengan konfigurasi sistem yang sebanding.
2. Memastikan tidak ada aplikasi lain yang berjalan secara signifikan selama proses benchmarking.
3. Melakukan pengujian CPU menggunakan sysbench CPU test.
4. Melakukan pengujian memori menggunakan sysbench memory test.
5. Melakukan pengujian disk menggunakan sysbench file I/O test.

Setiap jenis pengujian dilakukan sebanyak tiga kali untuk memperoleh hasil yang lebih stabil dan mengurangi pengaruh fluktuasi performa sistem.

3.6 Teknik Pengumpulan Data

Data dikumpulkan dari hasil keluaran (output) sysbench pada setiap pengujian. Output tersebut berisi metrik performa yang relevan sesuai dengan jenis pengujian yang dilakukan.

Data yang dikumpulkan meliputi waktu eksekusi, throughput, dan metrik lain yang berkaitan dengan performa CPU, RAM, dan disk. Seluruh data dicatat secara sistematis dan disusun dalam bentuk tabel agar memudahkan proses analisis.

Pengumpulan data dilakukan secara konsisten pada kedua platform untuk menjaga keseragaman hasil.

3.7 Teknik Analisis Data

Analisis data dilakukan menggunakan metode analisis deskriptif. Metode ini digunakan untuk membandingkan hasil pengujian performa antara WSL dan Linux Ubuntu pada Virtual Machine berdasarkan data yang diperoleh.

Data dianalisis dengan memperhatikan nilai hasil setiap pengujian dan kecenderungan performa masing-masing platform. Untuk memperjelas hasil perbandingan, data disajikan dalam bentuk tabel dan grafik.

Hasil analisis ini digunakan sebagai dasar dalam menarik kesimpulan mengenai perbedaan performa CPU, RAM, dan disk pada kedua lingkungan sistem operasi.

BAB 4
HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian CPU

Pengujian CPU dilakukan menggunakan perintah `sysbench cpu run` dengan parameter tambahan untuk menguji kemampuan komputasi prosesor. Pengujian ini mengukur kecepatan eksekusi dalam events per second (jumlah operasi per detik). Berikut adalah hasil pengujian CPU pada tiga kali percobaan:

Tabel 4.1: Hasil Pengujian CPU (Events per Second)

Lingkungan	Uji-1	Uji-2	Uji-3	Rata-rata
WSL	2203.60	1961.46	2225.39	2130.15
VM	1212.02	1279.38	1150.45	1213.95

Grafik 4.1: Perbandingan Performa CPU

Berdasarkan hasil pengujian CPU, dapat diamati bahwa Windows Subsystem for Linux (WSL) menunjukkan performa yang secara signifikan lebih tinggi dibandingkan dengan Linux Ubuntu pada Virtual Machine. Rata-rata kecepatan eksekusi pada WSL mencapai 2130.15 events per second, sedangkan pada Virtual Machine hanya mencapai 1213.95 events per second. Perbedaan ini menunjukkan bahwa WSL memiliki keunggulan hampir dua kali lipat dalam hal kemampuan komputasi CPU dibandingkan dengan lingkungan virtualisasi penuh.

Performa WSL yang lebih tinggi dapat dijelaskan oleh arsitektur WSL yang terintegrasi langsung dengan kernel Windows, memungkinkan akses yang lebih efisien ke sumber daya CPU fisik. WSL 2 menggunakan teknologi virtualisasi ringan dengan kernel Linux yang dioptimalkan, yang mengurangi overhead dibandingkan dengan virtualisasi penuh pada Virtual Machine. Selain itu,

WSL dapat memanfaatkan fitur-fitur optimasi Windows seperti penjadwalan proses yang lebih efisien untuk beban kerja komputasi.

Sementara itu, Virtual Machine mengalami overhead virtualisasi yang lebih signifikan karena setiap permintaan dari sistem operasi tamu harus melalui lapisan hypervisor untuk diterjemahkan ke perangkat keras fisik. Hal ini menyebabkan penurunan performa yang nyata, terutama pada operasi komputasi intensif seperti yang diuji dengan Sysbench. Meskipun Virtual Machine menyediakan isolasi penuh dan lingkungan yang lebih terpisah, hal ini datang dengan biaya performa yang harus dibayar.

4.2 Hasil Pengujian Memori

Pengujian memori dilakukan menggunakan perintah `sysbench memory run` untuk mengukur kecepatan transfer data pada RAM. Metrik yang diukur adalah throughput dalam satuan MiB per second (Megabinary bytes per second). Berikut adalah hasil pengujian memori pada tiga kali percobaan:

Tabel 4.2: Hasil Pengujian Memori (Transfer MiB/sec)

Lingkungan	Uji-1	Uji-2	Uji-3	Rata-rata
WSL	3583.11	3115.49	3583.11	3427.24
VM	1700.72	2212.67	2227.24	2046.88

Grafik 4.2: Perbandingan Performa Memori

Hasil pengujian memori menunjukkan pola yang konsisten dengan pengujian CPU, di mana Windows Subsystem for Linux (WSL) kembali menunjukkan performa yang lebih unggul dibandingkan dengan Linux Ubuntu pada Virtual Machine. Rata-rata throughput memori pada WSL mencapai 3427.24 MiB per second, sementara pada Virtual Machine hanya mencapai

2046.88 MiB per second. Perbedaan ini menunjukkan bahwa WSL memiliki kecepatan transfer memori sekitar 67% lebih tinggi dibandingkan dengan lingkungan virtualisasi.

Keunggulan WSL dalam pengujian memori dapat dijelaskan oleh mekanisme manajemen memori yang lebih efisien. WSL berbagi memori secara dinamis dengan sistem host Windows, memungkinkan alokasi dan akses memori yang lebih langsung ke perangkat keras fisik. Arsitektur WSL 2 yang menggunakan teknologi virtualisasi ringan meminimalkan overhead dalam operasi memori, sehingga menghasilkan throughput yang lebih tinggi.

Di sisi lain, Virtual Machine menggunakan alokasi memori yang terisolasi dan terbatas, dengan setiap akses memori harus melalui lapisan virtualisasi dari hypervisor. Meskipun teknik modern seperti memory ballooning dan direct memory access telah mengurangi overhead ini, tetap ada penalti performa yang signifikan dibandingkan dengan akses memori langsung. Hasil pengujian menunjukkan bahwa Virtual Machine mengalami bottleneck pada operasi memori, terutama pada percobaan pertama di mana throughput hanya mencapai 1700.72 MiB per second.

4.3 Hasil Pengujian Disk

Pengujian disk dilakukan menggunakan perintah `sysbench fileio` dengan mode pengujian random read/write untuk mengukur performa input/output pada media penyimpanan. Pengujian ini menghasilkan dua metrik utama: kecepatan baca (Read) dan kecepatan tulis (Write) dalam satuan MiB per second. Berikut adalah hasil pengujian disk pada tiga kali percobaan:

Tabel 4.3: Hasil Pengujian Disk Read (MiB/s)

Lingkungan	Uji-1	Uji-2	Uji-3	Rata-rata
WSL	8.84	9.40	16.51	11.58
VM	1.72	2.10	1.97	1.93

Tabel 4.4: Hasil Pengujian Disk Write (MiB/s)

Lingkungan	Uji-1	Uji-2	Uji-3	Rata-rata
WSL	5.89	6.26	11.01	7.72
VM	1.15	1.40	1.31	1.29

Grafik 4.3: Perbandingan Performa Disk Read

Grafik 4.4: Perbandingan Performa Disk Write

Pengujian disk menunjukkan perbedaan performa yang paling dramatis antara kedua lingkungan. Windows Subsystem for Linux (WSL) secara konsisten mengungguli Linux Ubuntu pada Virtual Machine dengan margin yang sangat besar. Untuk operasi baca (read), WSL mencapai rata-rata 11.58 MiB per second, sementara Virtual Machine hanya mencapai 1.93 MiB per second - perbedaan hampir enam kali lipat. Untuk operasi tulis (write), WSL mencapai rata-rata 7.72 MiB per second, sedangkan Virtual Machine hanya 1.29 MiB per second - perbedaan hampir enam kali lipat juga.

Performa disk yang jauh lebih baik pada WSL terutama disebabkan oleh arsitektur sistem file yang digunakan. WSL mengakses file melalui sistem file NTFS milik Windows secara langsung, dengan lapisan translasi yang minimal. WSL 2 menggunakan teknologi 9P file system protocol yang dioptimalkan untuk performa, memungkinkan akses yang efisien ke file yang disimpan pada drive Windows. Selain itu, WSL dapat memanfaatkan cache file sistem Windows dan optimasi I/O lainnya yang sudah terintegrasi.

Sebaliknya, Virtual Machine mengalami overhead yang sangat signifikan pada operasi disk karena setiap permintaan I/O harus melalui beberapa lapisan abstraksi. File sistem pada Virtual Machine biasanya disimpan dalam format virtual disk image (seperti VMDK atau VDI), yang menambah latency dan mengurangi throughput. Virtualisasi disk pada hypervisor juga menambah overhead tambahan, terutama untuk operasi I/O acak (random read/write) seperti yang diuji dalam penelitian ini. Hasil pengujian menunjukkan bahwa Virtual Machine mengalami bottleneck yang parah pada operasi disk, dengan performa yang sangat rendah dibandingkan dengan WSL.

4.4 Analisis Perbandingan

Tabel 4.5: Ringkasan Hasil Benchmarking

Parameter	WSL Performance	VM Performance	Selisih	Keunggulan
CPU	2130.15 events/sec	1213.95 events/sec	+75.5%	WSL
Memori	3427.24 MiB/sec	2046.88 MiB/sec	+67.4%	WSL
Disk Read	11.58 MiB/s	1.93 MiB/s	+500%	WSL
Disk Write	7.72 MiB/s	1.29 MiB/s	+498%	WSL

Berdasarkan hasil pengujian yang telah dilakukan pada tiga aspek utama (CPU, memori, dan disk), dapat dianalisis bahwa Windows Subsystem for Linux (WSL) secara konsisten menunjukkan performa yang lebih unggul dibandingkan dengan Linux Ubuntu pada Virtual Machine. Pola ini terlihat pada semua parameter yang diuji, dengan perbedaan yang paling signifikan terjadi pada operasi disk I/O.

Performa WSL yang lebih baik dapat dijelaskan oleh beberapa faktor kunci. Pertama, arsitektur WSL yang terintegrasi erat dengan sistem host Windows memungkinkan akses yang lebih langsung ke sumber daya perangkat keras. WSL 2 menggunakan teknologi virtualisasi ringan dengan kernel Linux yang dioptimalkan, yang mengurangi overhead dibandingkan dengan virtualisasi penuh pada Virtual Machine. Kedua, WSL dapat memanfaatkan optimasi sistem yang sudah ada pada Windows, seperti manajemen memori yang efisien dan cache file sistem yang canggih. Ketiga, akses ke sistem file NTFS secara langsung memberikan keunggulan signifikan pada operasi I/O dibandingkan dengan virtual disk image yang digunakan pada Virtual Machine.

Di sisi lain, Virtual Machine menunjukkan performa yang lebih rendah terutama karena overhead virtualisasi yang signifikan. Setiap permintaan dari sistem operasi tamu harus melalui lapisan hypervisor, yang menambah latency dan mengurangi throughput. Meskipun Virtual Machine memberikan isolasi yang lebih baik dan lingkungan yang lebih terpisah, hal ini datang dengan biaya performa yang harus dibayar, terutama untuk operasi yang intensif seperti komputasi CPU dan I/O disk.

Implikasi dari temuan ini penting untuk dipertimbangkan dalam konteks penggunaan praktis. Untuk pengembangan perangkat lunak dan tugas-tugas yang membutuhkan performa tinggi, WSL mungkin menjadi pilihan yang lebih baik karena memberikan akses yang lebih efisien ke sumber daya sistem. Namun, untuk skenario yang membutuhkan isolasi penuh, kompatibilitas perangkat keras tertentu, atau pengujian lingkungan yang benar-benar terpisah, Virtual Machine tetap memiliki nilai meskipun dengan performa yang lebih rendah. Penting juga untuk dicatat bahwa hasil ini spesifik untuk konfigurasi pengujian yang digunakan, dan performa relatif dapat bervariasi tergantung pada faktor-faktor seperti spesifikasi hardware, konfigurasi virtualisasi, dan jenis beban kerja.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan mengenai perbandingan performa antara Windows Subsystem for Linux (WSL) dan Linux Ubuntu pada Virtual Machine, dapat ditarik beberapa kesimpulan sebagai berikut:

Pertama, pada aspek performa CPU, WSL menunjukkan keunggulan yang signifikan dengan rata-rata kecepatan eksekusi sebesar 2130.15 events per second, sementara Virtual Machine hanya mencapai 1213.95 events per second. Ini menunjukkan bahwa WSL memiliki kemampuan komputasi yang hampir dua kali lipat lebih baik dibandingkan dengan lingkungan virtualisasi penuh.

Kedua, dalam pengujian performa memori, WSL kembali mengungguli Virtual Machine dengan rata-rata throughput sebesar 3427.24 MiB per second, dibandingkan dengan 2046.88 MiB per second pada Virtual Machine. Perbedaan ini menunjukkan bahwa WSL memiliki akses memori yang lebih efisien dan cepat, dengan keunggulan sekitar 67% dibandingkan dengan lingkungan virtualisasi.

Ketiga, pada pengujian performa disk, perbedaan antara kedua lingkungan menjadi paling mencolok. WSL mencapai rata-rata kecepatan baca sebesar 11.58 MiB per second dan kecepatan tulis sebesar 7.72 MiB per second, sementara Virtual Machine hanya mencapai 1.93 MiB per second untuk baca dan 1.29 MiB per second untuk tulis. Ini menunjukkan keunggulan WSL yang hampir enam kali lipat dalam operasi I/O disk.

Secara keseluruhan, penelitian ini membuktikan bahwa Windows Subsystem for Linux (WSL) memberikan performa yang secara signifikan lebih baik dibandingkan dengan Linux Ubuntu pada Virtual Machine dalam hal CPU, memori, dan disk I/O. Keunggulan WSL terutama disebabkan oleh arsitektur yang lebih efisien dengan overhead virtualisasi yang lebih rendah, serta integrasi yang lebih baik dengan sistem file dan sumber daya host Windows.

5.2 Saran

Berdasarkan hasil penelitian dan kesimpulan yang telah diperoleh, berikut adalah beberapa saran yang dapat dipertimbangkan:

Pertama, bagi pengguna yang membutuhkan performa tinggi untuk pengembangan perangkat lunak, komputasi ilmiah, atau tugas-tugas yang membutuhkan akses I/O intensif, disarankan untuk menggunakan Windows Subsystem for Linux (WSL) daripada Virtual Machine konvensional. WSL menawarkan keseimbangan yang baik antara performa dan kompatibilitas dengan ekosistem Linux.

Kedua, meskipun WSL menunjukkan performa yang lebih baik, Virtual Machine tetap memiliki nilai dalam skenario tertentu yang membutuhkan isolasi penuh, pengujian lingkungan yang benar-benar terpisah, atau kompatibilitas dengan perangkat keras tertentu. Pengguna harus mempertimbangkan kebutuhan spesifik mereka dalam memilih antara WSL dan Virtual Machine.

Ketiga, untuk penelitian lebih lanjut, disarankan untuk melakukan pengujian dengan variasi konfigurasi yang lebih luas, termasuk penggunaan hypervisor yang berbeda (seperti Hyper-V, VMware, VirtualBox), konfigurasi resource yang bervariasi, dan jenis beban kerja yang lebih beragam. Selain itu, pengujian dapat diperluas ke aspek lain seperti konsumsi daya, performa jaringan, dan kompatibilitas aplikasi.

Keempat, disarankan juga untuk melakukan pengujian dengan versi WSL yang lebih baru atau fitur-fitur khusus yang mungkin tersedia, serta membandingkan dengan teknologi kontainer seperti Docker yang juga populer untuk isolasi lingkungan pengembangan.

Kelima, bagi pengembang dan administrator sistem, disarankan untuk melakukan evaluasi performa berdasarkan kasus penggunaan spesifik mereka, karena hasil benchmarking mungkin bervariasi tergantung pada konfigurasi sistem dan jenis aplikasi yang digunakan.

DAFTAR PUSTAKA

Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.

Microsoft. (2023). *Windows Subsystem for Linux Documentation*. Diakses dari https://docs.microsoft.com/en-us/windows/wsl/

Ubuntu. (2023). *Ubuntu 22.04 LTS Documentation*. Diakses dari https://ubuntu.com/server/docs

Sysbench Documentation. (2023). https://github.com/akopytov/sysbench

VMware. (2023). *Virtual Machine Performance Guidelines*. Diakses dari https://docs.vmware.com/

VirtualBox. (2023). *User Manual*. Diakses dari https://www.virtualbox.org/manual/

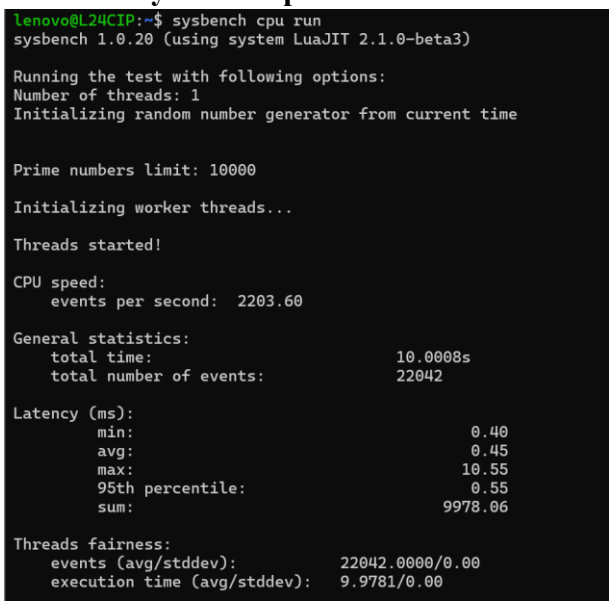
LAMPIRAN

PERCOBAAN KE-1

1. PENGUJIAN WINDOWS WLS (UBUNTU)

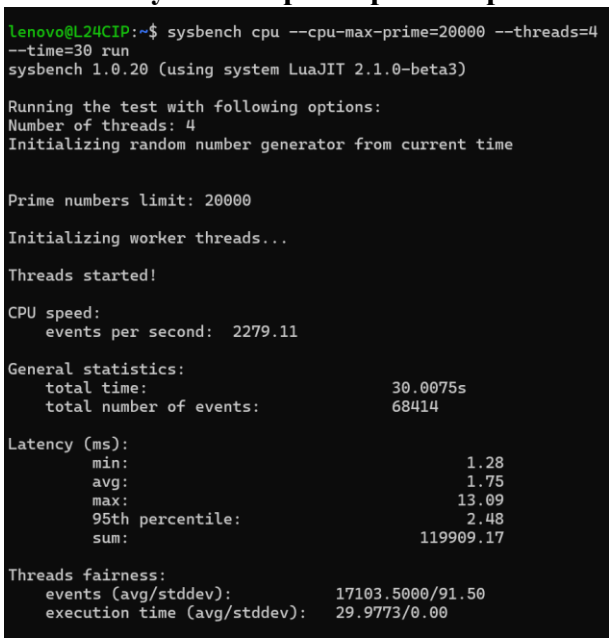
1.1 CPU Benchmark

Jalankan sysbench cpu run



```
lenovo@L24CIP:~$ sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2203.60

General statistics:
    total time:                          10.0008s
    total number of events:              22042

Latency (ms):
         min:                                    0.40
         avg:                                    0.45
         max:                                   10.55
         95th percentile:                        0.55
         sum:                                 9978.06

Threads fairness:
    events (avg/stddev):           22042.0000/0.00
    execution time (avg/stddev):   9.9781/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run



```
lenovo@L24CIP:~$ sysbench cpu --cpu-max-prime=20000 --threads=4
--time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2279.11

General statistics:
    total time:                          30.0075s
    total number of events:              68414

Latency (ms):
         min:                                    1.28
         avg:                                    1.75
         max:                                   13.09
         95th percentile:                        2.48
         sum:                               119909.17

Threads fairness:
    events (avg/stddev):           17103.5000/91.50
    execution time (avg/stddev):   29.9773/0.00
```

1.2 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
lenovo@L24CIP:~$ sysbench cpu --cpu-max-prime=20000 --threads=4
--time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2279.11

General statistics:
    total time:                          30.0075s
    total number of events:              68414

Latency (ms):
         min:                                    1.28
         avg:                                    1.75
         max:                                   13.09
         95th percentile:                        2.48
         sum:                               119909.17

Threads fairness:
    events (avg/stddev):           17103.5000/91.50
    execution time (avg/stddev):   29.9773/0.00
```

1.3 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
lenovo@L24CIP:~$ sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Reusing existing file test_file.0
Reusing existing file test_file.1
Reusing existing file test_file.2
Reusing existing file test_file.3
Reusing existing file test_file.4
Creating file test_file.5
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2063597568 bytes written in 10.53 seconds (186.93 MiB/sec).
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
lenovo@L24CIP:~$ sysbench fileio --file-test-mode=rndrw run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      565.78
    writes/s:                     377.19
    fsyncs/s:                     1209.88

Throughput:
    read, MiB/s:                  8.84
    written, MiB/s:               5.89

General statistics:
    total time:                          10.0710s
    total number of events:              21561

Latency (ms):
         min:                                    0.01
         avg:                                    0.46
         max:                                    8.05
         95th percentile:                        0.87
         sum:                                 9956.19

Threads fairness:
    events (avg/stddev):           21561.0000/0.00
    execution time (avg/stddev):   9.9562/0.00
```

jalankan sysbench fileio cleanup (membersihkan file)

```
No bytes written.
lenovo@L24CIP:~$ sysbench fileio cleanup
```

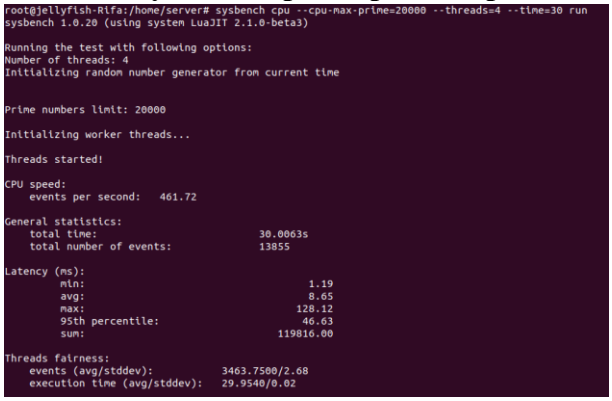
2. PENGUJIAN LINUX UBUNTU

1.1 CPU Benchmark

Jalankan sysbench cpu run

```
server@Rifa-VirtualBox:~$ sysbench --version
sysbench 1.0.20
server@Rifa-VirtualBox:~$ sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1212.02

General statistics:
    total time:                          10.0003s
    total number of events:              12122

Latency (ms):
         min:                                    0.44
         avg:                                    0.82
         max:                                   62.70
         95th percentile:                        4.10
         sum:                                 9976.88

Threads fairness:
    events (avg/stddev):           12122.0000/0.00
    execution time (avg/stddev):   9.9769/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run



```
root@jellyfish-Rifa:/home/server# sysbench cpu --cpu-max-prime=20000 --threads=4 --time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   461.72

General statistics:
    total time:                          30.0063s
    total number of events:              13855

Latency (ms):
         min:                                    1.19
         avg:                                    8.65
         max:                                  128.12
         95th percentile:                       46.63
         sum:                                119816.00

Threads fairness:
    events (avg/stddev):           3463.7500/2.68
    execution time (avg/stddev):   29.9540/0.02
```

1.2 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
root@jellyfish-Rifa:/home/server# sysbench memory run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 17421144 (1741534.60 per second)

17012.84 MiB transferred (1700.72 MiB/sec)


General statistics:
    total time:                          10.0022s
    total number of events:              17421144

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   24.79
         95th percentile:                        0.00
         sum:                                 4238.99

Threads fairness:
    events (avg/stddev):           17421144.0000/0.00
    execution time (avg/stddev):   4.2390/0.00
```

1.3 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Reusing existing file test_file.0
Reusing existing file test_file.1
Reusing existing file test_file.2
Reusing existing file test_file.3
Reusing existing file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
```

```
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2063597568 bytes written in 35.58 seconds (55.32 MiB/sec).
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-test-mode=rndrw run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      110.01
    writes/s:                     73.34
    fsyncs/s:                     234.79

Throughput:
    read, MiB/s:                  1.72
    written, MiB/s:               1.15

General statistics:
    total time:                          10.3612s
    total number of events:              4205

Latency (ms):
         min:                                    0.01
         avg:                                    2.38
         max:                                   39.05
         95th percentile:                        5.77
         sum:                                 9990.95

Threads fairness:
    events (avg/stddev):           4205.0000/0.00
    execution time (avg/stddev):   9.9909/0.00
```

Jalankan sysbench fileio cleanup (membersihkan file)

```
root@jellyfish-Rifa:/home/server# sysbench fileio cleanup
```

PERCOBAAN KE-2

1. PENGUJIAN WINDOWS WLS (UBUNTU)

1.1 CPU Benchmark

Jalankan sysbench cpu run

```
lenovo@L24CIP:~$ sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1961.46

General statistics:
    total time:                          10.0003s
    total number of events:              19619

Latency (ms):
         min:                                    0.46
         avg:                                    0.51
         max:                                   11.45
         95th percentile:                        0.61
         sum:                                 9976.18

Threads fairness:
    events (avg/stddev):           19619.0000/0.00
    execution time (avg/stddev):   9.9762/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run

```
lenovo@L24CIP:~$ sysbench cpu --cpu-max-prime=20000 --threads=4 --tim
e=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2421.26

General statistics:
    total time:                          30.0012s
    total number of events:              72647

Latency (ms):
         min:                                    1.17
         avg:                                    1.65
         max:                                   20.27
         95th percentile:                        2.26
         sum:                               119926.19

Threads fairness:
    events (avg/stddev):           18161.7500/177.20
    execution time (avg/stddev):   29.9815/0.00
```

1.2 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
lenovo@L24CIP:~$ sysbench memory run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 31908898 (3190264.56 per second)

31161.03 MiB transferred (3115.49 MiB/sec)


General statistics:
    total time:                          10.0002s
    total number of events:              31908898

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    3.45
         95th percentile:                        0.00
         sum:                                 4012.05

Threads fairness:
    events (avg/stddev):           31908898.0000/0.00
    execution time (avg/stddev):   4.0121/0.00

lenovo@L24CIP:~$
```

1.3 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
lenovo@L24CIP:~$ sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Reusing existing file test_file.0
Reusing existing file test_file.1
Reusing existing file test_file.2
Reusing existing file test_file.3
Reusing existing file test_file.4
Reusing existing file test_file.5
Reusing existing file test_file.123
Reusing existing file test_file.124
Reusing existing file test_file.125
Reusing existing file test_file.126
Reusing existing file test_file.127
No bytes written.
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
lenovo@L24CIP:~$ sysbench fileio --file-test-mode=rndrw run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      601.32
    writes/s:                     400.88
    fsyncs/s:                     1284.70

Throughput:
    read, MiB/s:                  9.40
    written, MiB/s:               6.26

General statistics:
    total time:                          10.0757s
    total number of events:              22919

Latency (ms):
         min:                                    0.01
         avg:                                    0.43
         max:                                   15.43
         95th percentile:                        0.84
         sum:                                 9958.59

Threads fairness:
    events (avg/stddev):           22919.0000/0.00
    execution time (avg/stddev):   9.9586/0.00
```

Jalankan sysbench fileio cleanup (membersihkan file)

```
lenovo@L24CIP:~$ sysbench fileio clenup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Unknown command: clenup
```

2. PENGUJIAN LINUX UBUNTU

2.1 CPU Benchmark

Jalankan sysbench cpu run

```
root@jellyfish-Rifa:/home/server# sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1279.38

General statistics:
    total time:                          10.0023s
    total number of events:              12798

Latency (ms):
         min:                                    0.44
         avg:                                    0.78
         max:                                   28.50
         95th percentile:                        3.62
         sum:                                 9976.64

Threads fairness:
    events (avg/stddev):           12798.0000/0.00
    execution time (avg/stddev):   9.9766/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run

```
root@jellyfish-Rifa:/home/server# sysbench cpu --cpu-max-prime=20000 --threads=4 --time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   549.14

General statistics:
    total time:                          30.0028s
    total number of events:              16480

Latency (ms):
         min:                                    1.11
         avg:                                    7.27
         max:                                  110.34
         95th percentile:                       44.17
         sum:                               119890.56

Threads fairness:
    events (avg/stddev):           4120.0000/10.61
    execution time (avg/stddev):   29.9726/0.01
```

2.2 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
root@jellyfish-Rifa:/home/server# sysbench memory run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 22660616 (2265777.26 per second)

22129.51 MiB transferred (2212.67 MiB/sec)


General statistics:
    total time:                          10.0001s
    total number of events:              22660616

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   18.71
         95th percentile:                        0.00
         sum:                                 3618.39

Threads fairness:
    events (avg/stddev):           22660616.0000/0.00
    execution time (avg/stddev):   3.6184/0.00
```

2.3 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2147483648 bytes written in 34.16 seconds (59.96 MiB/sec).
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-test-mode=rndrw
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
```

```
Threads started!


File operations:
    reads/s:                      134.43
    writes/s:                     89.62
    fsyncs/s:                     293.32

Throughput:
    read, MiB/s:                  2.10
    written, MiB/s:               1.40

General statistics:
    total time:                          10.2643s
    total number of events:              5183

Latency (ms):
         min:                                    0.01
         avg:                                    1.93
         max:                                   38.58
         95th percentile:                        5.37
         sum:                                 9977.38

Threads fairness:
    events (avg/stddev):           5183.0000/0.00
    execution time (avg/stddev):   9.9774/0.00
```

Jalankan sysbench fileio cleanup (membersihkan file)

```
root@jellyfish-Rifa:/home/server# sysbench fileio cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)
```

PERCOBAAN KE-3

1. PENGUJIAN WINDOWS WLS (UBUNTU)

1.1 CPU Benchmark

Jalankan sysbench cpu run

```
lenovo@L24CIP:~$ sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2225.39

General statistics:
    total time:                          10.0005s
    total number of events:              22258

Latency (ms):
         min:                                    0.44
         avg:                                    0.45
         max:                                    0.88
         95th percentile:                        0.48
         sum:                                 9989.60

Threads fairness:
    events (avg/stddev):           22258.0000/0.00
    execution time (avg/stddev):   9.9896/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run

```
lenovo@L24CIP:~$ sysbench cpu --cpu-max-prime=20000 --threads=4 --time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3373.89

General statistics:
    total time:                          30.0008s
    total number of events:              101227

Latency (ms):
         min:                                    1.11
         avg:                                    1.18
         max:                                   13.30
         95th percentile:                        1.23
         sum:                               119938.75

Threads fairness:
    events (avg/stddev):           25306.7500/370.82
    execution time (avg/stddev):   29.9847/0.00
```

1.4 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
lenovo@L24CIP:~$ sysbench memory run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 36697935 (3669100.29 per second)

35837.83 MiB transferred (3583.11 MiB/sec)


General statistics:
    total time:                          10.0004s
    total number of events:              36697935

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    0.24
         95th percentile:                        0.00
         sum:                                 3745.17

Threads fairness:
    events (avg/stddev):           36697935.0000/0.00
    execution time (avg/stddev):   3.7452/0.00
```

1.5 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
lenovo@L24CIP:~$ sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2147483648 bytes written in 10.18 seconds (201.20 MiB/sec).
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
lenovo@L24CIP:~$ sysbench fileio --file-test-mode=rndrw run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      1056.66
    writes/s:                     704.44
    fsyncs/s:                     2260.68

Throughput:
    read, MiB/s:                  16.51
    written, MiB/s:               11.01

General statistics:
    total time:                          10.0485s
    total number of events:              40293

Latency (ms):
         min:                                    0.00
         avg:                                    0.25
         max:                                    4.82
         95th percentile:                        0.58
         sum:                                 9962.72

Threads fairness:
    events (avg/stddev):           40293.0000/0.00
    execution time (avg/stddev):   9.9627/0.00
```

Jalankan sysbench fileio cleanup (membersihkan file)

```
lenovo@L24CIP:~$ sysbench fileio cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

1 PENGUJIAN LINUX UBUNTU

1.2 CPU Benchmark

Jalankan sysbench cpu run

```
root@jellyfish-Rifa:/home/server# sysbench cpu run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1150.45

General statistics:
    total time:                          10.0004s
    total number of events:              11506

Latency (ms):
         min:                                    0.44
         avg:                                    0.87
         max:                                   36.60
         95th percentile:                        3.75
         sum:                                 9987.65

Threads fairness:
    events (avg/stddev):           11506.0000/0.00
    execution time (avg/stddev):   9.9876/0.00
```

Jalankan sysbench cpu - -cpu-max-prime=20000 - -threads=4 - -time=30 run

```
root@jellyfish-Rifa:/home/server# sysbench cpu --cpu-max-prime=20000 --threads=4 --t
ime=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   537.54

General statistics:
    total time:                          30.0026s
    total number of events:              16128

Latency (ms):
         min:                                    1.12
         avg:                                    7.42
         max:                                  123.14
         95th percentile:                       44.98
         sum:                                119681.66

Threads fairness:
    events (avg/stddev):           4032.0000/9.57
    execution time (avg/stddev):   29.9204/0.02
```

1.3 Memory Benchmark (RAM)

Jalankan sysbench memory run

```
root@jellyfish-Rifa:/home/server# sysbench memory run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 22814388 (2280688.95 per second)

22279.68 MiB transferred (2227.24 MiB/sec)


General statistics:
    total time:                          10.0020s
    total number of events:              22814388

Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   13.34
         95th percentile:                        0.00
         sum:                                 3665.32

Threads fairness:
    events (avg/stddev):           22814388.0000/0.00
    execution time (avg/stddev):   3.6653/0.00
```

1.4 File I/O Benchmark

Jalakan sysbench fileio - -file-total-size=2G prepare

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-total-size=2G prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2147483648 bytes written in 37.37 seconds (54.80 MiB/sec).
```

Jalakan benchmark disk sysbench fileio - -file-test-mode=rndrw run

```
root@jellyfish-Rifa:/home/server# sysbench fileio --file-test-mode=rndrw run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!
```

```
File operations:
    reads/s:                      125.85
    writes/s:                     83.90
    fsyncs/s:                     270.87

Throughput:
    read, MiB/s:                  1.97
    written, MiB/s:               1.31

General statistics:
    total time:                          10.4827s
    total number of events:              4913

Latency (ms):
         min:                                    0.00
         avg:                                    2.03
         max:                                   31.12
         95th percentile:                        6.21
         sum:                                 9971.57

Threads fairness:
    events (avg/stddev):           4913.0000/0.00
    execution time (avg/stddev):   9.9716/0.00
```

Jalankan sysbench fileio cleanup (membersihkan file)

```
root@jellyfish-Rifa:/home/server# sysbench fileio cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

TABEL DAN GRAFIK PERBANDINGAN

TABEL

1. PERBANDINGAN PENGUJIAN PERTAMA

Parameter Pengujian	Metrik	Windows WSL (Ubuntu)	Linux Ubuntu (VirtualBox)
CPU Benchmark	Speed (events/sec)	2203.60	1212.02
Memory (RAM)	Transfer (MiB/sec)	3583.11	1700.72
Disk I/O	Read (MiB/s)	8.84	1.72
Disk I/O	Written (MiB/s)	5.89	1.15

Analisis :

- CPU: Windows WSL unggul hampir 2x lipat (2203 vs 1212) dibandingkan Linux di VirtualBox.
- RAM: Throughput memori WSL jauh lebih tinggi (3583 MiB/s) dibandingkan VirtualBox (1700 MiB/s).
- Disk I/O: Performa disk pada Linux VirtualBox sangat rendah (1.72 MiB/s Read), kemungkinan karena *overhead* virtualisasi pada disk image.

2. PERBANDINGAN PENGUJIAN KEDUA

Parameter Pengujian	Metrik	Windows WSL (Percobaan 2)	Linux Ubuntu VM (Percobaan 2)
CPU Benchmark	Speed (events/sec)	1961.46	1279.38
Memory (RAM)	Transfer (MiB/sec)	3115.49	2212.67
Disk I/O	Read (MiB/s)	9.40	2.10
Disk I/O	Written (MiB/s)	6.26	1.40

Analisis:

Konsistensi terlihat di sini. Windows WSL tetap unggul di semua aspek. Linux VM mengalami sedikit kenaikan performa Disk Read (dari 1.72 di Percobaan 1 menjadi 2.10 di Percobaan 2), namun masih jauh tertinggal dibandingkan WSL.

3. PERBANDINGAN PENGUJIAN KETIGA

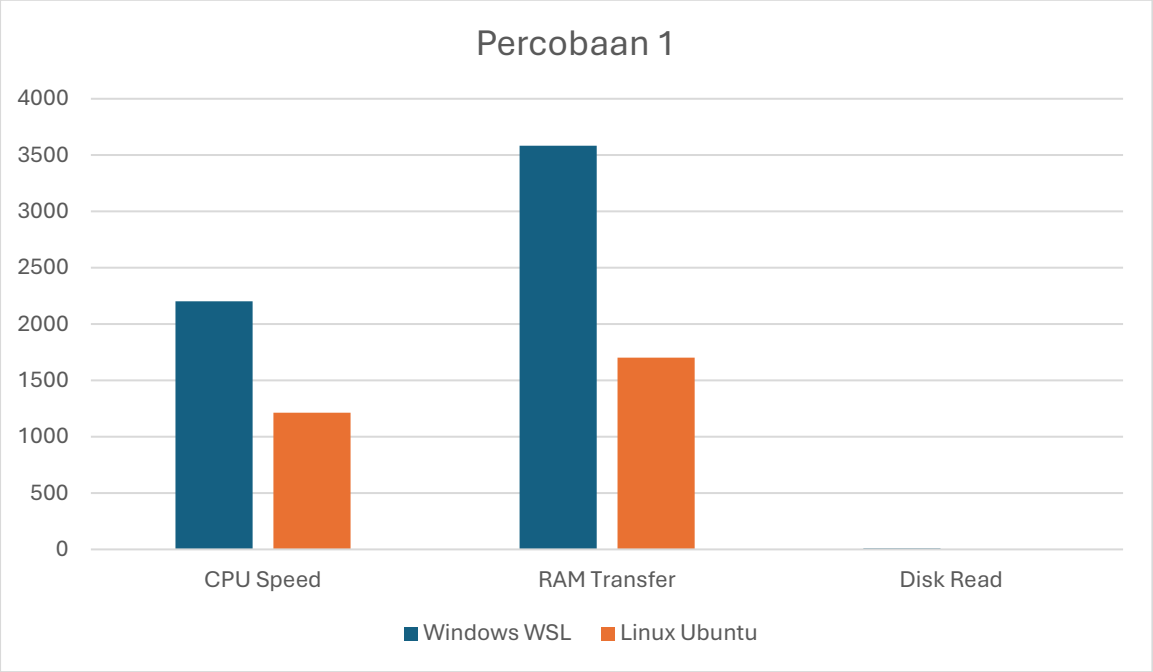
Parameter Pengujian	Metrik	Windows WSL (Percobaan 3)	Linux Ubuntu VM (Percobaan 3)
CPU Benchmark	Speed (events/sec)	2225.39	1150.45
Memory (RAM)	Transfer (MiB/sec)	3583.11	2227.24
Disk I/O	Read (MiB/s)	16.51	1.97
Disk I/O	Written (MiB/s)	11.01	1.31

Analisis :

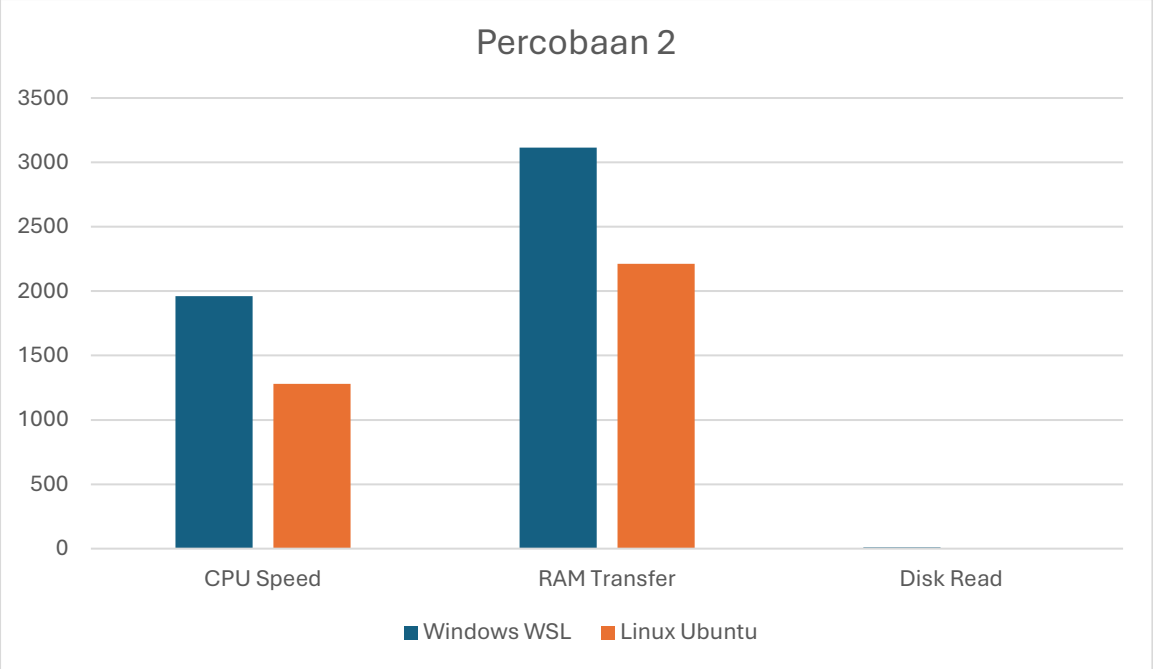
Pada percobaan ketiga ini, pola performa kembali terkonfirmasi. Windows WSL menunjukkan stabilitas performa yang tinggi (CPU ~2225 dan Disk Read ~16.51). Sementara itu, Linux di VirtualBox tertinggal cukup jauh, terutama pada sektor Disk I/O yang hanya mampu mencapai 1.97 MiB/s, menunjukkan adanya *bottleneck* (hambatan) yang signifikan pada virtualisasi storage.

GRAFIK

1. **CPU, RAM, DISK I/O (KE-1)**



2. **CPU, RAM, DISK I/O (KE-2)**



3. CPU, RAM, DISK I/O (KE-3)