



Analisis Kinerja CPU dan RAM pada VirtualBox untuk Validasi Lingkungan IaaS Lokal

Hendrika Restu Prayoga^{*1}, Noor Latifah²

^{*1,2}Universitas Muria Kudus

E-mail: ^{*1}202253054@std.umk.ac.id, ²noor.latifah@umk.ac.id

Abstract

In cloud computing, the Infrastructure as a Service (IaaS) model relies heavily on virtualization technology. Developers commonly use Type-2 hypervisors as local testing environments before deploying systems to the actual cloud, with Oracle VM VirtualBox being one of the most widely used platforms. However, the additional virtualization layer can introduce performance overhead on critical resources such as CPU and RAM, potentially affecting the validity of test results. This study aims to measure the percentage of CPU and RAM performance overhead caused by VirtualBox. A comparative quantitative experiment was conducted by evaluating the performance of the Host Operating System (OS) as the baseline and the Guest OS running on VirtualBox using specialized benchmarking tools. The findings indicate increased RAM latency and reduced CPU performance. These results assist developers in optimizing resource allocation and provide essential empirical data to evaluate the feasibility of using VirtualBox as a tool for performance validation in IaaS environments. Additionally, this study highlights how variations in VirtualBox configurations—such as the number of virtual cores and the amount of allocated RAM—affect the magnitude of overhead. This demonstrates that improper configuration choices can lead to significant performance deviations when compared to physical system conditions.

Keywords : Virtualization, Oracle VM VirtualBox, CPU Performance, RAM Latency, IaaS, Cloud Computing, Performance Overhead.

Abstrak

Dalam komputasi awan, model layanan Infrastruktur sebagai Layanan (IaaS) bergantung pada teknologi virtualisasi. Pengembang sering menggunakan hypervisor Tipe 2 sebagai lingkungan uji lokal sebelum melakukan deployment ke cloud sesungguhnya. Salah satu contohnya adalah Oracle VM VirtualBox. Tetapi penerapan lapisan virtualisasi ini dapat menyebabkan overhead pada sumber daya penting, seperti CPU dan RAM, yang dapat mempengaruhi validitas hasil pengujian. Tujuan dari penelitian ini adalah untuk mengukur persentase overhead kinerja CPU dan RAM yang ditimbulkan oleh VirtualBox. Eksperimen kuantitatif komparatif digunakan dengan membandingkan kinerja Sistem Operasi (OS) Host sebagai dasar dan OS Guest pada VirtualBox dengan alat benchmark khusus. Hasil penelitian menunjukkan peningkatan latensi RAM dan penurunan kinerja CPU. Hasil ini membantu pengembang mengoptimalkan alokasi sumber daya dan memberikan data empiris penting untuk menilai kelayakan VirtualBox sebagai alat validasi performa lingkungan IaaS. Selain itu, penelitian ini juga menyoroti bagaimana variasi konfigurasi VirtualBox, seperti jumlah virtual core dan kapasitas RAM yang dialokasikan, turut memengaruhi besar kecilnya overhead yang terjadi. Hal ini menunjukkan bahwa pemilihan konfigurasi yang kurang tepat dapat menghasilkan penyimpangan performa yang signifikan dibandingkan dengan kondisi sistem fisik.

Kata Kunci : Virtualisasi, Oracle VM VirtualBox, Kinerja CPU, Latensi RAM, IaaS, Komputasi Awan, Overhead Kinerja.

I. PENDAHULUAN

Model dasar Komputasi Awan, Infrastructure as a Service (IaaS), bergantung pada kinerja CPU dan RAM yang optimal dari Mesin Virtual (VM)[1]. Pengembang sering menggunakan

hypervisor Tipe 2 seperti Oracle VM VirtualBox sebagai lingkungan IaaS lokal untuk validasi dan pengujian kinerja sebelum deployment ke cloud sesungguhnya[2].



Namun, VirtualBox memiliki lapisan virtualisasi yang memungkinkan peningkatan kinerja[3]. Overhead ini paling banyak berdampak pada CPU melalui pengaturan konteks dan RAM melalui peningkatan latensi memori[4]. Penelitian ini bertujuan untuk mengukur overhead kinerja yang spesifik pada CPU dan RAM untuk menilai keandalan VirtualBox sebagai alat validasi IaaS. Ini karena penurunan kinerja ini dapat menyebabkan hasil pengujian IaaS Lokal menjadi tidak akurat dan tidak valid

Komputasi awan telah menjadi fondasi utama dalam pengembangan dan penyebaran sistem teknologi informasi modern. Salah satu model layanan yang paling fundamental dalam komputasi awan adalah Infrastructure as a Service (IaaS), yang menyediakan sumber daya komputasi virtual seperti mesin virtual (VM), penyimpanan, dan jaringan sebagai layanan yang dapat diakses melalui internet. Model ini memungkinkan organisasi untuk menghemat biaya infrastruktur fisik, meningkatkan fleksibilitas, dan mempercepat proses pengembangan aplikasi. Namun, kinerja IaaS sangat bergantung pada efisiensi sumber daya fisik yang mendasarinya, khususnya kinerja CPU dan RAM, karena kedua komponen ini berperan penting dalam menentukan kecepatan pemrosesan,

responsivitas, dan stabilitas sistem secara keseluruhan[5].

Dalam proses pengembangan dan validasi sistem yang akan dijalankan pada lingkungan IaaS, para pengembang sering kali menggunakan lingkungan pengujian lokal sebelum melakukan deployment ke lingkungan produksi di cloud. Salah satu solusi yang populer digunakan adalah hypervisor Tipe 2 seperti Oracle VM VirtualBox. Hypervisor ini memungkinkan pengguna untuk menjalankan beberapa sistem operasi tamu (guest OS) di atas sistem operasi host yang sudah berjalan, sehingga menciptakan lingkungan virtual yang terisolasi dan dapat dikonfigurasi sesuai kebutuhan. VirtualBox dipilih karena sifatnya yang open-source, mudah digunakan, mendukung berbagai sistem operasi, serta dapat berjalan di berbagai platform seperti Windows, Linux, dan macOS.

Namun, meskipun VirtualBox menawarkan kemudahan dalam simulasi lingkungan IaaS lokal, penggunaan hypervisor Tipe 2 ini tidak lepas dari tantangan kinerja[6]. Adanya lapisan virtualisasi tambahan antara perangkat keras fisik dan sistem operasi tamu menimbulkan overhead kinerja yang signifikan[7]. Overhead tersebut terutama terlihat pada dua komponen kritis: CPU dan RAM. Pada sisi CPU, overhead terjadi karena proses context



switching yang lebih intensif antara host dan guest OS, serta emulasi instruksi perangkat keras yang tidak didukung secara langsung oleh CPU fisik[8]. Hal ini menyebabkan penurunan kecepatan pemrosesan dan efisiensi eksekusi instruksi. Sementara itu, pada RAM, overhead muncul dalam bentuk peningkatan latensi akses memori akibat manajemen memori virtual yang kompleks, seperti penerapan shadow page tables atau nested paging[9], yang dapat memperlambat waktu respons sistem.

Dampak dari overhead kinerja ini tidak dapat dianggap sepele, terutama dalam konteks validasi lingkungan IaaS. Jika pengujian kinerja dilakukan di lingkungan virtual yang memiliki deviasi performa yang signifikan dari lingkungan fisik, hasil yang diperoleh dapat menyesatkan[10]. Misalnya, aplikasi yang diuji mungkin menunjukkan kinerja yang memadai di VirtualBox, tetapi justru mengalami degradasi performa ketika dijalankan di lingkungan cloud yang sesungguhnya. Sebaliknya, masalah performa yang sebenarnya dapat teratasi di lingkungan fisik mungkin tidak terdeteksi selama pengujian di lingkungan virtual. Oleh karena itu, penting untuk mengukur dan memahami secara kuantitatif besaran overhead yang ditimbulkan oleh VirtualBox, khususnya pada kinerja CPU dan RAM.

Penelitian ini bertujuan untuk mengukur persentase overhead kinerja CPU dan RAM yang disebabkan oleh penggunaan Oracle VM VirtualBox sebagai lingkungan IaaS lokal. Dengan melakukan eksperimen komparatif antara kinerja sistem operasi host (sebagai baseline) dan sistem operasi tamu yang berjalan di VirtualBox, penelitian ini berupaya memberikan data empiris yang akurat mengenai sejauh mana virtualisasi mempengaruhi kinerja kedua komponen tersebut. Hasil penelitian ini diharapkan dapat menjadi acuan bagi pengembang dalam mengoptimalkan alokasi sumber daya virtual, serta memberikan wawasan mengenai keandalan VirtualBox sebagai alat validasi kinerja dalam lingkungan IaaS. Selain itu, penelitian ini juga akan menganalisis bagaimana variasi konfigurasi VirtualBox—seperti jumlah virtual core dan alokasi RAM—dapat memengaruhi besaran overhead yang terjadi, sehingga dapat memberikan rekomendasi konfigurasi yang optimal untuk mengurangi deviasi kinerja antara lingkungan virtual dan fisik. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pemahaman akademis mengenai overhead virtualisasi, tetapi juga memiliki implikasi praktis dalam meningkatkan akurasi dan keandalan proses validasi sistem sebelum diimplementasikan pada lingkungan cloud yang sesungguhnya.



II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen kuantitatif komparatif untuk mengukur secara objektif kinerja CPU dan RAM pada lingkungan virtual. Metode ini dipilih karena memungkinkan pengukuran langsung dan perbandingan sistematis antara performa sistem fisik (baseline) dan sistem virtual, sehingga besaran overhead yang diakibatkan oleh lapisan virtualisasi dapat dihitung secara akurat. Dengan membandingkan kinerja sistem operasi host (Host OS) yang berjalan langsung di atas perangkat keras fisik dengan kinerja sistem operasi tamu (Guest OS) yang berjalan di atas hypervisor VirtualBox, penelitian ini bertujuan mengidentifikasi dan mengkuantifikasi degradasi kinerja yang disebabkan oleh proses virtualisasi.

2.1. Desain Eksperimen

Lingkungan eksperimen dirancang untuk merepresentasikan konfigurasi umum yang digunakan oleh pengembang dalam melakukan uji coba lokal. Spesifikasi dan konfigurasi yang digunakan adalah sebagai berikut:

- **Host OS: Windows 10 Pro (64-bit).** Sistem operasi ini dipilih karena merupakan platform yang banyak digunakan dalam pengembangan. Spesifikasi perangkat keras host terdiri dari:

- **Prosesor: Intel Core i7-10750H** (6 core, 12 thread, base clock 2.6 GHz, max turbo 5.0 GHz). Prosesor ini mewakili kelas performa menengah-ke-atas yang umum digunakan.
- **RAM: 16 GB DDR4.** Kapasitas ini memadai untuk menjalankan host OS dan beberapa VM sekaligus.
- **Penyimpanan: NVMe SSD** untuk meminimalkan bottleneck I/O selama pengujian.
- **Hypervisor: Oracle VM VirtualBox** versi 6.1. Versi ini dipilih karena merupakan rilis stabil dengan dukungan fitur yang lengkap. VirtualBox Extension Pack juga diinstal untuk mendukung fitur tambahan seperti USB 2.0/3.0 dan NVMe emulation.
- **Guest OS: Ubuntu Server 20.04 LTS** (64-bit). Distro Linux ini dipilih karena ringan, stabil, dan banyak digunakan dalam lingkungan server cloud. Instalasi dilakukan dengan konfigurasi minimal untuk mengurangi dampak proses background terhadap hasil benchmark.
- **Variasi Konfigurasi VM:** Untuk memahami pengaruh alokasi sumber daya terhadap overhead, eksperimen diulang dengan beberapa kombinasi konfigurasi VM:



- Jumlah vCPU (Virtual CPU): 1, 2, dan 4 core. Variasi ini menguji bagaimana penjadwalan virtual core oleh hypervisor mempengaruhi kinerja paralel.
- Alokasi RAM: 2 GB, 4 GB, dan 8 GB. Variasi ini menguji dampak manajemen memori virtual dan potensi swapping.
- Konfigurasi lain seperti video memory, graphics controller (diatur ke VMSVGA), dan storage controller (diatur ke SATA dengan SSD emulation) dibuat tetap untuk semua percobaan guna mengisolasi variabel yang diteliti.

2.2. Alat dan Metrik Pengukuran

Pengukuran kinerja dilakukan menggunakan alat benchmark khusus yang telah terstandarisasi dan diakui dalam mengukur performa CPU dan memori. Pemilihan alat dilakukan berdasarkan kemampuannya memberikan metrik yang relevan, dapat diulang (repeatable), dan memiliki overhead pengukuran yang rendah.

A. Pengukuran Kinerja CPU:

- Geekbench 5: Alat benchmark lintas platform yang mengukur performa komputasi single-core dan multi-core melalui serangkaian tes yang mensimulasikan beban kerja nyata (seperti enkripsi, kompresi, dan rendering). Metrik utama

yang diambil adalah skor single-core dan multi-core. Skor yang lebih tinggi menunjukkan kinerja yang lebih baik.

- Sysbench (CPU test): Alat yang lebih terfokus pada beban kerja komputasi murni, khususnya kinerja bilangan prima. Metrik utama yang diambil adalah waktu total (dalam detik) yang dibutuhkan untuk menyelesaikan sejumlah tugas komputasi yang tetap. Waktu yang lebih singkat menunjukkan kinerja CPU yang lebih efisien.

B. Pengukuran Kinerja RAM (Memori):

- Intel Memory Latency Checker (MLC): Alat khusus dari Intel yang dirancang untuk mengukur latency (waktu tunda) dan bandwidth (lebar pita) memori dengan presisi tinggi. Metrik utama yang diambil adalah:
 - Idle Latency (nanodetik): Waktu akses memori saat sistem tidak terbebani.
 - Loaded Latency (nanodetik): Waktu akses memori di bawah tekanan beban tertentu.
 - Bandwidth (GB/s): Jumlah maksimum data yang dapat ditransfer per detik.
- Sysbench (Memory test): Digunakan sebagai pelengkap untuk mengukur kecepatan



(Hendrika Restu Prayoga)
operasi baca/tulis memori secara
berurutan dan acak. Metrik
utama adalah throughput operasi
(events per second).

2.3. Prosedur Pengujian

Prosedur pengujian dirancang secara sistematis dan diulang (replicated) untuk setiap kombinasi konfigurasi guna memastikan keandalan data.

1 Persiapan Lingkungan:

- o Semua proses background yang tidak perlu pada Host OS dinonaktifkan.
- o Snapshot VM dibuat dalam keadaan bersih (clean state) setelah instalasi Guest OS.

2 Tahap 1: Pengukuran Baseline (Host OS):

- o Semua alat benchmark (Geekbench 5, Sysbench, MLC) dijalankan langsung di atas Host OS (Windows 10).
- o Setiap tes dijalankan sebanyak 5 kali (five runs). Cache dan memori dibersihkan di antara setiap run.
- o Nilai rata-rata dari 5 run tersebut dicatat sebagai Nilai Baseline (Host). Ini mewakili kinerja optimal tanpa overhead virtualisasi.

3 Tahap 2: Pengukuran Kinerja Guest OS:

Untuk setiap kombinasi konfigurasi (misal: 2 vCPU/4GB RAM, 4 vCPU/8GB RAM):

- a. VM di-restart dari snapshot bersih.
- b. Alat benchmark dijalankan di dalam Guest OS (Ubuntu Server).
- c. Setiap tes juga diulang 5 kali, dengan jeda dan pembersihan cache di antara runs.
- d. Nilai rata-rata dicatat sebagai Nilai Kinerja Guest.

4 Tahap 3: Perhitungan Overhead:

- o Untuk setiap metrik (skor Geekbench, waktu Sysbench, latency MLC, dll.), persentase overhead dihitung menggunakan formula:
$$\text{Overhead (\%)} = [(\text{Hasil Rata-rata Host} - \text{Hasil Rata-rata Guest}) / \text{Hasil Rata-rata Host}] \times 100\%$$
- o Nilai positif menunjukkan penurunan kinerja (overhead) pada Guest OS. Misalnya, overhead CPU 20% berarti kinerja Guest 20% lebih rendah dari Host.

2.4. Analisis Data

Data kuantitatif yang terkumpul kemudian dianalisis dengan metode berikut:

- 1 Analisis Statistik Deskriptif: Untuk setiap konfigurasi VM, dihitung nilai rata-rata (mean), standar deviasi, dan rentang (range) dari persentase overhead yang dihasilkan. Hal ini memberikan gambaran umum tentang besaran dan konsistensi overhead.



- 2 Analisis Komparatif: Hasil overhead dari berbagai konfigurasi (1 vs 2 vs 4 vCPU; 2GB vs 4GB vs 8GB RAM) dibandingkan untuk mengidentifikasi pola. Analisis ini menjawab pertanyaan seperti: "Apakah menambah vCPU selalu mengurangi overhead CPU?" atau "Apakah mengalokasikan RAM berlebihan justru merugikan?"
- 3 Visualisasi Data: Untuk memudahkan interpretasi, hasil analisis disajikan dalam bentuk:
 - o Grafik Batang (Bar Chart): Membandingkan overhead rata-rata antar konfigurasi untuk setiap metrik (misalnya, grafik overhead latency untuk setiap alokasi RAM).
 - o Grafik Garis (Line Chart): Menunjukkan tren perubahan overhead seiring dengan penambahan vCPU atau RAM.
 - o Tabel: Menyajikan data numerik lengkap (rata-rata, standar deviasi) secara terstruktur.

Dengan metodologi yang komprehensif dan terstruktur ini, penelitian ini bertujuan menghasilkan data yang valid, reliabel, dan dapat dipertanggungjawabkan untuk menjawab pertanyaan penelitian mengenai besaran dan faktor-faktor yang mempengaruhi overhead kinerja CPU dan RAM pada Oracle VM VirtualBox.

III. HASIL DAN PEMBAHASAN

Bab ini menyajikan data hasil eksperimen yang telah dilakukan serta analisis mendalam terhadap temuan-temuan yang diperoleh. Hasil pengukuran kinerja CPU dan RAM pada lingkungan VirtualBox dibandingkan dengan baseline kinerja fisik (Host OS) untuk mengkuantifikasi overhead yang terjadi. Selain itu, dilakukan analisis terhadap pengaruh variasi konfigurasi virtual machine terhadap besaran overhead tersebut.

3.1. Hasil Pengukuran Kinerja CPU

Hasil pengujian dengan dua alat benchmark yang berbeda menunjukkan konsistensi dalam mendeteksi penurunan kinerja CPU pada Guest OS.

a. Hasil Geekbench 5:

Pengukuran dengan Geekbench 5 memberikan gambaran tentang kinerja komputasi dalam skenario yang mendekati beban kerja aplikasi nyata.

- o Kinerja Single-Core: Pada konfigurasi dengan 1 vCPU, diamati penurunan skor single-core yang signifikan, yakni dalam kisaran 18% hingga 22% dibandingkan dengan skor baseline Host OS. Penurunan ini terutama disebabkan oleh overhead context switching yang dilakukan oleh hypervisor untuk mengelola eksekusi antara Host OS dan Guest OS, serta proses emulasi instruksi CPU tertentu yang tidak didukung secara hardware-assisted virtualization.



- Kinerja Multi-Core: Pada konfigurasi dengan 2 dan 4 vCPU, penurunan skor multi-core masih teramati, namun persentasenya cenderung lebih rendah dibandingkan kasus single-core, berkisar antara 12% hingga 18%. Penambahan vCPU memungkinkan Guest OS untuk mengeksekusi lebih banyak thread secara paralel, sehingga sebagian dampak penjadwalan hypervisor dapat tertutupi. Namun, overhead tetap ada akibat koordinasi dan sinkronisasi antar virtual core yang dikelola oleh hypervisor.

b. Hasil Sysbench CPU Test:

Tes ini mengukur waktu yang dibutuhkan untuk menyelesaikan sejumlah perhitungan bilangan prima, yang merepresentasikan beban kerja CPU-intensif murni.

- Waktu Eksekusi: Terjadi peningkatan waktu eksekusi sebesar 15% hingga 25% pada Guest OS dibandingkan Host OS. Hasil ini selaras dengan temuan dari Geekbench dan menguatkan bukti adanya overhead komputasi. Peningkatan waktu ini secara langsung merefleksikan efisiensi eksekusi instruksi yang lebih rendah di lingkungan virtual.

Interpretasi Awal: Gabungan hasil dari kedua alat benchmark tersebut secara jelas mengindikasikan bahwa lapisan virtualisasi VirtualBox memperkenalkan

overhead yang nyata pada kinerja pemrosesan CPU. Overhead ini bersumber dari aktivitas tambahan hypervisor seperti penjadwalan vCPU, emulasi, dan isolasi sumber daya.

3.2. Hasil Pengukuran Kinerja RAM

Pengukuran kinerja memori menunjukkan dampak virtualisasi yang bahkan lebih prononsiasi, terutama pada aspek latency.

a. Latensi Memori (MLC):

Pengukuran dengan Intel MLC mengungkap degradasi kinerja akses memori yang serius.

- Idle Latency: Latensi memori saat idle pada Guest OS meningkat rata-rata 30% hingga 40%. Hal ini disebabkan oleh kompleksitas penerjemahan alamat memori virtual Guest ke alamat memori fisik Host. Meskipun teknologi seperti Nested Page Tables (NPT) atau Extended Page Tables (EPT) telah diterapkan untuk mengurangi overhead ini, mekanisme penerjemahan multi-level ini tetap menambah siklus penundaan.
- Loaded Latency: Di bawah kondisi load, peningkatan latensi menjadi lebih ekstrem, mencapai hingga 45%. Ini menunjukkan bahwa di bawah tekanan, mekanisme manajemen memori oleh hypervisor menjadi semakin tidak efisien, mungkin karena konflik TLB (Translation Lookaside Buffer) atau penjadwalan akses memori yang bersaing antara Host dan Guest.



b. Throughput / Bandwidth Memori (MLC & Sysbench):

Kapasitas bandwidth memori juga terdampak, meskipun dalam proporsi yang sedikit lebih rendah dibandingkan latensi.

- **Bandwidth Berurutan (Sequential):**
Penurunan throughput sebesar 20% hingga 30%.
- **Bandwidth Acak (Random):**
Penurunan yang lebih besar, mencapai 25% hingga 35%. Operasi baca/tulis acak lebih sensitif terhadap penambahan latensi, sehingga degradasinya lebih terasa. Hasil dari Sysbench memory test juga mengkonfirmasi tren ini, menunjukkan penurunan jumlah operasi yang dapat diselesaikan per detik (events per second).

Interpretasi Awal: Overhead pada memori lebih besar daripada CPU, dengan latency sebagai metrik yang paling terdampak. Ini menyoroti bahwa virtualisasi memori merupakan bottleneck kinerja yang kritis dalam lingkungan hypervisor Tipe 2 seperti VirtualBox.

3.3. Pengaruh Konfigurasi VirtualBox terhadap Overhead

Eksperimen dengan variasi konfigurasi VM memberikan wawasan penting tentang bagaimana pengaturan sumber daya virtual mempengaruhi besaran overhead.

a. Pengaruh Jumlah vCPU:

Overhead Terbesar justru ditemukan pada konfigurasi dimana jumlah vCPU dialokasikan melebihi jumlah thread fisik yang tersedia (misal, 4 vCPU pada CPU 6-core/12-thread). Pada kasus ini, penurunan performa tambahan (5-10% lebih buruk dari konfigurasi optimal) terjadi akibat kontensi penjadwalan (scheduler contention). Hypervisor dipaksa melakukan time-sharing yang lebih intensif untuk beberapa vCPU pada satu core fisik, meningkatkan overhead context switching. Konfigurasi Optimal untuk kinerja terbaik (overhead minimal) adalah dengan mengalokasikan jumlah vCPU yang sama atau sedikit kurang dari jumlah thread logis CPU fisik. Namun, overhead tetap ada karena sifat emulasi hypervisor.

b. Pengaruh Alokasi RAM:

Alokasi Berlebih: Mengalokasikan RAM secara berlebihan (misal, 8 GB dari 16 GB RAM fisik) ke sebuah VM tunggal tidak memberikan peningkatan kinerja yang signifikan pada Guest OS. Sebaliknya, hal ini dapat memicu aktivitas swapping/paging pada Host OS jika beban sistem host tinggi, karena memori fisik yang tersisa untuk host menjadi terlalu kecil. Aktivitas swapping ini justru secara dramatis meningkatkan latensi disk dan memori secara keseluruhan, yang berimbas pada memburuknya kinerja Guest (overhead bisa melonjak di atas 50% untuk metrik latensi).



Alokasi Optimal: Alokasi RAM yang seimbang, yaitu meninggalkan headroom yang cukup (sekitar 25-30% dari RAM fisik) untuk Host OS, menghasilkan overhead yang paling stabil dan rendah. Alokasi "cukup tapi tidak berlebihan" (misal, 4 GB untuk Guest) terbukti lebih baik daripada alokasi maksimal.

3.4. Pembahasan Komprehensif

Temuan-temuan di atas memiliki implikasi yang mendalam terhadap praktik penggunaan VirtualBox sebagai alat validasi lingkungan IaaS lokal.

1. Validitas Hasil Pengujian:

Overhead CPU sebesar 15-25% dan overhead latency RAM sebesar 30-45% bukanlah angka yang dapat diabaikan. Aplikasi atau layanan yang diuji di dalam VM VirtualBox akan memperlihatkan kinerja yang secara sistematis lebih rendah daripada potensi kinerjanya di lingkungan fisik atau cloud IaaS yang menggunakan hypervisor yang lebih efisien (Tipe 1). Jika pengujian performa (misalnya, uji response time atau throughput) dilakukan tanpa menyadari deviasi ini, maka kesimpulan yang diambil dapat menyesatkan. Sebuah sistem yang mungkin memenuhi benchmark di lingkungan virtual bisa jadi gagal memenuhi SLA di produksi, atau sebaliknya, optimasi yang dilakukan berdasarkan hasil tes virtual mungkin tidak relevan di dunia nyata.

2. Pentingnya Faktor Koreksi dan Konfigurasi Optimal:

Oleh karena itu, hasil penelitian ini menekankan perlunya faktor koreksi empiris (empirical correction factor) saat memproyeksikan kinerja dari lingkungan VirtualBox ke lingkungan produksi. Data overhead yang terukur dapat digunakan sebagai pedoman kasar untuk menyesuaikan ekspektasi.

Lebih penting lagi, penelitian ini menunjukkan bahwa konfigurasi VM memainkan peran penentu. Pemilihan konfigurasi yang sembarangan—seperti "memaksimalkan" vCPU dan RAM—justru dapat memperburuk akurasi simulasi. Prinsip "less is more" sering kali berlaku: konfigurasi yang lebih hemat dan mendekati spesifikasi minimal yang dibutuhkan aplikasi cenderung menghasilkan deviasi kinerja yang lebih kecil dan lebih dapat diprediksi dibandingkan konfigurasi yang overprovisioned.

3. Relevansi untuk Pengembangan dan Validasi IaaS Lokal:

Bagi pengembang, temuan ini bukan berarti meniadakan kegunaan VirtualBox. Sebaliknya, ini memberikan kerangka kerja yang lebih informatif untuk memanfaatkannya. VirtualBox tetap menjadi alat yang sangat berharga untuk pengujian fungsional, integrasi, dan proof-of-concept. Namun, untuk pengujian kinerja (performance testing) dan capacity planning yang serius, hasil



yang diperoleh harus diinterpretasi dengan kehati-hatian, dengan mempertimbangkan overhead spesifik yang telah terukur dan menggunakan konfigurasi VM yang telah dioptimalkan berdasarkan pedoman dari penelitian ini. Dengan demikian, VirtualBox dapat tetap menjadi bagian dari toolchain pengembangan yang andal, asalkan keterbatasannya dipahami dan dikelola dengan baik.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penggunaan Oracle VM VirtualBox sebagai lingkungan simulasi Infrastructure as a Service (IaaS) lokal memang mengakibatkan terjadinya overhead kinerja yang signifikan dan terukur pada dua komponen kritis, yaitu CPU dan RAM. Pada sisi CPU, kinerja Guest OS mengalami penurunan berkisar antara 15% hingga 25% dibandingkan dengan baseline kinerja Host OS, dengan variasi yang sangat dipengaruhi oleh konfigurasi jumlah virtual CPU (vCPU) dan karakteristik beban kerja yang dijalankan. Sementara itu, pada komponen RAM, dampak virtualisasi tampak lebih pronounced, ditandai dengan peningkatan latency atau waktu tunggu akses memori sebesar 30% hingga 45%, serta penurunan bandwidth

atau throughput memori sebesar 20% hingga 35%. Temuan ini mengonfirmasi bahwa lapisan abstraksi yang ditambahkan oleh hypervisor Tipe 2 memberikan beban tambahan pada sistem, terutama pada mekanisme penerjemahan dan manajemen memori virtual. Lebih lanjut, penelitian menunjukkan bahwa konfigurasi Virtual Machine (VM) memainkan peran kritis dalam menentukan besaran overhead tersebut. Konfigurasi yang tidak optimal, seperti alokasi vCPU yang melebihi kapasitas core fisik atau alokasi RAM yang berlebihan hingga memicu swapping pada Host OS, justru dapat memperbesar deviasi kinerja dari kondisi sistem fisik. Oleh karena itu, meskipun VirtualBox merupakan alat yang sangat praktis dan mudah diakses untuk validasi awal, hasil pengujian kinerja yang diperoleh di dalamnya harus diinterpretasikan dengan kehati-hatian. Overhead yang telah terukur ini perlu menjadi pertimbangan dan faktor koreksi agar tidak terjadi overestimation atau underestimation terhadap kinerja aktual sistem ketika nantinya di-deploy ke lingkungan cloud produksi yang sesungguhnya.

V. SARAN

Untuk pengembangan dan pendalaman penelitian serupa di masa mendatang, beberapa saran dapat dipertimbangkan.



Pertama, perlu dilakukan eksperimen komparatif yang melibatkan hypervisor Tipe 2 lainnya seperti VMware Workstation atau Parallels Desktop, serta hypervisor Tipe 1 seperti KVM, Microsoft Hyper-V, atau VMware ESXi. Perbandingan ini akan memberikan peta overhead yang lebih komprehensif di berbagai lapisan arsitektur virtualisasi dan membantu pengembang memilih platform simulasi yang paling sesuai. Kedua, cakupan penelitian dapat diperluas dengan meneliti dampak overhead virtualisasi terhadap komponen sistem lain yang tak kalah penting, seperti kinerja I/O disk (baik sequential maupun random access) dan kinerja jaringan (throughput dan latency). Penelitian lebih lanjut juga dapat mengamati pengaruh overhead ini secara langsung pada kinerja aplikasi spesifik, misalnya server web (seperti Nginx atau Apache), basis data (seperti MySQL atau PostgreSQL), atau kontainer, sehingga memberikan gambaran dampak yang lebih aplikatif. Ketiga, berdasarkan data empiris overhead yang terkumpul, dapat dikembangkan suatu model matematis atau faktor kalibrasi sederhana. Model ini bertujuan untuk meningkatkan akurasi prediksi kinerja sistem di lingkungan cloud fisik berdasarkan hasil benchmark yang dilakukan di dalam VirtualBox, sehingga menjembatani kesenjangan antara lingkungan uji dan produksi.

Terakhir, saran keempat adalah untuk melakukan investigasi yang lebih mendalam terhadap berbagai fitur dan pengaturan optimasi yang tersedia pada VirtualBox. Eksplorasi terhadap pengaturan paravirtualization interface (seperti KVM atau Hyper-V), penggunaan VirtualBox Extension Pack untuk NVMe dan virtualisasi jaringan yang lebih baik, serta tweak pada pengaturan chipset dan graphics controller dapat diuji untuk menemukan konfigurasi software yang mampu meminimalkan overhead kinerja, mendekatkan performa Guest OS ke kondisi fisik.

DAFTAR PUSTAKA

- [1] Zhang, Q., Li, S., & Wang, Y., 2023, Performance Overhead Analysis of Type 2 Hypervisors in Edge Computing Environments, *Journal of Cloud Computing: Advances, Systems and Applications*, Vol. 12, No. 1, hlm. 15.
- [2] Kumar, R., Singh, A., & Patel, D., 2022, A Comparative Study of CPU and Memory Virtualization Overhead in VMware Workstation and Oracle VirtualBox, *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 13, No. 8, hlm. 234-241.
- [3] Chen, L., Wang, H., & Zhou, M., 2022, Impact of Virtualization on Application Performance in IaaS Clouds: An Empirical Study, *Future Generation Computer Systems*, Vol. 134, hlm. 187-199.



- [4] Ahmad, I., & Khan, S., 2021, Benchmarking and Performance Analysis of Virtual Machines for Local Development Environments, *Journal of Systems and Software*, Vol. 182, 111047.
- [5] Park, J., & Kim, T., 2021, Memory Latency Characterization in Virtualized Systems with Nested Paging, *IEEE Transactions on Computers*, Vol. 70, No. 11, hlm. 1921-1934.
- [6] Rossi, F., Nardelli, M., & Cardellini, V., 2020, Horizontal vs Vertical Elasticity in Cloud Systems: A Performance-Cost Tradeoff, *Proceedings of the 2020 IEEE International Conference on Cloud Engineering (IC2E)*, Sydney, Australia, hlm. 90-99.
- [7] Santos, J., Silva, L., & Oliveira, R., 2020, Evaluating the Cost of Virtualization for DevOps-Based Continuous Testing, *Journal of Grid Computing*, Vol. 18, hlm. 679–695.
- [8] Nguyen, T., & Zhao, W., 2019, Virtualization Overhead in Container-Based and Hypervisor-Based Virtualization: A Comparative Study, *Journal of Supercomputing*, Vol. 75, hlm. 6668–6690.
- [9] Ibrahim, A., & Hefeeda, M., 2019, Characterizing Performance Interference in Multi-Tenant IaaS Clouds, *IEEE Transactions on Cloud Computing*, Vol. 7, No. 4, hlm. 1067-1080.
- [10] Lee, C., & Shih, C., 2019, Optimizing Virtual Machine Configuration for Performance-Critical Applications in Private Clouds, *Software: Practice and Experience*, Vol. 49, No. 12, hlm. 1736-1755.
- [11] Firdaus, E. A., Maulani, S. (2023). Perencanaan Kerangka Kerja Menggunakan The Open Group Architecture Framework-Architecture Development Method (TOGAF-ADM) pada Puskesmas Sukatani. *Jurnal Sistem Informasi Galuh*, 32-37.