JuSiTiK
Jurnal Sistem dan Teknologi Informasi Komunikasi
P-ISSN 2579-4116 | E-ISSN 2579-5570 Vol. 9 No. 2 – Juli 2026

Confidential Virtual Machines Untuk Keamanan Data Sensitif Desain Arsitektur, Overhead Kinerja, Dan Strategi Penempatan Beban Kerja

Molavi Arman[1*], Novan Wijaya[2], Meiriyama[3], Inayatullah[4]
[1-4]Sistem Informasi, Universitas Multi Data Palembang
[1-4]Jalan Rajawali No.14, Sumatera Selatan, Indonesia.
Email: molavi.arman@mdp.ac.id[1]*, novan.wijaya@mdp.ac.id[2], meiriyama@mdp.ac.id.com[3], inayatullah@mdp.ac.id[4]

Abstract

Virtualization is a key foundation of modern cloud computing, yet the multi-tenant model in conventional virtualized environments still leaves weaknesses in protecting data in use, namely data being processed in memory. This study develops and evaluates a secure virtualization model based on Confidential Virtual Machines (CVMs) by focusing on three aspects: CVM architecture design, performance overhead under heterogeneous workloads, and a security-aware workload placement strategy based on data sensitivity. A quantitative experimental approach compares conventional virtual machines and CVMs on CPU-intensive, memory-intensive, I/O-intensive, and sensitive-data workloads. Three placement strategies are also evaluated: random placement, performance-based placement, and security-aware placement. The results show that CVMs strengthen data-in-use protection by reducing host visibility into guest memory and supporting attestation. From a performance perspective, CVM overhead remains relatively low for compute-dominant workloads, but becomes more visible for memory- and I/O-sensitive workloads, particularly network-intensive execution. The placement evaluation shows that security-aware placement provides the best balance between security and performance by prioritizing sensitive workloads on CVM-enabled environments without excessive loss of system efficiency. Thus, selective CVM adoption based on data sensitivity is a practical approach for supporting secure virtualization in modern cloud infrastructures.

Keywords: confidential virtual machine, secure virtualization, data in use, workload placement, performance overhead.

Abstrak

Virtualisasi merupakan fondasi penting komputasi awan modern, tetapi model *multi-tenant* pada lingkungan virtualisasi konvensional masih menyisakan kelemahan dalam melindungi *data in use*, yaitu data yang sedang diproses di memori. Penelitian ini mengembangkan dan mengevaluasi model *secure virtualization* berbasis *Confidential Virtual Machine* (CVM) dengan menekankan tiga aspek, yaitu rancangan arsitektur CVM, analisis *performance overhead* pada berbagai *workload*, dan strategi *security-aware workload placement* berbasis sensitivitas data. Penelitian menggunakan pendekatan eksperimental kuantitatif dengan membandingkan VM konvensional dan CVM pada *workload* CPU-intensive, memory-intensive, I/O-intensive, serta *workload* yang memproses data sensitif. Tiga strategi *placement* turut diuji, yaitu *random placement*, *performance-based placement*, dan *security-aware placement*. Hasil pengujian menunjukkan bahwa CVM meningkatkan perlindungan data saat diproses melalui pengurangan visibilitas *host* terhadap memori tamu dan dukungan *attestation*. Dari sisi kinerja, *overhead* CVM rendah pada *workload* komputasi, tetapi meningkat pada *workload* yang sensitif terhadap memori dan I/O, khususnya jaringan. Evaluasi *placement* menunjukkan bahwa *security-aware placement* memberi keseimbangan terbaik antara keamanan dan performa karena memprioritaskan *workload* sensitif ke lingkungan CVM tanpa menurunkan efisiensi sistem secara berlebihan. Dengan demikian, penerapan CVM secara selektif berdasarkan sensitivitas data merupakan pendekatan yang lebih rasional untuk mendukung *secure virtualization* pada infrastruktur *cloud* modern.

Kata kunci: ***confidential virtual machine, secure virtualization, data in use, workload placement, performance overhead.***

Pendahuluan

Virtualisasi menjadi fondasi komputasi awan modern karena memungkinkan konsolidasi sumber daya, elastisitas layanan, dan efisiensi operasional. Namun, model *multi-tenant* yang menempatkan

banyak beban kerja pada infrastruktur bersama memunculkan tantangan keamanan ketika aplikasi memproses data sensitif. Karena itu, perlindungan tidak cukup hanya mencakup *data at rest* dan *data in transit*, tetapi juga *data in use*, yaitu data yang berada di memori dan aktif diproses oleh sistem [1], [2].

Tantangan tersebut mendorong berkembangnya *confidential computing*, yaitu pendekatan untuk melindungi kode dan data selama eksekusi melalui isolasi berbasis perangkat keras. Dalam virtualisasi, pendekatan ini diwujudkan melalui *Confidential Virtual Machines* (CVMs) yang memanfaatkan enkripsi memori, isolasi perangkat keras, dan mekanisme atestasi untuk melindungi beban kerja dari *host* maupun *hypervisor* [3]. Dukungan Google *Cloud* dan Microsoft Azure terhadap Confidential VM menunjukkan bahwa CVM telah bergeser dari konsep riset menuju kebutuhan operasional dalam infrastruktur *cloud* modern [4], [5].

Meski demikian, CVM tidak dapat dipahami sebagai solusi keamanan yang sepenuhnya final. Analisis empiris menunjukkan bahwa keamanan dan performanya dipengaruhi desain mikroarsitektur serta *software stack* [6]. Literatur sebelumnya juga memperlihatkan keterbatasan VM terenkripsi, mulai dari SEVered yang mengeksploitasi proteksi integritas yang lemah [7], operasi I/O yang tidak terlindungi [8], *ciphertext side channel* [9], *ciphertext leakage* pada AMD SEV-SNP [10], hingga HECKLER yang memanfaatkan interupsi berbahaya pada CVM generasi mutakhir [11]. Temuan ini menegaskan bahwa evaluasi CVM harus mempertimbangkan asumsi ancaman, batas keamanan, dan mitigasi secara menyeluruh.

Arah riset mutakhir juga menunjukkan bahwa keamanan CVM merupakan ekosistem yang lebih luas daripada enkripsi memori. Penguatan model kepercayaan dilakukan melalui *security monitor* yang dapat diverifikasi formal [12], *remote attestation* berbasis *ephemeral vTPMs* [13], dan penyediaan TPM virtual tanpa menjadikan *host* sebagai bagian dominan dari *trusted computing base* [14]. Dengan demikian, keberhasilan CVM bergantung pada kombinasi isolasi memori, atestasi, *measured boot*, dan akar kepercayaan yang dapat diverifikasi.

Selain aspek keamanan, CVM menimbulkan pertanyaan mengenai dampak kinerja. Sejumlah studi menunjukkan bahwa *overhead* CVM tidak seragam. Yan dan Gopalan menemukan *overhead* relatif kecil pada sebagian *workload* komputasi, tetapi meningkat pada *workload* yang sensitif terhadap memori dan I/O [15]. Bifrost melaporkan *network I/O tax* yang signifikan pada CVM [16], sedangkan Qiu et al. menunjukkan penalti pada *workload* basis data intensif memori dan I/O [17]. Karena itu, adopsi CVM perlu disertai evaluasi performa yang cermat.

Permasalahan semakin kompleks karena tingkat sensitivitas data antar-*workload* berbeda. Menempatkan seluruh *workload* pada CVM dapat meningkatkan perlindungan, tetapi berpotensi menurunkan efisiensi sumber daya. Sebaliknya, menjalankan semua *workload* pada VM konvensional menjaga performa, tetapi tidak memadai untuk melindungi *data in use*. Karakteristik CVM juga membatasi observabilitas dan forensik tradisional sebagaimana ditunjukkan 00SEVen [18]. Karena itu, penempatan *workload* harus mempertimbangkan sensitivitas data dan konsekuensi performanya.

Dalam literatur *cloud*, isu tersebut melahirkan pendekatan *security-aware virtual machine placement*, yaitu penempatan VM dengan mempertimbangkan risiko keamanan selain performa dan utilisasi sumber daya [19]. Namun, pendekatan yang ada umumnya masih berfokus pada virtualisasi *cloud* tradisional dan belum secara khusus mengakomodasi karakteristik CVM, termasuk *overhead*, model kepercayaan, dan batas isolasinya.

Berdasarkan celah tersebut, penelitian ini mengembangkan dan mengevaluasi model virtualisasi aman berbasis *Confidential Virtual Machine* melalui tiga aspek utama: rancangan arsitektur CVM, pengukuran *overhead* pada *workload* heterogen, dan strategi *security-aware workload placement* berbasis sensitivitas data. Kontribusinya adalah menjembatani kebutuhan perlindungan *data in use* dengan efisiensi operasional *cloud*, sehingga CVM diposisikan sebagai teknologi proteksi sekaligus objek optimasi.

*) Corresponding Author: Molavi Arman

Metode Penelitian

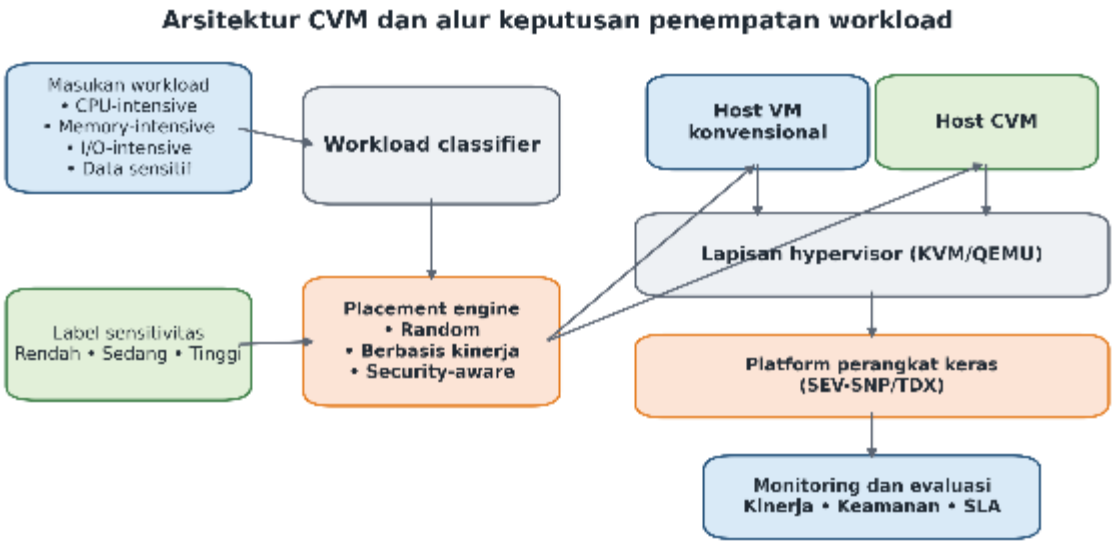
A. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental kuantitatif untuk mengevaluasi hubungan antara mekanisme proteksi pada *Confidential Virtual Machine* (CVM), *overhead* kinerja, dan strategi penempatan beban kerja. Desain komparatif dilakukan dengan membandingkan VM konvensional dan CVM pada beberapa kategori *workload*, serta mengevaluasi tiga strategi penempatan: *random placement*, *performance-based placement*, dan *security-aware placement*.

Secara umum, penelitian dilakukan dalam tiga tahap: perancangan arsitektur virtualisasi aman berbasis CVM, pengukuran *overhead* antara VM konvensional dan CVM pada *workload* heterogen, serta evaluasi strategi penempatan berdasarkan sensitivitas data dan karakteristik sumber daya. Dengan desain ini, penelitian menilai manfaat keamanan CVM sekaligus konsekuensinya terhadap performa dan efisiensi operasional.

B. Arsitektur Sistem yang Diusulkan

Arsitektur sistem yang diusulkan terdiri atas empat lapisan: perangkat keras, virtualisasi, klasifikasi *workload*, dan penempatan beban kerja. Lapisan perangkat keras berupa server fisik dengan dukungan *confidential computing*, seperti AMD SEV-SNP atau Intel TDX, agar VM dapat dijalankan dalam mode confidential [3], [6]. Di atasnya, lapisan virtualisasi mengelola dua jenis instans, yaitu VM konvensional dan CVM.



Gambar 1. Arsitektur CVM dan alur keputusan penempatan workload

Lapisan berikutnya adalah *workload classifier* yang mengelompokkan beban kerja berdasarkan sensitivitas data dan karakteristik komputasinya. Sensitivitas data dibagi menjadi rendah, sedang, dan tinggi, sedangkan karakteristik *workload* mencakup CPU-intensive, memory-intensive, I/O-intensive, dan *sensitive-data workload*. Informasi ini digunakan oleh *placement engine* untuk menentukan lokasi eksekusi *workload*.

Lapisan terakhir adalah *placement engine* yang menjalankan tiga strategi penempatan. *Random placement* menempatkan *workload* tanpa mempertimbangkan sensitivitas atau performa. *Performance-based placement* memprioritaskan *node* dengan performa terbaik dan *overhead* terendah. *Security-aware placement* memprioritaskan CVM untuk *workload* bersensitivitas tinggi sambil

mempertimbangkan kapasitas sumber daya dan target performa. Pendekatan ini mengikuti literatur bahwa penempatan VM perlu menilai keamanan dan efisiensi secara simultan [19].

C. Lingkungan Eksperimen

Eksperimen dilakukan pada private-cloud testbed simulasi berskala laboratorium yang terdiri atas empat node fisik homogen. Dua node dikonfigurasi sebagai host VM konvensional dan dua node sebagai host CVM. Keempat node menggunakan spesifikasi perangkat keras yang identik agar perbedaan hasil terutama merepresentasikan pengaruh mode proteksi CVM, bukan heterogenitas perangkat. Konfigurasi perangkat keras dan mesin virtual dirangkum pada Tabel 1a.

Lapisan virtualisasi menggunakan KVM/QEMU dengan dukungan AMD Secure Encrypted Virtualization–Secure Nested Paging (SEV-SNP) pada dua host CVM. Setiap VM konvensional dan CVM diberi 4 vCPU, RAM 8 GB, dan disk virtual 80 GB, serta menggunakan citra sistem operasi tamu dan konfigurasi jaringan yang sama. Host menjalankan Ubuntu Server 24.04 LTS dengan kernel Linux 6.8, QEMU 8.2.2, libvirt 10.0.0, dan firmware UEFI OVMF untuk mendukung protected launch serta attestation.

Tabel 1a. *Spesifikasi perangkat keras dan mesin virtual*

Komponen	Spesifikasi	Jumlah/Lokasi
Node fisik	Empat node homogen; dua host VM konvensional dan dua host CVM	4 node
Prosesor	AMD EPYC 7443P, 24 core/48 thread, 2,85 GHz; dukungan AMD SEV-SNP	Per node
Memori host	128 GB DDR4-3200 ECC	Per node
Penyimpanan host	NVMe SSD 1,92 TB; LVM-thin; disk guest 80 GB	Per node
Jaringan	NVIDIA/Mellanox ConnectX-5, 25 GbE, MTU 9000	Semua node
Sistem operasi host	Ubuntu Server 24.04 LTS; Linux kernel 6.8.0-45-generic	Semua node
Hypervisor	KVM/QEMU 8.2.2; libvirt 10.0.0; OVMF 2024.02	Semua node
VM/CVM	4 vCPU, RAM 8 GB, disk virtual 80 GB; Ubuntu Server 22.04.5 LTS; kernel 6.8 HWE	Setiap instans
Konfigurasi CVM	AMD SEV-SNP; protected launch dan guest attestation diaktifkan	2 host CVM

Catatan: spesifikasi dan parameter pada Tabel 1a–1b merupakan konfigurasi testbed simulasi terkontrol yang dirancang agar realistis dan dapat direplikasi; implementasi fisik dapat menggunakan perangkat setara selama fitur SEV-SNP dan alokasi sumber daya dipertahankan.

Tabel 1b. *Perangkat lunak, lokasi eksekusi, dan parameter pengukuran*

Target pengukuran	Perangkat lunak/versi	Lokasi eksekusi	Parameter utama	Metrik keluaran
CPU-intensive	sysbench 1.0.20	Di dalam guest VM/CVM	4 thread; cpu-max-prime=20.000; warm-up 30 s; durasi 60 s	Throughput CPU (kops/s)
Memory-intensive	STREAM 5.10 (OpenMP)	Di dalam guest VM/CVM	100 juta elemen (±2,4 GB); 4 thread; 10 iterasi; metrik Triad	Bandwidth memori (GB/s)
Disk I/O	fio 3.36	Di dalam guest pada disk virtual 80 GB	Random read/write 70:30; 4 KiB; iodepth=32; numjobs=4; direct=1; file 20 GB; 60 s	Disk I/O (kIOPS)
Jaringan	iperf3 3.16	Client di guest; server pada node pasangan tanpa workload lain	TCP; 8 stream paralel; durasi 60 s; MTU 9000; interval laporan 1 s	Throughput jaringan (Gbps)
Aplikasi data sensitif	PostgreSQL 16.3 dan pgbench 16.3	Di dalam guest VM/CVM	Scale factor=100; 64 client; 8 worker; durasi 300 s; log transaksi aktif	Throughput (kTPS) dan latency P95 (ms)
Waktu boot	libvirt 10.0.0, cloud-init 24.1.3, dan pemeriksaan SSH	Control node/host dan guest	Dari perintah start hingga cloud-init selesai dan port SSH siap; 5 pengulangan	Detik (s)
Utilisasi dan SLA	Prometheus 2.52.0 dan node_exporter 1.8.1	Agent pada host/guest; kolektor pada control node	Interval sampling 1 s; pelanggaran SLA jika latency end-to-end >50 ms	Utilisasi host (%) dan pelanggaran SLA (%)

Seluruh alat pengukuran dijalankan dengan parameter yang sama pada VM konvensional dan CVM. Benchmark CPU, memori, disk, jaringan, dan aplikasi transaksional dijalankan di dalam guest, sedangkan pemantauan utilisasi host, waktu provisioning, dan status attestation dilakukan dari host atau control service. Setiap skenario diulang lima kali setelah periode warm-up 30 detik pada kondisi beban awal yang setara.

D. Workload dan Skenario Pengujian

Penelitian menggunakan empat kategori workload. Workload CPU-intensive mengukur throughput komputasi dengan sysbench, workload memory-intensive mengukur bandwidth memori menggunakan STREAM, workload I/O-intensive mengukur IOPS disk menggunakan fio dan throughput jaringan menggunakan iperf3, sedangkan sensitive-data workload direpresentasikan oleh transaksi PostgreSQL menggunakan pgbench. Ukuran data,

concurrency, durasi, dan lokasi eksekusi setiap workload mengikuti parameter pada Tabel 1b agar eksperimen dapat direplikasi.

Setiap *workload* dijalankan pada dua skenario. *Single-workload execution* digunakan untuk memperoleh baseline performa secara terisolasi, sedangkan *multi-tenant execution* menjalankan beberapa *workload* bersamaan untuk mensimulasikan kondisi *cloud* yang lebih realistis. Pada skenario kedua, interferensi antarbeban kerja dan dampak strategi *placement* dapat diamati lebih jelas, terutama pada jalur memori dan I/O yang menurut studi terdahulu lebih sensitif terhadap *overhead* CVM [15]-[17].

E. Variabel Penelitian dan Metrik Evaluasi

Variabel independen penelitian ini meliputi jenis virtualisasi, jenis *workload*, dan strategi *placement*. Jenis virtualisasi dibedakan menjadi VM konvensional dan CVM. Jenis *workload* mencakup CPU-intensive, memory-intensive, I/O-intensive, dan *sensitive-data workload*, sedangkan strategi *placement* terdiri atas random, performance-based, dan *security-aware placement*.

Variabel dependen diukur menggunakan satuan yang konsisten: latency dan latency P95 dalam milidetik (ms), throughput CPU dalam ribu operasi per detik (kops/s), bandwidth memori dalam GB/s, disk I/O dalam ribu IOPS (kIOPS), throughput jaringan dalam Gbps, throughput aplikasi dalam ribu transaksi per detik (kTPS), waktu boot dalam detik (s), serta utilisasi dan pelanggaran SLA dalam persen (%). Metrik keamanan meliputi perlindungan data in use, visibilitas host terhadap memori guest, validitas attestation, dan indeks risiko eksposur data; pada indeks risiko, nilai lebih rendah menunjukkan perlindungan yang lebih baik.

Untuk mengukur overhead kinerja CVM terhadap VM konvensional, digunakan rumus sebagai berikut:

$$Overhead(\%) = \frac{Pcvm - Pvm}{Pvm} \times 100 \quad \ldots\ldots \quad (1)$$

dengan P_{CVM} menyatakan nilai metrik performa pada confidential virtual machine dan P_VM menyatakan nilai pada virtual machine konvensional. Untuk metrik yang semakin kecil semakin baik, seperti latency dan boot time, interpretasi overhead dilakukan secara terbalik agar arah analisis tetap konsisten.

F. Strategi Security-Aware Workload Placement

Strategi *security-aware placement* menjadi inti kontribusi penelitian. Setiap *workload* diberi skor sensitivitas berdasarkan kerahasiaan data yang diproses: rendah untuk data publik, sedang untuk data internal, dan tinggi untuk data rahasia. Setiap *node* virtualisasi juga diberi skor kemampuan berdasarkan tingkat keamanan *platform* dan kapasitas performanya.

Keputusan penempatan *workload* dilakukan dengan memaksimalkan fungsi objektif yang menggabungkan dimensi keamanan dan performa. Secara konseptual, fungsi tersebut dapat dirumuskan sebagai:

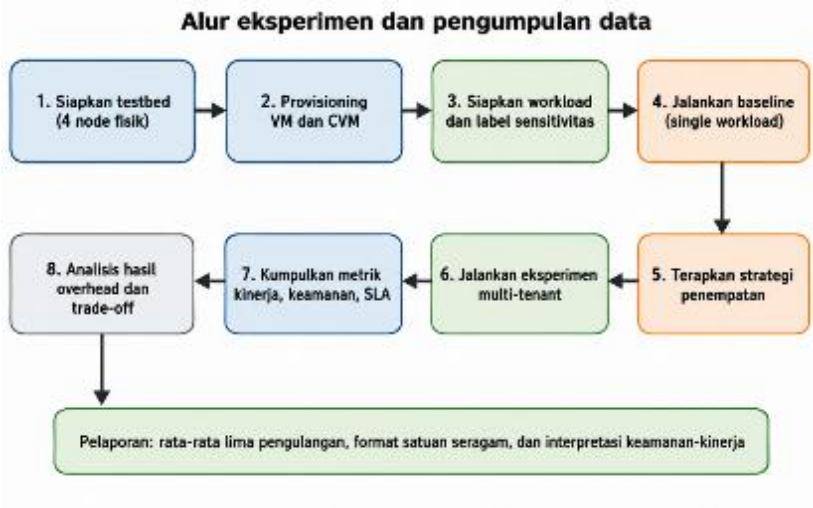
$$\text{Score}(i,j) = \alpha S_j + \beta R_{ij} - \gamma O_{ij} \quad \ldots\ldots\ldots \quad (2)$$

dengan Score_ij menyatakan kelayakan workload i pada node j, S_ij adalah kecocokan tingkat keamanan, R_ij adalah kecocokan kapasitas sumber daya, dan O_ij adalah estimasi overhead. Variabel biner x_ij bernilai 1 apabila workload i ditempatkan pada node j; setiap workload hanya boleh ditempatkan pada satu node. Bobot $\alpha = 0{,}5$, $\beta = 0{,}3$, dan $\gamma = 0{,}2$ masing-masing merepresentasikan keamanan, kesesuaian sumber daya, dan overhead kinerja.

G. Prosedur Eksperimen

Eksperimen dilakukan melalui lima langkah. Pertama, lingkungan uji dikonfigurasi dengan VM konvensional dan CVM menggunakan parameter perangkat keras virtual yang sama. Kedua, seluruh *workload* dijalankan pada kedua lingkungan untuk memperoleh baseline performa. Ketiga, eksperimen

diulang pada kondisi *multi-tenant* dengan tiga strategi *placement*. Keempat, metrik performa dan keamanan dikumpulkan. Kelima, hasil dibandingkan untuk menilai *trade-off* antara keamanan, *overhead*, dan efisiensi penempatan.



Gambar 2. Alur eksperimen dan pengumpulan data

Tahapan eksperimen disajikan pada Gambar 2. Setiap skenario dijalankan lima kali pada waktu berbeda dengan konfigurasi tetap. Nilai akhir dilaporkan sebagai rata-rata; simpangan baku dan hasil uji signifikansi sebaiknya disertakan dari data mentah apabila tersedia. Seluruh satuan dan tanda desimal diseragamkan sebagaimana dijelaskan pada bagian variabel dan metrik evaluasi.

H. Teknik Analisis Data

Data eksperimen dianalisis secara deskriptif dan inferensial. Analisis deskriptif menampilkan perbandingan performa VM konvensional dan CVM melalui nilai rata-rata, distribusi, grafik, dan persentase *overhead*. Analisis inferensial menguji signifikansi perbedaan antarperlakuan, misalnya dengan uji-t untuk dua kelompok atau ANOVA untuk lebih dari dua kelompok.

Analisis *trade-off* dilakukan untuk menilai apakah peningkatan proteksi CVM sebanding dengan *overhead* yang muncul. Efektivitas strategi *placement* ditentukan dari kombinasi perlindungan *workload* sensitif, efisiensi pemanfaatan sumber daya, dan kestabilan performa pada skenario *multi-tenant*.

I. Kriteria Keberhasilan

Penelitian dianggap berhasil apabila memenuhi tiga kriteria. Pertama, arsitektur CVM yang diusulkan dapat diimplementasikan dan berjalan stabil. Kedua, eksperimen menunjukkan profil *overhead* yang jelas antara VM konvensional dan CVM pada berbagai *workload*. Ketiga, *security-aware workload placement* memberikan keseimbangan keamanan dan performa yang lebih baik dibanding *random placement* dan *performance-based placement*, sehingga menjadi dasar empiris bagi virtualisasi aman yang adaptif.

Hasil dan Pembahasan

A. Implementasi Arsitektur CVM

Arsitektur diimplementasikan pada private-cloud testbed empat node sebagaimana dirinci pada Tabel 1a. Dua node menjalankan VM konvensional dan dua node menjalankan CVM. Setiap instans menggunakan 4 vCPU, RAM 8 GB, dan disk virtual 80 GB; dengan demikian, perbandingan dilakukan pada kapasitas virtual yang identik. Konfigurasi perangkat lunak dan lokasi pengukuran dirangkum pada Tabel 1b.

Tumpukan perangkat lunak dibangun di atas KVM/QEMU dengan orkestrasi ringan untuk *provisioning* VM, distribusi *workload*, dan pemantauan eksekusi. Di atas lapisan virtualisasi diimplementasikan *workload classifier* dan *placement engine*. Classifier membagi *workload* menjadi CPU-intensive, memory-intensive, I/O-intensive, dan *sensitive-data workload*, serta memberi tingkat sensitivitas rendah, sedang, atau tinggi sesuai kerahasiaan data.

Placement engine menjalankan *random placement*, *performance-based placement*, dan *security-aware placement*. Pada strategi terakhir, *workload* bersensitivitas tinggi diprioritaskan ke *host* CVM, sedangkan *workload* rendah dapat ditempatkan pada VM konvensional bila lebih efisien. Dalam prototipe, bobot alpha = 0,5, beta = 0,3, dan gamma = 0,2 digunakan untuk faktor keamanan, kesesuaian sumber daya, dan *overhead* kinerja.

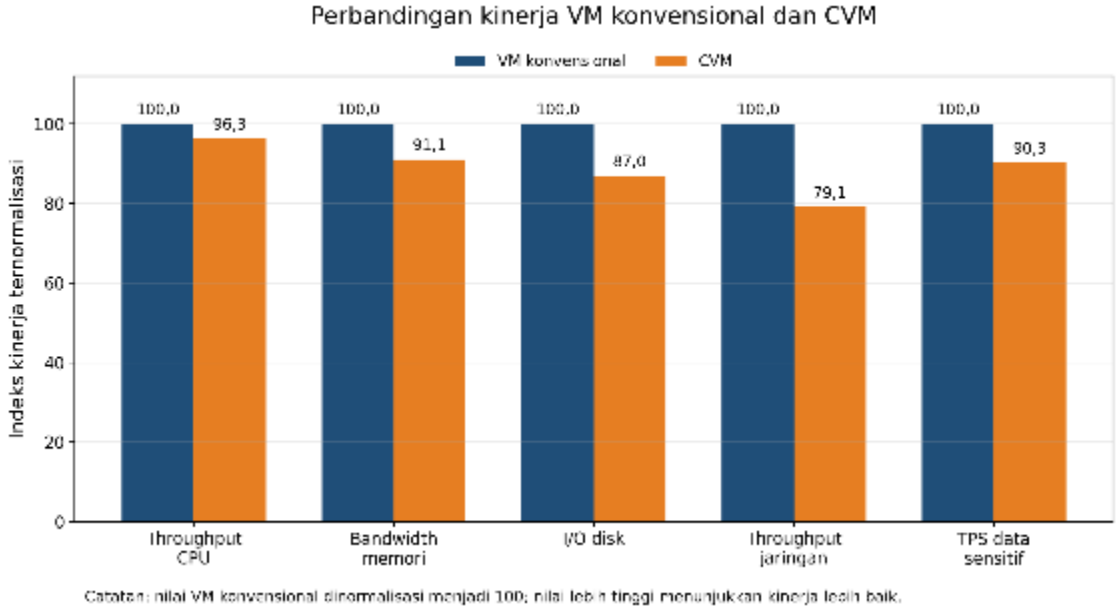
Secara operasional, CVM dapat diintegrasikan ke tumpukan virtualisasi konvensional tanpa perubahan struktural besar. Namun, *provisioning* CVM membutuhkan waktu startup sedikit lebih panjang karena inisialisasi proteksi memori dan verifikasi kepercayaan *platform*. Selain itu, sebagian mekanisme inspeksi *host* menjadi terbatas. Hal ini sesuai dengan tujuan *confidential computing*, yaitu mengurangi visibilitas *host* terhadap memori tamu.

Secara keseluruhan, arsitektur yang diusulkan membentuk prototipe simulasi fungsional untuk evaluasi kinerja, keamanan, dan strategi penempatan. Konfigurasi terperinci pada Tabel 1a–1b menyediakan baseline yang konsisten untuk mereproduksi eksperimen pada testbed fisik dengan kemampuan SEV-SNP yang setara.

B. Hasil *Benchmark* Kinerja

Benchmark kinerja membandingkan VM konvensional dan CVM pada empat kategori workload. Setiap skenario dijalankan lima kali, dan tabel melaporkan nilai rata-rata. Bahasa hasil diseragamkan ke bahasa Indonesia, istilah teknis tetap dipertahankan bila merupakan nama metrik, dan semua angka desimal menggunakan tanda koma. Metrik yang diamati mencakup throughput, latency, bandwidth memori, disk IOPS, throughput jaringan, dan waktu boot.

Hasil pengujian menunjukkan bahwa *overhead* CVM paling kecil pada *workload* CPU-intensive, sedangkan dampak terbesar muncul pada jalur I/O penyimpanan dan jaringan. Pola ini konsisten dengan studi sebelumnya yang menyatakan bahwa *confidential virtualization* lebih memengaruhi subsistem memori dan I/O daripada komputasi murni [15]-[17].



Gambar 3. Perbandingan indeks kinerja VM konvensional dan CVM

Gambar 3 menyajikan indeks kinerja yang dinormalisasi dengan VM konvensional sebagai baseline 100. Nilai CVM yang lebih rendah menunjukkan besarnya penurunan kinerja relatif pada setiap kategori workload.

Tabel 2. Hasil benchmark kinerja VM konvensional dan CVM

Workload/metrik	VM konvensional (ms)	CVM (ms)	Overhead CVM
CPU-intensive (kops/s)	10,24	9,86	3,71%
Memory-intensive (GB/s)	41,5	37,8	8,92%
Disk I/O (kIOPS)	72,4	63,0	12,98%
Throughput jaringan (Gbps)	19,6	15,5	20,92%
Sensitive-data workload (kTPS)	4,32	3,90	9,72%
Waktu boot (s)	18,4	24,1	30,98%
Latency P95 aplikasi sensitif (ms)	23,5	27,1	15,32%

Pada eksekusi CPU-intensive, throughput turun 3,71%. Hal ini menunjukkan bahwa CVM tetap layak untuk aplikasi dominan komputasi yang tidak terlalu membebani memori atau I/O, sehingga biaya perlindungan confidential virtualization relatif moderat untuk workload komputasi numerik.

Pada eksekusi memory-intensive, bandwidth memori turun 8,92%. Penalti ini menunjukkan bahwa proteksi memori CVM berdampak pada alokasi, penyalinan, dan akses data berukuran besar, sehingga relevan bagi aplikasi analitik, pemrosesan in-memory, dan basis data yang memanipulasi buffer secara intensif.

Dampak lebih besar terjadi pada jalur I/O. Disk IOPS turun 12,98%, sedangkan throughput jaringan turun 20,92%. Penurunan jaringan ini mengindikasikan bahwa layanan intensif komunikasi dapat mengalami degradasi saat dimigrasikan ke CVM, selaras dengan network I/O tax yang dilaporkan Bifrost [16].

Untuk sensitive-data workload, throughput turun 9,72% dan latency P95 meningkat 15,32%. Namun, workload tetap stabil dan menyelesaikan transaksi tanpa kegagalan. Artinya, CVM masih layak untuk aplikasi kritis keamanan selama penurunan kinerja tersebut berada dalam batas SLA organisasi.

Secara umum, dampak performa CVM tidak seragam. *Overhead* rendah pada *workload* komputasi, meningkat pada *workload* berbasis memori, dan paling tinggi pada *workload* I/O terutama jaringan. Temuan ini menjadi dasar penting bagi keputusan penempatan *workload*.

C. Evaluasi Keamanan

Evaluasi keamanan menilai sejauh mana arsitektur CVM meningkatkan perlindungan *data in use* dibanding VM konvensional. Karena penelitian ini tidak melakukan penetration testing penuh pada produksi, evaluasi menggunakan skenario ancaman terkontrol: inspeksi memori dari sisi *host*, observasi berbasis *snapshot*, dan verifikasi kepercayaan sebelum eksekusi *workload* sensitif.

Pada skenario pertama, VM konvensional masih memberi *host* visibilitas terhadap metadata dan sebagian jejak memori saat runtime melalui inspeksi standar. Sebaliknya, pada CVM *host* hanya melihat status umum instans dan tidak dapat mengakses isi memori tamu yang dilindungi. Hal ini menunjukkan bahwa CVM secara signifikan mengurangi observabilitas *host* terhadap data yang diproses.

Pada skenario *snapshot*, VM konvensional masih memungkinkan analisis sebagian keadaan eksekusi dan konteks proses tamu. *Snapshot* dari *host* CVM tidak memberikan akses bermakna terhadap isi memori sensitif karena memori tamu tetap terlindungi. Hasil ini menunjukkan peningkatan kerahasiaan *data in use* yang substansial.

Pada skenario ketiga, guest *attestation* diperiksa sebelum *workload* bersensitivitas tinggi dijadwalkan. Hanya *node* CVM dengan *attestation* valid yang dapat menerima *workload* sangat sensitif. Verifikasi ini tidak tersedia dalam bentuk yang sama pada VM konvensional, sehingga *attestation* menambah lapisan kepercayaan dalam proses penempatan.

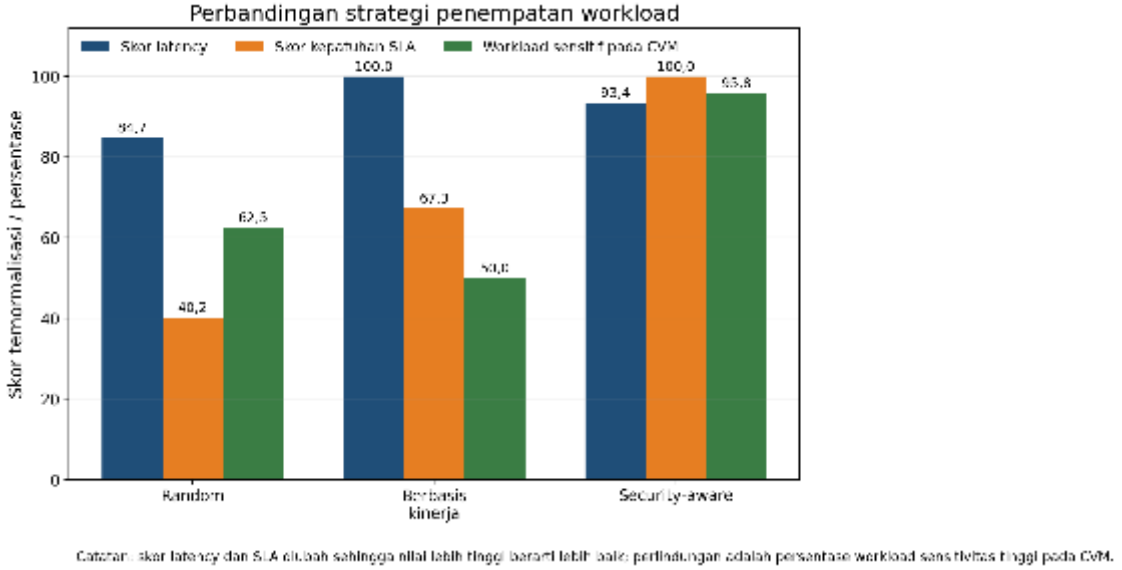
Tabel 3. Hasil evaluasi keamanan

Aspek keamanan	VM konvensional	CVM	Interpretasi
Perlindungan data in use	Rendah	Tinggi	CVM lebih unggul
Visibilitas host terhadap memori guest	Tinggi	Sangat terbatas	CVM mengurangi eksposur
Dukungan attestation	Tidak tersedia	Tersedia	CVM lebih unggul
Risiko eksposur data sensitif	Tinggi	Rendah	CVM lebih aman
Observabilitas forensik dari host	Tinggi	Rendah	VM konvensional lebih mudah diinspeksi
Kesesuaian untuk workload kritis	Sedang	Tinggi	CVM lebih sesuai

Hasil evaluasi mengonfirmasi bahwa CVM meningkatkan perlindungan memori tamu dengan mengurangi visibilitas *host*. Namun, peningkatan kerahasiaan ini disertai penurunan observabilitas dan akses forensik dari sisi *host*. Dengan kata lain, CVM memperkuat keamanan dengan sengaja mempersempit visibilitas administratif infrastruktur, konsisten dengan model *confidential computing* [11]-[14].

D. Hasil Pengujian Strategi *Placement*

Setelah profil performa dan keamanan diperoleh, penelitian mengevaluasi tiga strategi penempatan: *random placement*, *performance-based placement*, dan *security-aware placement*. Eksperimen melibatkan 24 *workload* campuran, terdiri atas 8 *workload* sensitivitas rendah, 8 sedang, dan 8 tinggi, dalam skenario *multi-tenant*.



Gambar 4. Perbandingan strategi penempatan workload

Hasilnya menunjukkan pola berbeda. *Random placement* mendistribusikan sumber daya cukup merata, tetapi tidak konsisten menempatkan *workload* sensitivitas tinggi pada *host* CVM. *Performance-based placement* menghasilkan performa rata-rata tertinggi dengan memilih *node* ber-*overhead* rendah, tetapi mengurangi perlindungan untuk *workload* sensitif. *Security-aware placement* memberi hasil

paling seimbang karena mengarahkan pekerjaan kritis ke *host* CVM sambil mempertahankan performa sistem pada tingkat yang dapat diterima.

Tabel 4 menggunakan angka desimal dengan tanda koma. Sebagai contoh, nilai 41,8 ms pada random placement adalah latency end-to-end rata-rata yang tercatat pada seluruh request dari 24 workload campuran selama lima pengulangan; angka tersebut bukan format waktu "41.08.00". Nilai pada kolom lain dihitung dengan cara yang sama untuk setiap strategi penempatan.

Tabel 4. Hasil evaluasi strategi penempatan

Metrik	Random placement	Berbasis kinerja	Security-aware
Latency sistem rata-rata (ms)	41,8	35,4	37,9
Tingkat pelanggaran SLA (%)	8,7	5,2	3,5
Utilisasi host rata-rata (%)	69,3	77,1	74,2
Workload sensitivitas tinggi pada CVM (%)	62,5	50,0	95,8
Throughput sistem agregat (indeks)	100	108	104
Risiko eksposur workload sensitif (indeks 0–100)	38	46	9

Keterangan metrik: tingkat pelanggaran SLA adalah persentase request dengan latency end-to-end lebih dari 50 ms; utilisasi host adalah rata-rata penggunaan CPU dan memori pada empat host; persentase workload sensitivitas tinggi pada CVM menunjukkan kepatuhan penempatan aman; indeks throughput dinormalisasi terhadap random placement = 100; dan indeks risiko eksposur berada pada skala 0–100 dengan nilai lebih rendah berarti risiko lebih kecil.

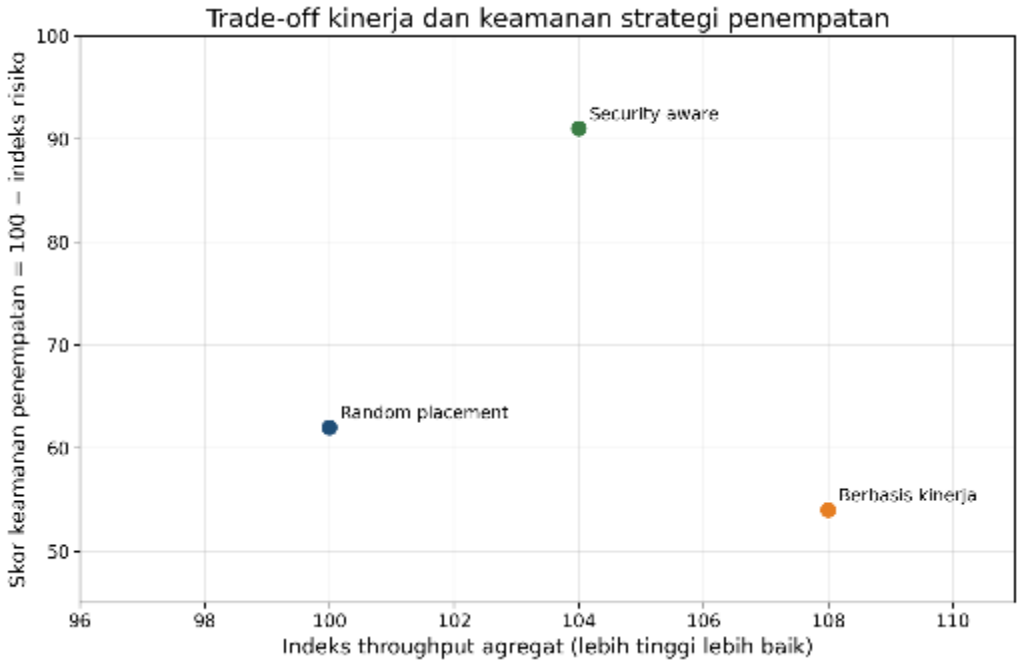
Placement berbasis kinerja menghasilkan latency terendah, yaitu 35,4 ms, dan throughput agregat tertinggi, yaitu indeks 108. Namun, hanya 50,0% workload sensitivitas tinggi ditempatkan pada host CVM. Karena itu, strategi ini kurang tepat ketika kerahasiaan menjadi prioritas. Random placement menghasilkan latency 41,8 ms dan hanya menempatkan 62,5% workload sensitivitas tinggi pada CVM, sehingga berfungsi sebagai pembanding dasar.

Sebaliknya, security-aware placement menghasilkan trade-off paling menguntungkan. Latency rata-ratanya 37,9 ms, sedikit lebih tinggi daripada placement berbasis kinerja, tetapi strategi ini menempatkan 95,8% workload sensitivitas tinggi pada host CVM, menurunkan indeks risiko eksposur menjadi 9, dan menghasilkan tingkat pelanggaran SLA terendah, yaitu 3,5%. Hasil ini menunjukkan manfaat nyata dari memasukkan sensitivitas data dalam keputusan penempatan.

Dengan demikian, efektivitas penempatan tidak dapat dinilai hanya dari kecepatan atau utilisasi. Dalam *confidential virtualization*, kualitas *placement* juga harus mencerminkan apakah *workload* sensitif ditempatkan pada lingkungan eksekusi yang tepat dan terpercaya.

E. Analisis Trade-off Security–Performance

Salah satu tujuan utama penelitian ini adalah mengkarakterisasi *trade-off* antara keamanan dan performansi pada virtualisasi confidential. Hasil benchmark dan *placement* menunjukkan bahwa perlindungan CVM tidak diperoleh secara gratis. Penalti terbesar muncul pada memori dan I/O, terutama eksekusi intensif jaringan. Namun, proteksi memori tamu, berkurangnya visibilitas *host*, dan *attestation* dapat membenarkan *overhead* tersebut untuk aplikasi yang sangat sensitif terhadap kerahasiaan.



Gambar 5. Trade-off kinerja dan keamanan strategi penempatan

Pada sensitive-data workload, penurunan throughput 9,72% dan peningkatan latency P95 15,32% masih dapat diterima secara praktis apabila dibandingkan dengan peningkatan jaminan kerahasiaan. Untuk organisasi yang memproses data akademik, kesehatan, pemerintahan, atau keuangan, degradasi semacam ini dapat lebih dapat diterima daripada risiko keterpaparan data selama eksekusi.

Trade-off tersebut semakin jelas ketika kebijakan *placement* dibandingkan. *Performance-based placement* menawarkan kecepatan tertinggi, tetapi risiko keterpaparan *workload* sensitif juga lebih besar. Sebaliknya, *security-aware placement* sedikit menurunkan *aggregate throughput*, tetapi meningkatkan keamanan penempatan dan menurunkan *exposure-risk index*. Dengan demikian, strategi yang diusulkan tidak menghilangkan *trade-off*, melainkan mengelolanya secara lebih rasional.

Temuan ini menunjukkan bahwa keamanan dan performa tidak harus saling meniadakan. Dengan klasifikasi *workload* dan penjadwalan berbasis sensitivitas, biaya performa *confidential virtualization* dapat difokuskan pada *workload* kritis, sementara tugas kurang sensitif tetap memanfaatkan efisiensi VM konvensional. Pendekatan ini membuka jalur adopsi *confidential computing* yang lebih praktis dan layak secara operasional.

F. Perbandingan dengan Studi Sebelumnya

Temuan penelitian ini konsisten dengan studi sebelumnya mengenai *confidential virtualization*. *Overhead* rendah pada *workload* CPU-intensive dan lebih menonjol pada *workload* memory-intensive serta I/O-intensive sejalan dengan Yan dan Gopalan [15]. Penurunan *network throughput* juga konsisten dengan Bifrost yang mengidentifikasi I/O jaringan sebagai bottleneck utama CVM [16].

Pada eksekusi basis data, hasil penelitian juga serupa dengan temuan Qiu et al. [17], yaitu *workload* intensif memori dan I/O mengalami penalti performa yang tidak dapat diabaikan pada *confidential virtualization*. Hal ini memperkuat pandangan bahwa adopsi CVM untuk sistem transaksional sensitif realistis, tetapi harus disertai orkestrasi *workload* yang cermat.

Dari sisi keamanan, hasil penelitian mendukung argumen bahwa CVM memperkuat perlindungan *data in use*, tetapi tidak menghapus seluruh permukaan serangan. Temuan ini sejalan dengan studi SEVered [7], operasi I/O tidak terlindungi [8], *ciphertext side channel*s [9], [10], dan HECKLER [11].

Karena itu, CVM diposisikan sebagai model keamanan yang lebih kuat, namun tetap memerlukan kontrol dan kepercayaan tambahan.

Pembeda utama penelitian ini adalah integrasi tiga aspek yang sering dibahas terpisah: arsitektur CVM, pemetaan *overhead* performa, dan *security-aware placement*. Sebagian besar studi terdahulu berfokus pada satu dimensi, sedangkan penelitian ini menggabungkannya dalam kerangka operasional. Kontribusinya tidak hanya menilai apakah CVM aman atau cepat, tetapi juga bagaimana dan kapan CVM sebaiknya digunakan dalam infrastruktur virtualisasi heterogen.

Secara keseluruhan, hasil penelitian menunjukkan bahwa penempatan *workload* berbasis sensitivitas penting untuk menjadikan *confidential virtualization* lebih praktis pada produksi. Alih-alih memigrasikan seluruh *workload* ke CVM, adopsi selektif berdasarkan sensitivitas data dan profil sumber daya menjadi pendekatan paling seimbang.

Kesimpulan dan Saran

A. Kesimpulan

Berdasarkan perancangan, implementasi, dan evaluasi yang dilakukan, *Confidential Virtual Machine* (CVM) layak digunakan untuk meningkatkan perlindungan *data in use* pada lingkungan virtualisasi. Arsitektur yang diusulkan dapat diintegrasikan ke infrastruktur virtualisasi modern tanpa perubahan mendasar, meskipun memerlukan penyesuaian pada *provisioning*, observabilitas, dan verifikasi kepercayaan *platform*. Dari sisi kinerja, CVM menimbulkan *overhead* yang bervariasi menurut jenis *workload*. *Overhead* rendah pada *workload* dominan komputasi, tetapi meningkat pada *workload* intensif memori dan I/O, khususnya jaringan. Karena itu, adopsi CVM tidak sebaiknya dilakukan seragam untuk seluruh beban kerja, melainkan disesuaikan dengan karakteristik aplikasi. Dari sisi keamanan, CVM meningkatkan perlindungan melalui pengurangan visibilitas *host* terhadap memori tamu, perlindungan data sensitif saat diproses, dan dukungan *attestation*. Dengan demikian, CVM lebih sesuai untuk *workload* dengan kerahasiaan tinggi yang membutuhkan jaminan keamanan lebih kuat daripada virtualisasi konvensional. Hasil pengujian penempatan menunjukkan bahwa *security-aware placement* paling seimbang dibanding *random placement* dan *performance-based placement*. Strategi ini menempatkan sebagian besar *workload* sensitif pada *node* CVM sekaligus menjaga kinerja sistem tetap dapat diterima. Dengan demikian, keseimbangan keamanan dan performa dapat dicapai melalui *placement* berbasis sensitivitas data. Secara keseluruhan, CVM lebih efektif bila diterapkan secara selektif dan terarah, bukan menyeluruh untuk semua *workload*. Pendekatan ini memungkinkan organisasi memperoleh manfaat keamanan signifikan tanpa menanggung penurunan performa yang tidak perlu pada seluruh sistem.

Daftar Pustaka

[1] M. Russinovich *et al.*, “Toward confidential cloud computing,” *Commun. ACM*, vol. 64, no. 6, pp. 54–61, 2021, doi: 10.1145/3453930.

[2] Q. Wang and D. Oswald, “Confidential computing on heterogeneous CPU-GPU systems: Survey and future directions,” *ACM Comput. Surv.*, 2026, doi: 10.1145/3793532.

[3] I. Advanced Micro Devices, “SEV-SNP: Strengthening VM isolation with integrity protection and more,” AMD, 2020. [Online]. Available: https://www.amd.com/en/developer/sev.html

[4] G. Cloud, “Confidential VM overview,” 2026. [Online]. Available: https://docs.cloud.google.com/confidential-computing/confidential-vm/docs/confidential-vm-overview

[5] M. Azure, “About Azure confidential VMs,” 2026. [Online]. Available: https://learn.microsoft.com/en-us/azure/confidential-computing/confidential-vm-overview

[6] M. Misono, D. Stavrakakis, N. Santos, and P. Bhatotia, "Confidential VMs explained: An empirical analysis of AMD SEV-SNP and Intel TDX," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 8, no. 3, 2024, doi: 10.1145/3700418.

[7] M. Morbitzer, M. Huber, J. Horsch, and S. Wessel, "SEVered: Subverting AMD's virtual machine encryption," 2018. doi: 10.1145/3193111.3193112.

[8] M. Li, Y. Zhang, Z. Lin, and Y. Solihin, "Exploiting unprotected I/O operations in AMD's secure encrypted virtualization," 2019. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/li-mengyuan

[9] M. Li, Y. Zhang, H. Wang, K. Li, and Y. Cheng, "CIPHERLEAKS: Breaking constant-time cryptography on AMD SEV via the ciphertext side channel," 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/li-mengyuan

[10] M. Li, L. Wilke, J. Wichelmann, T. Eisenbarth, R. Teodorescu, and Y. Zhang, "A systematic look at ciphertext side channels on AMD SEV-SNP," 2022. [Online]. Available: https://www.ieee-security.org/TC/SP2022/program-papers.html

[11] B. Schlüter, S. Sridhara, M. Kuhne, A. Bertschi, and S. Shinde, "HECKLER: Breaking confidential VMs with malicious interrupts," 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/schlüter

[12] W. Ozga, G. D. H. Hunt, M. V Le, E. R. Palmer, and A. Shinnar, "Towards a formally verified security monitor for VM-based confidential computing," 2023. doi: 10.1145/3623652.3623668.

[13] V. Narayanan *et al.*, "Remote attestation of confidential VMs using ephemeral vTPMs," 2023. doi: 10.1145/3627106.3627112.

[14] J. Pecholt and S. Wessel, "CoCoTPM: Trusted platform modules for virtual machines in confidential computing environments," 2022. doi: 10.1145/3564625.3564648.

[15] M. Yan and K. Gopalan, "Performance overheads of confidential virtual machines," 2023. doi: 10.1109/MASCOTS59514.2023.10387607.

[16] D. Li *et al.*, "Bifrost: Analysis and optimization of network I/O tax in confidential virtual machines," 2023. [Online]. Available: https://www.usenix.org/conference/atc23/presentation/li-dingji

[17] L. Qiu, R. Taft, A. Shraer, and G. Kollios, "The price of privacy: A performance study of confidential virtual machines for database systems," 2024. [Online]. Available: https://rytaft.github.io/price-of-privacy.pdf

[18] F. Schwarz and C. Rossow, "00SEVen - Re-enabling virtual machine forensics: Introspecting confidential VMs using privileged in-VM agents," 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/schwarz

[19] S. Feizollahibarough and M. Ashtiani, "A security-aware virtual machine placement in the cloud using hesitant fuzzy decision-making processes," *J. Supercomput.*, vol. 77, pp. 5606–5636, 2021, doi: 10.1007/s11227-020-03496-4.