JUKOMTEK
(Jurnal Komputer Dan Teknologi)
e-ISSN 2961-9009
p-ISSN 2963-1289

ANALISIS PERFORMA DAN KEAMANAN SISTEM BACKUP DAN RESTORE PADA PRIVATE CLOUD BERBASIS CASAOS DI LINGKUNGAN VIRTUAL

Lukmanul Hakim[1], Rama Rara Isano[2], Yustian Servanda[3]

[1'2'3]Program Studi Teknologi Informasi, Universitas Mulia, Balikpapan, Kalimantan Timur
** [1]lukmanulhakim20040212@gmail.com, [2] ramaisano28@gmail.com, [3] yustians@universitasmulia.ac.id*

INFO ARTIKEL

Sejarah Artikel:
Diterima Tgl. 04/06/2026
Diperbaiki Tgl. 15/07/2026
Disetujui Tgl. 19/07/2026
Tersedia daring Tgl. 25/07/2026

e-ISSN 2961-9009
p-ISSN 2963-1289

DOI:
https://doi.org/10.64626/jukomtek.v5i2.681

Abstract: *Data loss caused by system failures and cyber threats requires a reliable data protection strategy for private cloud environments. This study aims to analyze the performance and security of a backup and restore system using Duplicati on a CasaOS-based private cloud in a virtual environment. An experimental method was employed by implementing CasaOS on Ubuntu Server running in VirtualBox, deploying Duplicati through Docker, resolving container network isolation, volume mapping, and privilege issues, and evaluating the performance of backup and restore processes using a 1.23 GB dataset. The results show that the proposed configuration successfully resolved network and permission constraints, enabling reliable backup and restore operations. The full backup process achieved an average throughput of 20.22 MB/s, while the incremental backup reached 87.42 MB/s, and the restore process was completed successfully without access failures. In conclusion, the integration of CasaOS, Docker, and Duplicati provides a secure, stable, and cost-effective backup and restore solution suitable for small- and medium-scale private cloud deployments.*

Keywords: *Private Cloud, CasaOS, Containerization, Data Security, Throughput*

Abstrak: Kerentanan kehilangan data akibat kegagalan sistem maupun ancaman siber memerlukan strategi perlindungan data yang andal pada lingkungan private cloud. Penelitian ini bertujuan untuk menganalisis performa dan keamanan sistem backup dan restore menggunakan Duplicati pada private cloud berbasis CasaOS di lingkungan virtual. Metode penelitian yang digunakan adalah metode eksperimen dengan mengimplementasikan CasaOS pada Ubuntu Server di VirtualBox, menjalankan Duplicati berbasis Docker, menyelesaikan kendala isolasi jaringan, volume mapping, dan hak akses kontainer, serta menguji performa proses backup dan restore menggunakan data uji berukuran 1,23 GB. Hasil penelitian menunjukkan bahwa konfigurasi yang diterapkan berhasil mengatasi kendala jaringan dan hak akses sehingga proses backup dan restore berjalan dengan baik. Proses full backup menghasilkan throughput rata-rata sebesar 20,22 MB/s, sedangkan incremental backup mencapai 87,42 MB/s, dan proses restore berhasil dilakukan tanpa kegagalan akses data. Kesimpulannya, integrasi CasaOS, Docker, dan Duplicati mampu menyediakan sistem backup dan restore yang aman, stabil, serta berbiaya rendah sehingga layak diterapkan sebagai

E-ISSN : 2776-9666
P-ISSN : 2776-1789

solusi perlindungan data pada lingkungan private cloud skala kecil hingga menengah.

Kata Kunci: Private Cloud, CasaOS, Kontainerisasi, Keamanan Data, Throughput

PENDAHULUAN

Kehilangan data digital akibat kerusakan perangkat keras, serangan perangkat pemeras (*ransomware*), ataupun kelalaian operasional manusia merupakan ancaman nyata dalam manajemen teknologi informasi saat ini. Pertumbuhan penggunaan cloud storage memberikan kemudahan dalam penyimpanan dan akses data, namun juga menimbulkan berbagai tantangan keamanan seperti akses tidak sah, kebocoran data, pengungkapan informasi sensitif, dan pelanggaran privasi. Oleh karena itu, organisasi mulai mempertimbangkan penerapan *private cloud* untuk meningkatkan kontrol terhadap data dan infrastruktur yang digunakan. Berdasarkan penelitian (Yang et al., 2020) ,keamanan data, integritas, kerahasiaan, dan ketersediaan informasi merupakan aspek utama yang harus diperhatikan dalam sistem penyimpanan berbasis cloud. Sebagai jalan keluar, platform *private cloud* menawarkan kendali penuh atas hak akses dan fisik infrastruktur penyimpanan secara mandiri.

CasaOS merupakan platform open-source yang berjalan di atas sistem operasi Linux dan menyediakan antarmuka grafis untuk memudahkan pengelolaan aplikasi berbasis kontainer. Dengan dukungan Docker, pengguna dapat melakukan instalasi dan pengelolaan layanan secara lebih sederhana tanpa harus menggunakan perintah terminal yang kompleks. Guna menjamin ketersediaan data yang tersimpan di dalam ekosistem tersebut, diperlukan sebuah sistem *backup* sekunder yang berjalan secara otomatis dan terisolasi dengan baik. Sistem *backup* pada penelitian ini menggunakan Duplicati karena mendukung enkripsi data sebelum proses penyimpanan serta mampu mengoptimalkan penggunaan ruang penyimpanan melalui mekanisme kompresi dan deduplikasi data. Pendekatan ini sejalan dengan konsep optimasi penyimpanan cloud yang memanfaatkan deduplikasi untuk mengurangi redundansi data tanpa mengurangi integritas informasi (Kavitha et al., 2025).

Penelitian ini memfokuskan diri pada perancangan, penyelesaian kendala teknis isolasi jaringan kontainer, dan pengujian performa fungsional dari mekanisme cadang-pulih (*backup* dan *restore*) data. Kontribusi penelitian ini terletak pada implementasi dan evaluasi integrasi CasaOS, Docker, dan Duplicati dalam lingkungan virtual untuk menghasilkan sistem *backup* dan *restore* yang aman serta mudah direplikasi pada skala kecil dan menengah. Melalui

E-ISSN : 2776-9666
P-ISSN : 2776-1789

pendekatan eksperimental ini, diharapkan model penataan sistem penyimpanan mandiri yang aman dapat diterapkan secara luas untuk kebutuhan komputasi skala kecil hingga menengah.

LANDASAN TEORI

Arsitektur Private Cloud dan CasaOS

Private cloud merupakan model layanan cloud yang dikelola secara khusus oleh satu organisasi sehingga memberikan kontrol yang lebih tinggi terhadap data, kebijakan keamanan, dan infrastruktur penyimpanan dibandingkan public cloud. Model ini banyak digunakan pada lingkungan yang membutuhkan perlindungan data sensitif dan pengelolaan sumber daya secara mandiri (Yang et al., 2020).

Konsep *private cloud* mengacu pada model komputasi awan yang menyediakan sumber daya komputasi secara eksklusif untuk satu organisasi. Menurut(Mell & Grance, 2012) ,model cloud memungkinkan akses jaringan yang luas, fleksibilitas sumber daya, serta pengelolaan layanan secara terpusat sesuai kebutuhan pengguna.

CasaOS berjalan di atas distribusi Linux sebagai lapisan abstraksi yang mengelola sistem penyimpanan lokal, manajemen berkas, serta instalasi aplikasi pihak ketiga dengan memanfaatkan kesederhanaan arsitektur Docker engine.

Penerapan *private cloud* memberikan fleksibilitas dalam pengelolaan layanan dan sumber daya komputasi karena seluruh infrastruktur dapat dikendalikan secara mandiri oleh organisasi. Model ini memungkinkan proses deployment layanan dilakukan secara lebih terstruktur sesuai kebutuhan operasional dan kebijakan keamanan yang diterapkan (Mohammed Maaz et al., 2023).

Selain memberikan kontrol yang lebih tinggi terhadap data, arsitektur penyimpanan cloud juga harus mempertimbangkan aspek skalabilitas dan keamanan agar mampu melayani kebutuhan penyimpanan yang terus meningkat. (Wahid et al., 2026) menyatakan bahwa keseimbangan antara keamanan dan skalabilitas menjadi faktor penting dalam perancangan infrastruktur cloud modern.

Keamanan Data dan Kontainerisasi

Dalam sistem cloud storage, keamanan data menjadi faktor yang sangat penting karena data tersimpan dan diakses melalui jaringan. Oleh sebab itu, aspek kerahasiaan, integritas, dan ketersediaan data perlu dijaga agar informasi tetap aman dan dapat digunakan ketika dibutuhkan. Untuk memenuhi kebutuhan tersebut, berbagai sistem cloud modern menerapkan teknik kriptografi, kontrol akses, serta mekanisme verifikasi integritas data guna melindungi informasi dari akses tidak sah maupun modifikasi yang tidak diinginkan.

Docker memisahkan aplikasi ke dalam lingkungan terisolasi yang disebut kontainer

E-ISSN : 2776-9666
P-ISSN : 2776-1789

agar dapat berjalan secara konsisten di berbagai infrastruktur sistem operasi host (Merkel, 2014).

Pendekatan kontainerisasi memungkinkan aplikasi dijalankan secara terisolasi tanpa memerlukan virtualisasi penuh seperti mesin virtual tradisional. Mekanisme ini memberikan efisiensi penggunaan sumber daya sekaligus mempermudah proses distribusi dan pemeliharaan aplikasi pada berbagai lingkungan sistem (Merkel, 2014).

Hasil penelitian (DYMORA & MAZUREK, 2025) menunjukkan bahwa teknologi kontainerisasi mampu memberikan penggunaan sumber daya yang lebih efisien dibandingkan pendekatan virtualisasi tradisional, terutama dalam implementasi layanan web yang membutuhkan skalabilitas dan performa tinggi. Penggunaan kontainer juga memungkinkan proses deployment aplikasi dilakukan dengan lebih cepat serta konsumsi sumber daya yang lebih rendah dibandingkan mesin virtual konvensional.

Perkembangan teknologi kontainer dalam satu dekade terakhir menunjukkan peningkatan penggunaan yang signifikan pada lingkungan cloud-native karena mampu meningkatkan portabilitas aplikasi dan efisiensi pengelolaan layanan. Teknologi ini telah menjadi salah satu fondasi utama dalam implementasi layanan modern berbasis cloud (Madhavapeddy et al., 2025).Dari sisi performa, pendekatan kontainerisasi juga dinilai mampu memberikan penggunaan sumber daya yang lebih efisien dibandingkan virtualisasi tradisional pada berbagai skenario layanan web dan server (DYMORA & MAZUREK, 2025). **Sistem Enkripsi dan Volume Duplicati**

Duplicati merupakan perangkat lunak *backup* open-source yang mendukung sistem operasi Linux, Windows, dan macOS. Duplicati menyediakan antarmuka berbasis web serta mendukung mode layanan (service mode) sehingga proses *backup* dapat dijalankan secara otomatis di latar belakang. Selain itu, Duplicati menggunakan basis data lokal untuk menyimpan konfigurasi, metadata, dan informasi proses pencadangan yang diperlukan saat melakukan pemulihan data (*Duplicati Documentation _ Duplicati*, n.d.).

Untuk menjaga kerahasiaan data, Duplicati mendukung penggunaan enkripsi AES-256 sebelum data disimpan pada media *backup*. Selain itu, Duplicati menerapkan mekanisme kompresi dan deduplikasi data untuk mengurangi redundansi serta meningkatkan efisiensi penggunaan ruang penyimpanan. Pada penelitian ini, ukuran volume arsip *backup* dikonfigurasi sebesar 50 MB untuk membagi data menjadi beberapa berkas cadangan yang lebih terstruktur dan mudah dikelola selama proses *backup* maupun *restore* ((Team, 2026)ne et al., 2025). Algoritma AES yang dikembangkan oleh Daemen dan Rijmen merupakan salah satu standar enkripsi simetris yang banyak digunakan untuk melindungi data digital karena

E-ISSN : 2776-9666
P-ISSN : 2776-1789

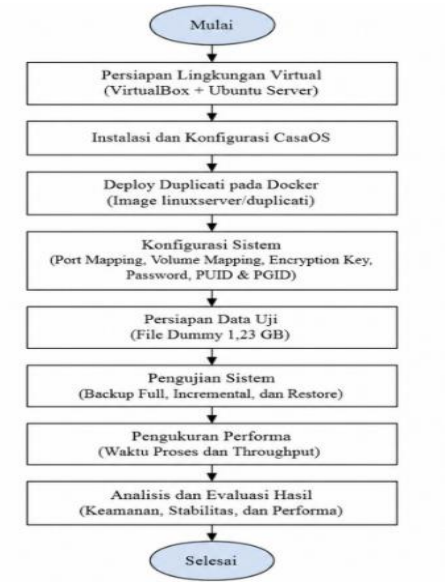
memiliki tingkat keamanan yang tinggi serta efisien untuk berbagai kebutuhan penyimpanan dan transmisi data (Daemen & Rijmen, 2002).

Penggunaan algoritma AES-256 banyak diterapkan pada sistem penyimpanan modern karena mampu memberikan tingkat perlindungan data yang tinggi dengan tetap mempertahankan efisiensi proses enkripsi dan dekripsi. Kombinasi teknik kriptografi yang kuat berperan penting dalam menjaga kerahasiaan informasi yang disimpan pada media cloud maupun sistem *backup* digital (Surya et al., 2025).

Penerapan mekanisme enkripsi dan kebijakan kontrol akses pada media penyimpanan cloud berperan penting dalam menjaga kerahasiaan data serta mengurangi risiko akses tidak sah terhadap informasi yang tersimpan (Mallick et al., 2026).

Teknik deduplikasi digunakan untuk mengurangi redundansi data dengan menyimpan hanya blok data yang unik sehingga kapasitas penyimpanan dapat dimanfaatkan secara lebih efisien tanpa memengaruhi integritas informasi (Kavitha et al., 2025).

METODE PENELITIAN



Gambar 1. Tahapan Penelitian

Penelitian ini dijalankan secara kronologis melalui tahapan eksperimen laboratorium komputer terstruktur. Tahap awal dimulai dengan membangun mesin virtual Ubuntu Server di dalam perangkat lunak *hypervisor* VirtualBox, diikuti oleh instalasi platform ekosistem CasaOS. Selanjutnya, layanan penunjang Duplicati digelar dari repositori resmi linuxserver/duplicati menggunakan perintah baris Docker CLI.

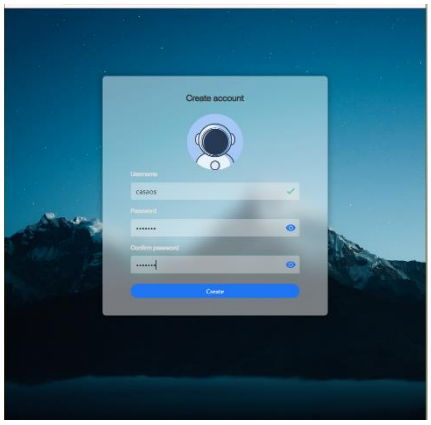
Pengujian fungsionalitas dan performa dilakukan dengan mempersiapkan berkas sampel data (*dummy data*) pada mesin laptop Windows host yang bertindak sebagai klien. Pengiriman berkas ke dalam direktori virtual diatur melalui manajemen folder bersama (*bind*

E-ISSN : 2776-9666
P-ISSN : 2776-1789

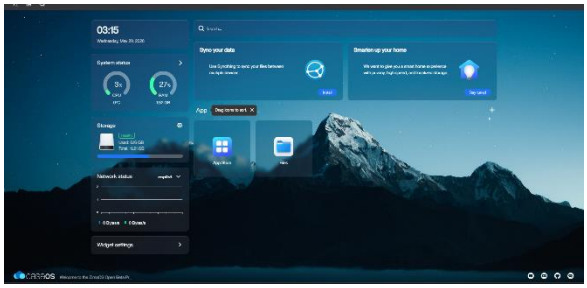
mount directory) ke jalur /config di dalam server. Pengambilan data metrik waktu diukur secara riil menggunakan alat pencatat waktu eksternal sejalan dengan peluncuran instruksi perintah eksekusi tugas pada sistem, seperti yang direpresentasikan pada alur struktur data Tabel 1.

Tabel 1. Struktur Variabel Pengujian Sistem

Variabel Uji	Batasan Ukuran Volume	Lokasi Target	Status Enkripsi
Sampel Data A	50 MB	Internal Container Path	AES-256 Enabled



Gambar 2. Tampilan Login CasaOS



Gambar 3. Halaman Beranda

HASIL DAN PEMBAHASAN

Deployment dan Resolusi Kendala Jaringan serta Hak Akses Kontainer

Pada fase awal *deployment* di server baru, layanan Duplicati sempat mengalami kegagalan total yang ditandai dengan munculnya pesan kesalahan *Connection reset by peer* saat diuji menggunakan utilitas perintah curl lokal. Investigasi melalui pembacaan data log kontainer menunjukkan bahwa sistem keamanan Duplicati menolak berjalan akibat tidak ditemukannya kunci enkripsi untuk basis data setelan internal. Masalah pemutusan koneksi sepihak ini terjadi karena konfigurasi dasar kontainer membatasi dirinya hanya untuk mendengarkan alamat internal localhost saja (Team, 2026)

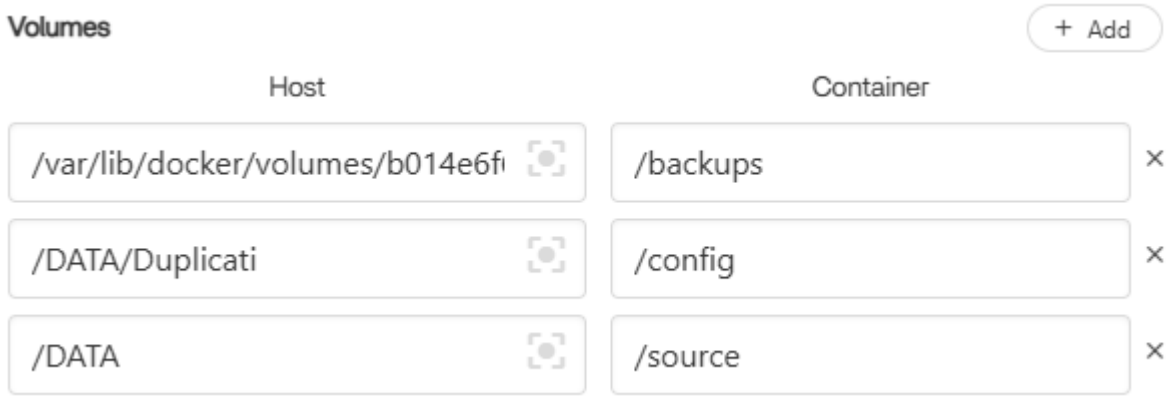
E-ISSN : 2776-9666
P-ISSN : 2776-1789

Untuk mengatasi hambatan tersebut, kontainer lama dibongkar dan dibangun kembali secara instan dengan menyisipkan parameter lingkungan keamanan yang baru secara eksplisit. Penambahan variabel -e SETTINGS_ENCRYPTION_KEY sukses menginisialisasi basis data sistem, sementara penegakan argumen --webservice-interface=any berhasil membuka gerbang isolasi kontainer sehingga server dapat diakses dari luar menggunakan IP lokal 10.150.49.1 (Duplicati Team, 2025). Selain itu, parameter --webservice-password diterapkan untuk menambahkan autentikasi pada antarmuka administrasi web sehingga akses konfigurasi sistem memerlukan kredensial pengguna (Duplicati Team, 2025).

```
casaos@casaos:~$ sudo docker run -d --name=duplicati --restart=always -p 8200:8200 -e CLI_ARGS="--we
bservice-interface=any --webservice-allowed-host=* --webservice-password=admin123" -e SETTINGS_ENCRY
PTION_KEY="rahasiakita123" -v /DATA/Duplicati:/config linuxserver/duplicati:latest
bf82e3d3a51be2c81f9e809127844863f9b5b57ac41a4ab7a4f46bec75da755e
```

Gambar 4. Implementasi Konfigurasi Duplicati Menggunakan Docker CLI

Selain kendala jaringan, ditemukan dua kendala kritis lain pada pemetaan penyimpanan lokal di dalam lingkungan CasaOS. Kendala pertama adalah kegagalan pembacaan direktori *source* akibat kesalahan *volume mapping* otomatis pada kontainer Docker yang mengarah pada direktori isolasi sistem (/var/lib/docker/volumes/...). Masalah ini diselesaikan dengan melakukan pemetaan ulang jalur *Host* secara manual menuju direktori penyimpanan utama server, yaitu /DATA.



Gambar 5. Perbaikan *Volume Mapping* pada Kontainer Duplicati untuk Mengatasi Kegagalan Pembacaan Direktori Source

Berdasarkan Gambar 5, konfigurasi volume pada CasaOS diubah secara manual dengan memetakan direktori host menuju direktori kontainer yang sesuai. Langkah ini dilakukan untuk mengatasi kegagalan pembacaan data akibat penggunaan volume Docker otomatis yang mengarah ke direktori isolasi sistem.

Kendala kedua muncul saat proses *restore* data (*restore*), di mana sistem menolak mengeksekusi berkas dengan pesan kesalahan *UnauthorizedAccessException: Access to the*

E-ISSN : 2776-9666
P-ISSN : 2776-1789

path is denied. Hal ini terjadi karena restriksi keamanan bawaan kernel Ubuntu terhadap hak milik *User ID* kontainer. Resolusi komprehensif dilakukan dengan mengonfigurasi variabel lingkungan (*Environment Variables*) PUID=0 dan PGID=0 pada pengaturan CasaOS untuk memberikan hak akses tingkat tertinggi (*Root*) secara permanen kepada layanan Duplicati ((Team, 2026)ne et al., 2025)

Gambar 6. Konfigurasi PUID=0 dan PGID=0 untuk Mengatasi Kendala Hak Akses pada Proses *Restore* Data

Berdasarkan Gambar 6, variabel lingkungan PUID dan PGID dikonfigurasi dengan nilai 0 untuk memberikan hak akses root kepada layanan Duplicati. Konfigurasi ini diperlukan karena proses *restore* sebelumnya gagal akibat pembatasan hak akses sistem operasi Ubuntu yang menyebabkan munculnya pesan kesalahan UnauthorizedAccessException. Setelah perubahan diterapkan dan kontainer dijalankan kembali, proses *restore* data dapat dilakukan tanpa kendala akses terhadap direktori tujuan.

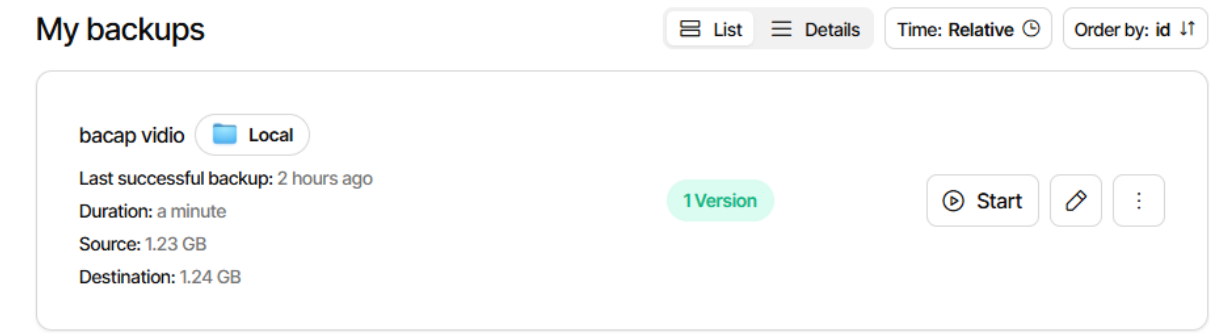
Analisis Performa Throughput Backup Data

Pengujian performa proses *backup* data dijalankan dengan mengunggah berkas video sampel berukuran besar bernama nessus.mp4 dengan kapasitas total sebesar 1,23 GB (1.230 MB). Pengaturan ukuran potongan volume tetap dipertahankan sebesar 50 MB untuk meminimalkan beban kerja memori utama server virtual saat melakukan kompresi data arsip secara paralel. Penggunaan ukuran volume yang seragam membantu proses pengelolaan arsip *backup* sehingga proses pencadangan dapat berlangsung secara konsisten pada lingkungan pengujian.

Tabel 2. Hasil Pengukuran Waktu Pengujian Performa Eksekusi Data

Skenario Uji	Ukuran Data asal (GB)	Ukuran Hasil Ekstraksi (GB)	Total Waktu Proses	Kecepatan (*Throughput*)	Status
Full Backup (Awal)	1,23	1,24	1 Menit 00 Detik (60,81 s)	20,22 MB/s	Sukses
Incremental Backup / *Root Data Restore* (Estimasi)	1,23	1,24	0 Menit 14 Detik (14,07 s)	87,42 MB/s	Sukses
	1,23	1,23	0 Menit 50 Detik (50,00 s)	24,60 MB/s	Sukses

E-ISSN : 2776-9666
P-ISSN : 2776-1789

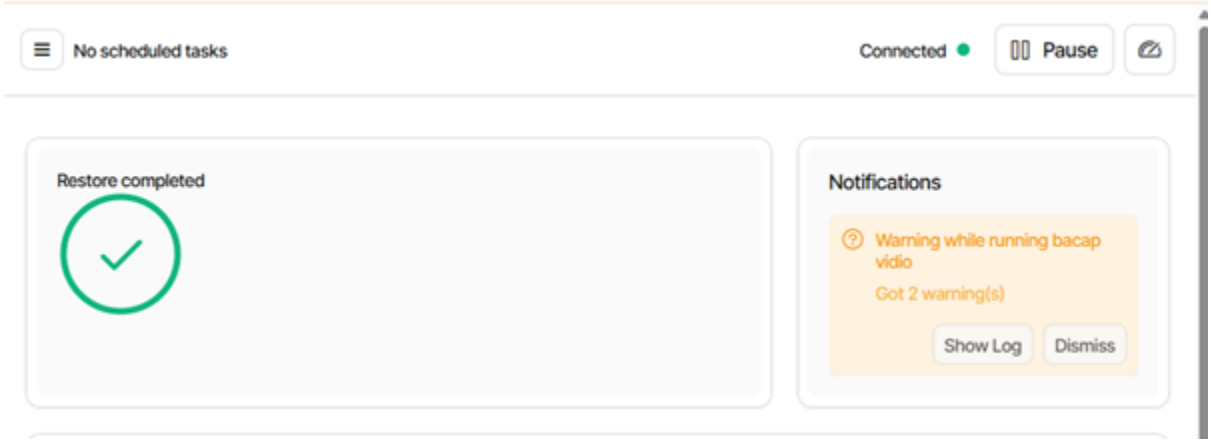


Gambar 7. Hasil Eksekusi Full *Backup* Data Menggunakan Duplicati

Berdasarkan Gambar 7, proses full *backup* terhadap berkas uji berukuran 1,23 GB berhasil diselesaikan oleh Duplicati dalam waktu sekitar satu menit. Hasil *backup* menghasilkan ukuran penyimpanan tujuan sebesar 1,24 GB. Perbedaan ukuran tersebut disebabkan oleh pembentukan metadata, indeks arsip, serta mekanisme enkripsi yang diterapkan secara otomatis selama proses *backup* berlangsung.

Selama proses pengujian tidak ditemukan gangguan layanan yang menyebabkan proses *backup* gagal. Hal ini menunjukkan bahwa lingkungan virtual yang digunakan cukup stabil untuk menjalankan layanan Duplicati. Temuan ini sejalan dengan penelitian (Rido et al., 2026) yang menunjukkan bahwa penerapan virtualisasi server dapat meningkatkan efisiensi pengelolaan sumber daya serta menjaga ketersediaan layanan pada infrastruktur teknologi informasi.

Temuan ini sejalan dengan penelitian (DYMORA & MAZUREK, 2025) yang menyatakan bahwa pendekatan kontainerisasi memberikan efisiensi penggunaan CPU dan memori yang lebih baik dibandingkan pendekatan virtualisasi tradisional pada berbagai layanan berbasis web. Efisiensi tersebut berkontribusi terhadap stabilitas layanan ketika sistem menerima beban kerja yang tinggi.



Gambar 8. Hasil Proses *Restore* Data pada Duplicati dengan Status *Restore* Completed

E-ISSN : 2776-9666
P-ISSN : 2776-1789

Berdasarkan Gambar 8, proses *restore* berhasil dijalankan dengan status "*Restore* completed" tanpa ditemukan kegagalan akses terhadap direktori tujuan. Hasil ini menunjukkan bahwa konfigurasi *volume mapping* serta pengaturan hak akses root melalui variabel PUID dan PGID telah berfungsi dengan baik sehingga data dapat dipulihkan secara utuh dari media *backup*.

Keberhasilan proses *backup* dan *restore* pada penelitian ini sejalan dengan penelitian (Arman & Wijaya, 2026) yang menunjukkan bahwa mekanisme pencadangan data pada lingkungan virtualisasi mampu meningkatkan ketersediaan data dan mendukung proses pemulihan sistem ketika terjadi gangguan operasional.

Berdasarkan rekaman data hasil pengujian pada Tabel 2, proses *backup* awal (*Full Backup*) terhadap objek uji berukuran 1,23 GB membutuhkan durasi waktu eksekusi sebesar 60,81 detik dengan laju transfer rata-rata (*throughput*) berada pada angka 20,22 MB/s. Hasil ini merepresentasikan efisiensi pemrosesan data lokal yang stabil. Terdapat sedikit pembengkakan ukuran data pada sisi *destination storage* menjadi 1,24 GB. Fenomena variasi ukuran ini dinilai wajar karena adanya penambahan metadata, indeks arsip, serta mekanisme enkripsi yang diterapkan selama proses *backup* berlangsung.

Analisis Performa Throughput Optimasi Incremental Data

Setelah implementasi variabel optimasi hak akses *Root* (PUID=0 dan PGID=0) diterapkan, dilakukan pengujian *backup* ulang (*Incremental Backup*) pada objek data yang sama. Hasil visualisasi performa menunjukkan lonjakan efisiensi waktu yang sangat signifikan, di mana proses pemindaian dan penyelarasan data selesai hanya dalam durasi 14,07 detik dengan kalkulasi logis *throughput* setara 87,42 MB/s.

Keberhasilan proses *backup* dan *restore* pada penelitian ini menunjukkan bahwa aspek ketersediaan data (availability) sebagai salah satu prinsip utama keamanan informasi dapat terpenuhi. Menurut (Yang et al., 2020), ketersediaan data merupakan komponen penting dalam sistem penyimpanan berbasis cloud karena menentukan kemampuan sistem untuk menyediakan data ketika dibutuhkan pengguna.

Kecepatan proses pemulihan data menjadi salah satu indikator penting dalam keberhasilan sistem *backup*. (Dilan Gomas & Rathnayake, 2026) menjelaskan bahwa kemampuan sistem dalam memenuhi target waktu pemulihan Recovery Time Objective (RTO) merupakan faktor penting untuk menjaga kontinuitas layanan ketika terjadi gangguan atau kehilangan data.

Salah satu faktor yang memengaruhi peningkatan performa tersebut adalah mekanisme deduplikasi yang digunakan Duplicati. Karena data yang telah dicadangkan sebelumnya tidak

E-ISSN : 2776-9666
P-ISSN : 2776-1789

dikirim ulang secara penuh, waktu pemrosesan dapat dipersingkat. Selain itu, penggunaan kontainer Docker juga membantu menjaga penggunaan sumber daya tetap ringan selama proses *backup* berlangsung.

(DYMORA & MAZUREK, 2025) menunjukkan bahwa teknologi kontainer seperti Docker dan Podman mampu memberikan performa tinggi dengan penggunaan sumber daya yang relatif efisien sehingga cocok digunakan pada layanan berbasis web maupun sistem penyimpanan data. Sistem tidak melakukan penyalinan ulang terhadap blok biner video yang identik, melainkan hanya melakukan proses pemindaian metadata (index scanning) terhadap tanda tangan hash berkas dan memperbarui tabel metadata pada basis data lokal ((Team, 2026)ne et al., 2025). Di sisi lain, konfigurasi hak akses root memungkinkan proses *backup* dan *restore* mengakses direktori tujuan tanpa mengalami kendala perizinan yang sebelumnya menyebabkan kegagalan proses.

KESIMPULAN

Implementasi rancang bangun perlindungan data berbasis CasaOS dan Duplicati mampu menyediakan mekanisme *backup* dan *restore* yang berjalan stabil serta mendukung perlindungan data melalui penerapan enkripsi pada proses penyimpanan. Melalui serangkaian eksperimen penanganan galat, ditemukan fakta teknis bahwa ketepatan deklarasi variabel enkripsi dan konfigurasi *binding port* eksternal merupakan faktor penentu utama keberhasilan eliminasi galat pemutusan jaringan kontainer. Hasil pengujian performa kuantitatif menunjukkan bahwa pembagian volume arsip standar sebesar 50 MB mampu menghasilkan kecepatan transfer data yang andal dan konstan, di mana proses *restore* selalu berjalan lebih responsif akibat minimnya beban enkripsi ulang pada sisi prosesor.

DAFTAR PUSTAKA

Arman, M., & Wijaya, N. (2026). *Implementasi Backup Image VM Pada Proxmox Virtual Environment (Promox VE) Untuk Kebutuhan Server*. *17*(01), 75–88.

Daemen, J., & Rijmen, V. (2002). The Design of Rijndael. *New York*, 255. http://portal.acm.org/citation.cfm?id=560131

Dilan Gomas, A. S., & Rathnayake, R. M. N. B. (2026). Optimizing Recovery Objectives (RTO and RPO) in Secure Linux NAS Environments: A Design Science Approach to Ransomware Resilience. *Asian Journal of Social Science and Management Technology*, *8*(1), 82–94. https://www.ajssmt.com/Papers/8018294.pdf

DYMORA, P., & MAZUREK, M. (2025). Performance Evaluation of Selected Containerization Methods in Web Services Applications. *Journal of Education, Technology and Computer Science*, *6*(36), 173–185.

E-ISSN : 2776-9666
P-ISSN : 2776-1789

https://doi.org/10.15584/jetacomps.2025.6.16

Kavitha, R., Shaik, M. S., Swarnalatha, N., Pujitha, M., Hussaini, S. A., Khan, S., & Ali, S. (2025). Data Deduplication-based Efficient Cloud Optimisation Technique: Optimizing Cloud Storage through Data Deduplication. *International Journal of Information Engineering and Electronic Business*, *17*(2), 129–146. https://doi.org/10.5815/ijieeb.2025.02.07

Madhavapeddy, A., Scott, D. J., Ferris, P., Gibb, R. T., & Gazagnaire, T. (2025). Functional Networking for Millions of Docker Desktops (Experience Report). *Proceedings of the ACM on Programming Languages*, *9*(ICFP). https://doi.org/10.1145/3747525

Mallick, B., Parida, P., Prasad, B., Nayak, C., Panda, M. K., Ali, N., & Kumar, N. M. (2026). Quantum-Secure Communication for Future Cyber-Physical and IoT Systems: A Systematic Review of Classical to Learning Approaches. *Computers*, *15*(6), 1–23. https://doi.org/10.3390/computers15060389

Mell, P., & Grance, T. (2012). The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology. *Public Cloud Computing: Security and Privacy Guidelines*, 97–101.

Merkel, D. (2014). Docker : Lightweight Linux Containers for Consistent Development and Deployment Docker : a Little Background Under the Hood. *Linux Journal*, *2014*(239), 2–7. http://delivery.acm.org.ezproxy.library.wisc.edu/10.1145/2610000/2600241/11600.html?ip=128.104.46.196&id=2600241&acc=ACTIVE SERVICE&key=066E7B0AFE2DCD37.066E7B0AFE2DCD37.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1557803890_216b4a0168a6b29b8f2e7a74

Mohammed Maaz, Md Akif Ahmed, Md Maqsood, & Dr Shridevi Soma. (2023). Development Of Service Deployment Models In Private Cloud. *Journal of Scientific Research and Technology*, (9), 1–12. https://doi.org/10.61808/jsrt74

Rido, R., Pahlevi, M. R., Maliki, I., & Iqbal, M. (2026). Implementasi Server Virtualisasi Menggunakan Proxmox untuk Manajemen Infrastruktur Jaringan dan Layanan Internet pada Institusi Pendidikan. *Jurnal Sistem Informasi, Sains Data, Dan Informatika*, *1*(1), 25–33. https://doi.org/10.56956/s6zv7r91

Surya, J., Louis, A., Rini, F., Mulyati, S., & Elzas, E. (2025). Cryptographic Framework for Cloud-Based Document Storage Using Aes-256 and Sha-256 Hybrid Systems. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *11*(2), 535–549. https://doi.org/10.33480/jitk.v11i2.7132

E-ISSN : 2776-9666
P-ISSN : 2776-1789

Wahid, W. N., Nurdianingsih, F., & Parker, J. (2026). Comparative Analysis of Cloud Storage Architectures for Scalability and Security. *Blockchain Frontier Technology (B-Front)*, *5*(2), 183–193. http://10.0.134.2/bfront.v5i2.857

Yang, P., Xiong, N., & Ren, J. (2020). Data Security and Privacy Protection for Cloud Storage: A Survey. *IEEE Access*, *8*, 131723–131740. https://doi.org/10.1109/ACCESS.2020.3009876