



Pengembangan Sistem *Backup Server* EPrint dengan Ubuntu 24.04 ke *Cloud Storage Nextcloud* (Studi Kasus: Sekolah Tinggi Pariwisata (STP) Sahid Surakarta)

Paryanta¹, Ernes Cahyo N.², Muhammad Johan Fatoni³

¹ Department of Computer System, Universitas Dharma AUB Surakarta, Indonesia

^{2,3} Department of Computer System, Universitas Dharma AUB Surakarta, Indonesia

Email author: paryanta@stmik-aub.ac.id¹, ermes.cahyo@undha-aub.ac.id², fatonijohann@gmail.com³

Article Info

Article history:

Received April 30, 2026

Revised May 6, 2026

Accepted May 26, 2026

Available: July 12, 2026

Published: July 30, 2026

Keywords:

Backup System;

EPrint;

Ubuntu 24.04;

Nextcloud;

Private Cloud;

Automation

ABSTRACT

Managing scientific archive data on the EPrints Server at STP Sahid Surakarta faces issues with manual backups using WinSCP. This procedure leads to potential human error and the risk of data loss due to local storage, thus threatening important institutional information. The purpose of this research is to develop a scheduled automatic backup system for an Ubuntu 24.04-based Eprints server, utilizing two Virtual Private Servers (VPS) and Nextcloud as cloud storage. The development method used is the Software Development Life Cycle (SDLC) Waterfall model, which includes analysis, design, implementation, and testing. The system implementation involves using bash scripts, cron, mysqldump, tar, and rclone for backup and restore functions. The system's functionality is verified through Black Box Testing. The test results show that the automatic backup system works consistently and as planned. Additionally, the restore function successfully recovered the entire data component and rolled back old data, complete with satisfactory activity log recording. This system provides a more efficient and controlled backup solution, addressing the weaknesses of the old system analyzed using the PIECES framework (performance, information, economy, control, efficiency, and service).

Corresponding Author:

Paryanta,

Universitas Dharma AUB Surakarta

Jl. W.M. Maramis No. 29 Surakarta

Email: paryanta@stmik-aub.ac.id



1. INTRODUCTION

Lembaga pendidikan perlu mengelola perlindungan data penting seperti jurnal, arsip akademik, dan dokumen data ilmiah yang perlu disimpan dan mudah diakses. Kerusakan/kehilangan data dapat sangat memengaruhi operasional institusi maupun reputasinya. Pada suatu penelitian dituliskan bahwa *backup* data adalah proses untuk memastikan data dapat dipulihkan jika terjadi kehilangan, baik karena kesalahan manusia, kerusakan perangkat keras, maupun serangan siber. Sistem *backup* yang baik harus mematuhi prinsip-prinsip dasar seperti otomatisasi, redundansi, keamanan, dan penyimpanan terpisah

dari sistem utama. Keamanan data adalah elemen yang sangat penting, yang mencakup tiga elemen utama dari Triad CIA yaitu kerahasiaan, integritas, dan ketersediaan. Ketiga prinsip ini digunakan agar menjaga data aman, tidak berubah selama pemulihan, dan selalu dapat diakses ketika dibutuhkan [5]. Di sisi lain, masih banyak lembaga menggunakan sistem *backup* yang masih rentan terhadap kesalahan manusia, seperti jadwal pencadangan yang tidak teratur dan sering hanya menggunakan penyimpanan lokal yang terletak di lokasi fisik yang sama dengan server utama. Kesalahan dalam proses pencadangan dan kegagalan perangkat menjadi salah satu faktor utama dalam kehilangan data. Situasi ini meningkatkan risiko kerusakan ganda dalam kasus kegagalan perangkat maupun bencana yang lain.

Hasil wawancara dengan *administrator server* di STP Sahid Surakarta dapat diketahui bahwa proses pencadangan *server EPrints* saat ini masih menggunakan software *WinSCP* melalui protokol *SFTP*. *WinSCP* digunakan sebagai jembatan oleh komputer *broadcasting* berbasis *Windows 11* ke *server virtual EPrints* berbasis *Debian 12*, yang digunakan untuk menyalin atau memindahkan file dari direktori *virtual server Debian 12* yang menjalankan layanan *EPrints* ke penyimpanan lokal di *hard disk* komputer *broadcasting* juga sebaliknya. Proses pencadangan ini memiliki beberapa masalah penting dengan penggunaan *WinSCP*. Salah satunya adalah koneksi *SSH* saat mengakses *server VirtualBox* untuk pencadangan tidak selalu stabil. Proses transfer file dapat tertunda atau gagal karena koneksi yang terputus. Selain itu, proses pencadangan sering terlewat atau tertunda karena tidak ada sistem otomatisasi, ini yang membuat *administrator server* sering lupa. Sebaliknya, penyimpanan file cadangan pada harddisk yang sama dengan *server* utama meningkatkan risiko kehilangan data. Pendapat peneliti lain menuliskan bahwa *backup* yang bergantung pada proses manual sangat rentan terhadap kesalahan manusia dan kemungkinan kerusakan data [4].

Penelitian ini difokuskan untuk membangun sistem pencadangan otomatis menggunakan dua VPS yang disediakan oleh penyedia layanan VPS *Idcloudhost*, yang memungkinkan sistem beroperasi dalam lingkungan online dan mereplikasi skenario yang lebih realistis. Kedua server VPS diinstal dengan *Ubuntu Server 24.04*. Pertama menjalankan *EPrints 3.4.5*, sebuah layanan repositori digital yang digunakan untuk mengelola data akademik, jurnal, dan arsip ilmiah. *Server EPrints* ini menjadi objek utama yang dicadangkan secara otomatis. Kemudian *server* kedua menjalankan *Nextcloud Server* versi 31.0.0, yang berfungsi sebagai *private cloud* untuk menyimpan hasil pencadangan data dari *server EPrints*. Pemilihan *Nextcloud* didasarkan pada kemampuannya menyediakan penyimpanan cloud secara gratis dan sangat sesuai untuk diakses oleh *administrator server* STP Sahid Surakarta. Pemilihan *Ubuntu 24.04* didasarkan pada beberapa alasan penting. *Ubuntu 24.04* juga memiliki paket-paket repository yang telah diperbarui ke versi terbaru, yang menjadikannya lebih baik dalam hal pemeliharaan sistem dan kompatibilitas dengan perangkat lunak terbaru. *Ubuntu 24.04* juga mendukung pembaruan keamanan selama lima tahun, yang sangat cocok untuk penggunaan jangka panjang dan meningkatkan stabilitas sistem. Suatu rumusan permasalahan pada penelitian ini adalah bagaimana pengembangan dan implementasi sistem pencadangan otomatis *Server EPrints* dengan *Ubuntu Server 24.04* dan *Nextcloud Server 31.0.0* pada lingkungan VPS dapat mengoptimalkan manajemen data arsip ilmiah dan mengurangi ketergantungan pada proses manual di STP Sahid Surakarta?

2. THE COMPREHENSIVE THEORETICAL BASIC / PROPOSED METHOD / ALGORITHM

2.1. Penelitian Terdahulu

Penelitian ini mengkaji dan berpedoman pada penelitian sebelumnya sebagai bahan pembandingan dalam melakukan sebuah penelitian. Beberapa penelitian terdahulu diantaranya adalah: a) Hendri & Subur Anugerah (2024) dengan judul “Membangun *Private Cloud* dengan *Nextcloud* dan *Ubuntu* untuk Solusi UMKM”. b) Dealva Arsyad et al (2024) dengan judul “Implementasi File Server Berbasis Cloud Menggunakan *Linux Ubuntu 24.04* dan *Nextcloud*”. c) Yahdi Kusnadi & Ardi Widiyanto (2024), dengan judul “Membangun Backup Data Penting Secara Otomatis Menggunakan *Bash Script* Pada *Red Hat Enterprise Linux*”. d) Wali et al (2022), dengan judul “Pengembangan Sistem *Monitoring Filesystem Server Database* Dengan *Alert* Menggunakan *Bot Telegram*”. e) Muhammad Yasir (2024), dengan judul “Implementasi *Cloud Computing* Berbasis *Ubuntu Server* Menggunakan *Nextcloud* sebagai Media Penyimpanan Data di CV. Rajawali Putri Jaya”.

3. METHOD

Metodologi adalah pendekatan ilmiah yang digunakan secara sistematis dan objektif untuk menyelidiki suatu masalah dengan tujuan mendapatkan pengetahuan yang bermanfaat dan dapat dipertanggungjawabkan [8]. Studi kasus adalah metode penelitian dari jenis penelitian kualitatif yang membantu mengungkap masalah atau kasus tertentu dalam konteks individu, kelompok, atau organisasi. Metode penelitian studi kasus adalah pendekatan untuk penelitian yang menggunakan pokok pertanyaan penelitian “bagaimana” atau “mengapa”. Studi kasus berguna apabila seseorang atau penelitian ingin memahami situasi atau masalah tertentu secara menyeluruh dan jika mereka dapat menemukan kasus yang kaya informasi. Dengan menggunakan studi kasus ini, peneliti dapat mengumpulkan data yang kaya dari berbagai sumber, termasuk observasi, wawancara, dan dokumentasi yang dibuat selama proses pengembangan sistem. Hasil studi kasus dapat dikatakan ilmiah jika dilakukan dengan cara yang sistematis, objektif, dan ilmiah [13].

Penelitian ini menggunakan metode kualitatif dengan jenis penelitian studi kasus untuk menggambarkan secara rinci proses perancangan dan implementasi sistem tanpa menggunakan data statistik. Peneliti memilih jenis studi kasus karena, adanya kegiatan observasi dan pengamatan terhadap satu subjek, yaitu proses pengelolaan sistem *backup* di STP Sahid Surakarta. Data dikumpulkan melalui wawancara, studi pustaka, dan dokumentasi yang dibuat selama proses pengembangan sistem. Karena tahapan sistem dapat dibangun secara sistematis dan berurutan, peneliti memilih metode pengembangan perangkat lunak SDLC dengan model *Waterfall*.

Objek penelitian pada Sekolah Tinggi Pariwisata (STP) Sahid Surakarta, dengan fokus pada aspek-aspek yang relevan dengan pengembangan sistem pencadangan server Eprints.

3.1. Alat

Untuk pembuatan sistem backup otomatis diperlukan kebutuhan untuk menunjang penerapannya, antara lain:

a. Kebutuhan Sumber Daya *Virtual Private Server* (VPS)

Merupakan kebutuhan alokasi sumber daya sistem minimal yang diberikan kepada masing-masing mesin virtual (*EPrints* dan *Nextcloud*) yang berjalan di lingkungan VPS Idcloudhost Tabel 1.

Tabel 1. Kebutuhan Sumber Daya *Virtual Private Server*

No.	Jenis Sumber Daya	Kebutuhan
1	Sistem Operasi Ubuntu 24.04 Processor 2 Core, Hard Disk (HDD) 20 GB, Random Access Memory (RAM) 12 GB	Berfungsi untuk <i>server Eprints</i> dan mengelola penyimpanan data <i>Cloud Storage Nextcloud</i> serta akses secara online

b. Perangkat Lunak

Meliputi aplikasi dan sistem operasi yang digunakan sebagai dasar pengembangan sistem dituliskan pada Tabel 2.

Tabel 2. Kebutuhan Perangkat Lunak

No.	Jenis	Nama Perangkat Lunak
1	Sistem operasi virtual server	Ubuntu 24.04
2	Layanan repositori digital	EPrint 3.4.5
3	Layanan cloud server lokal	Nextcloud versi 31.0.0
4	Akses remote server	Putty
5	Basis data	MySQL dan MariaDB
6	Programming Bash Script	Notepad ++
7	Alat desain UML	Draw.io
8	Browser web	Google Chrome dan Mozilla Firefox
9	Bot	Bot telegram
10	Sistem operasi laptop yang digunakan	Windows 10
11	Protokol jaringan kriptografi untuk komunikasi data	SSH

c. Perangkat Implementasi Sistem

Tabel 3 menuliskan spesifikasi perangkat keras laptop yang digunakan sebagai komputer host untuk menjalankan lingkungan virtual dan proses pengembangan sistem.

Tabel 3. Kebutuhan Perangkat Implementasi

No.	Perangkat Laptop	Kebutuhan
-----	------------------	-----------

1	Processor	Intel® Core™ i5 – 7200U Processor 2.50GHZ
2	Harddisk	1 TB
3	RAM	12 GB

d. Perangkat Jaringan Pendukung

Perangkat ini digunakan untuk menyediakan konektivitas *MiFi* hanya berfungsi sebagai sumber internet untuk perangkat laptop, bukan untuk konektivitas langsung antar-VPS. *MiFi* yang dibutuhkan adalah *MiFi Bolt-BL100* (4G LTE 2.4 Ghz / 5 Ghz).

3.2. Kebutuhan Data

Kebutuhan data yang perlu dicadangkan dalam melakukan backup sistem *Server EPrints* berbasis *Ubuntu 24.04* ke *Cloud Storage Nextcloud* antara lain: *Code Base EPrints* berisi struktur data direktori keseluruhan dari aplikasi *EPrints*, *EPrints* data direktori yang berisi file unggahan, dan *EPrints Database (MySQL)*.

3.3. Kebutuhan Informasi

Kebutuhan informasi yang akan dihasilkan dari sistem *backup* ini meliputi: status proses *backup* (berhasil/gagal), *log* aktivitas *backup*, dan waktu pelaksanaan *backup*, serta status proses *restore* dan *log* aktivitasnya yang dilakukan secara mandiri.

3.4. Teknik Pengumpulan Data

Mengenai jenis data, sumber data, dan pengumpulan data dijelaskan pada Tabel 4.

Tabel 4. Jenis Data, Sumber Data, dan Pengumpulan Data

Jenis Data	Sumber Data	Pengumpulan Data
Data Primer Data primer, juga dikenal sebagai data mentah, memungkinkan interpretasi yang akurat dan relevan serta mengurangi risiko distorsi data dibandingkan dengan data sekunder. Sangat penting untuk penelitian kualitatif dan kuantitatif agar dapat memberikan gambaran yang jelas dan akurat tentang fenomena yang dikaji [14]. Data primer adalah sumber informasi utama yang dikumpulkan langsung oleh peneliti dari informan atau responden tentang variabel penelitian melalui metode seperti observasi, wawancara, atau angket. Proses pengumpulannya melibatkan interaksi langsung peneliti dengan sumber data.	Data Primer Peneliti mendapatkan data ini dari hasil observasi dan wawancara kepada Bapak Wiyadi selaku <i>administrator server</i> di STP Sahid Surakarta.	Wawancara Peneliti memperoleh data dari hasil wawancara dengan Bapak Wiyadi, Administrator Server STP Sahid Surakarta, dan melakukan pengamatan terhadap sistem backup pada <i>server virtual Eprints</i> berbasis Debian 12. Dengan metode ini peneliti mendapatkan informasi detail tentang sistem <i>backup server EPrints</i> yang sedang berjalan.
Data Sekunder Data sekunder merupakan teknik pengumpulan dan analisis informasi yang sudah tersedia sebelumnya. Kelebihan data sekunder adalah mudah diakses, mudah diakses, dan menghemat waktu dan biaya. Namun, data sekunder juga memiliki kekurangan, seperti tidak akurat, tidak selalu sesuai dengan tujuan penelitian, dan dapat hilang. Akibatnya, sebelum menggunakan data, peneliti harus memastikan kualitasnya [14].	Data Sekunder Peneliti mendapatkan data ini dari hasil dokumentasi lewat analisis serta mencari dokumen, publikasi, jurnal, dan buku-buku digital yang relevan dengan penelitian ini.	Studi Pustaka Dalam hal ini peneliti mengumpulkan dan menganalisis informasi yang relevan dari berbagai sumber, termasuk buku, jurnal, dan makalah ilmiah untuk memperoleh pondasi teori tentang masalah yang akan diteliti.

3.5. Tahap-tahap Pengembangan Sistem

Peneliti menggunakan metode pengembangan SDLC dengan model Waterfall. Metode *Waterfall* adalah “metode yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisa, desain, pengkodean, pengujian dan pendukung (support)” [15]. Fungsi dari tiap bagian model *waterfall* adalah sebagai berikut:

a. Analisa Kebutuhan

Pada tahap ini peneliti melakukan identifikasi dan analisis terhadap kebutuhan sistem backup otomatis untuk server layanan penting di institusi pendidikan. Aktivitas meliputi observasi langsung di STP Sahid Surakarta dan wawancara dengan Bapak Wiyadi, selaku administrator server di STP Sahid Surakarta, serta studi pustaka untuk mendukung pemahaman teori.

b. Desain

Pada tahap ini peneliti merancang arsitektur sistem backup otomatis yang meliputi dua server virtual menggunakan layanan VPS, yaitu server Eprints (Ubuntu 24.04 + Eprints 3.4.5) sebagai objek pencadangan dan server Nextcloud (Ubuntu 24.04 + Nextcloud Server 31.0.0) sebagai *Cloud Storage* penyimpanan pribadi untuk media penyimpanan hasil pencadangan. Desain sistem mencakup *UML (Unified Modeling Language)* seperti diagram use case, activity diagram, sequence diagram, dan class diagram serta penjadwalan backup dengan cron, bash script, mysqldump, rclone, dan tar. Selain itu, perancangan sistem juga mencakup integrasi notifikasi otomatis melalui *Telegram Bot*, yang dirancang untuk memberikan informasi status proses *backup* (berhasil atau gagal) secara *real-time* kepada *Administrator Server* melalui pesan teks.

c. Implementasi Sistem

Pada tahap ini peneliti membangun sistem pada dua *Virtual Private Server (VPS)* yang disediakan oleh *Idcloudhost*. Tahapan ini meliputi instalasi *Ubuntu 24.04* (sudah include dari *idcloudhost*), *Nextcloud Server 31.0.0*, *Eprints 3.4.5*, bash script, setup cron, dan integrasi ke *Server Nextcloud* via *rclone* dan bot telegram.

d. Pengujian

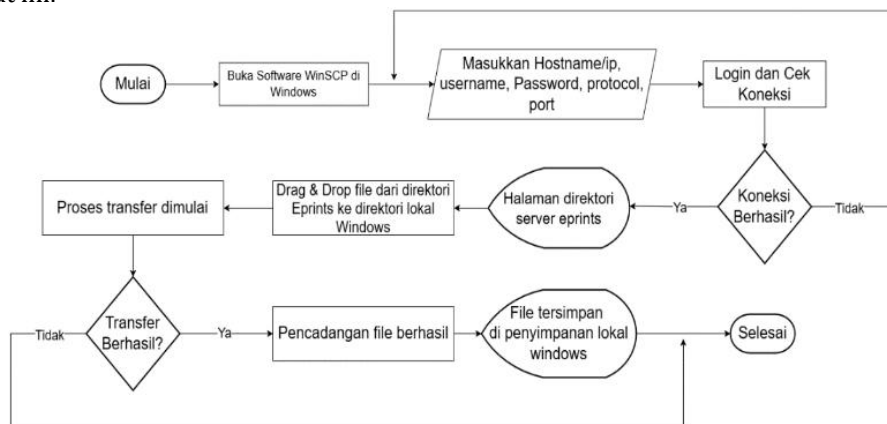
Peneliti melakukan pengujian untuk memastikan proses *backup* otomatis berjalan sesuai yang diharapkan, termasuk pengujian proses pencadangan file, serta kemampuan restore dan integrasi dengan *Nextcloud* sebagai media penyimpanan hasil *backup*. Pengujian dilakukan menggunakan teknik *Black Box Testing* dalam kondisi normal untuk menilai apakah sistem berjalan sesuai yang diharapkan.

e. Pendukung dan Pemeliharaan

Tahapan ini mempertimbangkan kemungkinan pengembangan sistem di masa mendatang, seperti notifikasi, enkripsi lanjutan, atau integrasi ke dashboard monitoring. Sistem juga dirancang agar mudah dipelihara oleh *Administrator Server*.

3.6. Hasil Analisis Sistem Lama

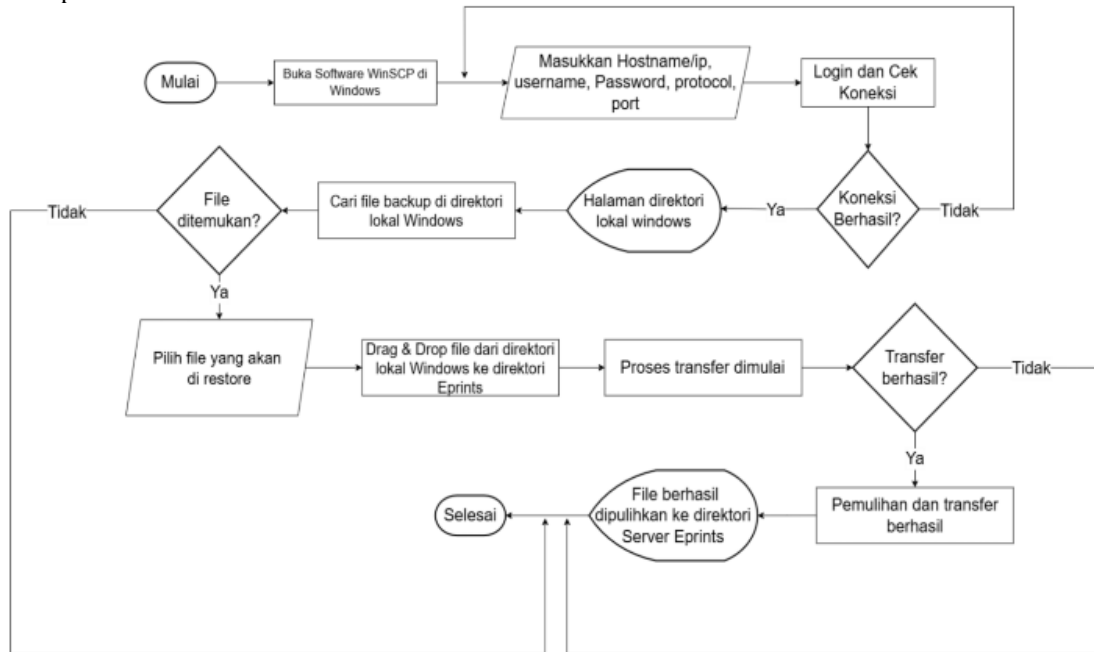
Gambar 1 menunjukkan bentuk *flowchart* sistem ini menggambarkan alur *backup* yang sedang berjalan saat ini.



Gambar 1. *Flowchart* Sistem Backup yang Sedang Berjalan

Pada Gambar 1 menunjukkan bagaimana program *WinSCP* dapat mengirim file dari *Server Eprints* ke penyimpanan lokal di *Windows*. Pertama, buka *WinSCP* dan masukkan informasi koneksi seperti nama *host/ip*, kata sandi, protokol, dan *port*. Sistem memulai proses transfer file dari *Server Eprints* ke penyimpanan lokal setelah login, memeriksa apakah koneksi berhasil. Jika berhasil, file disalin dan pencadangan selesai, tetapi jika gagal, transfer dianggap gagal. Proses akan berhenti jika koneksi gagal pada awalnya.

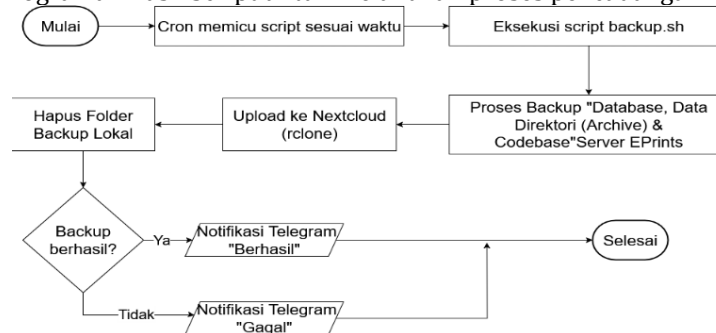
Flowchart sistem pada Gambar 2 menggambarkan alur *restore* yang sedang berjalan untuk *server Eprints*, yang mana pemulihan file dari penyimpanan lokal ke *Server Eprints* menggunakan *WinSCP* untuk proses pemulihan. Proses dimulai dengan memasukkan data *login (hostname, username, password, protocol, dan port)* dan memeriksa koneksi. Setelah koneksi berhasil, sistem memeriksa apakah ada *file backup* di penyimpanan lokal. Jika ditemukan, *file* tersebut dipilih dan dipindahkan ke *server Eprints*.



Gambar 2. *Flowchart* Sistem *Restore* yang Sedang Berjalan

3.7. Hasil Analisis Sistem Baru

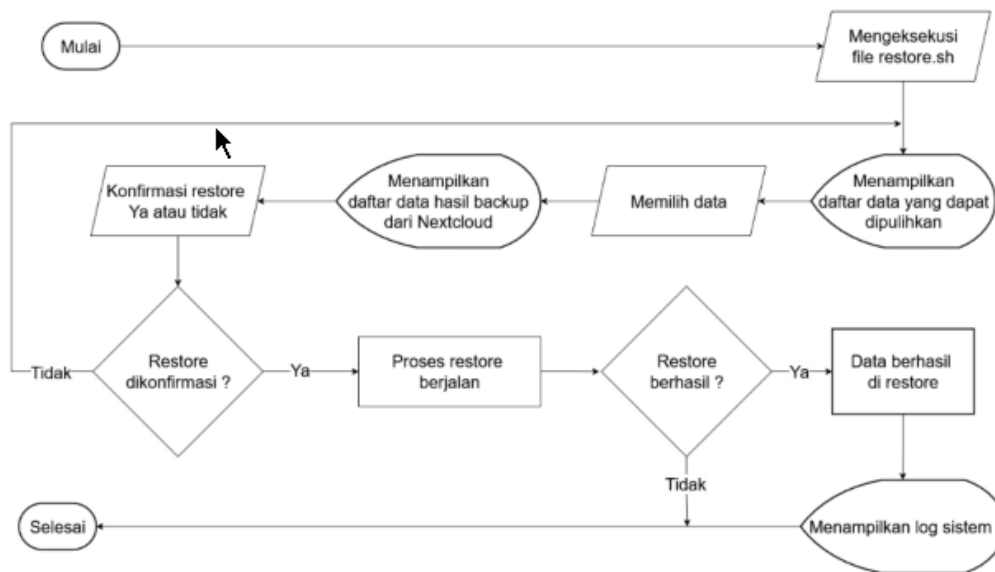
Flowchart program pada Gambar 3 menjelaskan alur pencadangan *server Eprints* yang memanfaatkan pemrograman *Bash Script* untuk melakukan proses pencadangannya.



Gambar 3. *Flowchart* Program *Backup* pada Sistem Baru

Program *backup* ini dijalankan otomatis melalui *cron* sesuai waktu yang telah dijadwalkan. *Script backup.sh* melakukan pencadangan *database, codebase*, dan data unggahan (*archive*) dari *server Eprints*, dan kemudian hasil *backup* dikompresi dan ditransfer ke *Nextcloud* menggunakan *Rclone*. Setelah proses transfer, log baru disimpan dan notifikasi *Telegram* dikirim untuk menandai keberhasilan ataupun kegagalan *backup*.

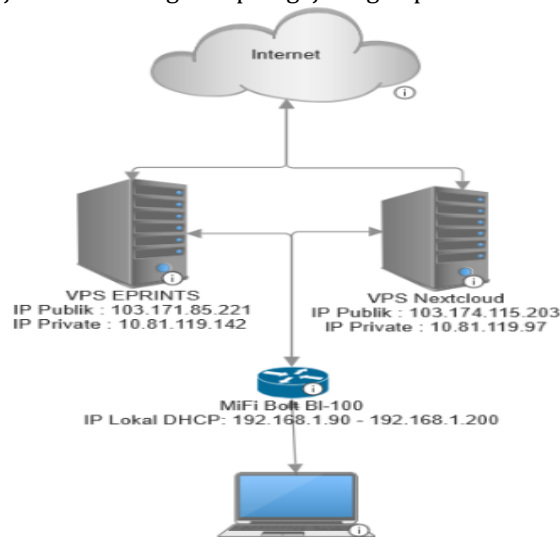
Gambar 4 menunjukkan *flowchart* program *Restore* di *server Eprints* yang terjadwal. Proses *restore* diawali dengan pengguna berhasil login sebagai *root* di *server Eprints*, kemudian pengguna mengeksekusi *script restore.sh*. Setelah itu sistem akan menampilkan daftar data yang dapat dipulihkan, di mana pengguna harus memilih data mana yang ingin dipulihkan. Selanjutnya, sistem akan menampilkan daftar data hasil backup dari *Nextcloud*. Pengguna diminta untuk mengonfirmasi apakah ingin melanjutkan proses pemulihan? Jika melanjutkan proses pemulihan, maka sistem melakukan proses pemulihan data hingga selesai.



Gambar 4. Flowchart Program Restore pada Sistem Baru

3.8. Perancangan Topologi Jaringan

Topologi jaringan dalam penelitian ini menggunakan topologi stars yang sederhana dan efektif diterapkan pada dua *server VPS* yang saling terhubung melalui Internet. VPS *EPrints* dan VPS *Nextcloud*, Keduanya secara default dapat berkomunikasi satu sama lain melalui IP Publik. Peneliti melakukan konfigurasi firewall supaya IP Private masing-masing VPS dapat berkomunikasi untuk proses pencadangan maupun pemulihan data menggunakan utilitas *rclone* untuk transfer datanya. *MiFi Bolt BL-100* berfungsi sebagai perangkat yang menyediakan sumber internet dari jaringan seluler yang menghubungkan laptop ke VPS *Idcloudhost*. Melalui *MiFi*, laptop dapat mengakses VPS melalui Internet publik dengan menggunakan alamat IP publik masing-masing VPS dengan *remote SSH server* VPS-nya. Gambar 5 menunjukkan rancangan topologi jaringan pada sistem baru.



Gambar 5. Topologi Jaringan pada Sistem Baru

4. RESULT DAN ANALISIS

4.1. Cara Kerja Sistem Backup Otomatis

Sistem *backup* otomatis yang dikembangkan untuk *Server EPrints* berfungsi tanpa memerlukan bantuan mandiri dari *Administrator Server* khusus untuk proses *backup*, untuk proses *restore* dilakukan secara mandiri oleh *Adminstrator Server*. Proses ini dimulai secara otomatis dengan memanfaatkan *cron* sebagai pengatur jadwal *script bash backup.sh*, serta *mysqldump* dan *tar* untuk pengarsipan data, dan *rclone* untuk mengirimkan data dengan aman ke *Nextcloud* memanfaatkan semua *tools*. Proses-proses yang dilakukan oleh sistem *backup* otomatis ini meliputi:

- Cron dikonfigurasi untuk memicu *script backup.sh* setiap 20 menit pada pukul 21.01 WIB. Konfigurasi ini, yang ditambahkan ke *crontab* pengguna *root* dengan baris `*/20 * * * * /root/backup.sh`.
- Lalu ketika *script backup.sh* dieksekusi oleh *cron*, langkah pertama adalah mencatat waktu mulai proses ke dalam *file log (backup.log)* dan membuat direktori *backup* lokal sementara. Sistem segera mengirimkan notifikasi awal melalui *Telegram Bot* untuk memberitahukan *Administrator Server* bahwa proses pencadangan sedang berjalan. Apabila terjadi kegagalan dalam pembuatan direktori awal, notifikasi kegagalan segera dikirim dan *script* akan dihentikan.
- Kemudian *script* menggunakan *mysqldump* untuk mencadangkan *database "repo" EPrints*. *File konfigurasi MySQL* sementara (`/tmp/eprints_db.cnf`) dibuat dengan kredensial *database* dan diamankan. *Database* di-*dump* ke *file .sql* di direktori sementara, lalu *file* konfigurasi dihapus, jika ada proses yang gagal, Kegagalan proses ini mengubah status *backup* menjadi "GAGAL", kemudian jika berhasil, status *backup* menjadi "BERHASIL".
- Setelah itu *script* mencadangkan *Code Base EPrints (/opt/eprints3)* dan *Data Direktori (archives/repo)*. Perintah *tar czvf* mengarsip dan mengompresi komponen-komponen ini menjadi *file .tar.gz* di direktori sementara. Kegagalan *tar* mengatur status *backup* menjadi "GAGAL", jika berhasil status *backup* menjadi "BERHASIL".
- Setelah selesai melakukan arsip data *EPrints*, termasuk basis data (*Database*), *Code base*, dan direktori unggahan (*Archive*), berhasil diarsipkan secara lokal dengan *mysqldump* dan *tar*, langkah berikutnya mengirimkan *file backup* tersebut ke *Nextcloud Cloud Storage*. Tugas ini dilaksanakan oleh *rclone*.
- Selanjutnya *script backup.sh* menghapus folder *backup* lokal sementara dari *Server EPrints*. Apabila proses penghapusan ini gagal, *error* juga akan ditangkap dan ditambahkan ke *ERROR_MESSAGE*. *Script* kemudian mengevaluasi status akhir dari seluruh proses *backup* dan mengirimkan notifikasi akhir melalui *Telegram Bot*. Jika BERHASIL, notifikasi sukses dikirimkan dari *Telegram Bot*, merangkum keberhasilan pencadangan semua komponen dan transfer ke *Nextcloud*. Jika GAGAL (ada *error* yang terkumpul di *ERROR_MESSAGE* dari salah satu atau beberapa tahapan), notifikasi kegagalan dikirimkan dari *Telegram Bot*, merinci pesan kesalahan yang terkumpul.

4.2. Cara Kerja Sistem Restore Otomatis

Berbeda dengan proses *backup*, sistem *restore* data dijalankan secara mandiri oleh *Administrator Server* yang memungkinkan fleksibilitas dalam memilih jenis data mana yang akan dipulihkan dan versi *backup* mana yang akan digunakan. Alur kerja umum sistem pemulihan data adalah sebagai berikut:

- Eksekusi *script restore* dilakukan *Administrator Server* didalam *Server Eprints*. *Script* ini kemudian menampilkan menu yang menunjukkan jenis data mana yang dapat dipulihkan, apakah itu *Archive*, *Database*, *Code Base*, atau semua komponen.
- Setelah itu, *Administrator Server* memilih *file backup* dan mengonfirmasi proses pemulihan.

Proses penting untuk memastikan bahwa metadata arsip ilmiah dapat dikembalikan adalah pemulihan *Database EPrints*. *Script* pemulihan dimulai setelah *Administrator Server* memilih opsi pemulihan *Database* dan *file.sql* yang sesuai dari daftar *backup* yang tersimpan pada *Nextcloud*.

4.3. Pengujian Sistem

Tujuan dari teknik pengujian ini adalah untuk memastikan bahwa sistem bekerja sesuai kebutuhan fungsional dan dapat mendeteksi kesalahan fungsi sistem yang terkait. Pengujian *Black Box* digunakan dalam penelitian ini ditujukan untuk memastikan bahwa sistem *backup* dan *restore* berfungsi sesuai yang diharapkan seperti yang direncanakan dalam tahap perancangan.

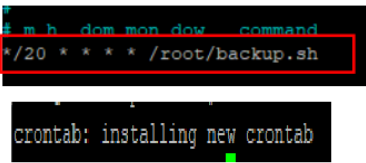
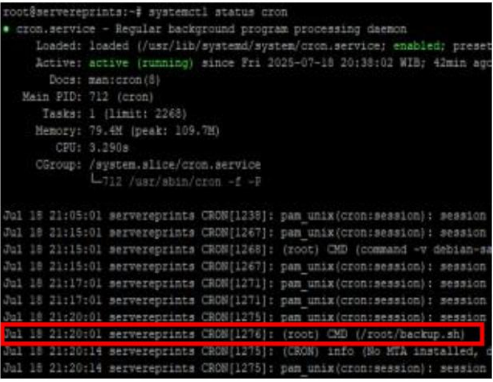
Pengujian dilakukan dengan serangkaian skenario yang mencakup fungsionalitas utama sistem *backup* dan *restore*, yaitu:

- Pengujian Fungsi *Backup* Otomatis, untuk memastikan *cron* menjadwalkan sesuai waktu yang ditentukan dalam *backup* data EPrints.
- Koneksi dan Integritas Nextcloud, untuk memberi verifikasi bahwa data dari proses *backup* ditransfer dan tersimpan dengan benar di Nextcloud sebagai penyimpanan *cloud* pribadi.
- Pengujian Fungsi *Restore* Data, untuk melakukan serangkaian pengujian untuk mengetahui kemampuan sistem dalam mengembalikan data EPrints dari penyimpan *Cloud Storage Nextcloud*.
- Verifikasi Pencatatan *Log* Aktivitas, untuk memeriksa apakah *log* sistem menyimpan semua *log* aktifitas *backup* dan *restore*, termasuk status keberhasilan atau kegagalan, dan waktu pelaksanaan.

4.4. Pengujian Fungsi Backup

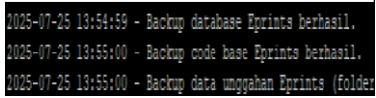
Tabel 5 Menuliskan Data Mengenai Pengujian Cron.

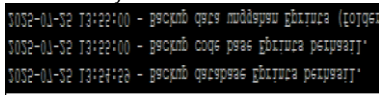
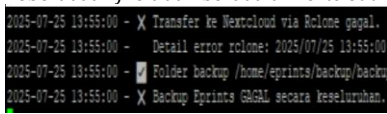
Tabel 5. Pengujian Konfigurasi Cron

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	Pengaturan Cron	Baris : <code>*/20 * * * *</code> <code>*/root/backup.sh</code> ditambahkan ke <i>crontab</i> .	Sistem menampilkan pesan "Crontab: installing new crontab".		Berhasil
2	Memantau proses Cron yang sedang aktif berjalan dan memicu <i>script backup.sh</i> sesuai jadwal	"systemctl status cron" menunjukkan status aktif	<i>Log</i> menunjukkan Eksekusi <i>script</i> setiap 20 menit.		Berhasil

Data mengenai pengujian pencadangan data EPrint (*database*, *code base*, dan data direktori) dituliskan pada Tabel 6.

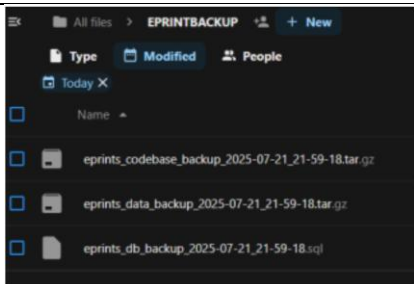
Tabel 6. Pengujian Backup Data

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	Cron mengeksekusi <i>script backup.sh</i> untuk mencadangkan	File <i>.tar.gz</i> untuk <i>Code Base</i> dan <i>Archive</i> terbentuk di direktori sementara <code>/home/eprints/backup</code> kemudian ditransfer ke <i>nextcloud</i>	File tercipta dan berhasil ditransfer ke <i>nextcloud</i>	Tercatat melalui <i>log backup.log</i> dan keberadaan file di Nextcloud 	Berhasil

	Code EPrints Archive	Base dan			
2	Cron mengeksekusi ccript backup.sh untuk mencadangkan database EPrints secara lokal	File .sql untuk database terbentuk di direktori sementara /home/eprints/backup kemudian ditransfer ke nextcloud	File .sql tercipta dan berhasil ditransfer ke nextcloud	Tercatat melalui log backup.log dan keberadaan file di Nextcloud. 	Berhasil
3	Cron mengeksekusi script backup.sh mencadangkan database EPrints secara lokal	File .tar.gz untuk Code Base, Database dan Archive terbentuk di direktori sementara /home/eprints/backup kemudian gagal di transfer ke nextcloud	File .sql dan .tar.gz tercipta serta gagal ditransfer ke nextcloud	Tercatat melalui log backup.log dan keberadaan file tidak berada di Nextcloud. 	Berhasil

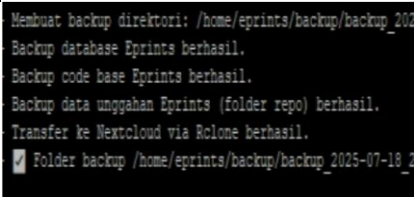
Tabel 7 Menuliskan Hasil Pengujian Transfer Data Ke Nextcloud Dengan Rclone.

Tabel 7. Pengujian Transfer Data

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	File backup berhasil ditransfer dari Server EPrints ke folder "EPRINTBACKUP" di Nextcloud	File backup muncul di Nextcloud dengan format nama dan ukuran sesuai.	File terunggah ke folder "EPRINTBACKUP" dengan timestamp yang benar, data lengkap, backup dengan timestamp yang sesuai		Berhasil

Data mengenai verifikasi pencatatan log aktivitas backup dituliskan pada Tabel 8.

Tabel 8. Pengujian Pencatatan Log

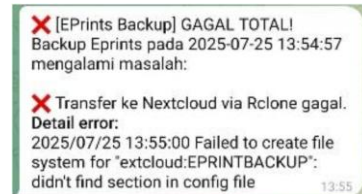
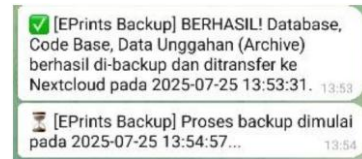
No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	File backup.log mencatat semua tahapan proses backup dengan detail (timestamp, status keberhasilan/kega galan)	Log mencatat semua tahapan backup dengan status keberhasilan	Log mencatat setiap tahapan backup dengan detail yang jelas		Berhasil

Tabel 9 menuliskan hasil verifikasi notifikasi bot telegram proses backup.

Tabel 9. Pengujian Notifikasi Bot Telegram

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	Cron mengeksekusi script backup.sh mencadangkan Code Base EPrints, Archive, dan Database	File data hasil backup berhasil ditransfer ke Nextcloud, serta notifikasi Bot Telegram berhasil	File data hasil backup berhasil ditransfer, notifikasi Bot Telegram berhasil	Tercatat melalui log backup.log, notifikasi bot telegram muncul dan keberadaan file di Nextcloud	Berhasil

- 2 *Cron* mengeksekusi *script backup.sh* mencadangkan *Code Base EPrints*, *Archive*, dan *Database*
- File data hasil backup* gagal ditransfer ke *Nextcloud*, serta notifikasi *Bot Telegram* gagal
- File data hasil backup* gagal ditransfer, notifikasi *Bot Telegram* gagal total
- Tercatat melalui *log backup.log*, notifikasi bot telegram muncul dan keberadaan *file* tidak berada di *Nextcloud*
- Berhasil



Data mengenai pembersihan pada folder backup sementara dituliskan pada Tabel 10.

Tabel 10. Pengujian Pembersihan Isi Folder Backup Sementara

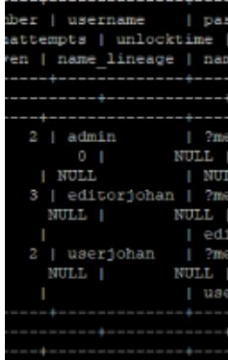
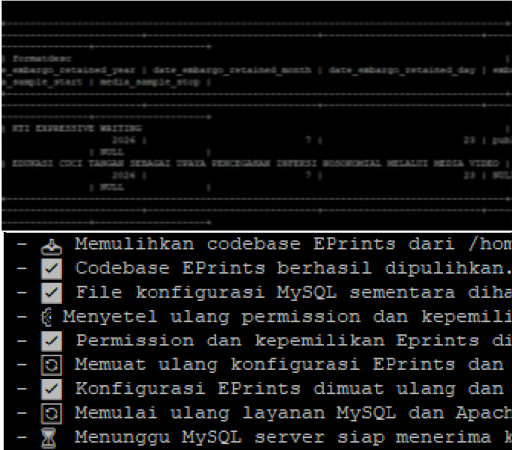
No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	Direktori <i>backup</i> lokal sementara di <i>Server Eprints</i> dihapus setelah transfer ke <i>Nextcloud</i> berhasil	Direktori <i>/home/eprints /backup</i> terhapus setelah transfer ke <i>Nextcloud</i>	Direktori <i>/home/eprints/b</i> ackup terhapus setelah transfer ke <i>Nextcloud</i> .		Berhasil

4.5. Pengujian Fungsi *Restore*

Tabel 11 Menuliskan hasil pengujian tiap komponen data.

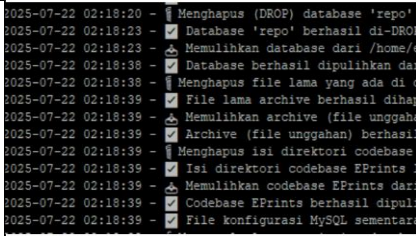
Tabel 11. Pengujian *Restore* Komponen Data

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	<i>Restore</i> Data Direktori (<i>Archive</i>). c. Simulasikan kehilangan data. d. Jalankan <i>restore.sh</i> . e. Pilih "1". "Restore <i>Archive</i> ".	Data di direktori unggahan dipulihkan sesuai kondisi <i>backup</i> terbaru	<i>File</i> PDF yang hilang kembali muncul. <i>Log</i> mencatat pemulihan.		Berhasil

2	<i>Restore Database EPrints.</i> a. <i>Restore dengan data baru.</i> b. <i>Jalankan restore.sh.</i> c. <i>Pilih "2". "Restore Database".</i>	<i>Database</i> dipulihkan sesuai <i>backup</i> terbaru.	Artikel yang dihapus kembali muncul dan user bisa login lagi		Berhasil
3	<i>Restore Code Base EPrints.</i> a. <i>Ubah file konfigurasi.</i> b. <i>Jalankan restore.sh.</i> c. <i>Pilih "3". "Restore Code Base".</i>	<i>Code Base</i> dipulihkan sesuai <i>backup</i> terbaru	Perubahan hilang dan aplikasi kembali berfungsi normal		Berhasil

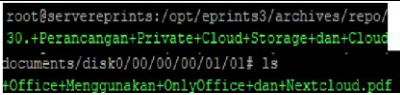
Data mengenai pengujian Restore untuk seluruh komponen data dituliskan pada Tabel 12.

Tabel 12. Pengujian *Restore* Seluruh Komponen Data

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	<i>Restore semua komponen data dari backup EPrints.</i> a. <i>Simulasikan restore semua data EPrints.</i> b. <i>Jalankan restore.sh.</i> c. <i>Pilih "4" "Restore SEMUA Komponen".</i>	Semua komponen (<i>Database, Code Base, dan Archive</i>) dipulihkan sesuai <i>backup timestamp</i> yang dipilih	Semua data dipulihkan dan aplikasi berfungsi normal. <i>Log</i> mencatat pemulihan semua komponen		Berhasil

Tabel 13 Menuliskan hasil pengujian Restore untuk data lama

Tabel 13. Pengujian *Restore* Data Lama

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Bukti / Keterangan	Status
1	<i>Restore Direktori (Archive) ke versi lama.</i> a. <i>Jalankan file restore.sh.</i>	Data dipulihkan ke versi lama, data baru hilang	Data baru hilang digantikan data lama. <i>Log</i> mencatat pemulihan versi lama.		Berhasil

- b. Pilih *backup archive* versi lama.

2. *Restore Database EPrints* ke versi lama.
- a. Data *Database* terbaru.
- b. Jalankan file *restore.sh*.
- c. Pilih *backup Database* versi lama.
- Database* dipulihkan ke kondisi *backup* lama.
- Data baru hilang.
- Data kembali ke kondisi lama.
- Log* mencatat pemulihan *Database*.

```
mysql> SELECT * FROM user;
+-----+-----+-----+-----+-----+-----+
| userid | rev_number | username | setttime | loginattempts | unlocktin |
| name_given | name_lineage | name_hc |
+-----+-----+-----+-----+-----+-----+
| 1 | 2 | admin | NULL | 0 | NULL |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Berhasil

```
30.+Perancangan+Private+Cloud+Storage+dan+Cloud+Off
23 | publication | accepted | NULL | NU
```

Data analisis PIECES mengenai hasil dan peningkatan sistem baru dituliskan pada Tabel 14.

Tabel 14. Hasil dan Peningkatan Sistem Baru

Item Analisis	Sistem Lama	Sistem Baru	Hasil dan Peningkatan Sistem baru
Kinerja (<i>Performance</i>)	Koneksi SSH tidak stabil, transfer file gagal.	Implementasi <i>cron& rclone</i> untuk otomatisasi.	<i>Backup</i> otomatis lancar dan konsisten tiap 20 menit, risiko gagal berkurang.
Informasi (<i>Information</i>)	Risiko kehilangan data tinggi, penyimpanan di <i>disk</i> yang sama dengan <i>server</i> .	Penggunaan <i>Nextcloud</i> untuk penyimpanan <i>backup</i> .	Keamanan data meningkat, <i>file</i> dengan <i>timestamp</i> memastikan integritas dan kemudahan pengelolaan.
Ekonomi (<i>Economics</i>)	Pencadangan manual memakan waktu dan tenaga.	Otomatisasi penuh dengan <i>cron</i> di <i>Linux</i> .	Mengurangi beban kerja, memungkinkan fokus pada tugas strategis dan efisiensi biaya jangka panjang.
Pengendalian (<i>Control</i>)	Pencadangan terjadwal, pengawasan kurang.	Penjadwalan otomatis dengan <i>cron</i> dan <i>log</i> terperinci.	Menjamin konsistensi jadwal, meningkatkan kontrol dan visibilitas terhadap proses <i>backup</i> .
Efisiensi (<i>Efficiency</i>)	Ketergantungan pada manusia, waktu pencadangan tidak pasti.	Otomatisasi <i>backup</i> dengan <i>cron</i> dan <i>bash script</i> .	Efisiensi tinggi, tidak perlu dilakukan mandiri karena <i>backup</i> otomatis, proses konsisten dan minim kesalahan.
Pelayanan (<i>Service</i>)	Pencadangan yang sedang berjalan belum memberikan kemudahan untuk <i>Administrator Server</i> dalam memantau status proses.	Notifikasi status proses <i>backup</i> via <i>Telegram Bot</i> dan akses <i>log</i> sistem interaktif.	<i>Administrator Server</i> mendapatkan informasi <i>realtime</i> mengenai status <i>backup</i> (berhasil/gagal), meningkatkan transparansi dan kemudahan pemantauan, serta memungkinkan respons cepat jika terjadi masalah, sehingga layanan <i>Eprints</i> dapat tetap tersedia.

5. DISCUSSION/CONCLUSION

Dari hasil implementasi dan analisis mengenai sistem baru untuk backup data pada STP Sahid Surakarta dapat disimpulkan bahwa *backup* data dapat dilakukan otomatis dan konsisten sesuai waktu yang ditetapkan dan risiko kegagalan sangat jauh berkurang. Mengurangi beban kerja *Administrator*

Server sehingga memungkinkan fokus pada tugas strategis dan efisiensi biaya jangka panjang. Keamanan data juga meningkat, dengan adanya informasi *timestamp* memastikan integritas dan kemudahan pengelolaan. Konsistensi jadwal terjamin serta dapat meningkatkan kontrol dan visibilitas terhadap proses *backup*. Karena backup otomatis, maka tidak perlu dilakukan. Dan *Administrator Server* mendapatkan informasi *realtime* mengenai status *backup* (berhasil/gagal) dan hal ini meningkatkan transparansi dan kemudahan pemantauan, serta memungkinkan respons cepat jika terjadi masalah.

REFERENCES

- [1] N. M. B. Aditya and J. N. U. Jaya, "Penerapan metode PIECES framework pada tingkat kepuasan sistem informasi layanan aplikasi MyIndihome," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 3, p. 325, 2022, doi: 10.30865/json.v3i3.3964.
- [2] R. Aditya, V. H. Pranatawijaya, and P. B. A. A. Putra, "Rancang bangun aplikasi monitoring kegiatan menggunakan metode prototype," *Journal of Information Technology and Computer Science*, vol. 1, no. 1, pp. 47-57, 2021.
- [3] F. Agostini, L. Bassi, J. Gasparetto, F. Giacomini, R. Miccoli, and E. Vianello, "Enhancing StoRM WebDAV data transfer performance with a new deployment architecture behind NGINX reverse proxy," *Proceedings of Science*, vol. 458, pp. 24-29, 2024.
- [4] M. Amin, "Private cloud storage sebagai media pencadangan data dan berbagi data secara real-time," *Jurnal INSTEK (Informatika Sains dan Teknologi)*, vol. 5, no. 2, pp. 219-228, 2020.
- [5] I. Arnomo, "Simulasi backup dan restore database repository," *Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol. 9, no. 2, pp. 92-99, 2019.
- [6] F. Breitingner, X. Zhang, and D. Quick, "A forensic analysis of Rclone and Rclone's prospects for digital forensic investigations of cloud storage," *Forensic Science International: Digital Investigation*, vol. 43, Art. no. 301443, 2022, doi: 10.1016/j.fsidi.2022.301443.
- [7] R. Dani, "Sistem informasi pendaftaran dan seleksi asisten dosen berbasis web menggunakan framework Laravel (Studi Kasus: Universitas Teknologi Digital Indonesia)," Undergraduate Thesis, Universitas Teknologi Digital Indonesia, 2023.
- [8] T. Dealva Arsyad, I. Agi Berutu, K. Rifail Azis, N. Talitha Jehian, and D. Kiswanto, "Implementasi file server berbasis cloud menggunakan Linux Ubuntu 24.04 dan Nextcloud," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 6, pp. 12595-12600, 2024, doi: 10.36040/jati.v8i6.12020.
- [9] S. Haris, "Pengelolaan repository artikel ilmiah dosen menggunakan aplikasi EPrints," 2020.
- [10] H. Hartini, S. Primaini, N. Nurhayani, and S. N. Darfero, "Pengembangan transaction monitoring pajak berbasis IoT dengan fitur script auto reboot studi kasus: PT. Pinisi Elektra," *Jurnal Ilmiah Informatika Global*, vol. 12, no. 1, pp. 7-14, 2021, doi: 10.36982/jiig.v12i1.1487.
- [11] M. W. Ilhami, W. V. Nurfajriani, A. Mahendra, R. A. Sirodj, and M. W. Afgani, "Penerapan metode studi kasus dalam penelitian kualitatif," *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 9, pp. 462-469, 2024.
- [12] T. Rukhmana, "Edu research for corporate learning and studies," *Jurnal Edu Research: Indonesian Institute for Corporate Learning and Studies (IICLS)*, vol. 2, no. 2, pp. 28-33, 2021.
- [13] A. Suryadi, "Rancang bangun sistem pengelolaan arsip surat berbasis web menggunakan metode Waterfall (Studi Kasus: Kantor Desa Karangrau Banyumas)," *Jurnal Khatulistiwa Informatika*, vol. 7, no. 1, pp. 13-21, 2019, doi: 10.31294/jki.v7i1.36.