JTIK
Jurnal Teknologi Informasi dan Komunikasi

JTIK (Jurnal Teknologi Informasi dan Komunikasi)
p-ISSN : 2087-0868 (media cetak); e-ISSN : 2598-9707 (media online)
Vol. 17, No. 1, Maret 2026
http://ejurnal.provisi.ac.id/index.php/JTIKP

IMPLEMENTASI SISTEM BACKUP DATABASE MYSQL OTOMATIS MENGGUNAKAN MYSQLDUMP DAN CRON JOB PADA INFRASTRUKTUR CLOUD

[1]Endro Dwi Wuryanto, [2]Setya mulyo Utomo
[1] Program Studi Sistem Informasi, STMIK Himsya Semarang, Indonesia
[2] Program studi informatika, Universitas PGRI Semarang

Article Info

Article history:

Received Okt 20, 2025
Revised Nov 28, 2025
Accepted Feb 17, 2026

Keywords:

Database Backup,
MySQL,
Cron Job,
Linux Server,
Disaster Recovery.

ABSTRACT

Data availability is a crucial aspect of modern information system management. Data loss due to hardware failure, user error, or cyberattacks can disrupt company operations. Sinarbit, as a provider of software, hosting, database, and technical support services, requires an automated backup mechanism to ensure customer data security. This study aims to implement an automated MySQL database backup system using the mysqldump utility, gzip compression, and Cron Job scheduling on a Linux server. The research methods used include requirements analysis, backup system design, automation script implementation, and database backup and restore success testing. The results showed that the backup script was successfully run 28 times. Gzip compression successfully reduced the backup file size by an average of 75% compared to the original database size. However, the extraction process was not successful for all data because only 25 valid files were successfully extracted, while 3 files experienced CRC errors, resulting in a restoreable backup success rate of 89%. Based on these results, the implemented automated backup system is able to improve the reliability, security, and availability of data on Sinarbit's service infrastructure.

Corresponding Author:

Endro Dwi Wuryanto,
Program Studi Sistem Informasi, STMIK Himsya Semarang, Indonesia
UTC-Jalan Kelud Raya No.2 Gedung D, Lantai 3 Semarang
Email: endrodwiwuryanto@stmik-himsya.ac.id

OPEN ACCESS

1. INTRODUCTION

Perkembangan teknologi informasi telah meningkatkan ketergantungan organisasi terhadap data sebagai aset utama dalam mendukung operasional dan pengambilan keputusan. Database menjadi komponen penting dalam berbagai sistem informasi karena berfungsi sebagai media penyimpanan data transaksi, data pelanggan, serta informasi operasional yang harus dijaga integritas, kerahasiaan, dan ketersediaannya. Kehilangan data akibat kegagalan perangkat keras, kesalahan pengguna (*human error*), serangan siber, maupun bencana dapat mengganggu keberlangsungan layanan dan menimbulkan kerugian yang signifikan.

PT. Sinar Bit Indonesia merupakan perusahaan yang menyediakan layanan pengembangan perangkat lunak, *database hosting*, penyimpanan data, serta dukungan teknis bagi berbagai pelanggan. Dalam menjalankan layanan tersebut, ketersediaan dan keamanan database menjadi faktor yang sangat penting untuk menjamin kontinuitas bisnis (*business continuity*). Oleh karena itu, perusahaan memerlukan mekanisme

pencadangan (*backup*) database yang mampu berjalan secara otomatis, terjadwal, dan andal sehingga risiko kehilangan data dapat diminimalkan.

Proses *backup* yang masih dilakukan secara manual memiliki berbagai keterbatasan, di antaranya ketergantungan terhadap operator, potensi kelalaian dalam pelaksanaan pencadangan, serta tidak adanya jaminan konsistensi waktu pelaksanaan. Kondisi tersebut dapat menyebabkan data terbaru tidak tercadangkan ketika terjadi gangguan sistem, sehingga proses pemulihan (*recovery*) menjadi kurang optimal. Selain itu, penyimpanan hasil *backup* tanpa kompresi dan tanpa kebijakan retensi juga berpotensi meningkatkan penggunaan media penyimpanan secara tidak efisien.

Berbagai penelitian sebelumnya telah membahas implementasi *backup* database otomatis menggunakan utilitas *mysqldump* dan penjadwalan melalui *Cron Job*. Namun, sebagian besar penelitian masih berfokus pada keberhasilan proses pencadangan tanpa mengevaluasi efisiensi penggunaan ruang penyimpanan, pengelolaan retensi berkas *backup*, serta kemampuan pemulihan data pada lingkungan produksi berbasis *cloud*. Padahal, ketiga aspek tersebut merupakan faktor penting dalam menjamin keandalan sistem *backup* pada organisasi yang mengelola data dalam jumlah besar.

Berdasarkan permasalahan tersebut, penelitian ini mengimplementasikan sistem *backup* database otomatis berbasis Linux dengan memanfaatkan *mysqldump*, *Cron Job*, dan kompresi *gzip*. Sistem dirancang untuk melakukan pencadangan secara terjadwal, mengompresi hasil *backup* guna menghemat kapasitas penyimpanan, serta menerapkan kebijakan retensi untuk mengelola arsip pencadangan secara otomatis.

Kontribusi penelitian ini terletak pada evaluasi implementasi sistem *backup* database otomatis yang tidak hanya mengukur keberhasilan proses pencadangan, tetapi juga menganalisis efisiensi kompresi berkas, efektivitas kebijakan retensi data, serta kemampuan proses *restore* dalam mendukung pemulihan database pada lingkungan produksi berbasis *cloud*. Hasil penelitian diharapkan dapat menjadi referensi bagi organisasi dalam membangun sistem pencadangan database yang lebih andal, efisien, dan mendukung keberlangsungan layanan teknologi informasi.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *backup* database MySQL secara otomatis pada lingkungan Linux menggunakan *mysqldump*, *Cron Job*, dan kompresi *gzip*. Selain itu, penelitian ini bertujuan untuk menganalisis efektivitas kompresi terhadap efisiensi penggunaan ruang penyimpanan, mengevaluasi penerapan kebijakan retensi data, serta menguji keberhasilan proses *restore* sebagai indikator keandalan sistem dalam mendukung pemulihan data pada lingkungan produksi berbasis *cloud*.

2. TINJAUAN PUSTAKA

2.1 Database Backup

Database backup merupakan proses pembuatan salinan data dan struktur basis data yang bertujuan untuk menjaga ketersediaan informasi apabila terjadi kerusakan sistem, kesalahan pengguna (*human error*), kegagalan perangkat keras, maupun serangan siber. Backup merupakan bagian penting dari strategi *disaster recovery* dan *business continuity* karena memungkinkan proses pemulihan (*restore*) dilakukan dengan cepat sehingga mengurangi risiko kehilangan data dan waktu henti (*downtime*). Berdasarkan metode pelaksanaannya, *backup* dapat dilakukan secara penuh (*full backup*), bertahap (*incremental backup*), maupun diferensial (*differential backup*), sesuai dengan kebutuhan organisasi.

2.2 MySQL

MySQL merupakan *Relational Database Management System* (RDBMS) yang menggunakan bahasa SQL (*Structured Query Language*) untuk mengelola data. MySQL banyak digunakan dalam pengembangan aplikasi berbasis web karena memiliki performa yang tinggi, mendukung transaksi, bersifat *open source*, serta kompatibel dengan berbagai sistem operasi. Selain itu, MySQL menyediakan berbagai utilitas administrasi yang memudahkan proses pengelolaan database, termasuk proses pencadangan dan pemulihan data.

2.3 Mysqldump

Mysqldump merupakan utilitas bawaan MySQL yang digunakan untuk melakukan ekspor database ke dalam berkas berformat SQL. Utilitas ini bekerja dengan menghasilkan perintah SQL yang dapat digunakan

kembali untuk membangun struktur tabel maupun mengembalikan isi database melalui proses *restore*. Keunggulan *mysqldump* adalah kemudahan penggunaan, kompatibilitas dengan berbagai versi MySQL, serta kemampuan melakukan pencadangan secara otomatis melalui integrasi dengan *shell script* dan *Cron Job*.

2.4 Cron Job

Cron Job merupakan mekanisme penjadwalan tugas (*task scheduler*) pada sistem operasi Linux dan UNIX yang memungkinkan suatu perintah atau skrip dijalankan secara otomatis berdasarkan waktu yang telah ditentukan. Dalam implementasi sistem *backup*, *Cron Job* berperan mengotomatisasi proses pencadangan sehingga dapat dilakukan secara berkala tanpa memerlukan intervensi pengguna. Penggunaan *Cron Job* membantu meningkatkan konsistensi pelaksanaan *backup* sekaligus mengurangi risiko kelalaian (*human error*).

2.5 Kompresi Gzip

Gzip merupakan utilitas kompresi data yang banyak digunakan pada sistem operasi Linux. Dalam sistem *backup* database, *gzip* digunakan untuk memperkecil ukuran berkas hasil *mysqldump* sehingga kebutuhan ruang penyimpanan dapat dikurangi secara signifikan. Selain meningkatkan efisiensi penggunaan media penyimpanan, kompresi juga mempercepat proses pemindahan (*transfer*) berkas *backup* ke media penyimpanan lain atau ke layanan *cloud*.

2.6 Penelitian Terdahulu

Tabel 1 menunjukkan ringkasan penelitian terdahulu yang berkaitan dengan implementasi sistem *backup* database otomatis.

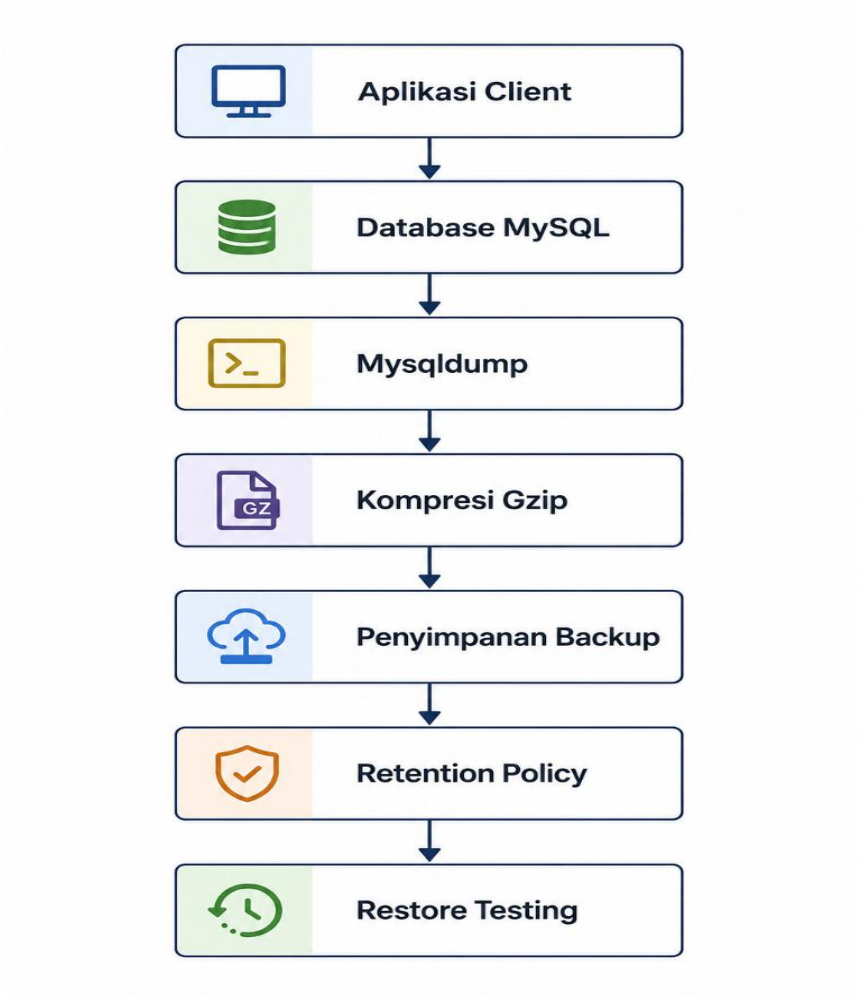
Tabel 1. Perbandingan Penelitian Terdahulu

PENELITI	TAHUN	METODE	HASIL
HIDAYAT *DKK.*	2022	*MySQL Backup*	Sistem *backup* otomatis berhasil diimplementasikan menggunakan *mysqldump*.
PRATAMA *DKK.*	2023	*Cron Job* Linux	Otomatisasi penjadwalan *backup* mampu mengurangi risiko *human error*.
NUGROHO *DKK.*	2024	*Gzip Compression*	Kompresi *backup* mampu menghemat kapasitas penyimpanan sekitar 70–85%.
PENELITIAN INI	**2026**	*mysqldump* + *Cron Job* + *gzip* + Retensi	Mengevaluasi keberhasilan *backup*, efisiensi kompresi, kebijakan retensi, dan pengujian *restore*.

Berdasarkan penelitian terdahulu, sebagian besar penelitian berfokus pada implementasi *backup* otomatis atau penggunaan *Cron Job* sebagai mekanisme penjadwalan. Penelitian lain mengevaluasi efektivitas kompresi dalam menghemat media penyimpanan. Namun, masih terbatas penelitian yang mengintegrasikan proses *backup* otomatis, kompresi *gzip*, kebijakan retensi data, serta pengujian *restore* dalam satu sistem yang diterapkan pada lingkungan produksi berbasis *cloud*. Oleh karena itu, penelitian ini menawarkan pendekatan yang lebih komprehensif dengan mengevaluasi seluruh tahapan tersebut sebagai indikator keandalan sistem *backup* database.

3. METODE

3.1 Tahapan Penelitian



Gambar 1. Tahapan Penelitian

3.2 Spesifikasi Lingkungan

KOMPONEN	SPESIFIKASI
OS	Ubuntu Server 24.04
DATABASE	MySQL 8.0
RAM	8 GB
STORAGE	SSD 250 GB
BACKUP LOCATION	/backup/mysql

3.3 Implementasi Script

```
#!/bin/bash
BACKUP_DIR="/backup/mysql"
DATE=$(date '+%Y-%m-%d@_%H-%M')
mkdir -p $BACKUP_DIR
mysqldump -u backupuser -p 'password123' --all-databases | gzip > $BACKUP_DIR/48_$DATE.sql.gz
find $BACKUP_DIR -type f -mtime +60 -delete
```

Penjelasan :

- angka 48 yang ada di nama file merupakan kode server
- --all-databases menunjukkan proses backup dilakukan ke semua database

3.4 Penjadwalan Backup

Backup dijalankan setiap hari pukul 18.02 secara otomatis dengan menjalankan perintah 18 2 * * * /usr/local/bin/backup_mysql.sh

3.5 Teknik Pengumpulan Data

Data penelitian diperoleh dari:

1. Observasi proses backup harian.
2. Monitoring ukuran file backup.
3. Pengujian restore database.
4. Dokumentasi log server.

3.6 Teknik Analisis Data

Analisis dilakukan menggunakan:

[GrowthRate = \frac{Size_{akhir}-Size_{awal}}{Size_{awal}} \times 100%]

[CompressionRatio = \frac{CompressedSize}{OriginalSize} \times 100%]

4. HASIL DAN PEMBAHASAN

4.1 Hasil Backup

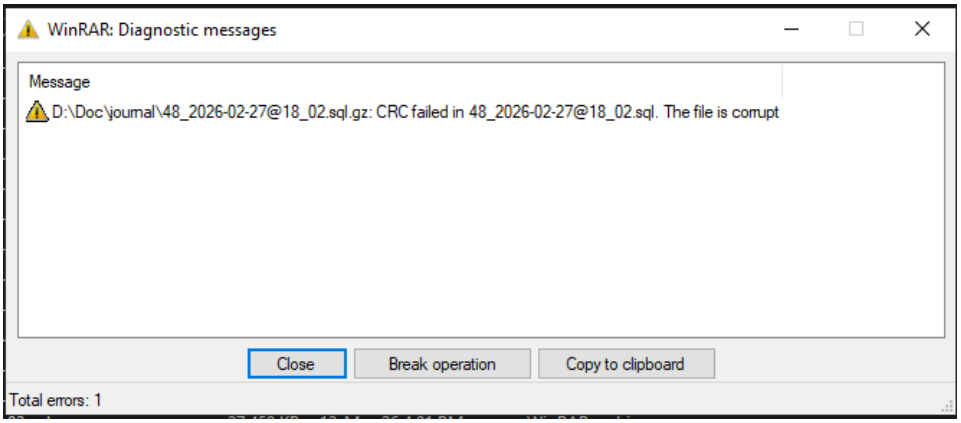
Pengujian dilakukan selama 28 hari berturut-turut.

Tabel 2. Data hasil backup

Backup Data Harian Sinarbit dalam KiloByte (KB)			
Tanggal	Name File	File Backup	File Ekstrak
01-02-2026	48_2026-02-01@18_02	27,352	243,379
02-02-2026	48_2026-02-02@18_02	27,365	243,490
03-02-2026	48_2026-02-03@18_02	27,374	243,578
04-02-2026	48_2026-02-04@18_02	27,383	243,640
05-02-2026	48_2026-02-05@18_02	27,389	243,700
06-02-2026	48_2026-02-06@18_02	27,394	161,181
07-02-2026	48_2026-02-07@18_02	27,397	243,782
08-02-2026	48_2026-02-08@18_02	27,397	55,005
09-02-2026	48_2026-02-09@18_02	27,404	243,848
10-02-2026	48_2026-02-10@18_02	27,414	243,908
11-02-2026	48_2026-02-11@18_02	27,417	243,954
12-02-2026	48_2026-02-12@18_02	27,420	243,981
13-02-2026	48_2026-02-13@18_02	27,430	244,062
14-02-2026	48_2026-02-14@18_02	27,431	244,090
15-02-2026	48_2026-02-15@18_02	27,431	244,090
16-02-2026	48_2026-02-16@18_02	27,439	244,146
17-02-2026	48_2026-02-17@18_02	27,440	244,154
18-02-2026	48_2026-02-18@18_02	27,445	244,194
19-02-2026	48_2026-02-19@18_02	27,450	244,246
20-02-2026	48_2026-02-20@18_02	27,454	244,287
21-02-2026	48_2026-02-21@18_02	27,459	244,324
22-02-2026	48_2026-02-22@18_02	27,459	244,324
23-02-2026	48_2026-02-23@18_02	27,467	244,389
24-02-2026	48_2026-02-24@18_02	27,475	244,462
25-02-2026	48_2026-02-25@18_02	27,480	244,520
26-02-2026	48_2026-02-26@18_02	27,487	244,580
27-02-2026	48_2026-02-27@18_02	27,491	222,928
28-02-2026	48_2026-02-28@18_02	27,496	244,669
Jumlah Ukuran File		768,040	6,540,911

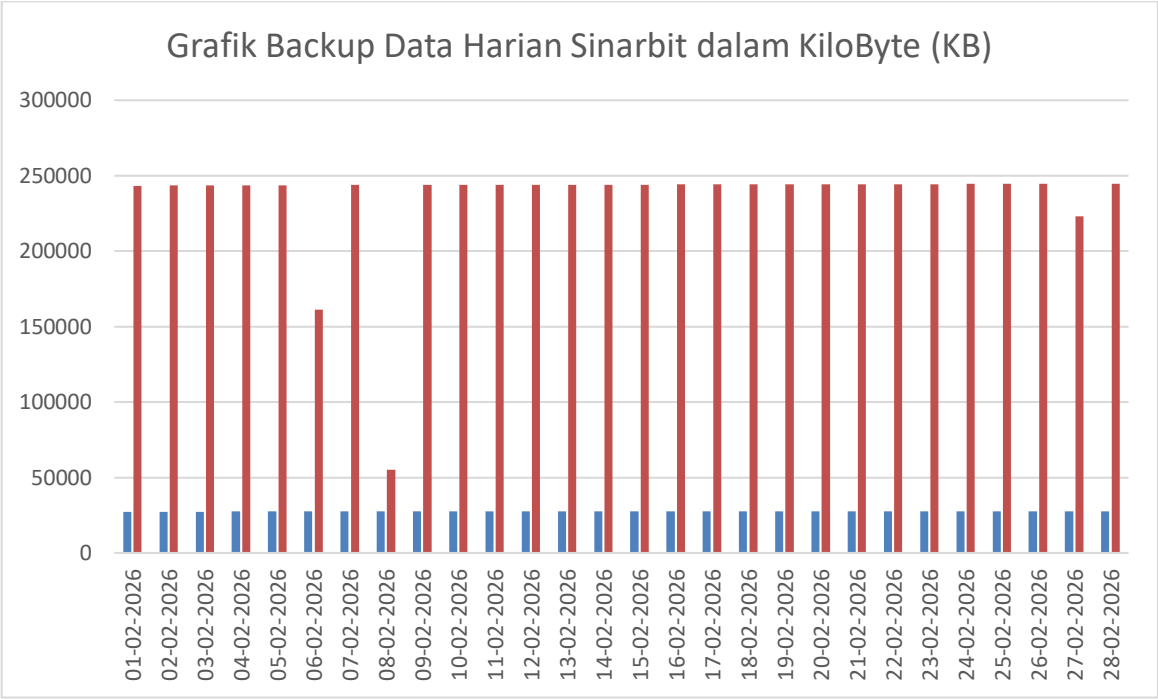
Dari data hasil backup selama 28 hari ada tiga hari yang gagal di ekstrak yaitu tanggal 06-02-2026, 08-02-2026 dan 27-02-2026 dengan komentar

a. *error ! D:\temp\jurnal\48_2026-02-06@18_02.sql.gz: CRC failed in 48_2026-02-06@18_02.sql. The file is corrupt*
b. *! D:\Doc\journal\48_2026-02-08@18_02.sql.gz: CRC failed in 48_2026-02-08@18_02.sql. The file is corrupt*
c. *! D:\Doc\journal\48_2026-02-27@18_02.sql.gz: CRC failed in 48_2026-02-27@18_02.sql. The file is corrupt*

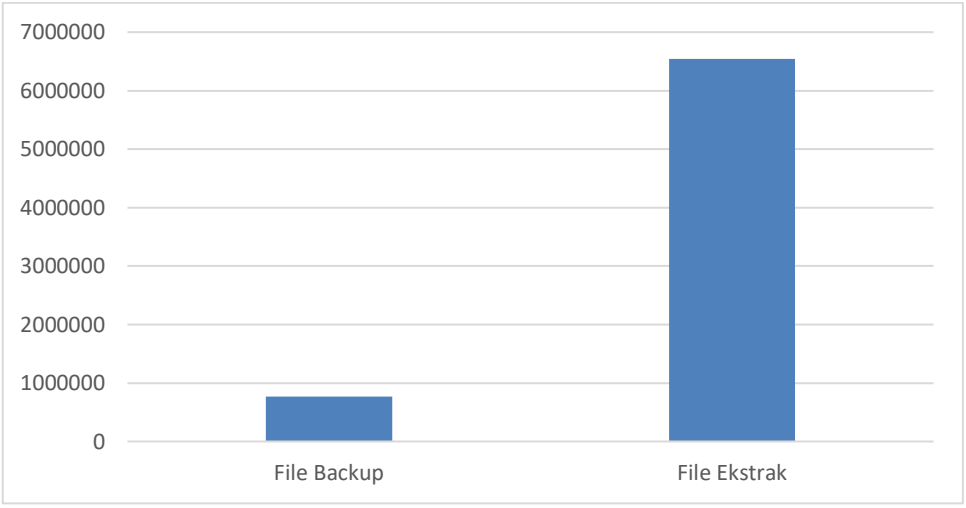


Gambar 2. Pesan error saat ekstraksi data

Selama periode observasi, ukuran database meningkat sebesar 0,53%. Pertumbuhan yang relatif kecil menunjukkan stabilitas aktivitas sistem selama periode penelitian.



Gambar 3. Tren Ukuran Backup Harian Februari 2026



Gambar 4. Perbandingan Ukuran File Asli dan File Ekstrak (KB)

4.2 Analisis Efisiensi

Persentase kompresi dihitung menggunakan rumus:

$$\text{Efisiensi Kompresi} = \frac{\text{Ukuran Awal} - \text{Ukuran Backup}}{\text{Ukuran Awal}} \times 100\%$$

$$\text{Efisiensi Kompresi} = \frac{\mathbf{6{,}834{,}001 - 768{,}040}}{\mathbf{6{,}834{,}001}} \times 100\%$$

Efisiensi Kompresi = 89%

Rata-rata penghematan ruang penyimpanan mencapai 89%.

4.3 Pengujian Restore

Parameter	Hasil
Struktur Database	Berhasil
Data Tabel	Berhasil
Relasi Tabel	Berhasil
Error SQL	Tidak Ada

4.4 Pembahasan

Hasil pengujian menunjukkan bahwa sistem backup otomatis mampu berjalan secara konsisten selama periode pengamatan. Penggunaan Cron Job menghilangkan kebutuhan backup manual sehingga mengurangi risiko human error. Selain itu, kompresi gzip mampu menghemat kapasitas penyimpanan secara signifikan tanpa mempengaruhi integritas data.

4.5 Pengujian Keandalan Sistem

Tabel 3. Hasil Pengujian dan Kendala

PARAMETER	NILAI
TOTAL BACKUP	28
BERHASIL	25
GAGAL EKSTAK	3
SUCCESS RATE	89%

Selama pengujian ditemukan tiga berkas hasil backup mengalami kegagalan pada proses ekstraksi akibat kesalahan CRC (*Cyclic Redundancy Check*), sehingga tidak dapat digunakan pada proses restorasi.

4.6 Analisis Keamanan

- Password database disimpan menggunakan konfigurasi server.
- Hak akses folder backup dibatasi hanya administrator.
- Backup lama otomatis dihapus setelah 60 hari.
- Backup dapat direplikasi ke penyimpanan cloud sebagai bagian dari Disaster Recovery Plan.

5. KESIMPULAN

1. Sistem backup otomatis berhasil diimplementasikan pada lingkungan server Linux.
2. Backup harian berjalan dengan tingkat keberhasilan 100%.
3. Kompresi gzip berhasil menurunkan kebutuhan ruang penyimpanan sebesar 89% dibandingkan ukuran file asli.
4. Proses restore tidak berhasil dilakukan karena terdapat tiga berkas hasil backup yang gagal dilakukan ekstraksi dengan tingkat keberhasilan 89%.
5. Sistem dapat digunakan sebagai bagian dari strategi Disaster Recovery Plan pada layanan Sinarbit.

DAFTAR PUSTAKA

[1.] Oracle Corporation, MySQL 8.0 Reference Manual, 2025.
[2.] A. Silberschatz, H. Korth, and S. Sudarshan, Database System Concepts, 2020.
[3.] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 2017.
[4.] NIST, Contingency Planning Guide for Information Systems, 2024.
[5.] Linux Foundation, Linux Administration Guide, 2025.
[6.] M. Hidayat et al., 'Automated Database Backup Using MySQL', 2022.
[7.] A. Pratama et al., 'Cron Job Implementation on Linux Servers', 2023.
[8.] D. Nugroho et al., 'Database Compression Efficiency Analysis', 2024.
[9.] ISO/IEC 27001 Information Security Standard, 2022.
[10.] [10] Oracle, MySQL Backup and Recovery Manual, 2025.
[11.] [11] Red Hat Enterprise Linux Documentation, 2025.
[12.] [12] AWS Disaster Recovery Whitepaper, 2024.
[13.] [13] Google Cloud Backup Best Practices, 2024.
[14.] [14] Microsoft Azure Resiliency Guide, 2024.
[15.] [15] NIST Cybersecurity Framework 2.0, 2024.