

Information Management of Student Scholarly Works through a Virtual Storage-Based Repository Integrated with SIAKAD: A Development and Evaluation Study at the Universitas Negeri Surabaya Library

¹Mutty Hariyati, Rina Juni Rianty², Aden Aris Hudiana³, Humayyun Nabila Ramadhani⁴

^{1,2,3,4}Universitas Negeri Surabaya

muttyhariyati@unesa.ac.id, rinarianty@unesa.ac.id², 26010845064@mhs.unesa.ac.id³,
humayyunramadhani@unesa.ac.id⁴

DOI: 10.30742/tb.v10i2.5919

Received: 11 September 2026

Revised: 1 October 2026

Accepted: 2 October 2026

ABSTRACT

Research Objective. The growth of scholarly works has increased storage pressure and server downtime at the Library of Universitas Negeri Surabaya. This study develops an end-to-end virtual storage repository integrated with SIAKAD and the student submission workflow. Its novelty lies in combining technical and library-service evaluation to improve storage scalability and operational efficiency. **Research Method.** A Research and Development (R&D) study using the Waterfall model collected data through literature reviews, observations, and focus group discussions (FGDs) with 4 librarians and 3 IT staff. The system was tested using Black-Box scenarios, Locust load testing, and the System Usability Scale (SUS), with 30 purposively selected respondents (12 librarians and 18 students). **Data Analysis.** Quantitative descriptive analysis covered functional success rates, response times, error rates at 100–1,500 concurrent users, operational changes, and SUS scores. **Results.** The system passed 96.2% of Black Box scenarios, with average response times remaining below the 1,000 ms benchmarks at all load levels, although throughput leveled off above 1,000 concurrent users. Compared with the previous manual process, submission time decreased by 75.7% (to 8.5 minutes), document retrieval time by 85.5%, and monthly downtime by 89.2%. The SUS score was 82.1 (grade A). **Conclusions.** Virtual storage is a feasible solution for university library repositories, supporting the management of scholarly works and access to information. Long-term scalability and digital preservation remain to be tested under real peak traffic and in other institutions.

Keywords: *Virtual Storage; Academic Repository Management; Waterfall Model; System Performance; System Usability Scale (SUS).*

A. INTRODUCTION

The rapid advancement of digital technology and the increasing penetration of the internet in higher education have transformed academic administration, prompting a shift from printed archives to digital data repositories (Clemence et al., 2025). However, at the Library of Universitas Negeri Surabaya (UNESA Library), reliance on physical storage and legacy server infrastructure has led to severe operational bottlenecks. Physical archives consume over 40% of the total

library floor area, incurring high maintenance costs and creating space inefficiencies. Furthermore, the volume of archived scholarly works grew exponentially from 410 GB in 2021 to 1,040 GB in 2025 (an average annual increase of 26.2%), causing traditional local servers to experience frequent downtime during peak access periods. To overcome these spatial and technological constraints, transitioning to a scalable, highly available virtual storage system is essential.

Previous studies highlight the imperative role of digital transformation and modern storage architectures in academic libraries. Bolton (2024) and Cox & Mazumdar (2024) emphasize that modernizing library infrastructure is vital for optimizing administrative workflows and managing expanding workloads. Zareef & Jabeen (2025) note that academic repositories require robust digital curation policies and flexible storage solutions to ensure long-term data preservation and continuous access. Furthermore, research by Jayakanth et al. (2022) and Yakubu et al. (2025) demonstrates that integrating cloud-based and virtualized environments significantly enhances system reliability, lowering service downtime while extending operational access to a 24/7 basis.

Macedo et al. (2021) explained that virtual storage technology abstracts physical storage devices into flexible, software-defined pools, enabling seamless elastic scalability up to Petabytes without requiring physical real estate expansion. By pairing object storage (S3-compatible) with modern web architectures, educational institutions can decouple data growth from hardware limits, enforce rigorous checksum-based integrity checks, and streamline retrieval performance (Salvi et al., 2025).

From a library and information science perspective, a repository of student scholarly works is not merely a storage facility but an information service: it encompasses the deposit of works, their metadata description, their preservation, and their retrieval by users (Pasqui, 2024; Zareef & Jabeen, 2025). When storage capacity or server availability is limited, these effects directly affect these services. Deposits are delayed, retrieval is slow or unavailable at peak periods, and physical archives continue to occupy space that could serve library users. Infrastructure decisions, therefore, belong to library information management and not only to information technology.

Existing studies leave three specific gaps. First, studies of digital transformation, digital curation, and technology adoption in academic libraries (Clemence et al., 2025; Harisanty et al., 2025; Zareef & Jabeen, 2025) mainly review policies, skills, and adoption factors, and they rarely report a system that was actually built and measured. Second, technical studies of object storage in education (Salvi et al., 2025) and architectural proposals for institutional repositories (He & Fang, 2025) examine infrastructure, but do not connect it to the workflow by which students deposit their scholarly works through the university academic information system (SIKAD). Third, few studies evaluate such a system both technically, in terms of functional correctness and behavior under concurrent use, and from the library side, in terms of service efficiency and perceived usability. The gap addressed here is therefore the lack of an end-to-end,

empirically evaluated R&D implementation of a virtual storage-based repository integrated with SIAKAD in an Indonesian university library.

Accordingly, this study aims to develop a virtual storage-based repository integrated with SIAKAD for managing students' scholarly works at the UNESA Library and to evaluate it. Specifically, the study aims to: (1) identify the functional and storage requirements of the repository together with librarians and IT staff; (2) design and develop the system on a scalable three-tier virtual storage architecture; (3) evaluate its functional correctness and its performance under increasing concurrent use; and (4) assess its effect on repository service indicators (submission time, retrieval time, downtime, and storage space) and its perceived usability among librarians and students. The novelty of the study lies in combining the development of an SIAKAD-integrated virtual storage repository with an evaluation that links technical performance to library service outcomes.

B. METHODS

This study employed a Research and Development (R&D) approach using the Waterfall software development model (W. Royce, 2021; W. W. Royce, 1970). The Waterfall model was chosen because the repository requirements could be specified in advance, and each phase produced a document or artifact that was reviewed before the next phase began. The study was conducted at UNESA Library, Lidah Wetan Campus, from February to June 2026, in five sequential phases: requirement analysis, system design, development and integration, testing and validation, and implementation.

Three groups took part in the study. The first consisted of 4 librarians and 3 IT staff members from the UNESA Library who served as FGD participants during the requirements analysis. They were purposively selected as staff directly responsible for managing, validating, or serving scholarly works (librarians) or for operating the library servers and the connection to SIAKAD (IT staff). The second group was the development team (programmers, UI/UX designers, and testers). The third group comprised 30 respondents for a limited trial testing and usability evaluation, consisting of 12 librarians and 18 students. These respondents were selected by purposive sampling using three criteria: (a) librarians who manage, validate, or serve scholarly works at UNESA Library, or students who had submitted, or were submitting, a thesis or final project; (b) having used the system to carry out the trial tasks of uploading, searching, and downloading a scholarly work before completing the questionnaire; and (c) willingness to take part.

Requirements were gathered from three sources. First, the existing deposit, storage, and retrieval process was observed using an observation sheet. Second, inventory data on scholarly works for 2021–2025 were compiled from the library records (Table 1). Third, FGDs were held with the participants described above in a single 90-minute session, following an FGD guide. The agreed requirements were documented in a Software Requirement Specification (SRS). The FGD guide covered current workflow problems, required repository functions, metadata and access rules, and integration with SIAKAD. The sessions were recorded and summarized by the research team, and the summaries were confirmed with the participants before the requirements were finalized.

Based on the SRS, a three-tier architecture was designed to separate the user interface, application services, and storage layer. The system was developed by a team of programmers, UI/UX designers, and testers using a Node.js/Express service architecture, S3-compatible object storage for files, a MySQL 8.0 database for metadata, an API Gateway, and an Nginx load balancer. The repository was integrated with SIAKAD through an application programming interface and deployed at repository.perpustakaan.unesa.ac.id over HTTPS.

Black Box test scenarios were derived from the functional requirements in the SRS. For each of the eight modules, the scenarios covered normal flows, invalid inputs, and boundary conditions (for example, non-PDF files or files larger than 50 MB), and the access rights of each user role, giving 79 scenarios in total; a module was considered valid when all its scenarios produced the expected output. Load testing with Locust simulated three repository activities, searching, downloading, and uploading, in a 3:2:1 ratio (Listing 3). Load was increased stepwise to 100, 250, 500, 1,000, and 1,500 concurrent users, each level running for 10 minutes with a spawn rate of 50 users per second. The upper bound of 1,500 users was chosen as a ceiling above the highest expected concurrency during peak deposit periods. Performance was judged against service level agreement (SLA) thresholds of an average response time of $\leq 1,000$ ms, a 95th percentile (P95) response time of $\leq 1,500$ ms, and an error rate of $\leq 5\%$, which were set by the research team together with the library IT staff as service targets. Usability was measured using the 10-item System Usability Scale (SUS) questionnaire (Brooke, 1995), rated on a 5-point scale.

To place the technical results in a service context, five indicators were compared between the manual process used before implementation and the virtual storage system: submission time, retrieval time, monthly downtime, storage capacity, and physical archive space. Times were recorded for 20 scholarly-work submission transactions and 30 document retrieval activities under each condition, with task durations measured from the start of a task to its completion, and downtime was taken from the server incident log for the periods before and after implementation. The comparison is descriptive: it has no control group and no statistical test.

Quantitative descriptive analysis was conducted using the pandas and matplotlib libraries in Python 3.11. Black Box results were expressed as the percentage of successful scenarios. Load testing results were assessed through average and P95 response times, throughput, and error rates at each load level. Before-and-after indicators were compared as percentage changes. SUS scores were calculated with the standard formula and interpreted against the benchmark mean of 68 and the curved grading scale of Sauro & Lewis (2016), in which scores of 80.8–84.0 correspond to grade A, with reference to the adjective ratings of Bangor et al. (2009).

C. RESULT AND DISCUSSION

1. Requirement Analysis Phase

The requirements analysis phase was conducted from February to March 2026 through field observations and Focus Group Discussions (FGDs) with

librarians and IT staff of the UNESA Library (see Methods for participants). Archive volume data of scholarly works over the past five years was collected to map data growth rates and identify bottlenecks within the local server. The mapping, as shown in Table I, reveals that the data volume of scholarly works grew from 410 GB in 2021 to 1,040 GB in 2025, reflecting an average annual growth rate of 26.2%. Meanwhile, the local server frequently experienced downtime during peak traffic periods.

Table I. Requirement Analysis Results: Volume Growth of UNESA's Scholarly Works (2021–2025)

Year	Number of Scholarly Works (Titles)	Archived Data Volume (GB)	Volume Growth (%)
2021	3,150	410	–
2022	3,480	512	24.9%
2023	3,820	645	26.0%
2024	4,210	812	25.9%
2025	4,690	1,040	28.1%
Average Growth/Year	+385 titles	+157.5 GB	26.2%

Note: Data were compiled from the inventory system of the UNESA Library, encompassing undergraduate theses, master's theses, doctoral dissertations, and research reports by faculty and administrative staff.

This growth trend is visualized in Figure I, which served as the baseline for determining an initial virtual storage capacity of 50 TB, with projected scalability reaching up to the petabyte scale.

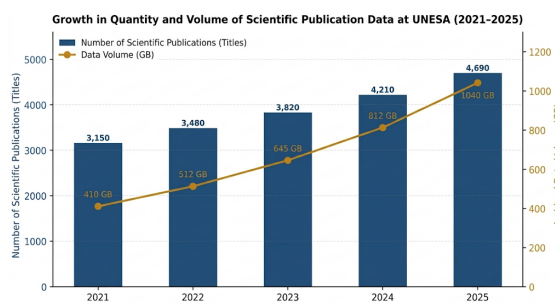


Figure I. Growth in the number and data volume of UNESA's scholarly works (2021–2025)

Sustained growth of this kind is to be expected as theses, dissertations, and research reports are deposited in digital form. For the library, the problem is not only where to store files but also whether the repository can continue accepting deposits and serving users during the busiest periods, when downtime can delay thesis deposits and interrupt access to scholarly works. Treating data growth as a service requirement rather than as a purely technical problem is consistent with the view that institutional data now underpin library development (Amusan & Tella, 2025; He & Fang, 2025). The initial capacity of 50 TB was therefore set as a planning baseline for future deposits, not as evidence that the repository can scale indefinitely (Salvi et al., 2025).

2. System Architecture Design

Based on the requirements analysis, a three-tier information system architecture was designed, separating the client, application, and storage layers. The storage layer utilizes S3-compatible object storage, allowing capacity to scale elastically without interrupting ongoing services. The system architecture design is illustrated in Figure 2.

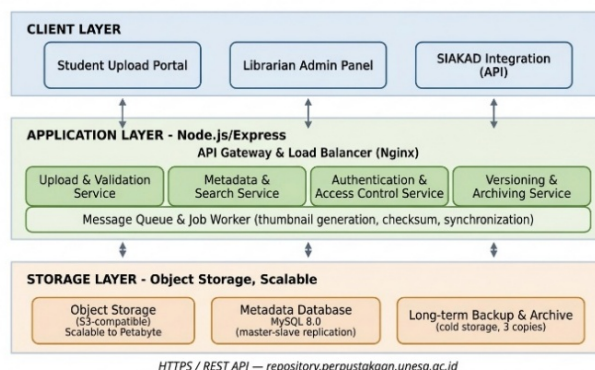


Figure 2. Information system architecture based on virtual storage at UNESA Library

The system was developed using a Node.js/Express service architecture equipped with an API Gateway and an Nginx load balancer, a MySQL 8.0 database for metadata management, and a message queue for processing asynchronous tasks such as checksum generation and synchronization. A core component of this architecture is the upload service, which validates file types and sizes before storing files in object storage with their corresponding metadata. A snippet of the source code for this service is presented below.

Listing 1. Upload service for scholarly works to object storage (Node.js/Express)

```
const { S3Client, PutObjectCommand } = require('@aws-sdk/client-s3');
const crypto = require('crypto');
const db = require('./db');

const s3 = new S3Client({ endpoint: process.env.STORAGE_ENDPOINT,
  region: 'auto' });
const BUCKET = 'karya-ilmiah-unesa';
const MAKS_UKURAN = 50 * 1024 * 1024; // 50 MB per berkas
const TIPE_DIIZINKAN = ['application/pdf'];

async function unggahKaryallmiah(req, res) {
  const berkas = req.file;
  const { judul, fakultas, penulisId } = req.body;

  if (!berkas || berkas.size > MAKS_UKURAN) {
    return res.status(422).json({ pesan: 'Berkas kosong atau melebihi 50 MB' });
  }
  if (!TIPE_DIIZINKAN.includes(berkas.mimetype)) {
    return res.status(422).json({ pesan: 'Hanya berkas PDF yang diizinkan' });
  }

  const checksum = crypto.createHash('sha256')
    .update(berkas.buffer).digest('hex');
  const kunci = `karya-ilmiah/${fakultas}/${Date.now()}_${berkas.originalname}`;

  await s3.send(new PutObjectCommand({
    Bucket: BUCKET, Key: kunci, Body: berkas.buffer,
    ContentType: berkas.mimetype, Metadata: { checksum },
  }));

  const [hasil] = await db.query(
```

```
'INSERT INTO karya_ilmiah
(judul, fakultas, penulis_id, storage_key, checksum, status)
VALUES (?, ?, ?, ?, ?, 'proses_validasi')',
[judul, fakultas, penulisId, kunci, checksum]
);

return res.status(201).json({ id: hasil.insertId, status: 'proses_validasi' });
}

module.exports = { unggahKaryaIlmiah };
```

The key design decision is to separate the repository into three layers: the interface users see, the application that handles deposits, searches, and access rules, and the storage layer where files are kept. Because the storage layer uses object storage that can be scaled without interrupting the service, the library can add capacity as its collection grows rather than replacing servers (Macedo et al., 2021; Salvi et al., 2025). Each file is also stored with a checksum, a digital fingerprint that enables later detection of corruption, thereby supporting the integrity requirement of digital preservation (Pasqui, 2024). By building these storage decisions into the deposit workflow, the design addresses the second limitation identified in the Introduction: the missing link between infrastructure and the way students submit their work. The checksum secures integrity only; other preservation functions, such as format migration and preservation policy, were not part of this system.

3. Interface Development and System Integration

The client-side upload interface was developed featuring a drag-and-drop mechanism alongside pre-submission validation for file types and sizes. To ensure clear user feedback, the interface is equipped with a real-time upload progress indicator. The implementation of this mechanism is presented in Listing 2.

Listing 2. File upload validation and progress indicator (JavaScript)

```
const dropzone = document.getElementById('dropzone');
const inputFile = document.getElementById('inputFile');
const progressBar = document.getElementById('progressBar');
const MAKS_UKURAN = 50 * 1024 * 1024;

['dragenter', 'dragover'].forEach(evt => {
  dropzone.addEventListener(evt, e => {
    e.preventDefault();
    dropzone.classList.add('dropzone--aktif');
  });
});

['dragleave', 'drop'].forEach(evt => {
  dropzone.addEventListener(evt, e => {
    e.preventDefault();
    dropzone.classList.remove('dropzone--aktif');
  });
});

dropzone.addEventListener('drop', e => unggah(e.dataTransfer.files[0]));
inputFile.addEventListener('change', e => unggah(e.target.files[0]));

function unggah(berkas) {
  if (!berkas || berkas.type !== 'application/pdf') {
    return tampilkanError('Hanya berkas PDF yang dapat diunggah.');
```

The implementation result of the system interface on the scholarly work management dashboard page is illustrated in Figure 3. This page presents a

summary of the total number of scholarly works, data volume, used capacity, validation status, and recent upload activity logs.

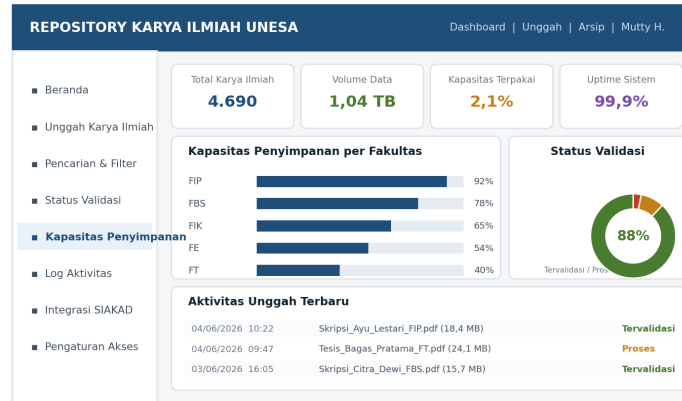


Figure 3. User interface dashboard for scholarly work management

For a library, the interface matters as much as the storage behind it, because a repository works only if students and staff can deposit their works without special assistance. Checking file type and size before upload and showing progress reduces failed submissions and uncertainty for depositors, and the dashboard gives library staff an overview of collection size, used capacity, and validation status. This supports the view that the adoption of library technology depends on practical usability and user needs rather than on technical sophistication (Harisanty et al., 2025; Zareef & Jabeen, 2025). Usability itself was measured only after implementation and with a limited group of respondents, as reported below.

a. Black Box Testing and Load Testing

Functional testing was conducted using the Black Box Testing method across eight core system modules. Out of 79 executed test scenarios, 76 scenarios performed as expected, while 3 scenarios required revision. Detailed test results are summarized in Table 2.

Table 2. System Functional Testing Results via Black Box Testing Method

No	Tested Module	Scenarios	Passed	Failed	Success Rate	Status
1	File Upload & Validation	14	14	0	100.0%	Valid
2	Metadata & Search	12	12	0	100.0%	Valid
3	Authentication & Access Control	10	9	1	90.0%	Revised
4	Scholarly Work Download	9	9	0	100.0%	Valid
5	Versioning & Archiving	8	7	1	87.5%	Revised
6	SIAKAD Integration (API)	10	9	1	90.0%	Revised
7	Backup & Restore	7	7	0	100.0%	Valid
No	Tested Module	Scenarios	Passed	Failed	Success Rate	Status
8	Notification & Activity Log	9	9	0	100.0%	Valid

Total	79	76	3	96.2%	Excellent
--------------	-----------	-----------	----------	--------------	------------------

Note: The three scenarios requiring revision comprised an active session persisting after token expiration, older document versions not automatically archiving to cold storage, and delayed SIAKAD metadata synchronization under slow network connections. All findings were resolved in the subsequent revision iteration.

Furthermore, Load Testing was performed using a concurrent-user simulation script to measure system resilience to traffic spikes. Testing was carried out incrementally, scaling from 100 to 1,500 concurrent users to simulate search, download, and upload activities. The load testing script is presented in Listing 3.

Listing 3. Concurrent user Load Testing simulation script (Python/Locust)

```
from locust import HttpUser, task, between
import random

class PenggunaRepository(HttpUser):
    wait_time = between(1, 3)
    kata_kunci = ["literasi digital", "big data", "manajemen informasi",
                 "virtual storage", "SDM perpustakaan"]

    @task(3)
    def cari_karya_ilmiah(self):
        q = random.choice(self.kata_kunci)
        self.client.get(f"/api/karya-ilmiah/search?q={q}", name="/search")

    @task(2)
    def unduh_karya_ilmiah(self):
        id_dokumen = random.randint(1, 4690)
        self.client.get(f"/api/karya-ilmiah/{id_dokumen}/download",
                       name="/download")

    @task(1)
    def unggah_karya_ilmiah(self):
        with open('sampel_skripsi.pdf', 'rb') as berkas:
            self.client.post("/api/karya-ilmiah/upload",
                            files={'file': berkas},
                            data={'judul': "Uji Beban Otomatis", "fakultas": "FIP"},
                            name="/upload")

# Dialankan bertahap: locust -f locustfile.py --users 1500 \
# --spawn-rate 50 --run-time 10m --host https://repository.perpustakaan.unesa.ac.id
```

The load testing results in Table 3 show that the average response time remained within the 1,000 ms SLA threshold at every load level tested (210–780 ms), and the P95 response time remained within the 1,500 ms threshold (340–1,420 ms). The error rate remained below the 5% limit even at the peak load of 1,500 users (2.40%). At the same time, the results show the first signs of saturation above 1,000 users, as discussed below. Figure 4 visualizes these results.

Table 3. Load Testing Results Across Various Concurrent User Levels

Concurrent Users	Average Response (ms)	P95 Response (ms)	Throughput (req/s)	Error Rate (%)
100	210	340	96.2	0.00
250	265	420	118.4	0.00
500	340	610	142.7	0.20
1,000	495	890	168.3	0.80
1,500	780	1,420	171.5	2.40
SLA Threshold	≤ 1,000 ms	≤ 1,500 ms	–	≤ 5%

Note: Testing was performed using Locust for a duration of 10 minutes per load level, with a user spawn rate of 50 users/second.

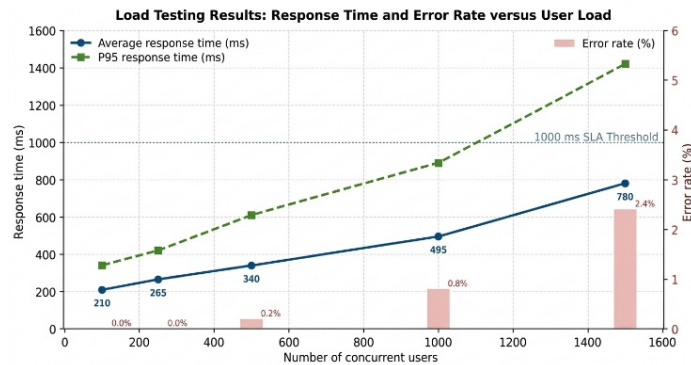


Figure 4. Load Testing results: response time and error rate against user load

The three failed functional scenarios all concerned access control, archiving, and integration, not the core deposit function. These are precisely the areas where a university repository must be reliable, because access rights protect students' works and the link with SIAKAD connects deposits to the academic record (Clemence et al., 2025). Finding and correcting them before implementation is the purpose of iterative functional testing.

The load results need a more cautious reading. Between 1,000 and 1,500 concurrent users, the number of users rose by 50% while throughput rose by only about 2% (from 168.3 to 171.5 requests per second); the average response time rose by about 58%, the P95 response time by about 60%, and the error rate tripled (from 0.80% to 2.40%). Because throughput has nearly stopped increasing while delays and errors grow, the system appears to be approaching capacity in this range, even though all SLA thresholds were still met. Above 1,500 users, searches, and downloads would likely become noticeably slower, and the test does not show what happens at that point. The test also used simulated traffic with a fixed mix of activities, so it cannot represent real peak periods such as thesis submission deadlines. The results, therefore, support the claim that the system can serve the concurrent use expected at this library, not that it scales without limits. In practice, the library should monitor actual peak concurrency and plan additional capacity before demand approaches this range (Amusan & Tella, 2025; Salvi et al., 2025).

b. System Performance Comparison and Usability Measurement

A performance comparison, shown in Table 4, was conducted between the manual process prior to implementation and the process using the virtual storage system. Times were recorded for 20 scholarly work submission transactions and 30 document retrieval activities under each condition. The comparison is descriptive, without a control group or statistical testing: it shows the size of the difference, not its cause.

Table 4. Manual System and Virtual Storage System Performance Comparison

Performance Indicator	Manual System (Before)	Virtual Storage (After)	Change
Scholarly work submission time	35.0 minutes	8.5 minutes	−75.7%
Document retrieval time	12.4 minutes	1.8 minutes	−85.5%
Monthly service downtime	6.5 hours	0.7 hours	−89.2%
Available storage capacity	1.5 TB (physical + local)	50 TB (scalable up to PB)	Increased
Physical archive space utilization	40% of total area	8% of total area	−32 percentage points

The most significant improvement occurred in monthly service downtime, which dropped by 89.2%, and in physical archive space utilization, which decreased from 40% to 8% of the total area, allowing the reclaimed space to be repurposed for collaborative areas.

System usability was measured using the System Usability Scale (SUS) questionnaire administered to 30 respondents, comprising 12 librarians and 18 students. To obtain each respondent's score, odd-item responses were first transformed by subtracting 1, whereas even-item responses were transformed by calculating 5 minus the response score. The transformed scores were then summed and multiplied by 2.5. Table 5 reports the item means; because the transformation is linear, the final score derived from these means equals the mean of the individual respondent scores.

Table 5. System Usability Measurement Results Using System Usability Scale (n = 30)

No	System Usability Scale Statement	Mean (1–5)	Score Contribution
1	I think that I would like to use this system frequently (+)	4.37	3.37
2	I found the system unnecessarily complex (−)	1.73	3.27
3	I thought the system was easy to use (+)	4.43	3.43
4	I think that I would need the support of a technical person to be able to use this system (−)	1.67	3.33
5	I found the various functions in this system were well integrated (+)	4.20	3.20
6	I thought there was too much inconsistency in this system (−)	1.60	3.40
7	I would imagine that most people would learn to use this system very quickly (+)	4.30	3.30
8	I found the system very cumbersome to use (−)	1.83	3.17
9	I felt very confident using/managing scholarly works on the system (+)	4.27	3.27
10	I needed to learn a lot of things before I could get going with this system (−)	1.90	3.10
Total Score Contribution		–	32.84
Final SUS Score (32.84 × 2.5) — Grade A (benchmark mean = 68)		–	82.1

Note: The (+) symbol denotes a positive statement with a score contribution = mean – 1, while the (–) symbol denotes a negative statement with a score contribution = 5 – mean. A SUS score of 82.1 corresponds to grade A on the curved grading scale of Sauro and Lewis (2016) and is above the benchmark mean of 68.

The before-and-after comparison indicates that the largest gains for library service came from faster deposit and retrieval and from fewer service interruptions. For students, a shorter submission process and more continuous access reduce the burden of depositing a thesis; for the library, reduced use of floor space shows how repository infrastructure can also affect the physical library, aligning with the changing role of libraries in the digital educational environment (Krasynskyi, 2025). These figures should be read with care. They come from a single institution, were compared before and after implementation without a control group, and may also reflect other changes made during the same period.

The SUS score of 82.1 (Grade A) indicates high perceived usability (Sauro & Lewis, 2016). This result is consistent with the item-level responses, with respondents reporting high confidence in using the system (Item 9, mean = 4.27) and low agreement that the system was unnecessarily complex (Item 2, mean = 1.73). This is relevant because repositories depend on users who are not technical specialists, and adoption in academic libraries rests on perceived usefulness and ease of use more than on technical strength alone (Harisanty et al., 2025; Zareef & Jabeen, 2025). Because the 30 respondents were purposively selected and included both librarians and students, the score reflects perceived usability only among these users, and the scores for the two groups are not reported separately.

c. Implications and Limitations

The study addressed the gap identified in the Introduction. Unlike reviews of adoption and digital curation (Harisanty et al., 2025; Zareef & Jabeen, 2025), it reports a working system with measured results; unlike technical studies of object storage (He & Fang, 2025; Salvi et al., 2025), it links the infrastructure to the deposit workflow through SIAKAD and to library service indicators. For digital repository management, the results show that a library can decouple repository growth from its physical, local infrastructure while keeping deposits connected to the academic information system. For access to information, shorter retrieval times and less downtime make scholarly works more reliably available to users. For library services, the dashboard and validation status support quality control of deposited works, and the space freed up by physical archives can be used for other services. For digital preservation, storing a checksum with each file, together with the backup and restore and the versioning and archiving functions tested in Table 2, provides basic conditions for preservation (Pasqui, 2024; Zareef & Jabeen, 2025); preservation policy, format migration, and long-term monitoring were not part of this system or its evaluation.

Several limitations should be noted. The study was conducted at one library over about five months (February–June 2026), so the results may not generalize to other institutions or to long-term use. Load was simulated up to 1,500 concurrent users with a fixed mix of activities, and the system approached saturation in this range. The before-and-after comparison was descriptive and

lacked a control group, so the improvements cannot be attributed solely to the system. Usability was measured with 30 purposively selected respondents, and the results for librarians and students are not reported separately. Finally, the long-term effectiveness of the system, including its contribution to preservation, has not been evaluated.

E. CONCLUSION

This study developed and evaluated a virtual storage-based repository, integrated with SIAKAD, for managing student scholarly works at UNESA Library. The system passed 76 of 79 functional scenarios (96.2%), maintained average response times within the 1,000 ms SLA up to 1,500 simulated concurrent users, although throughput leveled off above 1,000 users, and obtained a SUS score of 82.1 from 30 librarians and students. Compared with the previous manual process, submission time fell by 75.7%, retrieval time by 85.5%, and monthly downtime by 89.2%, and the share of floor area used for physical archives fell from 40% to 8%. These findings show that the system is feasible and beneficial for the deposit, retrieval, and continuity of service of a university repository. Because the evaluation was limited to one library, a short period, simulated load, and an uncontrolled before-and-after comparison, the study does not establish long-term scalability, digital preservation, or effectiveness in other institutions.

Future research should test the system under real peak-period traffic and with more than 1,000 concurrent users, examine preservation policies and functions alongside the technical infrastructure, and compare user groups and institutions. Extensions such as open metadata harvesting and AI-assisted retrieval (Asim et al., 2023; Douze et al., 2024) become worthwhile only after these limitations have been addressed.

REFERENCES

- Amusan, B. B., & Tella, A. (2025). Why Are Data so Important to Library Development Today? Implications for Practice. *Evidence Based Library and Information Practice*, 20(3), 65–74. <https://doi.org/10.18438/ebliip30802>
- Asim, M., Arif, M., Rafiq, M., & Ahmad, R. (2023). Investigating applications of Artificial Intelligence in university libraries of Pakistan: An empirical study. *The Journal of Academic Librarianship*, 49(6), 102803. <https://doi.org/10.1016/j.acalib.2023.102803>
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Stud.*, 4, 114–123.
- Bolton, S. (2024). Determining Equitable Liaison Librarian Workloads: An Investigation into the Conundrum. *College & Research Libraries*, 85(5). <https://doi.org/10.5860/crl.85.5.726>
- Brooke, J. (1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Clemence, O., Luambano, I., & Mwantimwa, K. (2025). Adoption and application of electronic records systems in higher learning institutions. *Information Development*, 41(2), 333–352. <https://doi.org/10.1177/026666669231158336>
- Cox, A. M., & Mazumdar, S. (2024). Defining artificial intelligence for librarians.

- Journal of Librarianship and Information Science*, 56(2), 330–340.
<https://doi.org/10.1177/09610006221142029>
- Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. (2024). The Faiss library. *Arxiv*.
<https://doi.org/https://doi.org/10.48550/arXiv.2401.08281>
- Harisanty, D., Anna, N. E. V., Putri, T. E., Firdaus, A. A., & Noor Azizi, N. A. (2025). Is adopting artificial intelligence in libraries urgency or a buzzword? A systematic literature review. *Journal of Information Science*, 51(2), 511–522.
<https://doi.org/10.1177/01655515221141034>
- He, Z., & Fang, W. (2025). Research data management in institutional repositories: an architectural approach using data lakehouses. *Digital Library Perspectives*, 41(1), 145–178. <https://doi.org/10.1108/DLP-02-2024-0022>
- Jayakanth, F., Byrappa, A. T., & Minj, F. (2022). Migration of a Research Library's ICT-Based Services to a Cloud Platform. *Information Technology and Libraries*, 41(1). <https://doi.org/10.6017/ital.v41i1.13537>
- Krasynskyi, O. (2025). Transformation of the Role of Libraries in the Digital Educational Environment. *Visnyk of Kharkiv State Academy of Culture*, 67, 138–153. <https://doi.org/10.31516/2410-5333.067.10>
- Macedo, R., Paulo, J., Pereira, J., & Bessani, A. (2021). A Survey and Classification of Software-Defined Storage Systems. *ACM Computing Surveys*, 53(3), 1–38. <https://doi.org/10.1145/3385896>
- Pasqui, V. (2024). Digital curation and long-term digital preservation in libraries. *JLIS.It*, 15(1), 109–125. <https://doi.org/10.36253/jlis.it-567>
- Royce, W. (2021). *Managing the Development of Large Software Systems (1970)* (pp. 321–332). <https://doi.org/10.7551/mitpress/12274.003.0035>
- Royce, W. W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*, 1–9.
- Salvi, A., Masram, B., Deokar, V., Kolte, P., & Dhas, S. (2025). Leveraging MinIO and FileCloud for On-Premise Storage-as-a-Service in Educational Institutions. *2025 Seventeenth International Conference on Contemporary Computing (IC3)*, 1–4. <https://doi.org/10.1109/IC366947.2025.11290695>
- Sauro, J., & Lewis, J. R. B. T.-Q. the U. E. (Second E. (2016). *Chapter 8 - Standardized usability questionnaires* (pp. 185–248). Morgan Kaufmann. <https://doi.org/https://doi.org/10.1016/B978-0-12-802308-2.00008-4>
- Yakubu, A. S., Kassim, A. M., & Husin, M. H. (2025). Moderating the role of trust toward intention to adopt cloud computing by academic libraries in North-Eastern Nigerian. *Journal of Librarianship and Information Science*, 57(2), 551–567. <https://doi.org/10.1177/09610006231222623>
- Zareef, M., & Jabeen, M. (2025). A systematic review of digital curation services in academic libraries: navigating policies, skills and challenges. *Digital Library Perspectives*, 41(3), 518–543. <https://doi.org/10.1108/DLP-10-2024-0158>