TSAQOFAH
Jurnal Penelitian Guru Indonesia

p-ISSN: 2810-0395
e-ISSN: 2810-0042

Terindeks : Dimensions, Scilit, Lens, Semantic, Garuda, Google Scholar, Base, etc

https://doi.org/10.58578/tsaqofah.v5i6.7264

PENGEMBANGAN SISTEM *REPOSITORY* DOKUMEN TEKNIS BERBASIS WEB DI PT SEMEN PADANG PADA UNIT *SITE ENGINEERING*

Development of a Web-Based Technical Document Repository System at PT Semen Padang in the Site Engineering Unit

Annur Wahid Muriyanto, Muhammad Adri, Dedy Irfan, Resmi Darni
Universitas Negeri Padang
annurwhd@gmail.com; mhd.adri@unp.ac.id

Article Info:

Submitted:	Revised:	Accepted:	Published:
Jul 20, 2025	Aug 8, 2025	Aug 20, 2025	Aug 25, 2025

Abstract

The advancement of information and communication technology has driven significant transformation in document management across various institutions, including corporations. This study aims to develop a web-based technical document repository system for the Site Engineering Unit at PT Semen Padang to address the limitations of manual management and improve efficiency in document retrieval and security. The method employed is software development using the Agile approach, encompassing planning, system design with Unified Modeling Language (UML), system development using the Vue.js and Laravel frameworks, testing through the BlackBox method, and implementation of User Acceptance Testing (UAT). The results indicate that the system successfully modeled actor interactions through UML and passed functional testing. All core features, such as login, document management, role settings, audit trail, and document search—operated effectively without errors. The study concludes that the implementation of a web-based technical document repository system significantly

Tsaqofah journal is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

enhances operational efficiency, accelerates retrieval processes, and strengthens document management security within the Site Engineering Unit at PT Semen Padang. The research also recommends further exploration of artificial intelligence integration to optimize system functionality in the future.

Keywords: Repository System; Technical Documents; UML; Document Management; Software Development

Abstrak: Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi signifikan dalam pengelolaan dokumen di berbagai institusi, termasuk perusahaan. Penelitian ini bertujuan untuk mengembangkan sistem *repository* dokumen teknis berbasis web di PT Semen Padang pada Unit Site Engineering guna mengatasi kendala pengelolaan manual serta meningkatkan efisiensi dalam pencarian dan pengamanan dokumen. Metode yang digunakan adalah pengembangan perangkat lunak dengan pendekatan Agile, mencakup tahapan perencanaan, desain menggunakan *Unified Modeling Language (UML)*, pengembangan sistem dengan framework Vue.js dan Laravel, pengujian menggunakan metode *BlackBox*, serta penerapan *User Acceptance Testing (UAT)*. Hasil penelitian menunjukkan bahwa sistem berhasil memodelkan interaksi aktor melalui UML dan telah teruji secara fungsional. Seluruh fitur utama seperti login, manajemen dokumen, pengaturan peran, *audit trail*, dan pencarian dokumen berfungsi dengan baik tanpa ditemukan *error*. Simpulan penelitian ini adalah bahwa penerapan sistem *repository* dokumen teknis berbasis web terbukti meningkatkan efisiensi operasional, mempercepat proses pencarian, serta memperkuat keamanan pengelolaan dokumen di lingkungan Unit Site Engineering PT Semen Padang. Penelitian ini juga merekomendasikan eksplorasi lanjutan terkait integrasi kecerdasan buatan untuk optimalisasi fungsi sistem di masa depan.

Kata Kunci: Sistem Repository; Dokumen Teknis; UML; Pengelolaan Dokumen; Pengembangan Perangkat Lunak

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam cara pengelolaan informasi dan dokumen di berbagai institusi, termasuk perusahaan. Pengelolaan dokumen teknis yang efektif menjadi kebutuhan utama dalam mendukung kelancaran operasional dan pengambilan keputusan berbasis data yang akurat. Kenyataannya, banyak perusahaan masih menghadapi kendala dalam pengelolaan dokumen secara manual yang rentan terhadap kehilangan, kerusakan, dan sulit diakses secara cepat (Sinaga, 2021). Penelitian ini yang dilakukan oleh (Rony & Anggraeni, 2023) menunjukkan bahwa tanpa sistem repository yang terstruktur, proses pencarian dokumen dapat memakan waktu lama dan berdampak langsung pada produktivitas kerja. Temuan ini sejalan dengan hasil studi (Nugroho, 2017) juga membuktikan bahwa penggunaan repository dokumen

berbasis web mampu memangkas waktu pencarian dokumen hingga 50% dibandingkan metode manual.

Hasil observasi selama Praktek Lapangan Industri (PLI) di PT Semen Padang, khususnya pada Unit *Site Engineering*, menunjukkan bahwa proses pengelolaan dokumen masih dilakukan secara semi-manual dengan memanfaatkan layanan *cloud storage* seperti OneDrive. Penyimpanan dokumen memang telah berpindah ke media digital, namun proses pengunggahan, pencarian, pengorganisasian, dan pengaturan hak akses tetap dilakukan secara manual oleh masing-masing pengguna. Dalam penggunaan penyimpanan berbasis *cloud*, proses pencarian dokumen memakan waktu yang cukup lama sehingga mengganggu efisiensi pekerjaan. Ketiadaan sistem pencatatan metadata yang terstruktur juga mengakibatkan pencarian menjadi lambat, risiko kehilangan atau tertukarnya versi dokumen tinggi, serta distribusi antar divisi kurang efisien.

Permasalahan bertambah dengan adanya ketergantungan terhadap personel tertentu yang memahami letak dan isi dokumen. Apabila personel tersebut tidak tersedia, proses pencarian dan pengelolaan dokumen menjadi terhambat. Ketergantungan pada *knowledge keeper* atau pegawai kunci dapat menghambat alur kerja dan meningkatkan risiko *bottleneck*. Situasi ini berpotensi menurunkan efisiensi kerja Unit *Site Engineering*, meningkatkan risiko kesalahan dalam pengambilan keputusan akibat keterlambatan akses informasi, serta menghambat pelestarian dokumen teknis yang bernilai strategis. Unit *Site Engineering* sendiri memiliki peran penting dalam mendukung proyek konstruksi dan pemeliharaan fasilitas yang memerlukan akses cepat terhadap dokumen teknis sebagai referensi utama.

Solusi yang relevan untuk mengatasi permasalahan tersebut adalah penerapan sistem *repository* dokumen berbasis web yang mampu menyediakan penyimpanan digital terorganisir, mendukung pencarian cepat berbasis metadata, serta memiliki fitur pelacakan histori dokumen untuk menjaga integritas data. Seperti dijelaskan oleh (Susilowati et al., 2022) menunjukkan bahwa penerapan repository digital di organisasi publik dapat mempercepat proses administrasi, meningkatkan keteraturan arsip, dan meminimalkan risiko kehilangan dokumen. Hal ini sejalan dengan pandangan (Alfredo & Susilowati, 2021) menegaskan bahwa *repository* tidak hanya berfungsi sebagai media penyimpanan, tetapi juga sebagai sarana pelestarian aset digital yang mendukung akses terbuka terhadap dokumen penting kapan saja dan di mana saja. Pengelolaan dokumen menjadi lebih terstruktur dengan adanya kontrol akses yang aman (Supriyatno & Hamidah, 2022).

Penerapan *repository* dokumen teknis berbasis web di lingkungan perusahaan diyakini dapat meningkatkan produktivitas kerja, mengurangi risiko kehilangan dokumen, serta mempercepat proses pengambilan keputusan berbasis informasi akurat. Seperti dijelaskan oleh (Suwanto, 2017) menekankan bahwa manajemen layanan *repository* yang tepat, seperti proses deposit dokumen dan komunikasi efektif dengan pengguna, dapat memastikan sistem berjalan optimal dan berdampak positif terhadap efisiensi kerja.

Pengembangan sistem *repository* dokumen teknis berbasis web yang dirancang untuk memenuhi kebutuhan Unit *Site Engineering* PT Semen Padang menjadi langkah strategis. Sistem ini akan memanfaatkan *framework* Laravel pada sisi *backend* dan Vue.js pada sisi *frontend* untuk memastikan antarmuka yang responsif serta mudah diakses dari berbagai perangkat. Perancangan basis data akan mendukung penyimpanan dokumen dalam berbagai format beserta metadata lengkap agar pencarian dapat dilakukan secara cepat dan akurat. Fitur manajemen hak akses pengguna, pelacakan histori perubahan dokumen, serta *audit trail* akan diintegrasikan untuk menjaga keamanan dan integritas data. Metodologi Agile Development dipilih agar pengembangan sistem dapat dilakukan secara iteratif dengan penyesuaian fitur berkelanjutan berdasarkan umpan balik pengguna di Unit *Site Engineering*. Dengan adanya sistem repository dokumen teknis berbasis web ini, diharapkan Unit *Site Engineering* dapat meningkatkan efisiensi operasional, meminimalisir risiko kehilangan dokumen, dan mempercepat pengambilan keputusan berbasis informasi yang terstruktur dan terpercaya.

METODE

Metode penelitian yang digunakan dalam pengembangan sistem repository dokumen teknis berbasis web ini mengikuti tahapan pengembangan perangkat lunak Agile yang meliputi *planning*, *design*, *development*, *testing*, *deployment*, dan *review/maintenance*. Pada tahap *planning*, dilakukan analisis kebutuhan sistem dengan mengidentifikasi proses bisnis, pelaku, permasalahan, dan solusi yang relevan. Analisis ini mencakup pemetaan aktivitas pengguna melalui tabel analisis proses, analisis pelaku, serta identifikasi masalah dan penyusunan solusi. Selanjutnya, tahap *design* dilakukan dengan pemodelan menggunakan UML yang terdiri dari *use case diagram*, *activity diagram* untuk admin dan user, serta *class diagram* yang menggambarkan struktur dan hubungan antar kelas. Perancangan basis data disusun dalam bentuk struktur tabel yang mencakup entitas pengguna, dokumen, klasifikasi, departemen, dan log aktivitas, disertai perancangan antarmuka untuk admin dan user yang mengutamakan kemudahan

penggunaan dan kelengkapan fitur. Tahap *development* mengimplementasikan desain tersebut dengan menggunakan *framework* Vue.js pada sisi *frontend*, Laravel pada sisi *backend*, dan MySQL sebagai basis data. Tahap akhir adalah *review/maintenance*, yaitu pemeliharaan berkala yang mencakup pemantauan performa, perbaikan bug, serta pembaruan konten dan fitur untuk memastikan sistem tetap optimal dan relevan dengan kebutuhan pengguna.



Gambar 1. Tahapan Meode Agile

Pengujian sistem dilakukan menggunakan pendekatan BlackBox Testing untuk memastikan setiap fungsi berjalan sesuai kebutuhan tanpa memeriksa struktur kode. Selain itu, dilakukan User Acceptance Testing (UAT) untuk mengevaluasi kesesuaian sistem dengan kebutuhan pengguna pada Unit Site Engineering.

HASIL

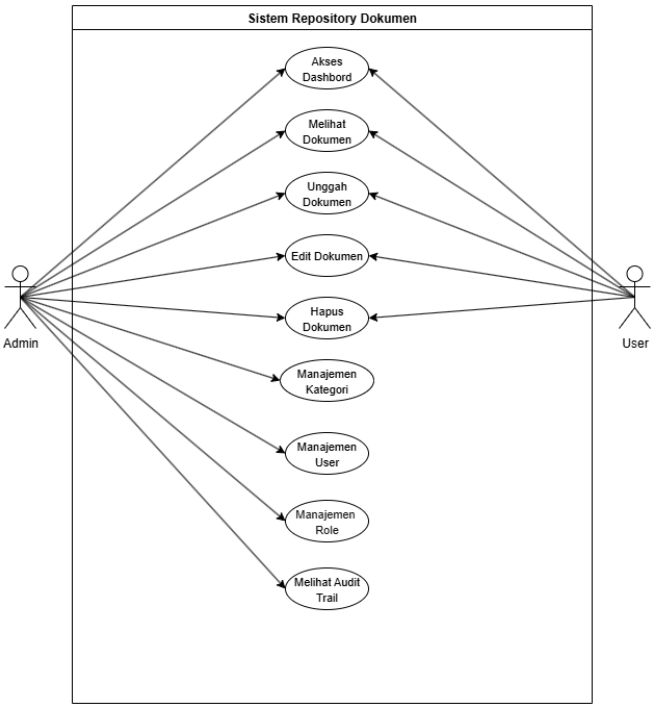
Perancangan Sistem

Perancangan Sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Perancangan sistem merupakan hasil modifikasi dari analisis yang nantinya akan diimplementasikan.

1.UML (*Unified Modeling Language*)

Berikut diagram – diagram UML yang digunakan dalam perancangan sistem *repository* ini.

a. *Use Case Diagram*

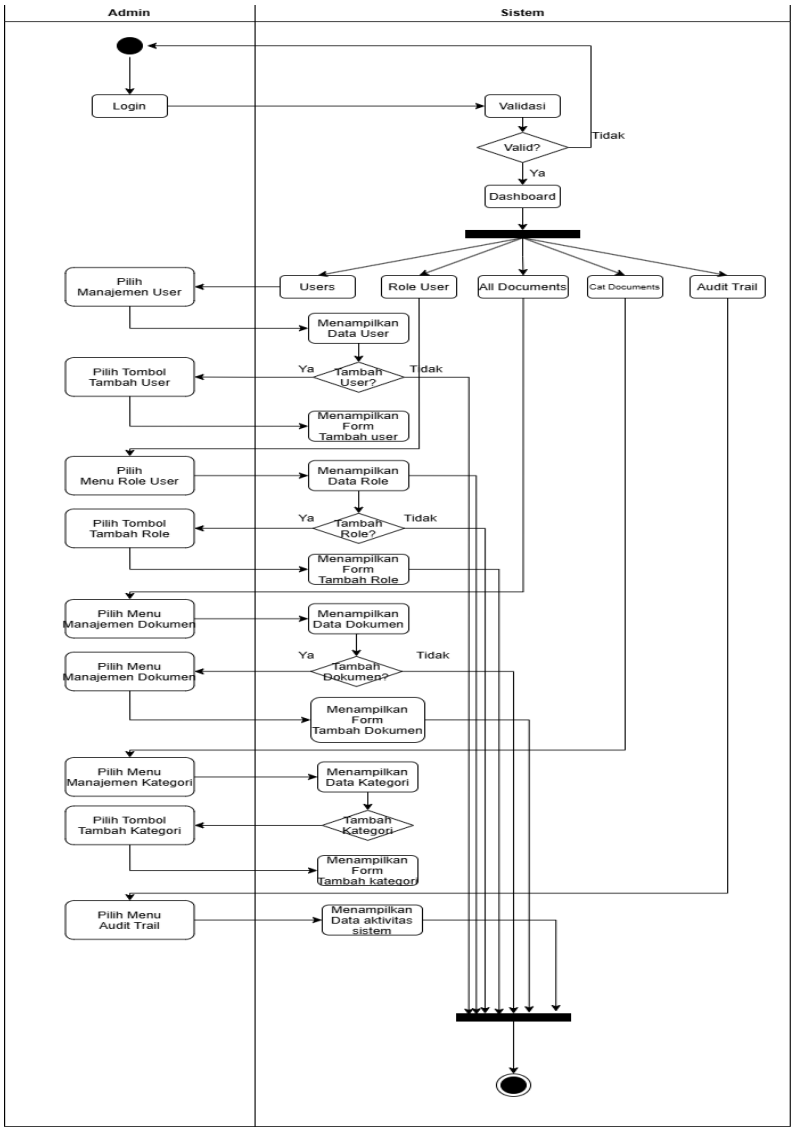


Gambar 2. *Use Case Diagram*

Use Case Diagram ini menggambarkan interaksi antara dua aktor utama, yaitu Admin dan *User*, dengan sistem repository dokumen teknis berbasis web. Aktor Admin memiliki hak akses penuh terhadap seluruh fungsi sistem, meliputi login, logout, manajemen dokumen (unggah, edit, hapus, lihat), manajemen kategori, manajemen pengguna, manajemen peran, serta pemantauan aktivitas melalui fitur audit trail. Admin juga bertanggung jawab terhadap pengelompokan dokumen dan pengaturan hak akses setiap pengguna dalam sistem. Sementara itu, aktor *User* memiliki hak akses terbatas yang disesuaikan dengan kebutuhan operasional. *User* dapat melakukan login, logout, serta mengunggah, melihat, mengedit, dan menghapus dokumen miliknya.*Activity Diagram*

Activity Diagram merupakan salah satu jenis diagram dalam UML (*Unified Model Language*) yang berfungsi untuk memodelkan alur kerja dari sistem. Diagram ini menunjukkan urutan aktivitas suatu proses dan memodelkan bagaimana aktivitas tersebut saling berhubungan satu sama lain.

b. *Activity Diagram* Admin



Gambar 3. *Activity Diagram* Admin

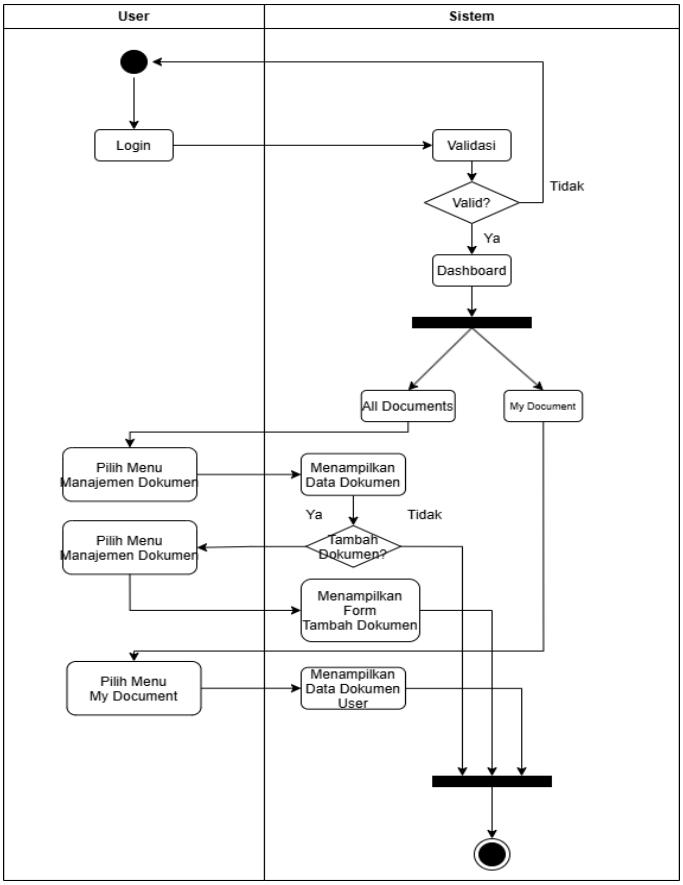
Activity diagram ini menggambarkan alur aktivitas Admin dalam sistem. Proses dimulai dari login, dilanjutkan dengan validasi kredensial oleh sistem. Jika valid, Admin diarahkan ke dashboard utama. Dari *dashboard*, Admin dapat mengakses beberapa menu, antara lain:

1) Manajemen *User*, menampilkan data pengguna serta opsi untuk menambah *User* baru.
2) Manajemen *Role*, menampilkan data peran pengguna dan menambahkan peran baru.
3) Manajemen Dokumen, mengelola dokumen teknis termasuk penambahan dokumen.

4) Manajemen Kategori, mengelola kategori dokumen dan menambahkan kategori baru.
5) *Audit Trail*, melihat riwayat aktivitas pengguna dalam sistem.

Setiap proses penambahan data diawali dengan pemilihan menu, dilanjutkan dengan tampilan data, dan jika dipilih, sistem menampilkan formulir penambahan data yang sesuai. Diagram ini mencerminkan alur kerja sistematis yang mendukung efisiensi pengelolaan sistem oleh Admin.

c. *Activity Diagram User*

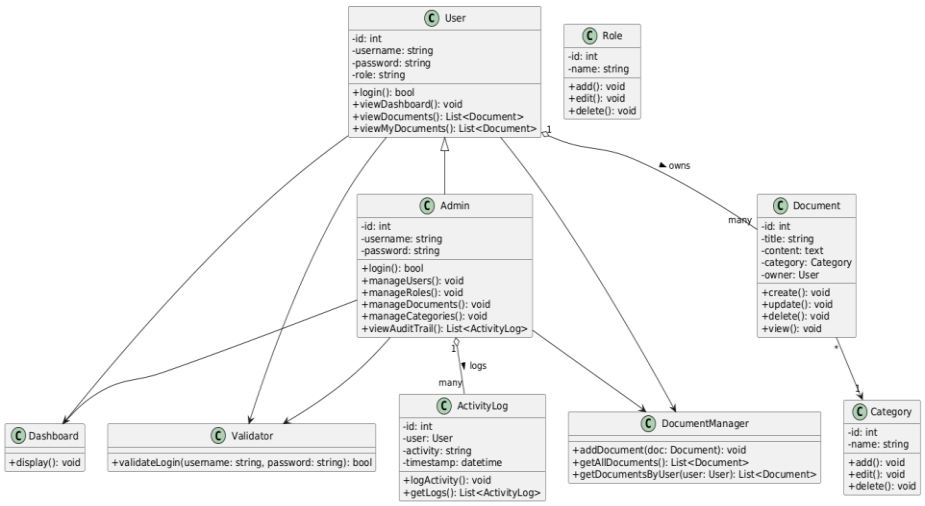


Gambar 4 *Activity Diagram User*

Activity diagram ini menggambarkan alur aktivitas pengguna (*User*) dalam sistem. Proses diawali dengan login, kemudian sistem melakukan validasi. Jika valid, pengguna diarahkan ke dashboard. Dari dashboard, pengguna dapat memilih menu Manajemen Dokumen untuk melihat dan menambah dokumen, atau memilih menu *My Document* untuk melihat dokumen milik sendiri. Penambahan dokumen dilakukan

melalui form yang ditampilkan setelah pengguna memilih opsi tambah dokumen. Diagram ini mencerminkan proses interaksi pengguna dengan sistem secara efisien sesuai hak aksesnya.

d. *Class Diagram*



Gambar 5. *Class Diagram*

Diagram kelas ini menggambarkan rancangan sistem manajemen dokumen berbasis web yang terdiri dari entitas utama seperti *User*, Admin, *Document*, *Category*, *Role*, *ActivityLog*, Validator, *Dashboard*, dan *DocumentManager*. Pengguna (*User*) memiliki atribut identitas dan peran, serta dapat melihat dokumen dan dasbor. Admin merupakan turunan dari *User* dengan hak akses tambahan untuk mengelola pengguna, peran, dokumen, dan kategori, serta memantau jejak aktivitas melalui *ActivityLog*. Setiap dokumen diklasifikasikan dalam satu *Category* dan dimiliki oleh satu *User*, dengan proses pengelolaan dilakukan melalui *DocumentManager*. Proses otentikasi ditangani oleh Validator, sementara *Dashboard* menampilkan antarmuka utama setelah login. Relasi antar kelas dirancang untuk mendukung prinsip enkapsulasi dan pemisahan tanggung jawab dalam sistem berbasis objek.

2.Basis Data

a. Struktur Tabel

1) Tabel *users*

Tabel 1. *users*

No	Field	Type	Width	Keterangan
1	Id	bigint		Primary Key
2	Name	varchar	255	
3	Email	varchar	255	Unique
4	password	varchar	255	
5	Role			

2) Tabel *documents*

Tabel 2. *documents*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	document_code	varchar	255	
3	title	varchar	255	
4	description	text		
5	file_size	varchar	255	
6	released_at	date		
7	department_id	bigint		Foreign Key ke departments
8	classification_id	bigint		Foreign Key ke classifications
9	created_by	bigint		Foreign Key ke *User*s
10	updated_by	bigint		Foreign Key ke *User*s
11	deleted_by	bigint		Foreign Key ke *User*s

3) Tabel *classifications*

Tabel 3. *classifications*

No	Field	Type	Width	Keterangan
1	Id	bigint		Primary Key
2	Name	varchar	255	
3	classification_code	varchar	255	

4) Tabel *departments*

Tabel 4. *departments*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	name	varchar	255	
3	department_code	varchar	255	
4	plant_id	bigint		Foreign Key ke tabel plants

5) Tabel *plants*

Tabel 5. *plants*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	name	varchar	255	

6) Tabel *in_progress_documents*

Tabel 6. *in_progress_documents*

No	Field	Type	Width	Keterangan
1	Id	bigint		Primary Key
2	Title	varchar	255	
3	description	text		
4	file_size	varchar	255	
5	created_by	bigint		FK ke *User*s
6	updated_by	bigint		FK ke *User*s
7	department_id	bigint		FK ke departments
8	classification_id	bigint		FK ke classifications

7) Tabel *in_progress_document_revisions*

Tabel 7. *in_progress_document_revisions*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	in_progress_ document_id	bigint		FK ke in_progress_documents
3	file_size	varchar	255	
4	comments	varchar	255	
5	created_by	bigint		FK ke *User*s
6	updated_by	bigint		FK ke *User*s

8) Tabel *in_progress_document_shares*

Tabel 8. *in_progress_document_shares*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	inprogress_doc_id	bigint		FK ke in_progress_documents
3	User_id	bigint		FK ke *User*s
4	shared_by	bigint		FK ke *User*s (nullable)

No	Field	Type	Width	Keterangan
5	permission	enum		
6	note	text		

9) Tabel *edms_documents*

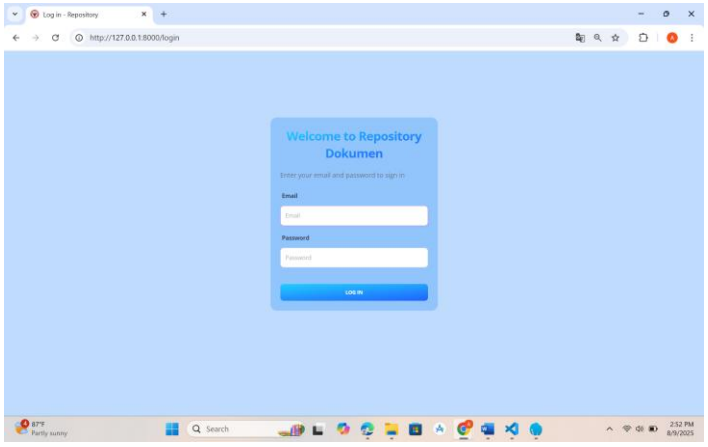
Tabel 9. *edms_documents*

No	Field	Type	Width	Keterangan
1	id	bigint		Primary Key
2	document_code	varchar	255	Unique
3	erf_code	varchar	255	
4	title	varchar	255	
5	job_type	varchar	255	
6	file_size	varchar	255	
7	created_by	bigint		FK ke *User*s
8	updated_by	bigint		FK ke *User*s
9	release_date	date		

3.Antar Muka (*Interface*)

Perancangan antarmuka ini berfungsi untuk merencanakan antarmuka yang akan digunakan pada sistem yang akan dibangun. Perencanaan antarmuka sebaiknya nyaman dilihat, mudah digunakan, dan lain sebagainya.

a. Antar Muka Login

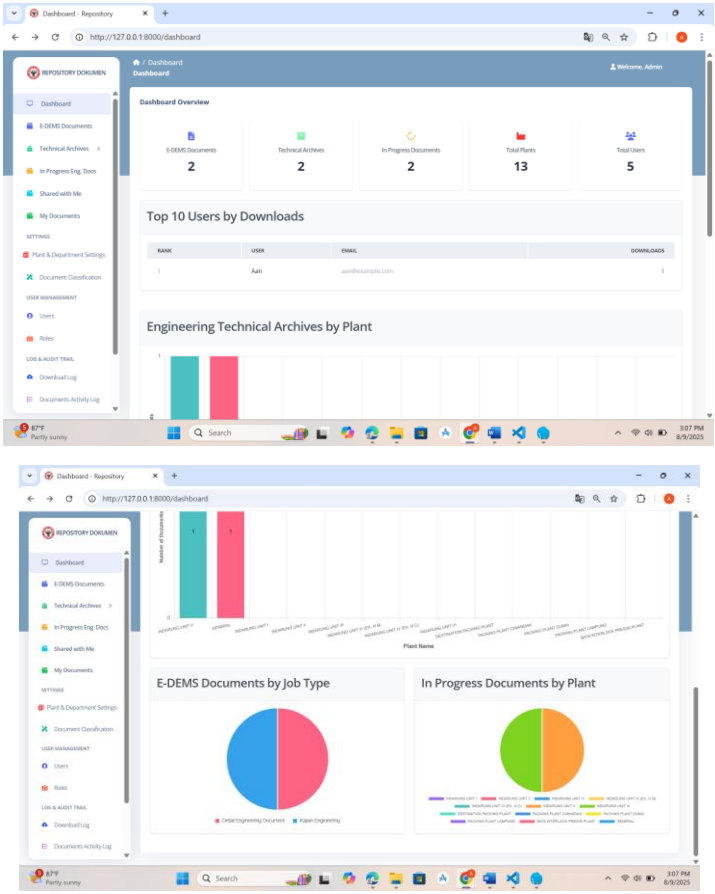


Gambar 6. Halaman Login Admin dan User

Halaman *login* merupakan halaman awal yang ditampilkan ketika pengguna mengakses web *repository* dokumen teknis. Pada halaman ini, pengguna diminta memasukkan *username* dan *password* untuk proses autentikasi. Jika data yang dimasukkan

sesuai, pengguna akan diarahkan menuju *dashboard* masing-masing sesuai peran (Admin/*User*). Bagi pengguna yang tidak memiliki akses, sistem akan menampilkan pesan peringatan. Halaman *login* ini berfungsi untuk menjaga keamanan akses ke dalam sistem *repository*.

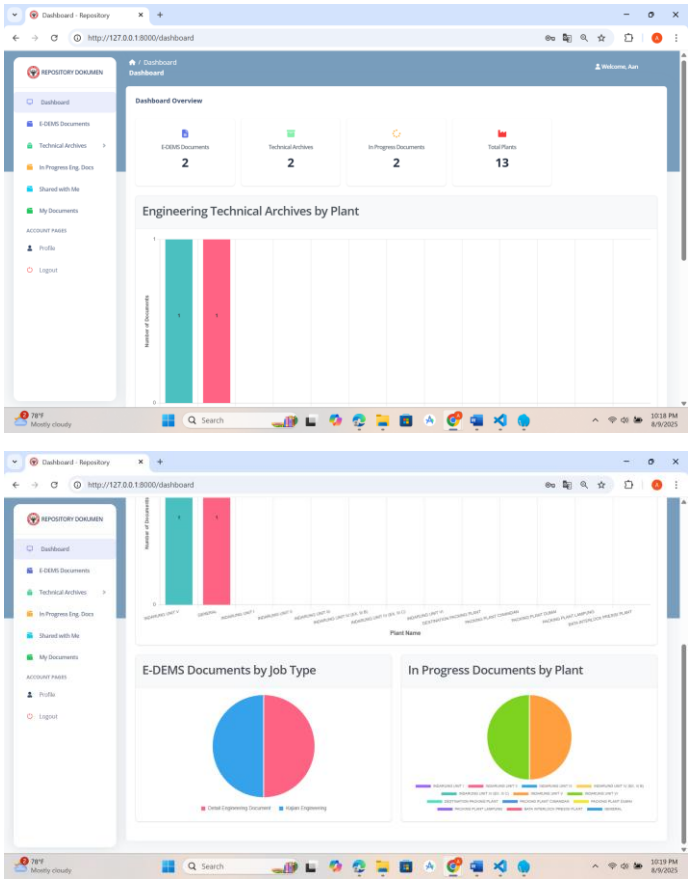
b. Antar Muka Dashboard Admin



Gambar 7. Halaman Dashboard Admin

Setelah berhasil melakukan login, pengguna akan diarahkan ke halaman dashboard. Halaman dashboard berfungsi sebagai halaman utama yang menampilkan ringkasan informasi terkait aktivitas dalam sistem repository. Untuk Admin, dashboard akan menampilkan informasi global seperti total dokumen yang tersimpan, jumlah dokumen dalam proses, aktivitas terbaru, serta statistik penggunaan sistem. Admin dapat memantau keseluruhan aktivitas repository dari berbagai plant dan departemen.

c. Antar Muka Dashboard *User*



Gambar 8. Halaman Dashboard User

Setelah berhasil melakukan *login*, pengguna akan diarahkan ke halaman *dashboard*. Halaman *dashboard* berfungsi sebagai halaman utama yang menampilkan ringkasan informasi terkait aktivitas dalam sistem *repository*. *Dashboard User* hanya menampilkan informasi yang relevan dengan dokumen milik *user* tersebut atau dokumen yang dibagikan kepadanya. *User* dapat melihat dokumen yang sedang dalam proses, dokumen yang telah diarsipkan, serta aktivitas yang berkaitan dengan akun mereka sendiri.

Uji Coba Sistem

Teknik pengujian *BlackBox Testing* digunakan selama fase pengujian untuk menilai fungsionalitas sistem dari sudut pandang pengguna. Tanpa memeriksa struktur kode internal, pendekatan ini difokuskan untuk memastikan bahwa setiap fitur sistem beroperasi sebagaimana mestinya.

1. Pengujian Halaman *Login*

Proses pengujian halaman *login* memiliki beberapa skenario seperti yang ditunjukkan pada tabel 10.

Tabel 10. *Test Case Halaman Login Admin dan User*

No	Fitur	Test Case	Langkah Uji	Expected Result	Status
1	*Login*	*Login* dengan data valid	Masukkan *username* & *password* valid	*Login* berhasil masuk ke *dashboard*	Sukses
		Login dengan data tidak valid	Masukkan *username* & *password* tidak valid	*Login* gagal dan muncul pesan gagal	Sukses

2. Pengujian Halaman Admin

Tabel 11 menunjukkan bagaimana proses pengujian halaman admin. Pengujian dilakukan terhadap semua menu yang terdapat pada sidebar seperti *Dashboard*, *E-DEMS Documents*, *Technical Archives*, *In Progress Eng. Docs*, *Shared with Me*, *My Document*, *Plant & Deaprtment* Settings, *Document Classification, Users, Roles, Download Log, Documents Activity Log, Profile* dan *Logout*.

Tabel 11. *Test Case Halaman Admin*

No	Fitur	Test Case	Langkah Uji	Expected Result	Status
1	*Dashboard*	Apakah Semua data dashboard ditampilkan dengan baik?	Klik Dashboard	Semua data dashboard ditampilkan dengan akurat	Sukses
2	*E-DEMS Documents*	Upload dokumen baru	Klik Upload, pilih file, klik Simpan	Dokumen muncul di daftar	Sukses
		Edit metadata dokumen	Klik Edit, ubah data, simpan	Data dokumen terupdate	Sukses
		Hapus dokumen	Klik Delete pada dokumen	Dokumen terhapus dari daftar	Sukses
		Cari dokumen	Ketik keyword di search bar	Daftar dokumen menampilkan hasil pencarian	Sukses
3	*Technical Archives*	Filter dokumen berdasarkan kategori	Pilih kategori di filter	Daftar dokumen terfilter	Sukses
4	*In Progress Eng. Docs*	Update status dokumen	Klik dokumen, ubah status, simpan	Status dokumen terupdate	Sukses
5	*Shared with Me*	Lihat dokumen dibagikan	Klik menu Shared with Me	Daftar dokumen yang dibagikan tampil	Sukses
6	*My Documents*	Lihat dokumen milik sendiri	Klik menu My Documents	Daftar dokumen pribadi tampil	Sukses

7	*Plant & Department Settings*	Tambah Plant/Department baru	Klik Add, isi data, simpan	Data Plant/Department bertambah	Sukses
8	*Document Classification*	Tambah klasifikasi dokumen	Klik Add, isi data, simpan	Klasifikasi dokumen bertambah	Sukses
9	*Users*	Tambah user baru	Klik Add User, isi data, simpan	User baru muncul di daftar	Sukses
		Edit data user	Klik Edit pada user, ubah data, simpan	Data user terupdate	Sukses
		Hapus user	Klik Delete pada user	User terhapus dari daftar	Sukses
		Reset password user	Klik Reset Password pada user	Password user berhasil direset	Sukses
10	*Roles*	Tambah role baru	Klik Add Role, isi data, simpan	Role baru muncul di daftar	Sukses
11	*Download Log*	Lihat histori download dokumen	Klik menu Download Log	Daftar log download tampil	Sukses
12	*Documents Activity Log*	Lihat histori aktivitas dokumen	Klik menu Documents Activity Log	Histori aktivitas tampil	Sukses
13	*Profile*	Update data profil	Klik menu Profile, ubah data, simpan	Data profil user terupdate	Sukses
14	*Logout*	Logout dari sistem	Klik tombol Logout	User berhasil keluar dan diarahkan ke login	Sukses

Berdasarkan pengujian yang ditunjukkan pada tabel 11 dapat dilihat bahwa semua fitur pada halaman admin berjalan dengan baik tanpa *error*.

3. Pengujian Halaman *User*

Pengujian halaman *user* dilakukan terhadap semua menu yang terdapat pada halaman *user* seperti *Dashboard*, *E-DEMS Documents*, *Technical Archives*, *In Progress Eng. Docs*, *Shared with Me*, *My Document*, *Profile* dan *Logout*. Tabel 12 menunjukkan pengujian *dashboard user*.

Tabel 12. *Test Case Halaman User*

No	Fitur	Test Case	Langkah Uji	Expected Result	Status
1	*Dashboard*	Apakah Semua data dashboard ditampilkan dengan baik?	Klik *Dashboard*	Semua data dashboard ditampilkan dengan akurat	Sukses

2	*E-DEMS Documents*	Cari dokumen	Ketik keyword di search bar	Daftar dokumen menampilkan hasil pencarian	Sukses
		Upload dokumen baru (jika diizinkan)	Klik Upload, pilih file, klik Simpan	Dokumen muncul di daftar	Sukses
		Edit dokumen milik sendiri	Klik Edit pada dokumen milik sendiri	Data dokumen terupdate	Sukses
		Hapus dokumen milik sendiri	Klik Delete pada dokumen milik sendiri	Dokumen terhapus dari daftar	Sukses
3	*Technical Archives*	Filter dokumen berdasarkan kategori	Pilih kategori di filter	Daftar dokumen terfilter	Sukses
4	*In Progress Eng. Docs*	Update status dokumen milik sendiri/dibagikan	Klik dokumen, ubah status, simpan	Status dokumen terupdate	Sukses
5	*Shared with Me*	Lihat dokumen dibagikan	Klik menu Shared with Me	Daftar dokumen yang dibagikan tampil	Sukses
6	*My Documents*	Lihat dokumen milik sendiri	Klik menu My Documents	Daftar dokumen pribadi tampil	Sukses
7	*Profile*	Update data profil	Klik menu Profile, ubah data, simpan	Data profil user terupdate	Sukses
8	*Logout*	Logout dari sistem	Klik tombol Logout	User berhasil keluar dan diarahkan ke login	Sukses

Berdasarkan pengujian yang ditunjukkan pada tabel 12 dapat dilihat bahwa semua fitur pada halaman *user* berjalan dengan baik tanpa *error*.

PEMBAHASAN

Pembahasan dalam artikel ini bertujuan untuk menjawab rumusan masalah yang diajukan, menginterpretasi temuan-temuan, serta memunculkan teori-teori baru atau modifikasi dari teori yang telah ada. Dalam konteks ini, perancangan sistem manajemen dokumen berbasis web menggunakan UML (Unified Modeling Language) berhasil memberikan solusi atas pertanyaan penelitian mengenai cara merancang sistem yang efisien dengan pengelolaan hak akses yang jelas antara Admin dan User. Temuan ini sejalan dengan penelitian sebelumnya, seperti yang dilakukan oleh Ambler (2004), yang mengungkapkan bahwa UML adalah alat yang efektif untuk memodelkan sistem perangkat lunak berorientasi objek. Penggunaan diagram UML, seperti Use Case Diagram, Activity Diagram, dan Class Diagram, terbukti efektif dalam menggambarkan interaksi antar aktor dan alur kerja sistem secara sistematis. Hasil pengujian dengan metode BlackBox dan UAT menunjukkan bahwa

seluruh fitur sistem berjalan dengan baik tanpa error. Hal ini memperkuat temuan bahwa perancangan berbasis UML yang dikombinasikan dengan framework modern (Laravel dan Vue.js) mampu menghasilkan sistem repository yang stabil, aman, dan sesuai kebutuhan pengguna. Dengan demikian, sistem ini tidak hanya efektif secara teoretis dalam desain, tetapi juga teruji secara praktis dalam penerapan di lapangan.

Interpretasi hasil temuan menunjukkan bahwa desain yang menggunakan diagram UML memungkinkan pemisahan tanggung jawab yang jelas antara Admin dan User, yang mengarah pada efisiensi pengelolaan dokumen. Penggunaan diagram seperti Activity Diagram menggambarkan alur aktivitas Admin dan User dalam mengakses dan mengelola dokumen, yang sejalan dengan teori tentang pembagian peran dalam sistem informasi (Zhang et al., 2008). Dalam hal ini, temuan yang diperoleh juga menegaskan pentingnya struktur tabel dalam database yang terorganisir dengan baik, seperti yang digarisbawahi oleh Hwang et al. (2011), yang mengemukakan bahwa manajemen basis data yang efisien adalah kunci utama dalam mendukung kinerja sistem manajemen dokumen. Selain itu, temuan dalam artikel ini memunculkan teori baru mengenai penerapan UML yang lebih dinamis dalam desain sistem manajemen dokumen berbasis web. Sebelumnya, UML lebih dikenal sebagai alat untuk mendesain sistem secara statis, namun dengan menggabungkan alur kerja yang melibatkan kolaborasi antara Admin dan User, sistem ini menunjukkan fleksibilitas yang lebih tinggi dalam merespons kebutuhan dinamis pengguna. Hal ini mengembangkan pandangan yang diajukan oleh Booch et al. (1999) tentang potensi UML sebagai alat yang lebih adaptif dalam mendesain sistem yang lebih kompleks dan interaktif.

Perbandingan dengan literatur terdahulu menunjukkan bahwa meskipun penggunaan UML dalam perancangan sistem manajemen dokumen telah banyak dibahas, integrasi antara alur kerja Admin dan User serta kolaborasi dalam pengelolaan dokumen secara real-time masih belum banyak diterapkan secara mendalam. Sebagai contoh, penelitian oleh Berenbach (2011) lebih berfokus pada penggunaan UML dalam desain arsitektur sistem, sedangkan penelitian ini menawarkan pendekatan yang lebih terperinci dan berorientasi pada efisiensi interaksi antar pengguna. Dengan demikian, temuan dalam artikel ini memberikan kontribusi baru terhadap perkembangan aplikasi UML dalam pengelolaan dokumen berbasis web.

Implikasi dari temuan ini sangat relevan bagi pengembangan sistem manajemen dokumen berbasis web di organisasi modern. Dengan memanfaatkan UML, sistem dapat berjalan lebih efisien, meminimalkan kesalahan, serta memastikan bahwa setiap pengguna

memiliki akses yang sesuai dengan peran mereka dalam sistem. Temuan ini memberikan landasan untuk penelitian lebih lanjut yang mengembangkan penggunaan teknologi baru, seperti kecerdasan buatan (AI) atau machine learning, dalam pengelolaan dokumen. Hal ini bertujuan untuk meningkatkan fungsionalitas dan efisiensi sistem, serta memberikan solusi yang lebih canggih dan adaptif dalam pengelolaan dokumen secara real-time, seperti yang disarankan oleh Chen et al. (2015). Penelitian lanjutan juga dapat melibatkan uji coba skenario yang lebih beragam untuk mengidentifikasi potensi pengembangan sistem yang lebih optimal, terutama dalam aspek keamanan dan performa sistem.

KESIMPULAN

Berdasarkan hasil penelitian, perancangan sistem repository dokumen teknis berbasis web dengan UML berhasil memodelkan interaksi antar aktor serta alur kerja sistem secara efisien. Implementasi sistem menggunakan Laravel dan Vue.js telah diuji dengan BlackBox Testing dan User Acceptance Testing (UAT), yang menunjukkan bahwa seluruh fitur berjalan baik tanpa error. Sistem ini terbukti meningkatkan efisiensi operasional, mempercepat pencarian dokumen, dan meningkatkan keamanan data. Penelitian selanjutnya disarankan untuk mengintegrasikan kecerdasan buatan (AI) atau machine learning guna meningkatkan fungsionalitas sistem.

DAFTAR PUSTAKA

Alfarisi, I. A., Priandika, A. T., & Puspaningrum, A. S. (2023). Penerapan Framework Laravel Pada Sistem Pelayanan Kesehatan (Studi Kasus: Klinik Berkah Medical Center). *Jurnal Ilmiah Computer Science*, *2*(1), 1–9. https://doi.org/10.58602/jics.v2i1.11

Amir, R. A. (2025). *SISTEM INFORMASI REPOSITORY STMIK CATUR SAKTI KENDARI BERBASIS WEB*. *10*(1).

Armehzan, H., Tuahuns, W. A., Rumboirusi, K. F., Hababuk, A. V. Y., Isawa, C. S., Krimadi, S., & Hasan, P. (2024). PENERAPAN UML UNTUK MENGOPTIMALKAN KINERJA PENJUALAN DI THE BEADERY JAYAPURA. *Bulletin of Network Engineer and Informatics*, *2*(2), 97. https://doi.org/10.59688/bufnets.v2i2.39

Dewi, K. C., Ciptayani, P. I., & Wijaya, I. W. R. (2018). Agile Project Management pada Pengembangan E-Musrenbang Kelurahan Benoa Bali. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *5*(6), 723–730. https://doi.org/10.25126/jtiik.2018561143

Fikri Rakawitan & Fadiyah Hani Sabila. (2023). Persiapan Peralatan dan Dokumen Sebelum Melaksanakan Aktivitas Bongkar Curah Kering pada PT Wahana Intradermaga Niaga Belawan. *Jurnal Riset Manajemen*, *1*(3), 372–382. https://doi.org/10.54066/jurma.v1i3.2604

Firnando, J., Franko, B., Tanzil, S. P., Wilyanto, N., Tan, C., & Kom, E. H. M. (2023). Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang. *JURNAL PENGABDIAN KEPADA MASYARAKAT*, *3*.

Irianto, I., Sudarmin, S., & Afrisawati, A. (2021). PENERAPAN METODE CUSTOMER RELATIONSHIP MANAGEMENT PADA PENJUALAN TOKO BAJU AZZAHRA. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, *4*(2), 191. https://doi.org/10.54314/jssr.v4i2.584

Kodali, N. . (2024). Tailwind CSS Integration in Angular: A Technical Overview. *International Journal of Innovative Research in Science,Engineering and Technology*, *13*(09), 1–14. https://doi.org/10.15680/IJIRSET.2024.1309092

M. Asep Rizkiawan, Harry Ramza, & Endy Sjaiful Alim. (2023). SISTEM INFORMASI PENCATATAN ASET DAN PEMINJAMAN BARANG MENGGUNAKAN METODE PENGEMBANGAN AGILE PADA BPTI UHAMKA. *Journal of Scientech Research and Development*, *5*(2), 461–473. https://doi.org/10.56670/jsrd.v5i2.217

Nistrina, K., & Sahidah, L. (2022). UNIFIED MODELLING LANGUAGE (UML) UNTUK PERANCANGAN SISTEM INFORMASI PENERIMAAN SISWA BARU DI SMK MARGA INSAN KAMIL. *Jurnal Sistem Informasi*, *04*.

Prakarsa, A. G., & Sujarwo, A. (2021). *Pemanfaatan Vue Js Pada Fitur Pengaturan Tambak Dalam Aplikasi Budi Daya Tambak Jala*.

Purwandari, N., & Firmansyah, B. (2023). Sistem Repository Dokumen Akreditasi Program Studi Berbasis Web pada Institut Bisnis dan Informatika Kosgoro 1957. *remik*, *7*(1), 196–210. https://doi.org/10.33395/remik.v7i1.11976

Setyawan, I., Anwariningsih, S. H., & Fitriyadi, F. (2024). *PEMBANGUNAN SISTEM REPOSITORI DOKUMEN UNIVERSITAS SAHID SURAKARTA*. *06*(02).

Sinaga, A. (2021). *Manajemen Sistem Informasi Instituational Repository*. *5*(1).

Siregar, L., Ramadhani, S., & Yulianto. (2024). Internship Information System at PT Biro Klasifikasi Indonesia Samarinda Branch. *TEPIAN*, *5*(3), 57–65. https://doi.org/10.51967/tepian.v5i3.3115

Sotnik, S., Manakov, V., & Lyashenko, V. (2023). *Overview: PHP and MySQL Features for Creating Modern Web Projects*. *7*(1).

Susilowati, M., Kurniawan, Y., & Wilujeng, L. L. (2022). Implementation of a Web-based Digital Repository Information System at the Office of Sumbersekar Village. *Journal of Community Practice and Social Welfare*, *2*(1), 11–23. https://doi.org/10.33479/jacips.2022.2.1.11-23

Suwanto, S. A. (2017). Manajemen Layanan Repository Perguruan Tinggi. *Lentera Pustaka: Jurnal Kajian Ilmu Perpustakaan, Informasi dan Kearsipan*, *3*(2), 165. https://doi.org/10.14710/lenpust.v3i2.16740

Tabrani, M., Suhardi, & Priyandaru, H. (2021). SISTEM INFORMASI MANAJEMEN BERBASIS WEBSITE PADA UNL STUDIO DENGAN MENGGUNAKAN FRAMEWORK CODEIGNITER. *Jurnal Ilmiah M-Progress*, *11*(1). https://doi.org/10.35968/m-pu.v11i1.598