Implementasi Template Engine pada Document Generator Laporan Akademik Mahasiswa Jurusan Bahasa dan Pariwisata

Octa Angga Kusuma [1)], Ongky Ramadona [2)], Muhammad Wahyudi [3)]
[1]Jurusan Bahasa dan Pariwisata, Politeknik Negeri Sriwijaya, Palembang
[2]Jurusan Akuntansi, Politeknik Negeri Sriwijaya, Palembang
[3]Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang
e-mail: [1]ooks_21@yahoo.co.id, [2]ongky@polsri.ac.id, [3]wahyudi@polsri.ac.id

Abstrak

Penyusunan laporan akademik mahasiswa di Jurusan Bahasa dan Pariwisata masih dilakukan secara manual, sehingga berpotensi menimbulkan kesalahan data, ketidakkonsistenan format, dan waktu penyusunan yang panjang. Penelitian ini bertujuan mengimplementasikan template engine pada aplikasi Document Generator berbasis web untuk mengotomatisasi penyusunan Laporan Akhir dan Laporan Kerja Praktik dengan format yang konsisten. Pengembangan menggunakan metode Research and Development (R&D) dengan model Prototype. Pengujian dilakukan melalui Black Box Testing, pengukuran efisiensi berdasarkan waktu penyusunan dan human error, serta User Acceptance Testing (UAT). Hasil Black Box Testing menunjukkan seluruh 11 skenario berhasil dijalankan sesuai hasil yang diharapkan. UAT terhadap 30 pengguna memperoleh nilai rata-rata 4,69 dari skala 5 atau 93,87%, dengan kesesuaian fungsi sebesar 95,33%, kemudahan penggunaan dan kecepatan proses masing-masing 94,00%, manfaat sistem 94,00%, serta kualitas dokumen 92,00%. Pengukuran efisiensi dirancang untuk membandingkan waktu penyusunan dan human error antara proses manual dan penggunaan sistem. Hasil penelitian menunjukkan bahwa penerapan template engine mendukung otomatisasi penyusunan dokumen akademik melalui pemetaan data ke template, sehingga menghasilkan dokumen yang lebih terstruktur, konsisten, dan sesuai kebutuhan pengguna.

Kata kunci*— Template Engine, Document Generator, Otomatisasi Dokumen Akademik, Laporan Akademik, Sistem Administrasi Akademik Berbasis Web.*

Abstract

The preparation of academic reports by students in the Department of Language and Tourism is still carried out manually, potentially leading to data entry errors, inconsistent formatting, and lengthy preparation times. This study aims to implement a template engine in a web-based Document Generator application to automate the preparation of Final Reports and Internship Reports with consistent formatting. The system was developed using the Research and Development (R&D) method with the Prototype model. Testing was conducted through Black Box Testing, efficiency measurement based on preparation time and human errors, and User Acceptance Testing (UAT). The Black Box Testing results showed that all 11 scenarios were successfully executed as expected. UAT involving 30 users obtained an average score of 4.69 out of 5, equivalent to 93.87%, with functional suitability at 95.33%, ease of use and process speed at 94.00% each, system usefulness at 94.00%, and document quality at 92.00%. Efficiency measurement was designed to compare preparation time and human errors between the manual process and the use of the system. The results indicate that implementing a template engine supports the automation of academic document preparation through data mapping into predefined templates, producing documents that are more structured, consistent, and aligned with user requirements.

Keywords*— Template Engine, Document Generator, Academic Document Automation, Academic Report, Web-Based Academic Administration System.*

Submitted: 28 Agustus 2026
Published: 30 September 2026

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi pengelolaan administrasi di lingkungan pendidikan tinggi, termasuk dalam pengelolaan dokumen akademik mahasiswa. Pemanfaatan sistem berbasis digital menjadi kebutuhan mendasar untuk meningkatkan efisiensi, akurasi, dan konsistensi administrasi akademik. Namun demikian, pada sejumlah institusi pendidikan, penyusunan dokumen akademik masih dilakukan secara manual sehingga proses administratif berjalan kurang efektif.

Laporan akhir dan laporan kerja praktik merupakan dua jenis dokumen akademik dengan intensitas penyusunan yang cukup tinggi di kalangan mahasiswa. Di Politeknik Negeri Sriwijaya, khususnya pada Jurusan Bahasa dan Pariwisata, penyusunan kedua dokumen tersebut sebagian besar masih dilakukan melalui aplikasi pengolah kata secara manual. Mahasiswa dituntut untuk menyesuaikan format penulisan, tata letak, identitas dokumen, serta struktur laporan secara berulang agar sesuai dengan pedoman akademik yang berlaku. Kondisi ini berpotensi menimbulkan ketidaksesuaian format, kesalahan penulisan data, serta memperpanjang waktu penyusunan maupun pemeriksaan dokumen.

Permasalahan tersebut tidak hanya berdampak pada mahasiswa, tetapi juga pada beban kerja administrasi jurusan dalam melakukan pengecekan dan standarisasi dokumen. Peningkatan jumlah mahasiswa yang menyusun laporan akhir dan laporan kerja praktik pada setiap semester turut memperbesar kompleksitas proses administrasi. Kondisi ini menunjukkan urgensi pengembangan sistem yang mampu mendukung proses penyusunan dokumen akademik secara otomatis dan terstruktur.

Salah satu pendekatan yang dapat diterapkan untuk menjawab permasalahan tersebut adalah penggunaan *template engine* pada aplikasi *document generator* berbasis web. *Template engine* memungkinkan pemisahan antara struktur tampilan dokumen dan data yang akan disajikan, sehingga proses pembentukan dokumen dapat dilakukan secara otomatis berdasarkan *template* yang telah ditetapkan sebelumnya. Pendekatan ini berpotensi menghasilkan dokumen dengan format yang lebih konsisten, menekan tingkat kesalahan penulisan, serta mempersingkat waktu penyusunan laporan akademik. Dinkelbach dkk. [1] menjelaskan bahwa *template engine* memungkinkan pemisahan antara data atau logika dengan keluaran melalui penggunaan model data dan *template*, yang kemudian diproses untuk menghasilkan dokumen keluaran secara otomatis. Dalam mekanisme tersebut, nilai aktual dari data dapat dipetakan ke parameter atau *placeholder* yang terdapat pada *template*, sehingga satu *template* dapat digunakan untuk menghasilkan berbagai dokumen dengan struktur yang tetap konsisten. Konsep serupa diterapkan dalam *document assembly*, di mana *document template* digunakan untuk merepresentasikan desain umum suatu dokumen dan dikombinasikan dengan data masukan untuk menghasilkan dokumen secara otomatis.

Beberapa penelitian terdahulu telah mengkaji implementasi *document generator* maupun otomasi *template* dokumen berbasis web, meskipun dengan konteks yang beragam. Haliza dkk. mengembangkan sistem otomasi *template* dokumen berbasis website dengan pendekatan metode *prototype* dan membuktikan bahwa sistem tersebut mampu mempercepat proses pengolahan dokumen dibandingkan metode manual [2]. Akan tetapi, cakupan dokumen yang diotomasi dalam penelitian tersebut masih bersifat umum dan belum menyentuh dokumen dengan struktur baku sebagaimana laporan akademik mahasiswa. Penelitian lain mengenai sistem *document generation* berbasis web menunjukkan bahwa penerapan arsitektur berbasis template dapat mempercepat proses pembuatan dokumen sekaligus menekan kesalahan formatting [3], namun pembahasannya lebih menitikberatkan pada aspek arsitektur sistem *generate* data secara umum tanpa mengaitkannya secara spesifik dengan kebutuhan format dokumen akademik. Adapun penelitian mengenai *template engine* pada konteks lain menunjukkan efektivitas teknologi tersebut dalam mengelola tampilan data secara dinamis dan terstruktur [4], meskipun penerapannya masih terbatas pada ranah visualisasi dataset, bukan pada dokumen administratif dengan ketentuan format yang lebih ketat seperti laporan akademik.

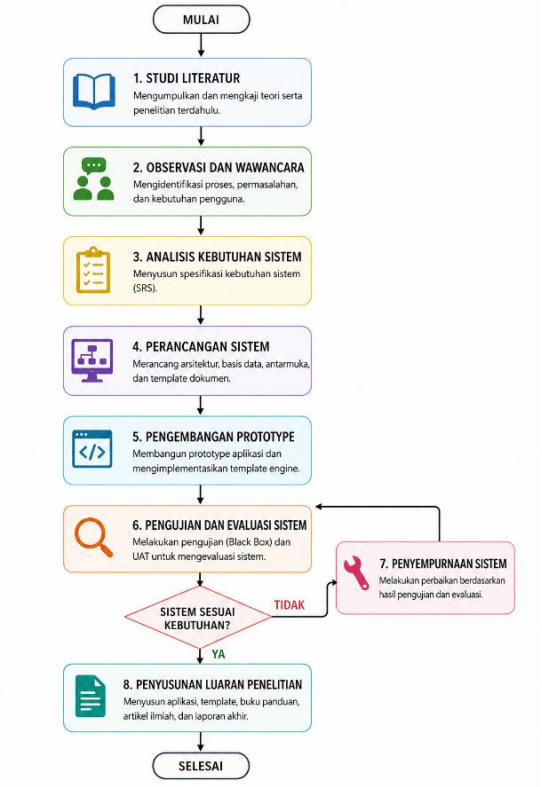
Ketiga penelitian tersebut secara konsisten menunjukkan bahwa *template engine* dan *document generator* efektif dalam mempercepat proses pembuatan dokumen serta mengurangi kesalahan format pada konteks masing-masing. Meskipun demikian, terdapat keterbatasan yang

Submitted: 28 Agustus 2026
Published: 30 September 2026

serupa di antara ketiganya, yaitu belum adanya penelitian yang secara khusus mengarahkan implementasi template engine pada penyusunan laporan akademik mahasiswa dengan format baku institusi pendidikan tinggi, terutama pada lingkungan perguruan tinggi vokasi yang umumnya memiliki ketentuan format laporan akhir dan laporan kerja praktik yang bersifat spesifik per jurusan. Integrasi sistem *generate* dokumen dengan kebutuhan format laporan sebagaimana berlaku di Jurusan Bahasa dan Pariwisata juga belum ditemukan pada penelitian-penelitian sebelumnya, sehingga solusi yang tersedia belum dapat diterapkan secara langsung tanpa penyesuaian yang signifikan. Permasalahan ini menjadi perhatian penting mengingat dokumen akademik dengan format baku dan bersifat berulang merupakan salah satu area yang berpotensi besar untuk diefisienkan melalui otomasi berbasis template, tetapi belum banyak dikaji secara khusus dalam penelitian terdahulu.

Berdasarkan permasalahan dan *research gap* tersebut, penelitian ini bertujuan mengimplementasikan *template engine* pada *Document Generator* berbasis web untuk mengotomatisasi penyusunan Laporan Akhir dan Laporan Kerja Praktik mahasiswa. Penelitian ini menguji bahwa penerapan *template engine* dapat meningkatkan efisiensi penyusunan dokumen, yang diukur melalui pengurangan waktu penyusunan minimal 50% dan penurunan *human error* minimal 80% dibandingkan proses manual. Selain itu, keberhasilan fungsi sistem dievaluasi melalui *Black Box Testing*, sedangkan penerimaan pengguna diukur menggunakan UAT dengan target minimal 80%. Tolok ukur tersebut digunakan sebagai dasar kuantitatif untuk mengevaluasi keberhasilan implementasi sistem.

2. METODE PENELITIAN



Gambar 1. Metode *Research and Development* (R&D) dengan model *Prototype*

Penelitian ini menggunakan *metode Research and Development* (R&D) dengan model *Prototype* untuk menghasilkan aplikasi *document generator* berbasis web. Pendekatan R&D sesuai digunakan dalam pengembangan produk perangkat lunak melalui tahapan pengembangan dan pengujian produk [5]. Metode di atas terdiri dari beberapa tahapan antara lain:

1. Studi Literatur

Submitted: 28 Agustus 2026
Published: 30 September 2026

Studi literatur dilaksanakan pada awal penelitian dengan mengumpulkan dan mengkaji berbagai referensi yang berkaitan dengan *template engine*, *document generator*, pengembangan aplikasi berbasis web, serta metode *Research and Development* (R&D). Kajian penelitian terdahulu juga digunakan untuk mengidentifikasi kebutuhan dan peluang pengembangan.

2. Observasi dan Wawancara

Pengumpulan data dilakukan melalui observasi, wawancara, dan kuesioner. Kegiatan dilakukan melalui observasi terhadap proses administrasi dan wawancara dengan staf administrasi serta mahasiswa. Wawancara dilakukan untuk mengidentifikasi kebutuhan pengguna, alur penyusunan laporan, serta kendala dalam proses manual. Responden terdiri atas 27 mahasiswa dan 3 staf administrasi yang dipilih menggunakan *purposive sampling*, dengan kriteria mahasiswa yang menyusun Laporan Akhir atau Laporan Kerja Praktik dan staf yang terlibat dalam pengelolaan dokumen akademik.

3. Analisis Kebutuhan Sistem

Berdasarkan data yang diperoleh, dilakukan analisis kebutuhan sistem untuk mengidentifikasi kebutuhan fungsional dan nonfungsional aplikasi. Tahap ini menghasilkan dokumen spesifikasi kebutuhan sistem (*Software Requirement Specification*/SRS) yang akan menjadi acuan pada proses perancangan.

4. Perancangan Sistem

Perancangan sistem dilakukan dengan mengembangkan *Document Generator* berbasis web yang mengintegrasikan data mahasiswa, data dosen pembimbing, data dosen penguji, data kerja praktik dan data laporan akhir dengan template akademik menggunakan *template engine*. Data seperti NPM, nama mahasiswa, kelas, judul laporan kerja praktik dan laporan akhir, dosen pembimbing, dosen penguji, tempat, ruangan, dan tanggal kegiatan disimpan dalam *database*. Alur integrasi sistem dirancang data pengguna → *database* → pemilihan *template* → *template engine* → rendering dokumen → validasi. Template menggunakan placeholder, seperti {{nama_mahasiswa}}, {{npm}}, dan {{judul_laporan}}, yang secara otomatis digantikan oleh data aktual saat proses *generate*. Mekanisme ini memungkinkan satu template digunakan untuk menghasilkan dokumen mahasiswa yang berbeda secara konsisten [6].

Keluaran sistem berupa dokumen laporan akhir dan laporan kerja praktik dalam format PDF sesuai *template* dan pedoman Jurusan Bahasa dan Pariwisata. Sistem melakukan validasi terhadap kelengkapan data dan memastikan seluruh *placeholder* telah terisi sebelum dokumen dihasilkan. Rancangan tersebut menjadi dasar pengembangan *prototype* dan implementasi *template engine*, yang selanjutnya diuji melalui pengujian fungsional dan *User Acceptance Testing* untuk memastikan kesesuaian sistem dengan kebutuhan pengguna [7].

5. Pengembangan Prototype

Prototype dikembangkan berdasarkan hasil analisis kebutuhan dan rancangan sistem dalam bentuk *Document Generator* berbasis web. Pengembangan meliputi fitur pengelolaan data mahasiswa, dosen pembimbing, dosen penguji, template, input data laporan, serta *generate dokumen*. *Template engine* diimplementasikan untuk menggabungkan data dengan *placeholder* pada template sehingga laporan akhir dan laporan kerja praktik dapat dibentuk secara otomatis. Model *prototype* dipilih karena memungkinkan pengembangan dan penyempurnaan sistem berdasarkan umpan balik pengguna secara bertahap [8]. *Prototype* menghasilkan dokumen laporan dalam format PDF melalui proses pengambilan data, *rendering* template, dan validasi hasil. *Prototype* selanjutnya digunakan sebagai dasar pengujian fungsional dan evaluasi pengguna sebelum dilakukan penyempurnaan sistem [9].

6. Pengujian dan Evaluasi Sistem

Pengujian dilakukan menggunakan *Black Box Testing* untuk memastikan fungsi login, pengelolaan data, *template*, *generate dokumen*, dan ekspor PDF berjalan sesuai kebutuhan sistem.

Submitted: 28 Agustus 2026
Published: 30 September 2026

Evaluasi dilakukan dengan membandingkan proses manual dan sistem berdasarkan waktu penyusunan, kesalahan format, serta keberhasilan generate dokumen. Selanjutnya dilakukan *User Acceptance Testing* (UAT) untuk mengukur penerimaan dan kesesuaian sistem berdasarkan penilaian pengguna [10], [11]. Keberhasilan sistem ditetapkan berdasarkan target penelitian, yaitu fungsi sistem minimal 95%, keberhasilan *generate* minimal 98%, pengurangan waktu penyusunan minimal 50%, penurunan kesalahan minimal 80%, kesesuaian format minimal 95%, dan penerimaan pengguna minimal 80%.

7. Penyempurnaan Sistem

Penyempurnaan sistem dilakukan berdasarkan hasil pengujian dan masukan pengguna dengan memperbaiki fungsi, tampilan, serta kesesuaian hasil *generate* dokumen. Sistem kemudian diuji kembali untuk memastikan hasil perbaikan telah memenuhi kebutuhan pengguna. Proses evaluasi dan perbaikan secara bertahap merupakan bagian dari pengembangan model prototype [12].

8. Penyusunan Luaran Penelitian

Luaran penelitian disusun setelah sistem selesai diuji dan disempurnakan, meliputi aplikasi *Document Generator*, *template* laporan akademik, buku panduan, artikel ilmiah, dan laporan akhir. Dokumentasi sistem dan dokumen hasil generate disiapkan sebagai bagian dari implementasi digitalisasi pengelolaan dokumen [13].

3. HASIL DAN PEMBAHASAN

3.1. Hasil Analisis Kebutuhan Sistem

Metode yang dilakukan antara lain observasi dan wawancara dengan pengguna yaitu staf administrasi dan mahasiswa di Jurusan Bahasa dan Pariwisata. Hasil observasi dan wawancara tersebut menunjukkan bahwa penyusunan laporan akhir dan laporan kerja praktik masih dilakukan secara manual menggunakan aplikasi pengolah kata. Permasalahan utama yang ditemukan meliputi pengisian identitas secara berulang, pengaturan format dokumen secara manual, ketidaksesuaian tata letak, serta waktu penyusunan dan pemeriksaan dokumen yang relatif lama.

Berdasarkan temuan tersebut, dirumuskan kebutuhan sistem berupa pengelolaan data mahasiswa, dosen pembimbing, dosen penguji, template laporan, serta pembuatan dokumen otomatis menggunakan template engine. Sistem juga dirancang menghasilkan dokumen dalam format PDF sehingga format laporan dapat distandarkan. Pendekatan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa pengembangan sistem berbasis web dapat membantu mempercepat pengelolaan informasi dan mendukung kebutuhan pengguna melalui pengujian penerimaan sistem [14].

3.2. Perancangan dan Implementasi Document Generator

3.2.1. Perancangan Sistem

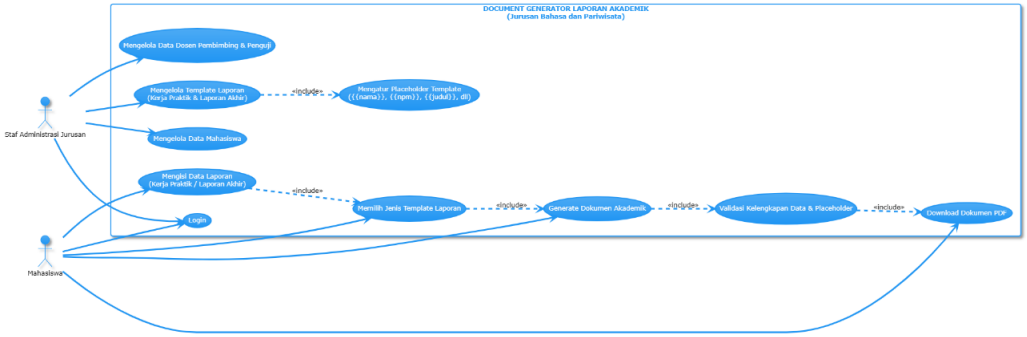
a) Use Case Diagram

Interaksi antara pengguna dengan fungsi-fungsi utama aplikasi dimodelkan menggunakan *Use Case Diagram* yang menunjukkan hak akses dan aktivitas masing-masing pengguna dalam sistem *document generator*. Diagram *use case* yang dirancang dapat dilihat pada Gambar 2.

Aplikasi *Document Generator* memiliki dua aktor utama, yaitu mahasiswa dan staf administrasi. Mahasiswa berfungsi sebagai pengguna yang mengisi data, memilih jenis laporan, dan melakukan *generate* dokumen akademik sesuai *template* yang tersedia. Staf administrasi berfungsi mengelola data mahasiswa dan *template* serta memastikan kesesuaian format dokumen dengan ketentuan akademik. Sistem kemudian

Submitted: 28 Agustus 2026
Published: 30 September 2026

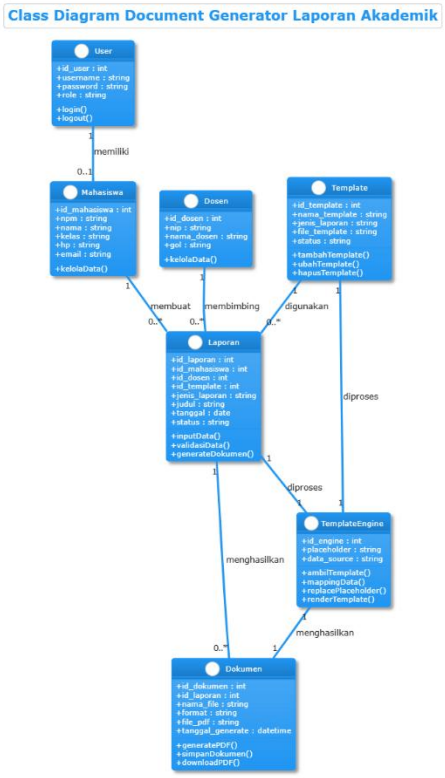
menggabungkan data dengan *template engine* dan menghasilkan dokumen dalam format PDF secara otomatis.



Gambar 2. *Use Case Diagram*

b) Class Diagram

Berdasarkan kebutuhan fungsional yang telah diidentifikasi, selanjutnya dirancang struktur kelas yang merepresentasikan data dan proses utama dalam aplikasi. *Class Diagram* digunakan untuk menggambarkan hubungan antar kelas, atribut, serta operasi yang mendukung pengelolaan data mahasiswa, dosen, *template*, dan dokumen hasil *generate*. Rancangan *class diagram* ditunjukkan pada Gambar 3.

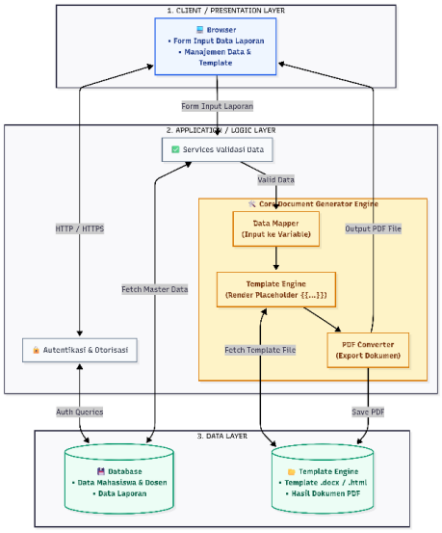


Gambar 3. *Class Diagram*

c) Arsitektur Sistem *Document Generator*

Arsitektur sistem dirancang untuk mengintegrasikan pengguna, aplikasi web, *database*, *template engine*, dan modul *document generator* dalam satu alur pengolahan dokumen. Perancangan arsitektur berfungsi sebagai *blueprint* yang menjadi acuan dalam pengembangan sistem agar komponen data, aplikasi, dan teknologi dapat terintegrasi sesuai kebutuhan [15]. Arsitektur sistem tersebut ditunjukkan pada Gambar 4.

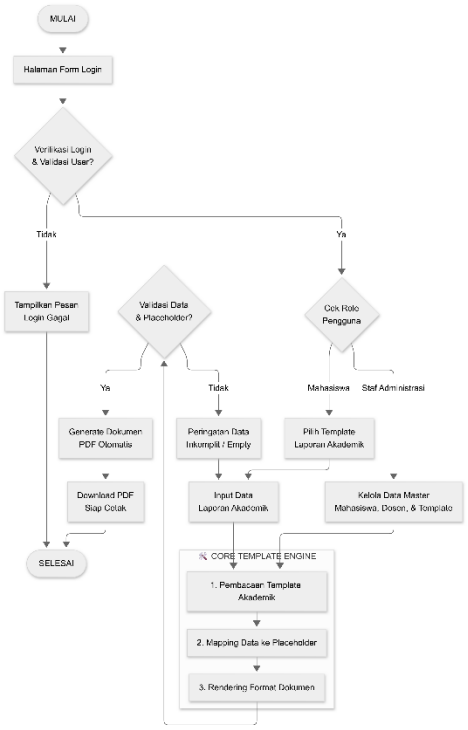
Submitted: 28 Agustus 2026
Published: 30 September 2026



Gambar 4. Arsitektur Sistem

d) Alur Sistem *Generate Document*

Pada tahap perancangan sistem, disusun alur kerja *document generator* berdasarkan kebutuhan fungsional yang telah diperoleh pada tahap analisis. Alur perancangan sistem tersebut menjadi acuan dalam menentukan proses dan fungsi utama aplikasi sebelum masuk ke tahap implementasi. Secara keseluruhan, alur sistem yang dirancang ditunjukkan pada Gambar 5.



Gambar 5. Alur Sistem

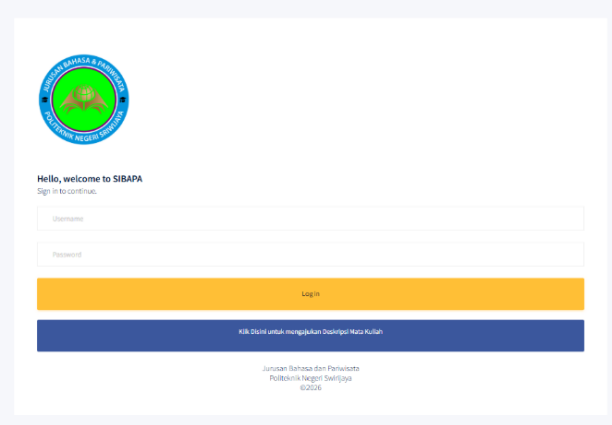
3.2.2. Implementasi Sistem

Implementasi sistem menghasilkan aplikasi *Document Generator* berbasis web untuk membantu penyusunan Laporan Akhir dan Laporan Kerja Praktik mahasiswa Jurusan Bahasa dan Pariwisata. Aplikasi menyediakan pengelolaan data mahasiswa, dosen pembimbing, dosen

Submitted: 28 Agustus 2026
Published: 30 September 2026

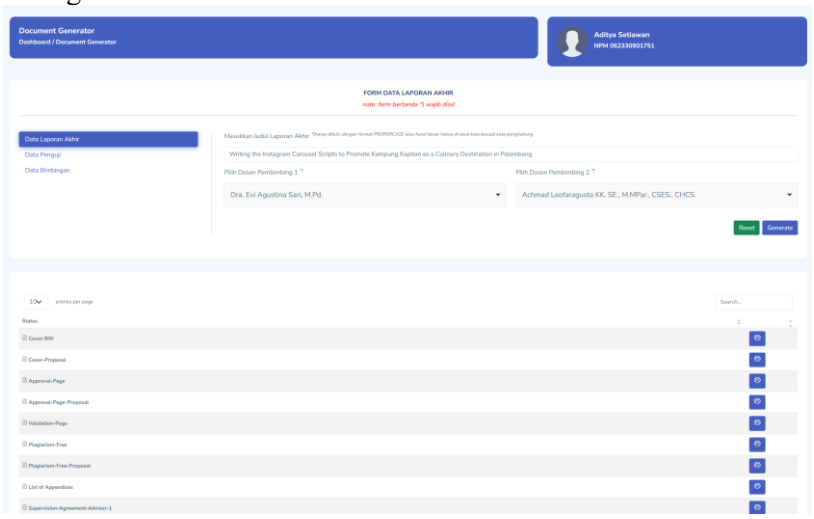
penguji, template, data laporan, serta proses *generate* dokumen. Pendekatan otomatisasi dokumen berbasis web dapat mengurangi proses input berulang dan kesalahan dalam pembuatan dokumen dibandingkan proses manual. Pada proses *generate*, data yang telah dimasukkan pengguna dipetakan ke *placeholder* pada template, kemudian diproses untuk menghasilkan dokumen PDF sesuai format yang ditentukan. Penggunaan aplikasi berbasis web untuk otomatisasi dokumen juga dapat menjaga keseragaman format dan meminimalkan kesalahan dalam dokumen yang dihasilkan [16]. Tampilan halaman aplikasi ditampilkan pada Gambar di bawah ini.

a. Halaman login merupakan halaman yang pertama kali diakses untuk masuk ke *dashboard* Aplikasi.



Gambar 6. Halaman Login

b. Menu *Document Generator* dibuat berdasarkan status semester mahasiswa yang melakukan login. Mahasiswa semester 5 akan memperoleh akses ke *Document Generator* Laporan Kerja Praktik, sedangkan mahasiswa semester 6 akan memperoleh akses ke *Document Generator* Laporan Akhir. Dengan pengaturan ini, mahasiswa hanya menggunakan *generator* sesuai dengan jenis laporan yang harus disusun pada semester tersebut. Document Generator terdiri dari Form Data Laporan Akhir, Data Penguji, dan Data Bimbingan.



Gambar 7. Form Data Laporan Akhir

c. Hasil *Document Generator* disesuaikan dengan kebutuhan dokumen pada masing-masing jenis laporan. Pada Laporan Akhir, sistem menghasilkan beberapa dokumen pendukung, yaitu *cover* laporan, *approval page, validation page, plagiarism free statement, list of appendices, supervision agreement advisor* 1 & 2, *supervision form, note advisor 1 & 2*,

Submitted: 28 Agustus 2026
Published: 30 September 2026

final report recommendation, final report exam revision, dan *implementation final report revision*. Sementara itu, pada Laporan Kerja Praktik, *Document Generator* menghasilkan *cover* laporan, *approval sheet, approval sheet examiner, list of appendices, daily activity log, assessment form, supervision sheet, advisor's notes, seminar recommendation, revision form,* dan *implementation revision form*. Tampilan hasil document generator dapat dilihat pada gambar berikut.



KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN BAHASA DAN PARIWISATA
Jalan Sungai Sahang, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
website : http://www.polsri.ac.id e-mail : info@polsri.ac.id

IMPLEMENTATION OF FINAL REPORT REVISION

The following student.

Name : Aditya Setiawan
NIM : 06233090175 1
Department : Language and Tourism
Final Report Title :
Writing the Instagram Carousel Scripts to Promote Kampung Kapitan as a Culinary Destination in Palembang

Has carried out revisions to the Final Report examined on _____, _____

The implementation of the Final Report revision has been approved by the Examiner Lecturer who provided the revision:

No.	Comments	Examiner Lecturer	Date	Signature
1.		Dr. Welly Ardiansyah, S.S., M.Pd.		
2.		Koryati, S.Pd., M.Pd.		
3.		Wirda Ningsih, S.Pd., M.A.		
4.		Genta Ulina Hutagalung, S.Pd., M.Pd.		

Palembang, ...
Chief Examiner.

Dr. Welly Ardiansyah, S.S., M.Pd.
NIP 197408172006041001

Gambar 8. Tampilan *Implementation of Final Report Revision*



No. Dok. : F-PBM-10 Tgl. Berlaku : 01 November 2024 No. Rev. : 02

KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN BAHASA DAN PARIWISATA
Jalan Sungai Sahang, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
website : http://www.polsri.ac.id e-mail : info@polsri.ac.id

STUDENT PRACTICAL WORK ASSESSMENT FROM COMPANY/INDUSTRY/INSTITUTION

Student Name : Dwi Juniani Saputri
Student ID (NPM) : 062430901742
Department/Study Program : Language and Tourism / D-III English Study
Company/Institution Name : Perum BULOG Kanwil Sumsel & Babel
Unit/Division/Section : Public Relation Division
Company/Institution Address : Jl. Perintis Kemerdekaan No.1, Duku, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30114
Practical Work Duration : 29 June 2026 to 29 September 2026

Number	Assessment Parameters	Assessment Criteria								Total Score**)
		Very Good		Good		Fair		Poor		
1	Ethics	10	9	8	7	6	5	4	3	
2	Discipline	10	9	8	7	6	5	4	3	
3	Self-confidence	10	9	8	7	6	5	4	3	
4	Cooperation	10	9	8	7	6	5	4	3	
5	Motivation	10	9	8	7	6	5	4	3	
6	Working Initiative	10	9	8	7	6	5	4	3	
7	Loyalty	10	9	8	7	6	5	4	3	
8	Work Responsibility	10	9	8	7	6	5	4	3	
9	Understanding and Ability in Performing and Completing Tasks	10	9	8	7	6	5	4	3	
10	Occupational Health and Safety	10	9	8	7	6	5	4	3	
										Total Score

*) Circle the appropriate rating number.
**) Write down the score according to the rating number.

Palembang, 31st July 2026

Acknowledged by,
Company/Institution Leader

Company/Institution Supervisor

Ihsan
ID -

Herli Iswanto
ID -

Note:
Please send this assessment form to Politeknik Negeri Sriwijaya as soon as the student has completed the Practical Work.

Gambar 9. Tampilan *Assessment Form*

Hasil implementasi menunjukkan bahwa data yang dimasukkan melalui aplikasi dapat dipetakan secara otomatis ke dalam *template* yang telah ditentukan. Proses tersebut menghasilkan dokumen PDF dengan struktur dan format yang telah disesuaikan dengan pedoman laporan akademik di Jurusan Bahasa dan Pariwisata.

3.3. Pengujian Sistem

Pengujian dilakukan untuk memastikan setiap fungsi pada *Document Generator* berjalan sesuai dengan kebutuhan yang telah ditetapkan. Pengujian terdiri atas *Black Box Testing* untuk memeriksa fungsi sistem berdasarkan input dan output yang dihasilkan serta *User Acceptance Testing* (UAT) untuk mengetahui tingkat penerimaan sistem oleh pengguna.

Pengujian dilakukan terhadap prototype *Document Generator* Laporan Akademik Mahasiswa untuk memastikan fungsi sistem, proses template rendering, dan keluaran dokumen berjalan sesuai kebutuhan. Selain pengujian fungsional, dilakukan evaluasi kuantitatif dengan membandingkan penyusunan laporan akhir dan laporan kerja praktik secara manual dengan menggunakan sistem. Pengujian menggunakan minimal 30 percobaan dokumen, dengan parameter waktu penyusunan, jumlah kesalahan data/format, keberhasilan *generate*, dan kesesuaian format. Selanjutnya dilakukan *User Acceptance Testing* (UAT) kepada minimal 30 pengguna yang terdiri dari mahasiswa dan pengguna administrasi untuk menilai kemudahan penggunaan, kesesuaian fungsi, dan manfaat sistem.

3.3.1. Black Box Testing

Black Box Testing dilakukan dengan memberikan input pada setiap fungsi utama sistem tanpa melihat kode program yang digunakan. Hasil *black box testing* dapat dilihat pada Tabel 1 di bawah ini.

Submitted: 28 Agustus 2026
Published: 30 September 2026

Tabel 1. Tabel Hasil *Blackbox Testing*

No.	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Login	Mahasiswa memasukkan username dan password yang benar	Sistem berhasil masuk ke halaman utama	Berhasil
2	Login	Mahasiswa memasukkan username/password yang salah	Sistem menolak login dan menampilkan pesan kesalahan	Berhasil
3	Menu *Document Generator*	Mahasiswa semester 5 login	Sistem menampilkan *generator* Laporan Kerja Praktik	Berhasil
4	Menu *Document Generator*	Mahasiswa semester 6 login	Sistem menampilkan *generator* Laporan Akhir	Berhasil
5	Template Form Laporan Akhir	Mahasiswa mengisi data laporan akhir, data pembimbing, data penguji	Sistem menyimpan semua data yang diinput	Berhasil
6	Template Form Kerja Praktik	Mahasiswa mengisi data laporan KP, data penilaian perusahaan, data penguji	Sistem menyimpan semua data yang diinput	Berhasil
7	Generate Laporan Akhir	Mahasiswa menjalankan generate data Laporan Akhir	Sistem menghasilkan seluruh dokumen sesuai template	Berhasil
8	Generate Laporan Kerja Praktik	Mahasiswa menjalankan generate data Laporan KP	Sistem menghasilkan seluruh dokumen sesuai template	Berhasil
9	*Placeholder*	Data pengguna diproses ke dalam *template*	*Placeholder* terisi dengan data yang sesuai	Berhasil
10	Export PDF	Pengguna melakukan generate dokumen	Dokumen dapat dihasilkan dalam format PDF	Berhasil
11	Data tidak lengkap	Pengguna melakukan generate tanpa melengkapi data wajib	Sistem memberikan peringatan dan tidak menghasilkan dokumen yang belum lengkap	Berhasil

3.3.2. *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap aplikasi *Document Generator* Laporan Akademik Mahasiswa. Pengujian melibatkan 30 pengguna yang terdiri dari mahasiswa dan staf administrasi jurusan. Pengujian dilakukan berdasarkan skenario penggunaan sistem dan penilaian terhadap aspek kemudahan penggunaan, kesesuaian fungsi, kecepatan proses, kualitas dokumen, dan manfaat sistem.

Tabel 2. Tabel Hasil *User Acceptance Testing*

Kategori UAT	Jumlah Pengguna	Diterima	Acceptance Rate
Mahasiswa	27	27	100%
Staf Administrasi Jurusan	3	3	100%
Jumlah	30	30	100%

Selanjutnya, penilaian pengguna dilakukan menggunakan skala Likert 1–5 terhadap lima aspek utama sistem.

Tabel 3. Hasil Penilaian Pengguna (Skala 1-5)

Aspek	Rata – Rata	Persentase
Kemudahan Penggunaan	4,70	94,00%
Kesesuaian Fungsi	4,77	95,33%
Kecepatan Proses	4,70	94,00%
Kualitas Dokumen	4,60	92,00%
Manfaat Sistem	4,70	94,00%
Rata-rata	4,69	93,87%

Hasil penelitian ini menunjukkan bahwa implementasi *template engine* pada aplikasi Document Generator memiliki karakteristik yang lebih spesifik dibandingkan penelitian terdahulu dalam hal penanganan dokumen dengan format yang bersifat baku (rigid document format). Haliza dkk. mengembangkan sistem otomasi *template* dokumen berbasis website yang menunjukkan kemampuan dalam mempercepat proses pengolahan dokumen dibandingkan metode manual. Namun, sistem tersebut diterapkan pada otomasi dokumen secara umum dan

Submitted: 28 Agustus 2026
Published: 30 September 2026

belum secara khusus diarahkan pada dokumen akademik yang memiliki struktur, tata letak, identitas, dan komponen dokumen yang harus mengikuti pedoman institusi. Sementara itu, Dong mengembangkan sistem document generation berbasis web yang menekankan aspek arsitektur dan proses pembentukan dokumen secara otomatis. Penelitian tersebut menunjukkan manfaat penggunaan pendekatan berbasis template dalam meningkatkan efisiensi document generation, tetapi belum secara spesifik membahas penerapan pada dokumen akademik dengan aturan format yang ketat dan berulang. Penelitian Sun dkk. juga menunjukkan pemanfaatan *template engine* untuk menghasilkan tampilan data secara dinamis, tetapi konteks penerapannya berfokus pada visualisasi dan penyajian dataset, bukan pada pembentukan dokumen administratif yang memiliki ketentuan format tetap.

Berbeda dengan penelitian tersebut, penelitian ini secara khusus mengimplementasikan *template engine* untuk menghasilkan Laporan Akhir dan Laporan Kerja Praktik mahasiswa yang memiliki struktur dan format baku sesuai pedoman Jurusan Bahasa dan Pariwisata. Sistem tidak hanya menggantikan data pada *placeholder*, tetapi juga mengintegrasikan data mahasiswa, dosen pembimbing, dosen penguji, data laporan, dan informasi kegiatan ke dalam *template* yang telah ditetapkan. Setiap jenis dokumen memiliki struktur dan komponen yang telah ditentukan, sehingga pengguna tidak perlu melakukan pengaturan tata letak secara manual. Mekanisme validasi kelengkapan data dan pemeriksaan pengisian *placeholder* sebelum proses *generate* juga membantu mencegah terbentuknya dokumen yang belum lengkap. Dengan demikian, kontribusi utama penelitian ini terletak pada penerapan *template engine* dalam konteks *rigid* document format, yaitu dokumen yang tidak hanya membutuhkan otomatisasi pengisian data, tetapi juga membutuhkan konsistensi struktur, tata letak, identitas, dan komponen dokumen sesuai standar akademik tertentu.

Hasil pengujian semakin memperkuat karakteristik tersebut. Black Box Testing menunjukkan bahwa seluruh 11 skenario fungsi berhasil dijalankan sesuai hasil yang diharapkan, sedangkan penilaian pengguna melalui UAT memperoleh rata-rata 4,69 dari skala 5 atau 93,87%. Aspek kesesuaian fungsi memperoleh nilai 95,33% dan kualitas dokumen sebesar 92,00%. Temuan tersebut menunjukkan bahwa sistem tidak hanya berfungsi sebagai pembangkit dokumen otomatis, tetapi juga mampu mempertahankan konsistensi format dokumen akademik sesuai kebutuhan pengguna. Oleh karena itu, dibandingkan penelitian terdahulu yang lebih berfokus pada otomasi dokumen secara umum, arsitektur document generation atau penyajian data berbasis template, penelitian ini memberikan kontribusi yang lebih terarah pada otomasi dokumen akademik dengan format yang *rigid* dan terstandar pada lingkungan perguruan tinggi vokasi.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *template engine* pada *Document Generator* berbasis web untuk mengotomatisasi penyusunan Laporan Akhir dan Laporan Kerja Praktik mahasiswa sesuai format akademik yang ditetapkan. Hasil *Black Box Testing* menunjukkan bahwa seluruh fungsi yang diuji dapat berjalan dengan baik. Berdasarkan pengukuran terhadap 30 responden, penggunaan *Document Generator* mengurangi waktu penyusunan dari rata-rata 15 menit menjadi 5 menit atau sebesar 66,67%, serta mengurangi *human error* dari 10 menjadi 2 kesalahan atau sebesar 80%. Hasil tersebut memenuhi tolok ukur penelitian berupa pengurangan waktu minimal 50% dan penurunan *human error* minimal 80%. Sementara itu, hasil UAT sebesar 93,87% menunjukkan tingkat penerimaan pengguna terhadap sistem dan digunakan sebagai indikator persepsi pengguna, bukan sebagai ukuran efisiensi. Dengan demikian, penerapan *template engine* mendukung proses penyusunan dokumen akademik yang lebih cepat, konsisten, dan sesuai dengan format yang telah ditetapkan.

5. SARAN

Berdasarkan hasil implementasi sistem, terdapat beberapa saran yang dapat digunakan untuk perbaikan dan pengembangan lebih lanjut, yaitu :

Submitted: 28 Agustus 2026
Published: 30 September 2026

1. Memperluas jenis dokumen akademik yang dapat dihasilkan, menambahkan fitur pengelolaan dan pembaruan *template* secara mandiri oleh staf administrasi jurusan.
2. Mengembangkan fitur integrasi dengan sistem informasi akademik yang sudah ada agar data mahasiswa dan dosen dapat diperoleh secara otomatis tanpa penginputan berulang.
3. Menguji dengan jumlah pengguna dan dokumen yang lebih besar serta pada jurusan lain perlu dilakukan untuk menguji skalabilitas dan memastikan sistem dapat diterapkan secara lebih luas di lingkungan perguruan tinggi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada P3M Politeknik Negeri Sriwijaya karena telah memberikan dukungan finansial terhadap penelitian ini.

DAFTAR PUSTAKA

[1] J. Dinkelbach, L. Razik, M. Mirz, A. Benigni, and A. Monti, "Template-based generation of programming language specific code for smart grid modelling compliant with CIM and CGMES," *The Journal of Engineering*, vol. 2023, no. 1, e12208, 2023, doi: **10.1049/tje2.12208**.

[2] Haliza, S. N., Ainnur, A., & Effendi, I. J. (2024). Design and Build a Website-Based Word Template Automation System using the Prototype Method. International Journal of Information System and Technology, 8(1), 45–53. IJISTECH Journal.

[3] Dong, S. (2026). Research on the Design of Document Generation System based on Web. Open Access Library Journal, 12(3), 1–10. Open Access Library Journal.

[4] Sun, Q., Micorescu, M., Ferreira, P., & Ouellette, B. F. F. (2022). STENCIL: A web templating engine for visualizing and sharing life science datasets. PLoS Computational Biology, 18(3), e1009859. PLoS Computational Biology_).

[5] Wajdi, M. F., & Tandililing, M. (2022). Implementasi metode Research and Development (R&D) pada aplikasi pengelolaan arsip. *Jurnal Informatika Progres, 14*(1), 33–40. https://doi.org/10.56708/progres.v14i1.320

[6] Chaudhary, R., Singh, K., Yadav, A., & Chandrevanshi, T. (2025). *Intelligent document automation: Integrating LLMs for dynamic template generation*. International Journal for Research in Applied Science and Engineering Technology. https://doi.org/10.22214/ijraset.2025.75203

[7] Puspita, M. A., Mardiana, M., Ariesta, R., & Muhammad, M. A. (2025). User acceptance testing dan performance testing pada pengembangan website Kanal Pengetahuan Dikti. *JIEET (Journal of Information Engineering and Educational Technology), 8*(2), 96–101. https://doi.org/10.26740/jieet.v8n2.p96-101

[8] Ramadhani, S. R., & Afriyansyah, M. F. (2024). Optimalisasi dan Evaluasi *User Acceptance* Dan *Usability* dengan Penerapan Prototyping pada Aplikasi Manajemen Produk dan Pesanan. *Jurnal Komputer Terapan, 10*(2), 131–141. https://doi.org/10.35143/jkt.v10i2.6421.

[9] Nifratama, A., Suratno, T., & Arsa, D. (2024). Analisis dan Evaluasi Pengujian pada Penerapan Metode Prototype dalam Software Engineering. *Jurnal Ilmiah Media Sisfo, 18*(1), 128–138. https://doi.org/10.33998/mediasisfo.2024.18.1.1649

[10] Wahyudi, I., Fahrullah, F., Alameka, F., & Haerullah, H. (2023). Analisis Blackbox Testing dan User Acceptance Testing terhadap Sistem Informasi Solusimedsosku. *Jurnal Teknosains Kodepena, 4*(1), 1–9. https://doi.org/10.54423/jtk.v4i1.54.

[11] Jibril, M., Zulrahmadi, & Amin, M. (2024). Pengujian Sistem Informasi E-Modul pada SMPN 1 Tempuling Menggunakan Black Box Testing. *Jurnal Perangkat Lunak, 6*(2), 327–332. https://doi.org/10.32520/jupel.v6i2.3326

Submitted: 28 Agustus 2026
Published: 30 September 2026

[12] Zulhajji, Z., Wahid, A., & Ikhsan, N. (2024). Pengembangan Prototype Sistem Peringatan Dini Bencana Banjir Berbasis Android. *Jurnal MediaTIK, 7*(2), 44–49. https://doi.org/10.59562/mediatik.v7i2.2226

[13] Khoirunisa, & Lestari, Y. (2024). Pengelolaan e-Archives melalui Document Management System berbasis website di PT. Bridgestone Tire Indonesia. *Jurnal Inovasi Administrasi Negara Terapan, 2*

[14] Wibowo, A., & Darwati, I. (2022). Perancangan sistem informasi pengelolaan sarana dan prasarana dengan pengujian User Acceptance Testing. *Reputasi: Jurnal Rekayasa Perangkat Lunak, 3*(1), 1–6.

[15] Widiadnyana, P., & Andreyana, P. V. (2023). Perancangan arsitektur TI untuk sistem informasi akademik universitas menggunakan Zachman Framework. *JITTER: Jurnal Ilmiah Teknologi dan Komputer, 4*(2), 1903–1911. https://doi.org/10.24843/JTRTI.2023.v04.i02.p22

[16] Darman, R. (2025). Pengembangan aplikasi berbasis web untuk pengelolaan data tekstual dan otomatisasi dokumen pertimbangan teknis pertanahan. *Jurnal Pertanahan, 15*(2), 121–141. https://doi.org/10.53686/jp.v15i2.296

Submitted: 28 Agustus 2026
Published: 30 September 2026