

Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)

DOI: <https://doi.org/10.35870/jtik.v10i3.6122>

Perancangan Sistem Informasi Alumni Menggunakan Pendekatan OOP untuk Meningkatkan Konektivitas Lulusan dan Efisiensi Administrasi Prodi Teknologi Rekayasa Perangkat Lunak

Alma' Abdhillah¹, Raden Mas Galih Pradityo², Muhammad Ramadhan³, Dika Aprilio Wibowo⁴, Aditya Wicaksono^{5*}, Endang Purnama Giri⁶

^{1,2,3,4,5*} Program Studi Teknologi Rekayasa Perangkat Lunak, Sekolah Vokasi, IPB University, Kabupaten Bogor, Provinsi Jawa Barat, Indonesia.

⁶ Program Studi Ilmu Komputer, Sekolah Sains Data, Matematika, dan Informatika, IPB University, Kabupaten Bogor, Provinsi Jawa Barat, Indonesia.

article info

Article history:

Received 15 December 2025

Received in revised form

20 January 2026

Accepted 1 February 2026

Available online July 2026.

Keywords:

Alumni Information System;

System Design; Object-

Oriented Programming;

Object-Oriented Analysis and

Design.

Kata Kunci:

Sistem Informasi Alumni;

Perancangan Sistem;

Pemrograman Berorientasi

Objek; Object-Oriented

Analysis and Design.

abstract

The management of alumni data for the Software Engineering Technology (SET) Study Program is currently facing challenges, including dispersed data, difficulty in updating, and administrative inefficiencies. This study aims to design a structured and centralized web-based Alumni Information System. The method used is Object-Oriented Analysis and Design (OOAD), which includes literature review, requirements identification, UML modeling (use case, activity, class, sequence diagrams), and interface design. The design results indicate that the system provides nine main functional features, which address the data management problems. The Object Oriented Programming (OOP) approach is applied to produce a system architecture that is modular, maintainable, and scalable. The designed system is estimated to increase administrative process efficiency by more than 90%, instantly reducing data search and verification time. Overall, this design is expected to enhance data accuracy, alumni connectivity, and operational efficiency of the study program.

abstrak

Pengelolaan data alumni Program Studi Teknologi Rekayasa Perangkat Lunak (TPL) saat ini menghadapi tantangan data yang tersebar, sulit diperbarui, dan inefisiensi administratif. Penelitian ini bertujuan merancang Sistem Informasi Alumni berbasis web yang terstruktur dan terpusat. Metode yang digunakan adalah Object-Oriented Analysis and Design (OOAD), mencakup studi literatur, identifikasi kebutuhan, pemodelan UML (use case, activity, class, sequence diagram), dan perancangan antarmuka. Hasil perancangan menunjukkan sistem menyediakan sembilan fitur fungsional utama yang mengatasi masalah pendataan. Pendekatan Object Oriented Programming (OOP) diterapkan untuk menghasilkan arsitektur sistem yang modular, maintainable, dan scalable. Sistem yang dirancang diestimasikan mampu meningkatkan efisiensi proses administratif hingga lebih dari 90%, memangkas waktu pencarian dan verifikasi data secara instan. Secara keseluruhan, perancangan ini diharapkan dapat meningkatkan akurasi data, konektivitas alumni, dan efisiensi operasional program studi.

Corresponding Author. Email: adityawicaksono@apps.ipb.ac.id ^{5}.

Copyright 2026 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. 

1. Pendahuluan

Transformasi teknologi informasi secara fundamental telah mengubah paradigma lembaga pendidikan dalam mengelola relasi dengan para lulusannya. Institusi pendidikan kini menjaga keterikatan dengan alumni melalui berbagai inisiatif, seperti pengadaan agenda pertemuan, penyediaan jejaring profesional, hingga program pendampingan karir (Occiano dkk., 2023). Pembangunan basis data alumni berbasis web menjadi instrumen krusial untuk menjamin aksesibilitas informasi, mengingat adopsi sistem informasi yang tepat berbanding lurus dengan peningkatan produktivitas operasional (Rahinda dkk., 2024). Implementasi platform digital mampu mempercepat distribusi data serta menyederhanakan birokrasi internal (Rahinda dkk., 2024). Secara fungsional, keberadaan sistem ini diharapkan mampu mengoptimalkan efektivitas pengumpulan data serta memfasilitasi pertukaran informasi mengenai agenda institusi dan peluang profesional (Ningsih & Mustakim, 2019). Selain itu, kebutuhan lulusan terhadap validasi administratif, seperti legalisasi dokumen akademik, menjadi faktor pendorong utama dalam menyediakan layanan yang responsif (Madhani dkk., 2022).

Meskipun urgensinya sangat nyata, masih banyak program studi yang belum memiliki infrastruktur pengelolaan alumni yang memadai. Kondisi tersebut memicu risiko terputusnya keterikatan antara lulusan dan almamater. Tanpa adanya kanal komunikasi yang terintegrasi, potensi hilangnya kontak antar alumni menjadi sangat besar akibat minimnya akses informasi (Izzati, 2023). Sebaliknya, riset membuktikan bahwa platform berbasis web efektif dalam mendorong kolaborasi dan memperkuat komunikasi dua arah (Anthony Jr., 2021). Tanpa adanya sistem yang terpusat, data alumni sering kali tersebar di berbagai kanal non-formal yang tidak terorganisir, sehingga menyulitkan program studi dalam memetakan profil lulusan maupun memberikan dukungan karir yang relevan. Berdasarkan identifikasi awal pada Program Studi Teknologi Rekayasa Perangkat Lunak, validasi data alumni secara manual saat ini memerlukan waktu 3–5 hari kerja, terutama saat persiapan pelaporan akreditasi. Selain itu, mayoritas data tidak diperbarui dalam kurun waktu dua tahun terakhir, yang

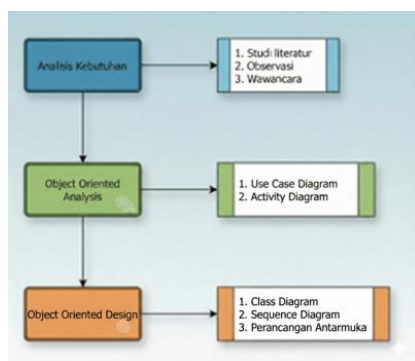
berimplikasi pada rendahnya akurasi informasi serta optimalisasi pemanfaatan profil alumni. Fenomena tersebut menegaskan kebutuhan akan sistem yang terintegrasi, sejalan dengan konsep *Alumni Tracking System* sebagai basis data efisien untuk pemantauan karir dan penguatan reputasi lembaga (Patil dkk., 2025). Beberapa institusi telah mengadopsi platform manajemen komersial seperti Graduway atau Almbase. Namun, solusi tersebut umumnya memerlukan biaya langganan yang tinggi dan dirancang untuk skala universitas besar, sehingga kurang efisien jika diterapkan pada level program studi. Selain itu, fitur yang tersedia sering kali terlalu umum dan belum sepenuhnya mengakomodasi kebutuhan spesifik, terutama dalam mendukung pelaporan internal dan standar akreditasi nasional. Oleh karena itu, diperlukan solusi alternatif berupa sistem yang dirancang secara spesifik sesuai karakteristik operasional program studi. Guna menjamin keberlanjutan sistem, pembangunan arsitektur perangkat lunak harus didasarkan pada struktur yang kokoh. Dalam pengembangan sistem modern, paradigma *Object Oriented Programming* (OOP) menjadi pilihan utama karena mendukung atribut kritis seperti *maintainability* dan *reusability* melalui prinsip polimorfisme, pewarisan (*inheritance*), enkapsulasi, dan abstraksi (Magableh dkk., 2024).

Penerapan OOP terbukti mampu meningkatkan modularitas dan skalabilitas sistem, memungkinkan pengembang membangun struktur perangkat lunak yang lebih terorganisir dan adaptif (Magableh dkk., 2024). Berbagai upaya pengembangan sistem alumni telah dilakukan untuk merespons tantangan ini. Kezia dkk. (2025) menawarkan solusi mutakhir melalui sistem pelacakan karir yang mengintegrasikan algoritma *Graph Network* untuk memvisualisasikan jejaring alumni secara interaktif. Di sisi lain, Ningsih & Mustakim (2019) lebih menitikberatkan pada pemenuhan kebutuhan fungsional dasar pendataan di tingkat program studi. Meskipun riset sebelumnya telah mengadopsi teknologi modern dan menerapkan metode OOP untuk mendukung modularitas kode, fokus utama studi tersebut umumnya terletak pada fitur visualisasi atau analisis koneksi (Kezia dkk., 2025). Terdapat celah di mana aspek arsitektur perangkat lunak sering kali hanya diposisikan sebagai pendukung, bukan sebagai fokus utama untuk menjamin keberlanjutan sistem administratif jangka

panjang. Penelitian ini hadir untuk mengisi ruang tersebut dengan menitikberatkan pada perancangan arsitektur sistem yang tangguh menggunakan paradigma OOP, guna memastikan perangkat lunak memiliki tingkat *maintainability* dan *reusability* yang tinggi. Penelitian ini bertujuan untuk merancang sistem informasi alumni berbasis web dengan menerapkan prinsip OOP guna menghasilkan arsitektur yang modular dan skalabel. Fokus utama adalah menyediakan platform yang mampu memenuhi standar administrasi program studi serta memfasilitasi komunikasi dua arah yang terstruktur. Melalui implementasi prinsip-prinsip tersebut, sistem yang dikembangkan diharapkan mampu mengatasi kendala data yang tersebar dan memperkuat hubungan berkelanjutan antara lulusan dengan almamater. Fitur utama yang dikembangkan mencakup manajemen profil, distribusi informasi kegiatan, serta penyediaan jejaring profesional, yang diharapkan memberikan dampak praktis bagi institusi dalam membangun infrastruktur digital yang efektif.

2. Metodologi Penelitian

Penelitian ini menggunakan pendekatan pengembangan sistem berbasis *Object-Oriented Programming* (OOP), di mana analisis dan perancangan sistem informasi alumni ini berlandaskan pada metodologi *Object-Oriented Analysis and Design* (OOAD). Metodologi ini dipilih karena mampu menghasilkan perancangan sistem yang terstruktur, modular, dan mudah dikembangkan. Proses penelitian terbagi ke dalam beberapa tahapan sistematis yang meliputi pengumpulan data, analisis kebutuhan, pemodelan berorientasi objek, serta perancangan arsitektur sistem.



Gambar 1. Metode Penelitian

Analisis Kebutuhan

Sebelum memasuki fase perancangan, langkah fundamental yang perlu dilaksanakan adalah analisis kebutuhan guna menetapkan spesifikasi layanan serta standar kualitas yang wajib dicapai oleh sistem. Kebutuhan fungsional menjelaskan apa saja layanan yang harus diberikan oleh perangkat lunak, sementara kebutuhan non-fungsional menetapkan standar kualitas yang harus menyertai layanan tersebut.

Studi Literatur

Studi Literatur dilakukan untuk memperoleh landasan teoritis dalam mengembangkan sistem berbasis *website* dan memahami konsep-konsepnya, seperti *Object Oriented Programming* (OOP) dan penerapan *Object Oriented Analysis and Design* (OOAD) yang mendukung modularitas dan efisiensi pemeliharaan. Penggunaan OOP dalam sistem informasi berbasis web meningkatkan modularitas program, memudahkan pemeliharaan, dan menghasilkan efisiensi dalam pengembangan aplikasi *website*.

Observasi

Observasi dilakukan untuk memetakan alur kerja serta mengamati kondisi nyata pengelolaan data alumni di Program Studi Teknologi Rekayasa Perangkat Lunak. Proses observasi dilaksanakan selama kurang lebih dua minggu di lingkungan ruang administrasi program studi, dengan fokus pada alur kerja pendataan alumni yang masih dilakukan secara manual. Hasil observasi menunjukkan bahwa data alumni tersebar pada berbagai dokumen dan platform informal, sehingga tidak mendukung kebutuhan pendataan formal dan pelaporan terstruktur.

Wawancara

Wawancara dilakukan secara langsung terhadap Sekretaris Prodi sebagai pemangku kepentingan utama sistem. Wawancara bertujuan untuk mengidentifikasi kebutuhan sistem, permasalahan yang dihadapi, serta ekspektasi terhadap sistem informasi alumni yang akan dirancang. Hasil wawancara menunjukkan kebutuhan fitur inti seperti registrasi dan autentikasi alumni, pembaruan profil, pencarian data alumni, serta pengelolaan konten informasi oleh admin.

Object Oriented Analysis

Object Oriented Analysis (OOA) merupakan tahap dalam pengembangan sistem di mana kebutuhan dan elemen utama sistem diidentifikasi lalu dimodelkan sebagai objek-objek yang merepresentasikan entitas dunia nyata. Fokus OOA adalah menggambarkan peran, atribut, perilaku, serta hubungan antar objek untuk memenuhi kebutuhan pengguna yang telah terdefinisi. Teknik OOA umumnya menggunakan diagram seperti *use case* untuk memetakan interaksi antara pengguna dan sistem, sehingga seluruh kebutuhan sistem terwakili secara jelas dan sistematis.

Object Oriented Design

Object Oriented Design (OOD) adalah proses perancangan solusi perangkat lunak yang menekankan kolaborasi antar objek, termasuk atribut, metode, dan hubungan antar objek untuk memenuhi kebutuhan yang diidentifikasi pada tahap analisis. Pada tahap ini, model konseptual dari OOA diterjemahkan menjadi rancangan teknis berupa *class diagram*, rancangan antarmuka, serta skema *database* atau struktur data, sesuai dengan *platform* target yang digunakan.

Untuk memastikan kesesuaian rancangan dengan kebutuhan pengguna dan prinsip OOP, dilakukan validasi rancangan dengan melibatkan sekretaris prodi rekayasa perangkat lunak. Validasi difokuskan pada evaluasi kelengkapan fitur, konsistensi penerapan prinsip OOP, serta keterpaduan antar diagram dan rancangan antarmuka. Masukan dari para ahli digunakan sebagai dasar penyempurnaan rancangan sistem.

3. Hasil dan Pembahasan

Hasil

Analisis Kebutuhan Sistem

Identifikasi persyaratan pengguna dilakukan secara sistematis untuk memastikan bahwa sistem yang dikembangkan mampu mengatasi hambatan utama dalam pengelolaan data alumni. Platform ini memegang peranan strategis dalam mengoptimalkan akurasi pengolahan data serta menyediakan akses informasi akademik yang terstruktur. Penentuan prioritas fitur dilakukan melalui metode MoSCoW (*Must have*, *Should have*, *Could have*, *Won't have*). Pendekatan ini menjamin bahwa kebutuhan dengan dampak signifikan terhadap operasional program studi mendapatkan perhatian utama. Fitur manajemen data alumni, autentikasi pengguna, dan fungsi pencarian ditetapkan sebagai kategori *Must have*. Sementara itu, fitur pendukung seperti pengelolaan konten publik dan kendali privasi diklasifikasikan sebagai *Should have* atau *Could have* berdasarkan tingkat urgensi fungsional bagi pengguna.

Kebutuhan Fungsional

Persyaratan fungsional menjadi landasan dalam menentukan rangkaian layanan dan prosedur teknis agar sistem beroperasi secara optimal. Kebutuhan ini berkaitan langsung dengan spesifikasi input, proses, dan output yang menjadi syarat utama dari pengguna sistem (Iskandar & Ratnasari, 2021). Fokus identifikasi diarahkan untuk meniadakan kendala pendataan manual yang selama ini menghambat produktivitas program studi.

Tabel 1. Daftar Kebutuhan Fungsional

ID	Nama Kebutuhan	Prioritas
FR-01	Autentikasi Login Admin	Tinggi
FR-02	Registrasi Alumni berdasarkan NIM	Tinggi
FR-03	Manajemen Profil Alumni	Tinggi
FR-04	Direktori dan Pencarian Alumni	Tinggi
FR-05	Dashboard Admin	Tinggi
FR-06	Ekspor Data	Tinggi
FR-07	Manajemen Konten Publik	Sedang
FR-08	Kontrol Privasi	Sedang
FR-09	Manajemen Multi-Admin	Rendah

Kebutuhan Non Fungsional

Definisi mengenai batasan, atribut, dan parameter kualitas yang ada pada sistem, di luar aspek fitur fungsionalnya, tercakup dalam lingkup kebutuhan

non-fungsional. Kebutuhan ini krusial untuk memastikan sistem dapat diandalkan, aman, efisien, dan berkelanjutan.

Tabel 2. Daftar Kebutuhan Non Fungsional

ID	Nama Kebutuhan	Priotitas
NFR-01	Keamanan	Tinggi
NFR-02	Kinerja	Tinggi
NFR-03	Kegunaan	Tinggi
NFR-04	Kompatibilitas	Tinggi
NFR-05	Desain Responsif	Tinggi
NFR-06	Branding	Sedang
NFR-07	Ketersediaan	Sedang
NFR-08	Skalabilitas	Sedang

Object Oriented Analysis

Object Oriented Analysis (OOA) berfokus pada identifikasi dan pemodelan objek serta interaksinya untuk mendefinisikan kebutuhan sistem secara mendalam. Pendekatan ini dipilih karena memberikan representasi entitas yang natural, mendukung prinsip OOP, serta memudahkan pemeliharaan dan pengembangan sistem. Analisis ini divisualisasikan melalui *Use Case Diagram* dan *Activity Diagram* yang menjadi landasan utama bagi perancangan arsitektur dan implementasi perangkat lunak.

Use Case Diagram

Hubungan antara pengguna (aktor) dan fitur atau kemampuan yang ditawarkan oleh suatu sistem digambarkan dalam diagram kasus penggunaan. *Use case diagram* berfungsi sebagai representasi visual untuk memetakan peran pengguna terhadap setiap fitur sistem informasi manajemen, sehingga memperjelas batasan dan wewenang pengguna dalam sistem.

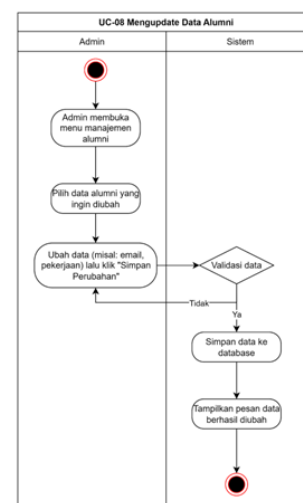


Gambar 2. *Use Case Diagram*

Use Case Diagram pada Gambar 2 memisahkan hak akses secara tegas antara Admin, Alumni, dan Pengunjung untuk menjaga keamanan data dan prinsip *Least Privilege*, namun tetap terintegrasi dalam satu basis data objek. Pemisahan ini krusial untuk memastikan bahwa data sensitif hanya diakses oleh Admin.

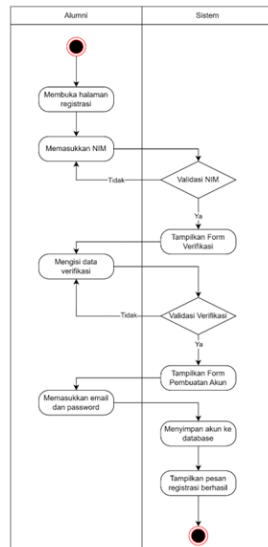
Activity Diagram

Activity diagram menggambarkan urutan aktivitas yang dilakukan oleh pengguna saat berinteraksi dengan *website*. Diagram ini fokus pada urutan aktivitas dan keputusan-keputusan yang terjadi selama proses berlangsung.



Gambar 3. *Activity Diagram* Memperbarui Data Alumni

Gambar 3 diatas menunjukkan *Activity Diagram* Memperbarui Data Alumni. Diagram ini menggambarkan bagaimana admin berinteraksi dengan sistem untuk mengelola data alumni secara terstruktur.



Gambar 4. *Activity Diagram* Registrasi Alumni

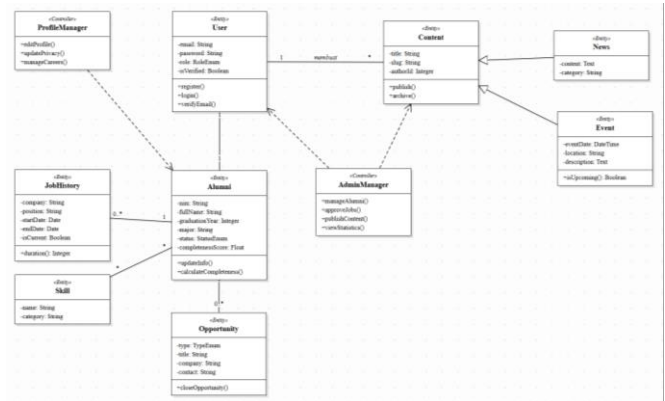
Gambar 4 diatas menggambarkan *Activity Diagram* Registrasi Alumni. Diagram ini menggambarkan alur aktivitas yang dilakukan oleh alumni selama proses registrasi berlangsung.

Object Oriented Design

Object Oriented Design (OOD) berfungsi mengatur tata letak perangkat lunak dengan berbasis pada manipulasi subsistem atau objek dalam sistem tersebut. Pada tahap ini, prinsip-prinsip OOP diterapkan secara konkret untuk memastikan arsitektur sistem yang kokoh dan mudah dipelihara.

Class Diagram

Class Diagram merupakan representasi visual dari struktur statis sistem yang mencakup definisi kelas, atribut, metode, dan relasi antar objek. Diagram ini bersifat statis karena hanya menggambarkan struktur dan jenis hubungan antar kelas, bukan proses interaksi yang terjadi saat *runtime*. *Class Diagram* relevan diterapkan pada proyek berbasis pemrograman berorientasi objek karena kemudahan pemahaman dan kesederhanaannya dalam merepresentasikan arsitektur sistem.

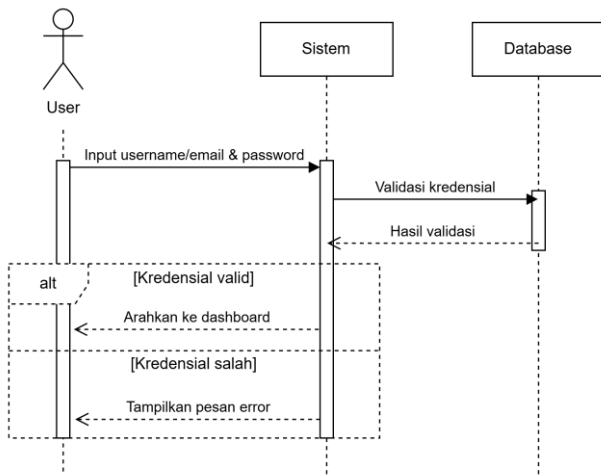


Gambar 5. *Class Diagram*

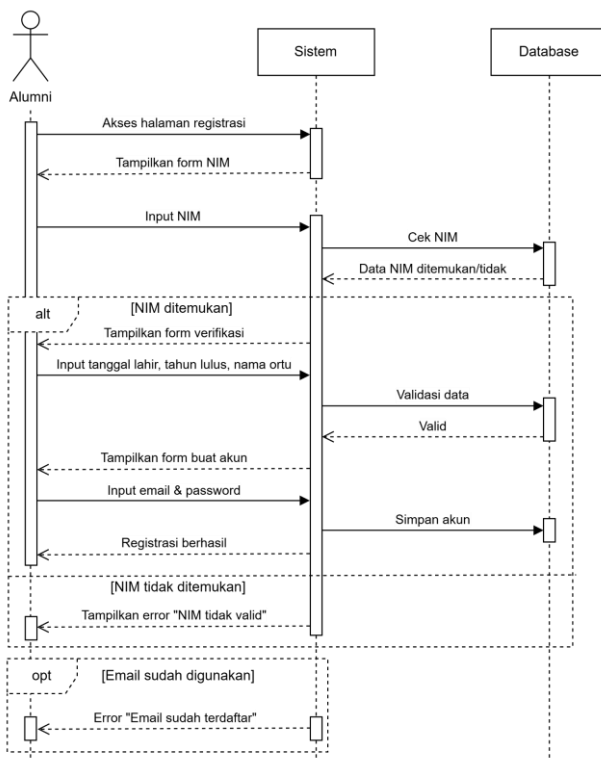
Berdasarkan Gambar 5, arsitektur sistem menerapkan prinsip OOP secara konkret untuk memastikan efisiensi dan keamanan kode. Prinsip *Inheritance* diterapkan pada kelas *User* yang menjadi induk bagi *Alumni*, serta kelas *Content* yang menurunkan atributnya ke *News* dan *Event*, sehingga mengurangi redundansi pada fungsi autentikasi dan manajemen konten. Sementara itu, keamanan data dijaga melalui *Encapsulation*, di mana atribut sensitif diatur bersifat *private* dan hanya dapat diakses melalui metode *public* yang tervalidasi, mencegah manipulasi data secara langsung dari luar sistem. Fleksibilitas sistem diperkuat dengan penerapan *Polymorphism* pada metode *publish()* di kelas *Content*, yang mampu beradaptasi menangani format data berbeda untuk berita maupun agenda kegiatan. Selain itu, struktur data yang logis dibangun melalui relasi *Association* dan *Composition*, terlihat pada hubungan antara *Alumni* dengan *JobHistory* dan *Skill*, yang memastikan rekam jejak karir dan kompetensi terkelola sebagai satu kesatuan utuh dengan profil alumni.

Sequence Diagram

Diagram yang menunjukkan bagaimana objek dalam sistem berinteraksi disebut diagram urutan [19]. Diagram ini menggambarkan sinyal yang dikirim bolak-balik antara objek sistem saat mereka berinteraksi untuk melaksanakan suatu aktivitas tertentu. Alur saat pengguna memasukkan kata sandi dan alamat *email* ditampilkan pada Gambar 6. Kredensial diverifikasi oleh sistem menggunakan basis data. Pengguna akan diarahkan ke dashboard jika kredensial valid; jika tidak, sistem akan menampilkan pesan kesalahan.

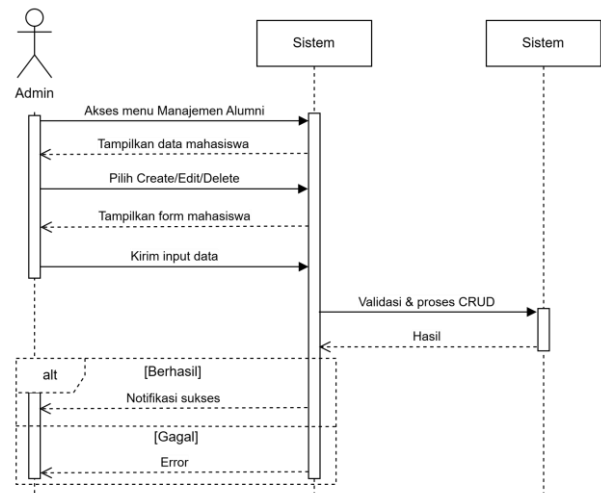
Gambar 6. *Sequence Diagram Login*

Alumni mengakses halaman registrasi dan memasukkan NIM seperti yang digambarkan pada Gambar 7. Sistem memeriksa ke *database* apakah NIM terdaftar. Jika ditemukan, alumni melengkapi data diri lalu membuat akun. Jika tidak ditemukan atau *email* sudah digunakan, sistem menampilkan pesan kesalahan.

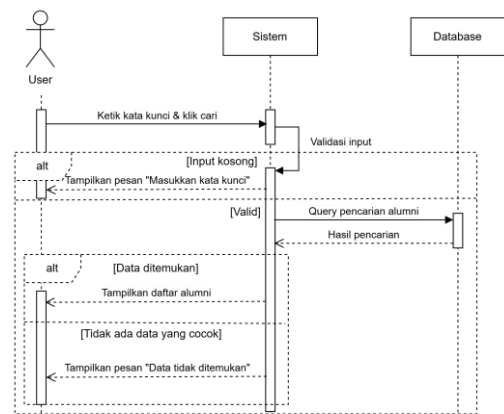
Gambar 7. *Sequence Diagram Registrasi*

Gambar 8 menunjukkan proses ketika admin mengelola data alumni. Admin membuka menu

manajemen alumni, lalu memilih aksi seperti menambah, mengedit, atau menghapus data. Sistem menampilkan *form*, menerima *input*, memvalidasi, kemudian memproses operasi CRUD ke *database*. Jika berhasil, sistem memberikan notifikasi sukses; jika gagal, sistem menampilkan pesan *error*.

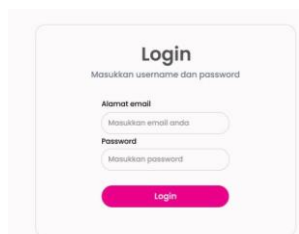
Gambar 8. *Sequence Diagram Manajemen Alumni*

Gambar 9 menjelaskan alur ketika pengguna melakukan pencarian data alumni. Pengguna mengetik kata kunci lalu menekan tombol cari. Sistem memvalidasi *input* jika kosong, sistem menampilkan pesan untuk mengisi kata kunci. Jika valid, sistem mengirim *query* ke *database* dan menerima hasil pencarian. Jika data ditemukan, sistem menampilkan daftar alumni; jika tidak ada yang cocok, sistem menampilkan pesan "Data tidak ditemukan."

Gambar 9. *Sequence Diagram Manajemen Pencarian Alumni*

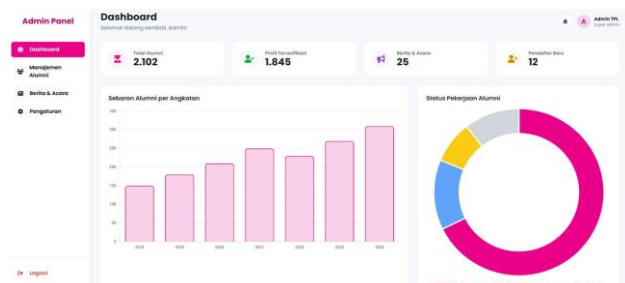
Perancangan Antarmuka

Perancangan Antarmuka adalah merancang *layout*, elemen-elemen interaktif, navigasi, dan interaksi antara pengguna dengan sistem agar tampilan antarmuka intuitif, efisien, dan sesuai kebutuhan pengguna.



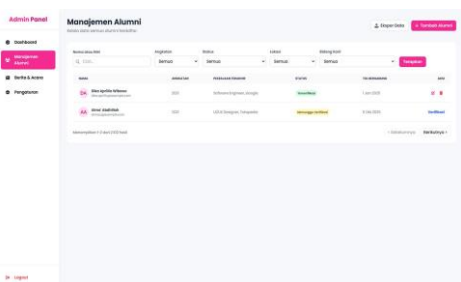
Gambar 10. Halaman Login

Halaman Login pada Gambar 10 berfungsi sebagai gerbang autentikasi tunggal (FR-01, FR-02) yang krusial untuk menjaga keamanan data (NFR-01). Desain ini memastikan hanya pengguna terdaftar (Alumni/Admin) yang dapat mengakses sistem, dengan fitur *forgot password* yang mendukung prinsip Kegunaan (NFR-03) dan menjaga ketersediaan akses pengguna.



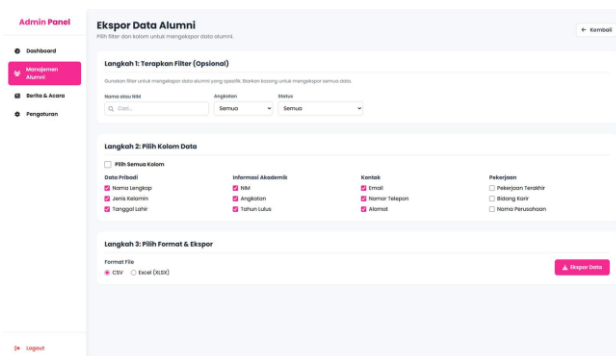
Gambar 11. Halaman Dashboard Statistik

Dashboard pada Gambar 11 dirancang khusus untuk menjawab kesulitan prodi dalam pelaporan akreditasi. Fitur visualisasi data menyajikan statistik kunci yang memangkas waktu pembuatan laporan sebaran lulusan dari hari menjadi hitungan detik.



Gambar 12. Halaman Filter Alumni

Gambar 12 adalah realisasi langsung dari kebutuhan Direktori dan Pencarian Alumni (FR-04). Fungsi *advanced filter* memungkinkan pengelola prodi mendapatkan data sebaran lulusan, *tracer study*, atau data akreditasi spesifik dalam hitungan detik, yang merupakan peningkatan efisiensi signifikan dibandingkan metode pencarian manual sebelumnya yang memakan waktu berhari-hari.



Gambar 13. Halaman Export Data Alumni

Halaman *Export Data Alumni* pada Gambar 13 dirancang untuk mendukung kebutuhan Ekspor Data (FR-06) dan pelaporan formal. Dengan opsi pemilihan kolom spesifik (Data Pribadi, Informasi Akademik, Kontak, Pekerjaan), antarmuka ini secara langsung mengakomodasi kebutuhan penyusunan borang akreditasi yang membutuhkan data terstruktur dan terpilah, sehingga mengurangi beban kerja admin dalam memformat ulang data secara manual.

Pembahasan

Identifikasi persyaratan sistem dilakukan secara sistematis guna memastikan platform yang dikembangkan mampu mengatasi hambatan utama dalam tata kelola data lulusan. Penentuan prioritas fungsi menggunakan metode MoSCoW (*Must have, Should have, Could have, Won't have*) menjamin fungsionalitas dengan dampak operasional tertinggi mendapatkan perhatian utama. Sejalan dengan temuan Iskandar & Ratnasari (2021), spesifikasi input, proses, dan output yang akurat menjadi syarat mutlak agar perangkat lunak beroperasi secara optimal dalam meniadakan kendala pendataan manual. Fokus utama diarahkan pada fitur manajemen profil, autentikasi, dan fungsi pencarian yang dikategorikan sebagai prioritas tinggi demi menjaga integritas informasi. Pemodelan sistem melalui *Object-Oriented Analysis* (OOA) memberikan representasi entitas yang natural

serta memudahkan pemeliharaan jangka panjang. Penggunaan *Use Case Diagram* memisahkan hak akses secara tegas antara Admin, Alumni, dan Pengunjung guna menerapkan prinsip *Least Privilege*. Arsitektur tersebut selaras dengan pemikiran Magableh dkk. (2024) yang menyatakan bahwa paradigma berorientasi objek sangat efektif dalam menjaga keamanan data sensitif melalui pembatasan wewenang pengguna. Pemisahan peran tersebut memastikan bahwa kendali administratif tetap terpusat tanpa mengabaikan aksesibilitas bagi alumni. Penerapan prinsip *Object-Oriented Design* (OOD) secara konkret terlihat pada struktur kelas yang modular. Penggunaan *Inheritance* pada kelas User dan Content berhasil mereduksi redundansi kode, sementara *Encapsulation* menjamin atribut sensitif hanya dapat dimanipulasi melalui metode publik yang tervalidasi. Fleksibilitas sistem diperkuat dengan *Polymorphism* pada metode publikasi konten, yang mampu beradaptasi menangani format data berbeda. Pendekatan arsitektural tersebut didukung oleh riset Kezia dkk. (2025), meskipun dalam studi tersebut fokus lebih diarahkan pada visualisasi jejaring, namun keduanya bersepakat bahwa struktur perangkat lunak yang kokoh adalah fondasi utama bagi keberlanjutan sistem administratif. Realisasi antarmuka pada sistem tersebut dirancang untuk menjawab kesulitan program studi dalam penyusunan laporan akreditasi. Fitur *Dashboard* Statistik dan *Advanced Filter* memungkinkan pengelola memetakan sebaran lulusan atau data *tracer study* dalam hitungan detik. Peningkatan efisiensi yang melampaui 90% dibandingkan metode manual mempertegas teori Patil dkk. (2025) mengenai urgensi *Alumni Tracking System* sebagai basis data efisien untuk pemantauan karir. Dengan demikian, sistem yang dirancang tidak hanya menjadi media komunikasi, tetapi juga berfungsi sebagai instrumen strategis dalam penguatan reputasi institusi melalui penyediaan data yang terstruktur dan terpilah secara otomatis.

4. Kesimpulan dan Saran

Penelitian ini berhasil merancang Sistem Informasi Alumni berbasis web menggunakan paradigma *Object Oriented Programming* (OOP) dengan metodologi *Object-Oriented Analysis and Design* (OOAD). Rancangan ini efektif mengatasi masalah pengelolaan

data alumni Program Studi Teknologi Rekayasa Perangkat Lunak (TPL) yang sebelumnya manual dan tersebar. Keberhasilan ini ditandai dengan perancangan fitur fungsional utama (autentikasi, manajemen profil, direktori, dashboard admin, dan ekspor data), yang ditopang oleh arsitektur *modular*, *maintainable*, dan *scalable* berkat penerapan prinsip-prinsip OOP. Meskipun demikian, penelitian ini masih memiliki keterbatasan karena hanya berada pada tahap analisis dan perancangan sistem, sehingga belum mencakup implementasi kode program serta pengujian fungsional, kinerja, dan keamanan sistem secara nyata. Oleh karena itu, saran pengembangan ke depan meliputi tiga fokus utama: melaksanakan implementasi rancangan dan pengujian menyeluruh (*user acceptance test*); mengembangkan fitur analitik lanjutan, seperti Business Intelligence dan visualisasi jejaring karir alumni; serta mengintegrasikan sistem dengan sistem akademik eksternal (SIKAD) untuk memverifikasi data dan meningkatkan efisiensi administrasi.

5. Daftar Pustaka

- Abdurrahman, U., Siregar, W. Y., & Siregar, R. (2025). IMPLEMENTASI OBJECT-ORIENTED PROGRAMMING (OOP) DALAM APLIKASI ORDER WASH BERBASIS DESKTOP MENGGUNAKAN VISUAL BASIC. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(4), 5956-5961. <https://doi.org/10.36040/jati.v9i4.13964>.
- Anthony Jr, B. (2021). Information flow analysis of a knowledge mapping-based system for university alumni collaboration: a practical approach. *Journal of the Knowledge Economy*, 12(2), 756-787.
- Dongmo, C. (2024). A Review of Non-Functional Requirements Analysis Throughout the SDLC. *Computers*, 13(12), 308.
- ISKANDAR, A. A. (2021). Sistem Informasi Manajemen Konfeksi Berbasis Web (Studi Kasus: Shofa Collection Tasikmalaya).

- Lubis, A. F., Nadha, D., Basri, F. K., & Pulungan, J. (2023, August). Analisa Perancangan Sistem Informasi Inventory menggunakan Metode OOAD Tatanan Cafe di Pekanbaru: Inventory Information System Design Analysis using OOAD Method at the Cafe Tatanan Pekanbaru. In *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat* (pp. 138-145).
- Madhani, E. I., Wicaksono, S. A., & Amalia, F. (2022). Pengembangan Sistem Informasi Pendataan Alumni berbasis Web pada SMA Negeri 3 Pontianak. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(4), 1940-1950.
- Magableh, A. A., Ata, H. A. B., Saifan, A. A., & Rawashdeh, A. (2024). Towards improving aspect-oriented software reusability estimation. *Scientific Reports*, 14(1), 13214.
- Ningsih, T., & Mustakim, M. (2019). Sistem Informasi Alumni Program Studi Sistem Informasi Uin Sultan Syarif Kasim Riau. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 5(2), 153-160. <http://dx.doi.org/10.24014/rmsi.v5i2.7641>.
- Prasetya, A. F., Sintia, S., & Putri, U. L. D. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan dan Informasi*, 1(1), 14-18.
- Prayitno, E., Suprawoto, T., & Perdana, I. J. (2024). Perancangan Ulang User Interface Sistem Informasi Akademik dengan Metode Design Thinking. *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, 22(2), 51-58. <https://doi.org/10.61805/fahma.v22i2.132>.
- Rachmatika, P. A., Ain, R. N., Wahyudinarti, E., & Fitri, A. S. (2025). Penerapan metode object oriented analysis and design pada aplikasi sistem informasi pelayanan masyarakat Surabaya 'MySurabaya'. *JITET: Jurnal Informatika dan Teknik Elektro Terapan*, 13(1), 1076–1083.
- Rahinda, A. S., Yunita, F., Ridha, M. R., & Usman, U. (2024). School information systems and alumni data at SMK Negeri 2 Tembilahan. *Jurnal Perangkat Lunak*, 6(1), 81–90.
- Ristias, A. A., Mulia, R. F., & Fajar, A. P. (2021). Sebuah Perancangan Aplikasi Codelife Berbasis Android Dengan Menggunakan Metode Object Oriented Analysis and Design (Ooad). In *Prosiding Seminar Nasional Teknologi dan Sistem Informasi* (Vol. 1, No. 1, pp. 293-302). <https://doi.org/10.33005/sitasi.v1i1.205>.
- Syaputra, S. M., Anwar, S., & Malau, Y. (2023). Perancangan Sistem Informasi Manajemen Tugas Harian Berbasis Website Pada PT. Mixtra Inti Tekindo. *Jurnal INSAN Journal of Information System Management Innovation*, 3(2), 62-71. <https://doi.org/10.31294/jinsan.v3i2.2904>.
- Yuanita, H. I., Wijayanto, B., & Cahyono, T. (2022). Frontend Development Of Course Scheduling System Integrated Sia At Engineering Faculty University Of Jenderal Soedirman Using Devops Method. *Jurnal Teknik Informatika (JUTIF)*, 3(2), 305-319. <https://doi.org/10.20884/1.jutif.2022.3.2.227>.