



## Implementasi Sistem Informasi Penelitian Dan Pengabdian Pada Fakultas Teknik Universitas Pelita Bangsa Berbasis *Microservices*

Fajar Agung Nughroho<sup>1</sup>, Dr. H. Dendy K. Pramudito<sup>2</sup>, M. Najamuddin Dw Miharja<sup>3</sup>  
<sup>1,2,3</sup> Program Studi Teknik Informatika, Fakultas Teknis Universitas Pelita Bangsa  
Bekasi, Indonesia

Email : [fajaragungn@pelitabangsa.ac.id](mailto:fajaragungn@pelitabangsa.ac.id), [doktor.haji.dendy@pelitabangsa.ac.id](mailto:doktor.haji.dendy@pelitabangsa.ac.id),  
[najamuddin.dwi@pelitabangsa.ac.id](mailto:najamuddin.dwi@pelitabangsa.ac.id)

Korespondensi email : [fajaragungn@pelitabangsa.ac.id](mailto:fajaragungn@pelitabangsa.ac.id)

---

### Abstrak

*This study aims to develop and implement a microservices-based information system for managing research and community service data at Universitas Pelita Bangsa. The previous system had several shortcomings, including manual and fragmented data management, leading to data duplication, slow reporting processes, and limited access to information for lecturers. This research employed Agile Development methodology to ensure that the developed system effectively meets user needs. The results indicate that the implementation of a microservices architecture significantly improves the efficiency and effectiveness of data management, accelerates the reporting process, and facilitates easier access to information for lecturers. Additionally, the system optimizes collaboration among stakeholders by providing seamless and integrated access to research and community service data. Consequently, this microservices-based information system has proven capable of enhancing the quality of faculty services, while also offering flexibility for future system development in line with the faculty's needs and growth.*

---

### Informasi Artikel

Diterima: 18 Juli 2024

Direvisi: 25 Agustus 2024

Dipublikasikan: 30 September 2024

---

### Keywords

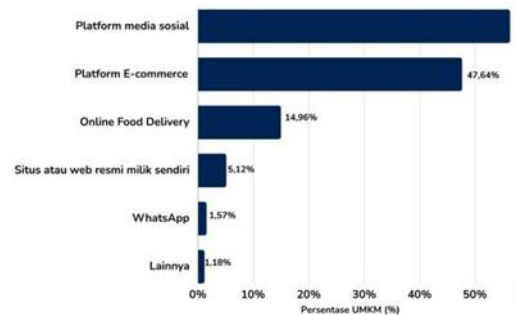
*Information Systems, Microservices, Data Management, Research.*

---

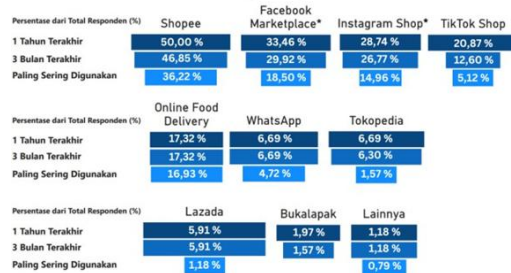


## I. Pendahuluan

Perkembangan teknologi informasi telah mendorong transformasi global, terutama melalui adopsi teknologi canggih seperti kecerdasan buatan (AI), *internet of things* (IoT), dan *cloud computing*. Inovasi-inovasi ini mempercepat digitalisasi dan meningkatkan efisiensi di berbagai sektor kehidupan [1]. Di Indonesia, teknologi informasi berperan penting dalam meningkatkan akses informasi, layanan publik, dan mendorong pertumbuhan ekonomi digital. Survei INDEF pada Desember 2023 terhadap 254 UMKM di berbagai wilayah menunjukkan bahwa platform digital turut mendukung peningkatan kinerja penjualan online UMKM [2]. Data terkait penjualan secara online yang ditunjukkan pada gambar 1. sebagai berikut :



**Gambar 1.** Tempat berjualan online dalam satu tahun terakhir  
Sumber: Indef (2024) [3]



**Gambar 2.** Jenis platform online  
Sumber: Indef (2024) [3]

Dalam Tridharma Perguruan Tinggi, dosen wajib menjalankan pendidikan, penelitian, dan pengabdian kepada masyarakat. Penelitian dan pengabdian berperan penting dalam meningkatkan kualitas akademik dan memberikan dampak nyata bagi masyarakat [4].

Pengelolaan data penelitian dan pengabdian di Fakultas Teknik Universitas Pelita Bangsa (UPB) masih manual dan terpisah, menyebabkan duplikasi data, sulitnya akses informasi, lambatnya pelaporan, serta terhambatnya analisis data dan koordinasi antar departemen [5][6].

Dengan berkembangnya teknologi informasi, dibutuhkan sistem terpusat untuk mengelola data penelitian dan pengabdian. Arsitektur *microservices* menjadi solusi karena menawarkan fleksibilitas, skalabilitas, dan kemudahan integrasi [7].

Implementasi arsitektur *microservices* di UPB dapat berbagai manfaat seperti mempercepat pengembangan layanan, memperbaiki pengelolaan data, dan meningkatkan analisis data secara *real-time* [8].

Penerapan sistem informasi berbasis arsitektur *microservices* di Fakultas Teknik Universitas Pelita Bangsa dapat membawa berbagai manfaat signifikan. Pertama, Sistem ini meningkatkan efisiensi pengelolaan data penelitian dan pengabdian dengan mengurangi duplikasi, mempercepat pelaporan, dan mempermudah akses informasi bagi dosen.. Kedua, sistem ini meningkatkan layanan fakultas melalui platform terintegrasi untuk manajemen penelitian, pengabdian, dan pelaporan. Ketiga, Arsitektur *microservices* mengoptimalkan skalabilitas sistem, memungkinkan pengembangan fleksibel dan



penyesuaian dengan kebutuhan serta pertumbuhan fakultas ke depan. Keempat, Sistem ini mendukung kolaborasi antar *stakeholder* dengan menyediakan akses mudah dan terpadu ke data penelitian dan pengabdian.

## II. Metodologi

### 2.1 Instrumen Penelitian

#### 1. Software

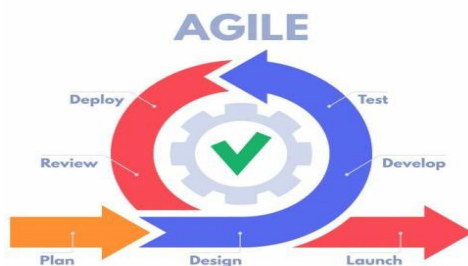
*Software* yang digunakan dalam penelitian ini terdiri dari *Visual Studio Code, Laragon, Node Js, Chrome Web Browser, Microsoft Edge Browser, Command Prompt.*

#### 2. Hardware

*Hardware* yang digunakan dalam penelitian ini terdiri dari Laptop Lenovo Ideapad 3i dengan *processor Intel Core i3 1115G4 2-Core / 4-Thread*

### 2.2 Metode yang Digunakan

Metode *Agile Development* satu pendekatan yang sering digunakan dalam perangkat lunak, dikenal juga sebagai sebagai SDLC (*Software Development Life Cycle*) yang berfokus pada proses iteratif, di mana aturan dan solusi dikembangkan melalui kolaborasi yang terstruktur dan terorganisir antar tim [88,89]



**Gambar 3.** *Agile development* Sumber: Setiawan (2021) [90]

#### 1. *Requirements* (Pengumpulan Kebutuhan)

Mengumpulkan kebutuhan sistem baru melalui interaksi dosen dan prodi menggunakan wawancara serta observasi, menghasilkan daftar kebutuhan untuk dasar desain sistem..

#### 2. *Design* (Desain)

Tahap desain mencakup perancangan arsitektur sistem dan antarmuka pengguna berdasarkan kebutuhan, termasuk diagram alur, struktur basis data, dan sketsa antarmuka. Berikut adalah elemen dan sketsa yang diterapkan :

##### a. *Landing Page*

Halaman pertama yang diakses oleh pengguna yaitu *Landing page, landing page* yang rapih mempermudah pengguna untuk menavigasi dan memamhani fungsi utama sistem.

Berikut deskripsi elemen dan fitur yang ada di *landing page* dapat dilihat pada tabel 1.

##### b. *Dashboard*

Dashboard ini halaman pertama yang dilihat pengguna dan memberikan ringkasan tentang status pengajuan, notifikasi terbaru, dan akses cepat ke fitur utama.

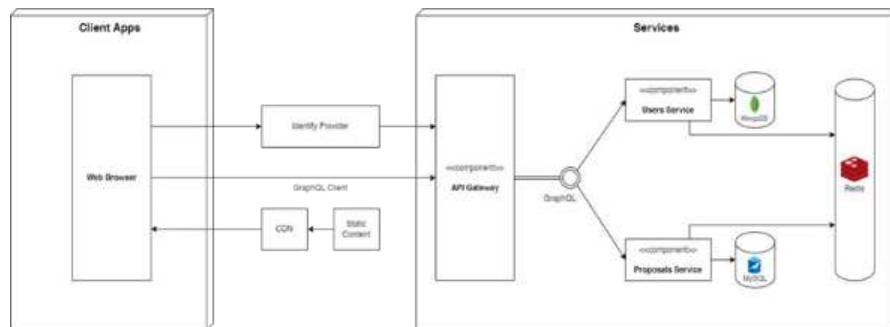
**Tabel 1.** Elemen *landing page*

<i>Header</i>	<ul style="list-style-type: none"> <li>- Logo: Terletak di sudut kiri atas, logo institusi atau sistem untuk membangun identitas dan kepercayaan.</li> <li>- Navigasi Utama: Menu navigasi dengan tautan ke halaman-halaman penting, seperti panduan, grafik penelitian dan pengabdian, riwayat pengajuan, berita dan pengumuman.</li> </ul>
<i>Banner Utama</i>	<ul style="list-style-type: none"> <li>- Deskripsi singkat sistem</li> <li>- Tombol untuk tindakan utama seperti "Ajukan Usulan".</li> </ul>
Grafik Penelitian & Pengabdian	<ul style="list-style-type: none"> <li>- Grafik Batang: Menunjukkan jumlah penelitian yang diajukan, diterima, dan ditolak dalam periode waktu tertentu.</li> <li>- Grafik Pai: Distribusi jenis usulan yang diajukan.</li> <li>- Filter: Opsi untuk memfilter data berdasarkan waktu, bidang, atau status pengajuan.</li> </ul>

**Tabel 2.** Elemen dashboard

Form pengajuan penelitian dan pengabdian	Bidang input teks yang meliputi identitas usulan, anggota, judul, <i>review</i> dan <i>submit</i>
Halaman Detail Usulan	Menampilkan detail dari usulan penelitian dan pengabdian
Halaman Persetujuan Prodi	Halaman ini digunakan oleh prodi untuk menyetujui atau menolak pengajuan usulan
Halaman Surat Rekomendasi Prodi	Halaman ini digunakan oleh prodi untuk mengirim surat rekomendasi kepada pengusul yang telah disetujui
Halaman Profile dan Account	Menampilkan detail identitas diri dan identitas akun

### c. Desain Arsitektur Sistem

**Gambar 4.** Arsitektur sistem

Desain arsitektur sistem penelitian dan pengabdian ini menggunakan React.Js, API berbasis GraphQL dan dua jenis *database* untuk kebutuhan yang berbeda.

#### 3. *Development* (Pengembangan)

Tahap pengembangan mengubah desain menjadi kode secara bertahap, dengan pengujian tiap bagian sebelum lanjut.

Peneliti berkolaborasi dengan *end user* untuk memastikan sistem sesuai kebutuhan.

#### 4. *Testing* (Pengujian)

Tahap pengujian ini berfungsi untuk memastikan sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini mencakup *unit testing*,

*contract testing, component testing, dan user acceptance testing (UAT).*

**5. Review (Ulasan)**

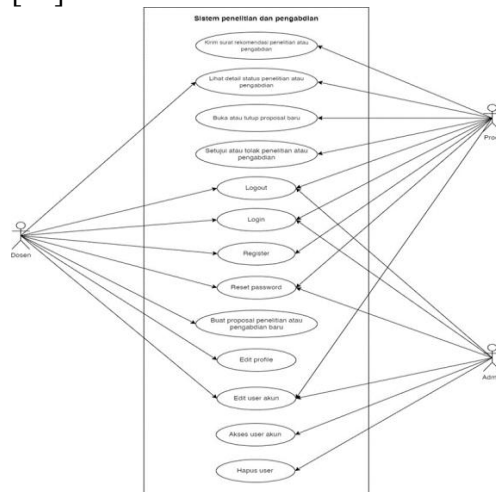
Tahap ulasan melibatkan evaluasi sistem dan pengumpulan umpan balik pengguna setelah tiap iterasi dan peluncuran, untuk mengidentifikasi perbaikan serta memastikan sistem berkembang sesuai kebutuhan dan kinerjanya efektif.

**2.3 Perancangan Sistem**

Proses ini dimulai dengan mengidentifikasi kebutuhan dan tujuan sistem berdasarkan analisis kebutuhan pengguna [9]. Selanjutnya, arsitektur sistem dirancang untuk menentukan struktur keseluruhan, termasuk komponen perangkat keras dan perangkat lunak, antarmuka pengguna, dan interaksi antar komponen [10].

**1. Use Case Diagram**

Use case diagram menampilkan berbagai use case dan aktor serta hubungan di antara mereka [11]. Diagram ini digunakan untuk memvisualisasikan use case statis dari suatu sistem. Use case menjelaskan fungsi atau tugas yang dilakukan oleh sistem tanpa merinci cara kerjanya [12].

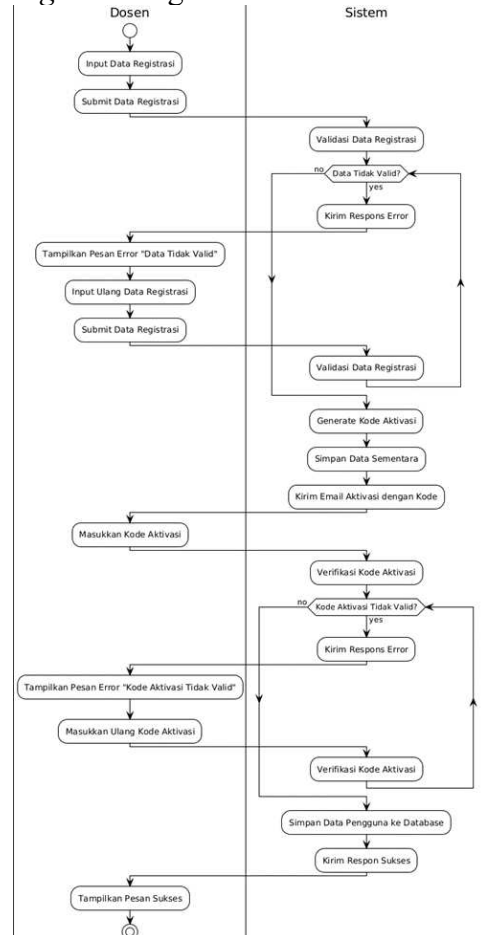


**Gambar 5.** Use case diagram sistem penelitian dan pengabdian

Diagram ini menunjukkan alur aktivitas pengguna dalam sistem, seperti pengajuan surat rekomendasi, melihat status, membuka/menutup, menyetujui/menolak proposal, serta manajemen akun (login, logout, register, reset password, dan edit profil). Sistem juga mendukung pembuatan proposal, pengelolaan hak akses, dan penghapusan akun. Diagram ini memberikan gambaran menyeluruh tentang interaksi pengguna dengan fitur sistem.

**2. Activity Diagram**

Berikut contoh diagram *activity register* dengan aktor dosen :



**Gambar 6.** Activity diagram register dengan aktor dosen

Pada metode ini disesuaikan dengan kebutuhan dan keperluan masing-masing fitur dan setiap dashboard akan

berbeda untuk penelitian ini disesuaikan dengan use case pada Gambar 3. 12.

### 3. *Sequence Diagram*

Diagram interaksi ini menunjukkan bagaimana objek berkomunikasi dalam urutan waktu tertentu, digunakan untuk memodelkan skenario spesifik dengan menampilkan alur pesan atau metode antar objek dalam menyelesaikan tugas [13][14]. *Sequence Diagram* dibuat sesuai dengan kebutuhan yang diperlukan.

### 4. *Class Diagram*

Class Diagram dapat dilihat pada gambar 7. *Class diagram* ini menggambarkan struktur sistem untuk penelitian dan pengabdian, mencakup pengguna dengan profil yang memiliki berbagai peran (Dosen, Prodi, Admin), data penelitian dan pengabdian, serta anggota yang terlibat. Setiap entitas seperti pengguna, penelitian, pengabdian, dan anggota memiliki atribut-atribut yang relevan dan terkait satu sama lain, dengan status usulan diatur sebagai enum (Submitted, Approved, Rejected) dan pengaturan status pembukaan pengajuan proposal.

### 5. *Database Diagram*

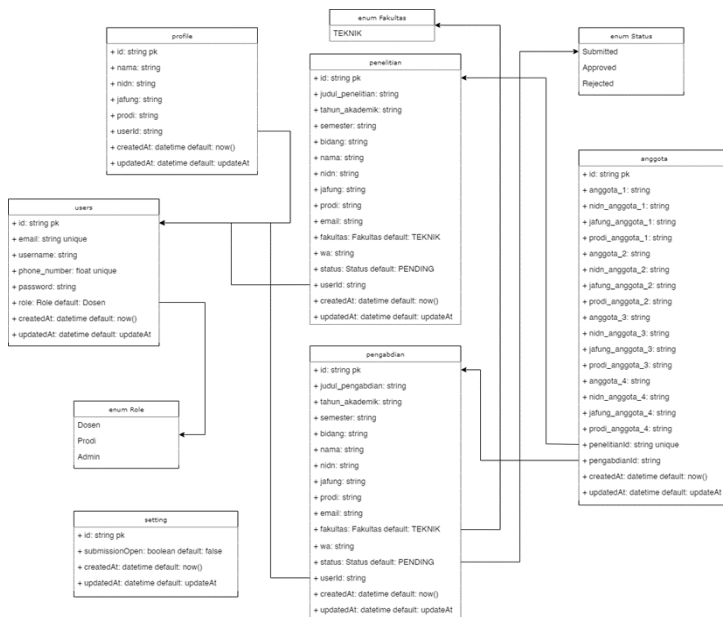
Skema *database users* dapat dilihat pada **gambar 8**. Database diagram tersebut menunjukkan desain skema basis data *NoSQL* dengan dua *collections: users* dan *profiles*.

Desain ini mencerminkan relasi *One-to-One* antara koleksi *users* dan *profiles*, di mana setiap pengguna memiliki satu profil terkait. Hal ini ditunjukkan dengan adanya *userId* di koleksi *profiles*, yang merujuk ke id di koleksi *users*.

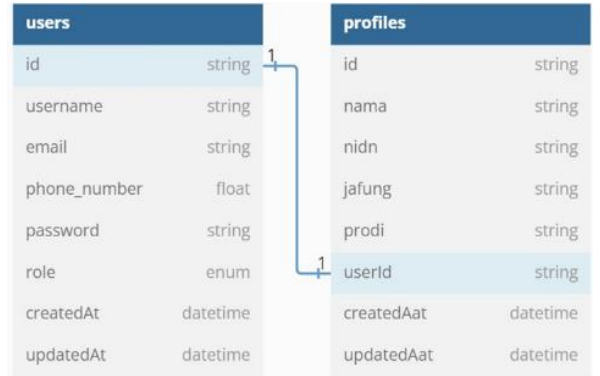
Skema database proposals dapat dilihat pada **gambar 9**. Desain diagram tersebut menunjukkan desain basis data dengan beberapa tabel utama: *users*, penelitian, pengabdian, anggota, dan *setting*. Berikut adalah penjelasan singkat mengenai masing-masing table.

### 6. *Deployment Diagram*

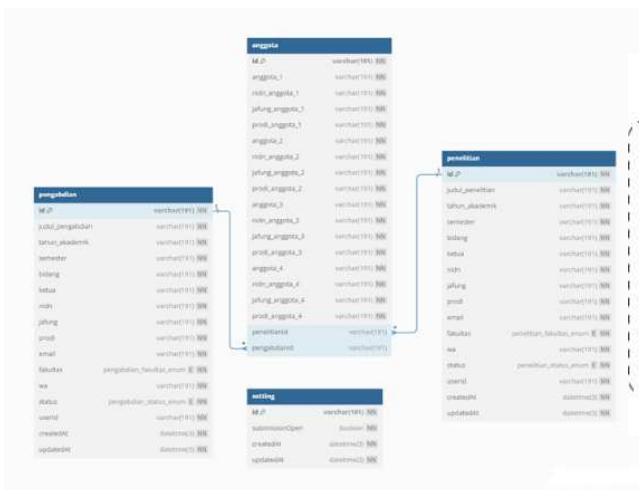
*Deployment diagram* yang ditampilkan dalam **gambar 10**, menggambarkan arsitektur sistem berbasis *microservices* yang terdiri dari beberapa komponen utama: *Client Apps*, *API Gateway*, *Services*, dan *Databases*.



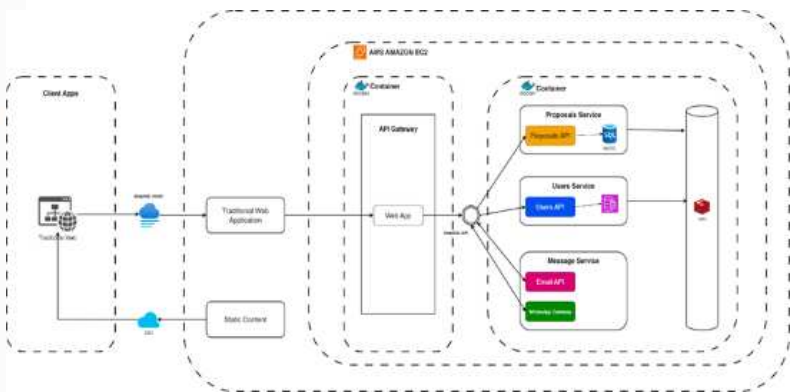
Gambar 7. Class diagram



Gambar 8. Skema database



Gambar 9. Skema database proposals



Gambar 10. Deployment diagram users

### III. Hasil dan Pembahasan

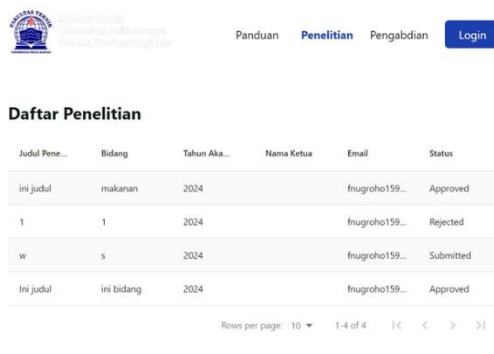
#### 3.1 Implementasi Sistem

Pada Implementasi sistem terdapat beberapa tampilan dari setiap dashboard yang dibuat seperti tampilan landing page, halaman penelitian, halaman pengabdian, halaman login, halaman register, halaman forgot password, halaman reset password, halaman verification code, halaman dashboard

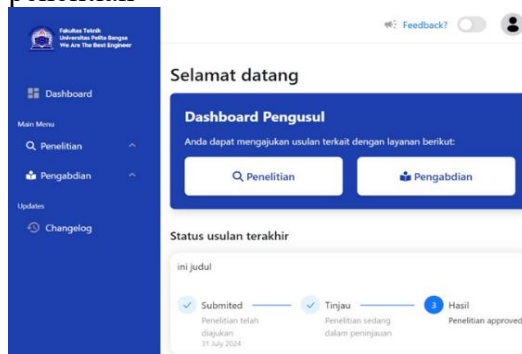
dosen, halaman dashboard prodi, halaman dashboard admin, halaman edit profile, halaman edit user account, halaman usulan penelitian atau pengabdian baru, halaman anggota, halaman judul, halaman review dan submit, halaman detail usulan, halaman detail usulan prodi, halaman surat rekomendasi prodi, halaman settings, halaman user management, tampilan grafik batang, tampilan grafik pie.



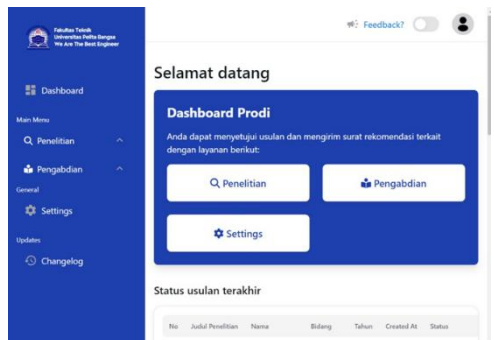
Gambar 11. Tampilan landing page



Gambar 12. Tampilan halaman penelitian



Gambar 13. Tampilan halaman dashboard dosen



Gambar 14. Tampilan halaman dashboard prodi

### 3.2 Pengujian Sistem

#### 1. Unit Testing

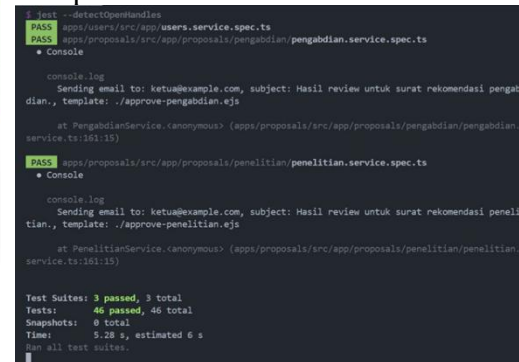
Pengujian unit dilakukan untuk memastikan bahwa setiap fungsi bekerja dengan benar sesuai dengan spesifikasi yang telah ditetapkan. Hasil pengujian diharapkan sesuai dengan output yang diharapkan untuk memastikan kualitas dan fungsionalitas aplikasi.

#### 2. Skenario Pengujian Unit

Berikut adalah skenario pengujian unit untuk *users service* :

##### a. Implementasi Pengujian Unit

Pada implementasi ini dilakukan pada *users service* dan *proposals service* yang menggunakan bahasa pemrograman *TypeScript* dan *framework Nest Js*. Dalam proses ini, perangkat lunak dijalankan untuk memastikan semua berfungsi sebagaimana mestinya. Dalam konteks ini, peneliti memilih menggunakan *Jest* untuk melakukan pengujian. *Jest* dikenal karena kecepatan menjalankan *test suite* secara paralel, cocok untuk pengujian komponen *Nest Js*.



Gambar 15. Unit testing result

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	19.45	6.28	8.4	77.04	
proposals/prisma	71.42	100	0	60	
prisma.service.ts	71.42	100	0	60	10-14
proposals/prisma/generated	91.37	37.5	0	91.37	
index.js	91.37	37.5	0	91.37	242-252
...sals/prisma/generated/runtime	16.45	5.26	7.46	52	
library.js	16.45	5.26	7.46	52	...11-112,123-126
proposals/src/app/email	57.14	100	0	40	
email.service.ts	57.14	100	0	40	22-31,45-54
proposals/src/app/entities	100	100	100	100	
enums.ts	100	100	100	100	
.../src/app/proposals/penelitian	91.66	80.64	75	91.37	
penelitian.service.ts	91.66	80.64	75	91.37	155-157,218-225
.../src/app/proposals/pengabdian	91.66	80.64	75	91.37	
pengabdian.service.ts	91.66	80.64	75	91.37	155-157,218-225
users/prisma	83.33	100	0	75	
prisma.service.ts	83.33	100	0	75	7
users/prisma/generated	90.9	50	0	90.9	
index.js	90.9	50	0	90.9	194-204
users/prisma/generated/runtime	16.42	5.29	7.39	52	
library.js	16.42	5.29	7.39	52	...11-112,123-126
users/src/app	72.17	64.51	68.42	71.68	
users.service.ts	72.17	64.51	68.42	71.68	...94-319,348-385
users/src/app/email	62.5	100	0	50	
email.service.ts	62.5	100	0	50	14-22
users/src/app/entities	100	100	100	100	
enums.ts	100	100	100	100	
users/src/app/utils	100	100	100	100	
sendToken.ts	100	100	100	100	

**Gambar 16.** Unit testing coverage  
b. *Contract Testing*

*Contract testing* memainkan peran krusial dalam memastikan bahwa perubahan pada satu layanan tidak menyebabkan kegagalan atau malfungsi pada layanan lain yang bergantung padanya. Konsep dasar dari *contract testing* adalah adanya "kontrak" yang disepakati antara penyedia layanan (*provider*) dan konsumen layanan (*consumer*), yang mendefinisikan harapan dan spesifikasi interaksi, seperti *format data*, *endpoint API*, dan jenis *response*.

Method	Result	Message	Pass
POST register	Pass	Should register user and return activation ...	1 0
POST login	Pass	Should login user and return user data an...	1 0
POST activate user	Pass	Should activate user and return user data	1 0
POST getUsers	Pass	Status code is 200	1 0
POST forgotPassword	Pass	Status code is 200	2 0
POST resetPassword	Pass	Response has forgotPassword field	1 0
POST updateUser	Pass	Status code is 200	1 0

**Gambar 17.** Users contract testing

Method	Result	Message	Pass
POST getAIPenelitian	Pass	Status code is 200	1 0
POST createPenelitian	Pass	Status code is 200	3 0
POST getPenelitianById	Pass	Status code is 200	1 0
POST getAIPengabdian	Pass	Status code is 200	1 0
POST createPengabdian	Pass	Status code is 200	3 0
POST getPengabdianById	Pass	Status code is 200	1 0
POST approve	Pass	Status code is 200	1 0
POST reject	Pass	Status code is 200	1 0
POST delete	Pass	Status code is 200	1 0

**Gambar 18.** Proposals contract testing

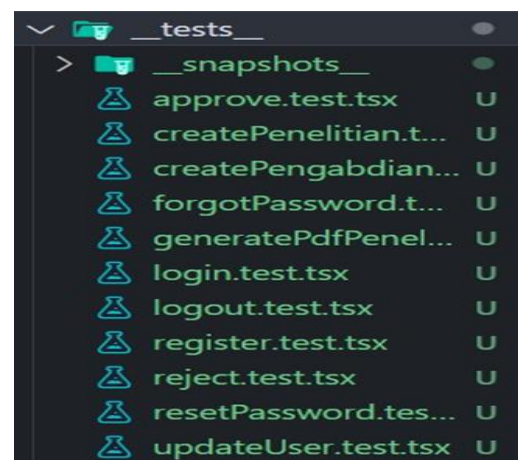
**Tabel 3.** Users contract testing

Register	Status	200
	Tujuan	Menguji endpoint registrasi pengguna baru.
	Deskripsi	Mengirimkan data pengguna baru dan mengharapkan respons dengan data aktivasi pengguna.
	Hasil	Pengujian berhasil (Pass). Pengguna berhasil terdaftar dan respons berisi data aktivasi pengguna.
activateUser	Status	200
	Tujuan	Menguji endpoint aktivasi
	Deskripsi	Mengirimkan data aktivasi pengguna dan mengharapkan respons dengan data pengguna yang telah diaktifkan.
	Hasil	Pengujian berhasil (Pass). Pengguna berhasil diaktifkan dan respons berisi data pengguna yang telah diaktifkan.
Login	Status	200
	Tujuan	Menguji endpoint login pengguna
	Deskripsi	Mengirimkan kredensial pengguna (email dan kata sandi) dan mengharapkan respons dengan data pengguna dan token autentikasi

	Hasil	Pengujian berhasil (Pass). Pengguna berhasil login dan respons berisi data pengguna dan token autentikasi.
getUsers	Status	200
	Tujuan	Menguji endpoint untuk mendapatkan daftar pengguna.
	Deskripsi	Mengirimkan permintaan untuk mendapatkan daftar semua pengguna dan mengharapkan respons dengan data pengguna.
	Hasil	Pengujian berhasil (Pass). Status kode respons adalah 200 dan respons berisi data pengguna.
forgotPassword	Status	200
	Tujuan	Menguji endpoint untuk permintaan lupa kata sandi.
	Deskripsi	Mengirimkan email pengguna untuk meminta reset kata sandi dan mengharapkan respons dengan informasi terkait reset kata sandi.
	Hasil	Pengujian berhasil (Pass). Status kode respons adalah 200 dan respons berisi field forgotPassword.
resetPassword	Status	200
	Tujuan	Menguji endpoint untuk reset kata sandi.
	Deskripsi	Mengirimkan data reset kata sandi (token dan kata sandi baru) dan mengharapkan respons dengan informasi sukses
	Hasil	Pengujian berhasil (Pass). Status kode respons adalah 200 dan kata sandi berhasil direset.
resetPassword	Status	200
	Tujuan	Menguji endpoint untuk memperbarui data pengguna.
	Deskripsi	Mengirimkan data yang diperbarui untuk pengguna dan mengharapkan respons dengan data pengguna yang diperbarui.
	Hasil	Pengujian berhasil (Pass). Status kode respons adalah 200 dan data pengguna berhasil diperbarui.

### c. Component Testing

Untuk mengimplementasikan *component testing* menggunakan *Jest* dan *React Testing Library*, peneliti akan membuat beberapa *file test* untuk menguji setiap komponen sesuai dengan skenario pengujian yang sudah ditentukan. Berikut adalah langkah-langkah dan contoh implementasinya untuk beberapa komponen:



Gambar 19. File test component

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
All files	65.27	55.35	29.05	66.66	
components/Auth	97.69	72	92.3	97.16	
forgotPassword.tsx	100	75	100	100	66-69
login.tsx	97.36	71.42	100	96.87	40
register.tsx	100	64.28	100	100	70-118
resetPassword.tsx	93.54	80	75	91.66	33-34
components/Dashboard/Account/update	70.96	51.06	35	66.66	...
update-akun.tsx	70.96	51.06	35	66.66	...85,104,1
.../Dashboard/Proposals/penelitian/create	50	25.92	17.07	52.94	
anggotaForm.tsx	8.33	0	0	10.93	24-196
data.ts	100	100	100	100	
formData.tsx	66.66	33.33	16.66	65.62	...05-146,1
identitasForm.tsx	92.3	50	66.66	92.3	41
index.tsx	82.25	52.17	60	81.03	...60,64,2
.../Dashboard/Proposals/pengabdian/create	50	25.92	17.07	52.94	
anggotaForm.tsx	8.33	0	0	10.93	24-196
data.ts	100	100	100	100	
formData.tsx	66.66	33.33	16.66	65.62	...05-146,1
identitasForm.tsx	92.3	50	66.66	92.3	41
index.tsx	82.25	52.17	60	81.03	...60,64,2
components/Dashboard/settings	75	100	0	75	
closed.tsx	75	100	0	75	4
components/hooks	100	75	100	100	
useUser.ts	100	75	100	100	9-12
graphql/actions	100	100	100	100	
auth.action.ts	100	100	100	100	
penelitian.action.ts	100	100	100	100	
pengabdian.action.ts	100	100	100	100	
settings.action.ts	100	100	100	100	
helpers	100	100	100	100	
schemas.ts	100	100	100	100	

**Gambar 20.** Code coverage component testing

Gambar tersebut menunjukkan laporan cakupan kode (code coverage) untuk file component yang diuji dalam gambar 4. Laporan ini memberikan wawasan tentang seberapa baik bagian-bagian kode yang diuji melalui tes otomatis. Berikut adalah beberapa poin penting dari laporan ini:

- 1) Cakupan kode total  
Dapat dilihat pada tabel 4. dibawah ini :

**Tabel 4.** Cakupan kode total component testing

% Stmts (Statements)	% Branch (Branches)	% Funcs (Functions)	% Funcs (Functions)
Menunjukkan persentase pernyataan dalam kode yang telah diuji. Secara keseluruhan, 65.27% pernyataan di semua file telah diuji.	Menunjukkan persentase cabang kondisi (seperti <i>if</i> atau <i>switch</i> ) yang telah diuji. Totalnya adalah 55.35%, Menunjukkan bahwa banyak kondisi yang belum diuji secara keseluruhan.	Persentase fungsi yang diuji adalah 29.05%, yang relatif rendah dibandingkan dengan pernyataan dan cabang kondisi.	Menunjukkan persentase baris kode yang telah diuji, dengan total 66.66%. Ini mencakup semua baris yang diuji dalam proyek ini.

2) Per-File/Cakupan Modul

- forgotPassword.tsx di components/Auth memiliki cakupan yang cukup baik dengan 67.69% pernyataan dan 97.16% baris kode diuji, tetapi hanya 55.35% cabang kondisi yang diuji.
- update-akun.tsx di components/Dashboard/Account/update memiliki cakupan yang lebih rendah dengan 70.96% pernyataan dan 71.7% baris kode diuji.
- create.tsx di components/Dashboard/Proposals/penelitian/create memiliki cakupan pernyataan sebesar 66.66% dan 65.62% baris kode diuji, dengan hanya 33.33% cabang kondisi yang diuji.
- create.tsx di components/Dashboard/Proposals/pengabdian/create memiliki cakupan pernyataan yang serupa dengan 66.66% pernyataan dan 65.62% baris kode diuji, dan 33.33% cabang kondisi yang diuji.
- settings.tsx di components/Dashboard/settings memiliki cakupan yang lebih baik dengan 75% pernyataan dan 66.66% baris kode diuji, tetapi hanya 50% cabang kondisi yang diuji.
- useUser.ts di components/hooks memiliki cakupan yang cukup baik dengan 75% pernyataan dan 100% baris kode diuji.
- auth.action.ts di graphql/actions memiliki cakupan penuh (100%) di

- semua metrik, menunjukkan bahwa semua kode di file ini diuji.
- penelitian.action.ts di graphql/actions juga memiliki cakupan penuh (100%) di semua metrik.
  - pengabdian.action.ts di graphql/actions memiliki cakupan penuh (100%) di semua metrik.
  - settings.action.ts di graphql/actions juga memiliki cakupan penuh (100%) di semua metrik.
  - helpers.ts di helpers memiliki cakupan penuh (100%) di semua metrik.
  - schemas.ts di schemas.ts juga memiliki cakupan penuh (100%) di semua metrik.

### 3) Uncovered Lines

Laporan ini juga mencantumkan nomor baris yang tidak tercakup oleh tes, misalnya,

*components/Auth/forgotPassword.tsx*

memiliki beberapa bagian yang tidak diuji (baris 66-69, 70-118, 33-34), *components/Dashboard/Account/update* (baris ...85, 104, 127-195), *components/Dashboard/Proposals/pencapaian/create* (baris ...05-146, 151, 155), *components/Dashboard/Proposals/pengabdian/create* (baris ...05-146, 151, 155), dan *components/Dashboard/settings* (baris 4).

### 4) Blackbox Testing

Blackbox Testing memungkinkan pengguna akhir menguji sistem dalam skenario dunia nyata, mencerminkan penggunaan sehari-hari. Mereka mengevaluasi fungsi dan fitur untuk memastikan semuanya berjalan lancar dan sesuai kebutuhan, mencakup dari tugas rutin hingga skenario kompleks.

**Tabel 5.** Pengujian *register blackbox testing*

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil
1	Register	Username:(Dikosongkan) Email:(Dikosongkan) Password:(Dikosongkan) PhoneNumber:(Dikosongkan)	Sistem menolak akses register dan menampilkan pesan: Data harus diisi!	Sesuai Harapan
2	Mengisi hanya username, lalu menekan tombol register	Username: newuser Email: (Dikosongkan) Password: (Dikosongkan) Phone Number: (Dikosongkan)	Sistem menolak akses register dan menampilkan pesan: "Semua field harus diisi"	Sesuai Harapan
3	Mengisi hanya email, lalu menekan tombol register	Username: (Dikosongkan) Email: newuser@example.com Password:(Dikosongkan) Phone Number: (Dikosongkan)	Sistem menolak akses register dan menampilkan pesan: "Semua field harus diisi"	Sesuai Harapan
4	Mengisi hanya password, lalu menekan tombol register	Username: (Dikosongkan) Email: (Dikosongkan) Password: password123 Phone Number: (Dikosongkan)	Sistem menolak akses register dan menampilkan pesan: "Semua field harus diisi"	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil
5	Mengisi hanya phone number, lalu menekan tombol register	Username: (Dikosongkan) Email: (Dikosongkan) Password: (Dikosongkan) Phone Number: 1234567890	Sistem menolak akses register dan menampilkan pesan: "Semua field harus diisi"	Sesuai Harapan
6	Mengisi email dengan format salah, lalu menekan tombol register	Username: newuser Email: wrongemailformat Password: password123 Phone Number: 1234567890	Sistem menolak akses register dan menampilkan pesan: "Email tidak valid"	Sesuai Harapan
7	Mengisi password kurang dari 8 karakter, lalu menekan tombol register	Username: newuser Email: newuser@example.com Password: short Phone Number: 1234567890	Sistem menolak akses register dan menampilkan pesan: "Password harus minimal 8 karakter"	Sesuai Harapan
8	Mengisi semua isian data register dengan benar, lalu menekan tombol register	Username: newuser Email: newuser@example.com Password: password123 Phone Number: 1234567890	Sistem menerima akses register dan menampilkan pesan: "Silahkan cek email anda untuk Aktivasi akun!"	Sesuai Harapan

Tabel 6. Pengujian aktivasi akun *blackbox testing*

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mengosongkan semua isian kode aktivasi, lalu menekan tombol verifikasi	Kode Aktivasi: (Dikosongkan)	Sistem menolak aktivasi dan menampilkan pesan: "Kode aktivasi harus diisi"	Sesuai Harapan
2	Mengisi kode aktivasi yang salah, lalu menekan tombol verifikasi	Kode Aktivasi: 1234	Sistem menolak aktivasi dan menampilkan pesan: "Kode aktivasi salah"	Sesuai Harapan
3	Mengisi kode aktivasi yang benar, lalu menekan tombol verifikasi	Kode Aktivasi: 5678	Sistem menerima aktivasi dan menampilkan pesan: "Akun berhasil diaktifkan"	Sesuai Harapan
4	Mengisi kode aktivasi yang benar dan kadaluarsa, lalu menekan tombol verifikasi	Kode Aktivasi: 9999	Sistem menolak aktivasi dan menampilkan pesan: "Expired"	Sesuai Harapan

**Tabel 4. 36** Pengujian *login blackbox testing*

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mengosongkan semua isian data login, lalu menekan tombol <i>login</i>	<i>Email: (Dikosongkan)</i> <i>Password: (Dikosongkan)</i>	Sistem menolak akses login dan menampilkan pesan: "Email dan password harus diisi"	Sesuai Harapan
2	Mengisi hanya email, lalu menekan tombol <i>login</i>	<i>Email: <a href="mailto:user@example.com">user@example.com</a></i> <i>Password: (Dikosongkan)</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan: " <i>Password</i> harus diisi"	Sesuai Harapan
3	Mengisi hanya password, lalu menekan tombol <i>login</i>	<i>Email: (Dikosongkan)</i> <i>Password: password123</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan: "Email harus diisi"	Sesuai Harapan
4	Mengisi email dengan format salah, lalu menekan tombol <i>login</i>	<i>Email: wrongemailformat</i> <i>Password: password123</i>	Sistem menolak akses <i>login</i> dan Menampilkan pesan: "Email tidak <i>valid</i> "	Sesuai Harapan
5	Mengisi email yang tidak terdaftar, lalu menekan tombol <i>login</i>	<i>Email: <a href="mailto:notfound@example.com">notfound@example.com</a></i> <i>Password: password123</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan: " <i>Email/password</i> tidak ditemukan"	Sesuai Harapan
6	Mengisi email yang benar dan password yang salah, lalu menekan tombol <i>login</i>	<i>Email: <a href="mailto:user@example.com">user@example.com</a></i> <i>Password: wrongpassword</i>	Sistem menolak akses <i>login</i> dan menampilkan pesan: " <i>Email/password</i> salah"	Sesuai Harapan
7	Mengisi email dan password yang benar, lalu menekan tombol <i>login</i>	<i>Email: <a href="mailto:user@example.com">user@example.com</a></i> <i>Password: password123</i>	Sistem menerima akses <i>login</i> dan mengarahkan ke <i>dashboard</i>	Sesuai Harapan

**Tabel 7.** Pengujian *forgot password blackbox testing*

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mengosongkan isian email, lalu menekan tombol <i>submit</i>	<i>Email: (Dikosongkan)</i>	Sistem menolak permintaan dan menampilkan pesan: "Email harus diisi"	Sesuai Harapan
2	Mengisi email yang tidak terdaftar, lalu menekan tombol	<i>Email: <a href="mailto:unknown@example.com">unknown@example.com</a></i>	Sistem Menolak permintaan dan menampilkan pesan: "Email	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
	<i>submit</i>		tidak ditemukan"	
3	Mengisi email yang terdaftar, lalu menekan tombol <i>submit</i>	<i>Email:</i> <a href="mailto:user@example.com">user@example.com</a>	Sistem menerima permintaan dan menampilkan pesan: "Silahkan cek email anda untuk <i>reset password</i> "	Sesuai Harapan

Tabel 8. Pengujian *reset password blackbox testing*

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mengosongkan semua isian <i>password</i> , lalu menekan tombol <i>submit</i>	<i>Password:</i> (Dikosongkan) <i>Confirm Password:</i> (Dikosongkan)	Sistem menolak permintaan dan menampilkan pesan: " <i>Password</i> dan <i>Konfirmasi Password</i> harus diisi"	Sesuai Harapan
2	Mengisi <i>password</i> baru dan konfirmasi <i>password</i> berbeda, lalu menekan tombol <i>submit</i>	<i>Password:</i> <i>password123</i> <i>Confirm Password:</i> <i>password456</i>	Sistem Menolak permintaan dan menampilkan pesan: " <i>Password</i> dan <i>Konfirmasi Password</i> tidak cocok"	Sesuai Harapan
3	Mengisi <i>password</i> baru dengan format yang benar, lalu menekan tombol <i>submit</i>	<i>Password:</i> <i>newpassword123</i> <i>Confirm Password:</i> <i>newpassword123</i>	Sistem menerima permintaan dan menampilkan pesan: " <i>Password</i> berhasil diubah, silakan login"	Sesuai Harapan

Tabel 9. Pengujian buat usulan baru blackbox testing

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mengosongkan semua isian pada <i>form</i> identitas usulan, lalu menekan tombol <i>next</i>	Semua <i>field</i> pada <i>form</i> identitas usulan kosong	Sistem menolak permintaan dan menampilkan pesan: "Data jangan ada yang kosong!"	Sesuai Harapan
2	Mengisi sebagian isian pada <i>form</i> identitas usulan, lalu menekan tombol <i>next</i>	<i>Field</i> semester dan tahun akademik diisi, tetapi ketua, nidn, prodi, dan jafung kosong	Sistem Menolak permintaan dan menampilkan pesan: "Data jangan ada yang kosong!"	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
3	Mengisi semua isian pada <i>form</i> identitas usulan, lalu menekan tombol <i>next</i>	Semua <i>field</i> pada <i>form</i> identitas usulan diisi	Sistem menerima permintaan dan berpindah ke <i>form</i> anggota	Sesuai Harapan
4	Mengisi sebagian isian pada <i>form</i> anggota, lalu menekan tombol <i>next</i>	Field <i>anggota_1</i> , <i>nidn_anggota_1</i> , <i>jafung_anggota_1</i> , dan <i>prodi_anggota_1</i> diisi, tetapi field anggota lain kosong	Sistem menerima permintaan dan berpindah ke <i>form</i> judul penelitian	Sesuai Harapan
5	Mengosongkan semua isian pada <i>form</i> judul usulan, lalu menekan tombol <i>next</i>	Semua <i>field</i> pada <i>form</i> judul usulan kosong	Sistem menolak permintaan dan menampilkan pesan: "Data jangan ada yang kosong!"	Sesuai Harapan
6	Mengisi semua isian pada <i>form</i> judul penelitian, lalu menekan tombol <i>next</i>	Semua <i>field</i> pada <i>form</i> judul penelitian diisi	Sistem menerima permintaan dan berpindah ke halaman <i>review &amp; submit</i>	Sesuai Harapan
7	Menekan tombol <i>submit</i> pada halaman <i>review &amp; submit</i> tanpa ada perubahan data	Semua <i>field</i> telah diisi lengkap	Sistem menerima permintaan, menyimpan data penelitian, dan menampilkan pesan sukses: "Usulan berhasil dibuat!"	Sesuai Harapan

**Tabel 10.** Pengujian generate surat rekomendasi prodi blackbox testing

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
1	Mencoba <i>generate</i> PDF untuk usulan yang belum disetujui	Status usulan: " <i>Submitted</i> "	Sistem menolak permintaan dan menampilkan pesan: " <i>Only approved can generate</i> PDF."	Sesuai Harapan
2	Mengisi nomor dokumen melalui <i>prompt</i> saat <i>generate</i> PDF	Status usulan: " <i>Approved</i> ", nomor dokumen dimasukkan melalui <i>prompt</i>	Sistem menerima nomor dokumen, menyimpan nomor tersebut, dan memulai proses <i>generate</i> PDF	Sesuai Harapan
3	<i>Generate</i> PDF untuk usulan yang telah disetujui	Status usulan: " <i>Approved</i> "	Sistem berhasil <i>generate</i> PDF, menampilkan <i>file</i> PDF pada <i>iframe</i> , dan menyediakan tombol unduh PDF	Sesuai Harapan

No	Skenario Pengujian	Uji Kasus	Hasil yang diharapkan	Hasil Pengujian
4	Mengirim <i>link</i> PDF melalui <i>WhatsApp</i> setelah <i>generate</i> PDF	Status usulan: " <i>Approved</i> ", PDF berhasil di- <i>generate</i>	Sistem berhasil mengirim pesan <i>WhatsApp</i> berisi <i>link</i> PDF kepada ketua usulan	Sesuai Harapan
5	Mencoba generate PDF tanpa ada data penelitian yang sesuai	Tidak ada data usulan yang sesuai dengan ID yang diberikan	Sistem menampilkan pesan error: " <i>Error loading data</i> "	Sesuai Harapan

Pada tabel yang telah disajikan sebelumnya, pengujian menggunakan metode *blackbox testing* untuk mengevaluasi fungsionalitas aplikasi. Tahap pengujian ini mencakup 7 modul dengan total 37 skenario pengujian.

Diketahui bahwa jumlah seluruh skenario adalah 37, dan jumlah skenario yang berhasil adalah 37. Maka, persentase keberhasilan aplikasi dapat dihitung sebagai berikut:

$$\frac{37}{37} \times 100\% = 100\%$$

Berdasarkan hasil perhitungan tersebut, dapat disimpulkan bahwa aplikasi memiliki persentase keberhasilan sebesar 100% berdasarkan pengujian *blackbox testing*.

#### IV. Kesimpulan

Implementasi sistem informasi untuk pengelolaan data penelitian dan pengabdian di Fakultas Teknik Universitas Pelita Bangsa memerlukan strategi komprehensif yang melibatkan pemilihan teknologi tepat, desain user-friendly, dan integrasi sistem yang baik. Implementasi arsitektur *microservices* meningkatkan fleksibilitas, dan responsivitas sistem, serta meminimalisir downtime. Selain itu, *microservices* mempermudah akses dan mengurangi duplikasi data, memastikan sumber data terpusat yang dapat diakses oleh semua layanan, sehingga

meningkatkan efisiensi dan integritas data di universitas.

Penelitian ini menyarankan agar implementasi sistem dilakukan dengan perencanaan matang, khususnya dalam desain arsitektur *microservices* yang fleksibel. Uji coba menyeluruh penting untuk memastikan sistem optimal dan bebas dari hambatan seperti integrasi kompleks atau isu keamanan. Ke depan, pengembangan fitur kolaborasi interaktif seperti real-time data sharing juga direkomendasikan untuk meningkatkan manfaat bagi dosen dan stakeholder.

#### Daftar Pustaka

- [1] S. Hendraputra et al., "Pengantar Teknologi dan Informasi," Yaya, 2021.
- [2] Alamsyah Agit, Hasmawati, Sahurri, Sari Yunus, and Fitriyani Syukri, "Peran Dan Kontribusi Teknologi Dalam Menunjang Potensi Ekonomi Masyarakat Di Wilayah Pedesaan," *Semin. Nas. Pariwisata dan Kewirausahaan*, vol. 2, pp. 263–271, 2023, doi: 10.36441/snpk.vol2.2023.126.
- [3] Institute for Development of Economics and Finance (INDEF), "Peran Platform Digital Terhadap Pengembangan Umkm Di Indonesia," pp. 1–46, 2024.

- [4] “PENELITIAN & PENGABDIAN”, [Online]. Available: <https://eksyar.pelitabangsa.ac.id/penelitian-pengabdian>
- [5] N. Siagian, T. E. Tamba, H. H. O. Situmorang, and H. Samosir, “Aplikasi Apotek Berbasis Web Menggunakan Arsitektur Microservices (Studi Kasus Apotek Glen, Kab.Toba),” *J. Appl. Technol. Informatics Indones.*, vol. 1, no. 2, pp. 22–28, 2021, doi: 10.54074/jati.v1i2.35.
- [6] M. M. R. Luntajo and F. Hasan, “Optimalisasi Potensi Pengelolaan Zakat di Indonesia melalui Integrasi Teknologi,” *Al-'Aqdu J. Islam. Econ. Law*, vol. 3, no. 1, p. 14, 2023, doi: 10.30984/ajiel.v3i1.2577.
- [7] Baehaqi, M. S. Basit, R. E. Indrajit, and R. D. Kurniawan, “Front End Learning Management System Development Using the Nextjs Framework,” *J. Tek. Inform.*, vol. 4, no. 4, pp. 899–911, 2023, doi: 10.52436/1.jutif.2023.4.4.1273
- [8] M. Iqbal, W. Handoko, and A. K. Syahputra, “Optimalisasi E-Learning Melalui Implementasi Microservice Untuk Peningkatan Skalabilitas Dan Efisiensi Pembelajaran Online,” *Jurdimas (Jurnal Pengabd. Kpd. Masyarakat) R.*, vol. 6, no. 3, pp. 439–444, 2023, doi: 10.33330/jurdimas.v6i3.2496.
- [9] F. Gusmiadi, I. Yasin, N. Penulis, K.: Finki, and G. Submitted, “Perancangan Sistem Pengelolaan Data Piutang pada PT Atosim Lampung,” *J. Teknol. Dan Sist. Inf.*, vol. 4, no. 2, pp. 121–126, 2023.
- [10] I. O. D. Brata, “ANALISIS DAN PERANCANGAN SISTEM,” *J. Akunt. Bisnis dan Ekon.*, 2021, doi: 10.33197/jabe.vol7.iss1.2021.629.
- [11] I. Sudiarta, I. Indrayana, I. Suasnawa, I. Atmaja, K. Indah, and P. Sunu, “User Requirement and Use Case Diagram for Traveler Tracking Application in Tourist Destination,” pp. 1376–1380, 2023, doi: 10.5220/0010965700003260.
- [12] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, “A Different Approach on Automated Use Case Diagram Semantic Assessment,” *Int. J. Intell. Eng. Syst.*, 2021, doi: 10.22266/IJIES2021.0228.46.
- [13] S. Al-Fedaghi, “UML Sequence Diagram: An Alternative Model,” *Int. J. Adv. Comput. Sci. Appl.*, 2021, doi: 10.14569/IJACSA.2021.0120576.
- [14] C. Alvin, B. Peterson, and S. Mukhopadhyay, “Static generation of UML sequence diagrams,” *Int. J. Softw. Tools Technol. Transf.*, 2021, doi: 10.1007/s10009-019-00545-z.