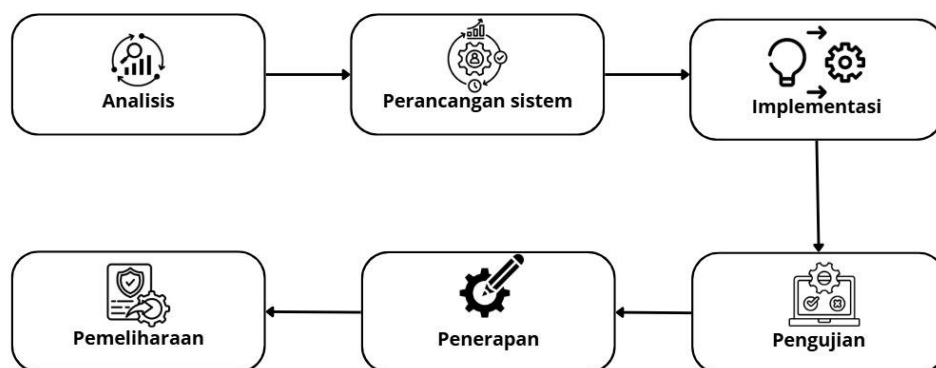


BAB II

PRODUK

2.1 Metode Penelitian

Metode ini menggunakan metode *waterfall*, metode *waterfall* merupakan suatu pendekatan dalam pengembangan perangkat lunak yang dilakukan secara berurutan, di mana setiap tahapannya mengalir seperti air terjun. Setiap fase dalam proses ini harus dilalui secara sistematis dengan tujuan akhir untuk membangun sebuah perangkat lunak komputer[10]. Tahapan pengembangan sistem berikut disusun untuk memastikan bahwa sistem yang dibangun dapat memenuhi kebutuhan pengguna serta selaras dengan tujuan dari pengembangan aplikasi *ILEDIN*, dapat dilihat sebagai berikut:



Gambar 2. 1 Perancangan Aplikasi

Berdasarkan gambar berikut dapat diketahui bahwa terdapat beberapa langkah dalam penelitian ini, penjelasan untuk masing-masing langkah adalah sebagai berikut:

2.1.1 Analisis kebutuhan

Tahap awal dalam pengembangan sistem informasi akreditasi dimulai dengan proses identifikasi kebutuhan pengguna. Langkah ini dilakukan melalui studi lapangan dan analisis terhadap permasalahan yang dihadapi oleh pengguna dalam proses akreditasi. Hasil dari tahapan ini menunjukkan bahwa terdapat berbagai kendala dalam pengumpulan, pengelolaan, dan pelaporan data akreditasi yang masih dilakukan secara manual dan kurang terstruktur. Oleh karena itu, diperlukan sistem yang mampu mendukung proses akreditasi secara lebih efektif, efisien, dan terintegrasi guna menunjang kelancaran pelaksanaan akreditasi pada Politeknik Harapan Bersama. Berdasarkan survei dari pihak SPMI ditemukan bahwa :

- a. Selama ini dalam melakukan proses akreditasi, SPMI masih menggunakan cara manual dengan menggunakan *excell*.
- b. Membutuhkan pengelolaan data secara cepat dan akurat untuk meningkatkan efesiensi waktu proses akreditasi
- c. Diperlukan sebuah sistem input dan output data dari pihak Prodi dan Dosen dalam pengawasan SPMI.

Kebutuhan fungsional :

- a. Manajemen akun (membuat akun untuk setiap Prodi).
- b. Pengelolaan dan pengawasan data oleh SPMI dari data yang dimasukan Prodi dan Dosen.
- c. Pengaturan akses *form* oleh SPMI.

- d. Melakukan manajemen tahun akademik.

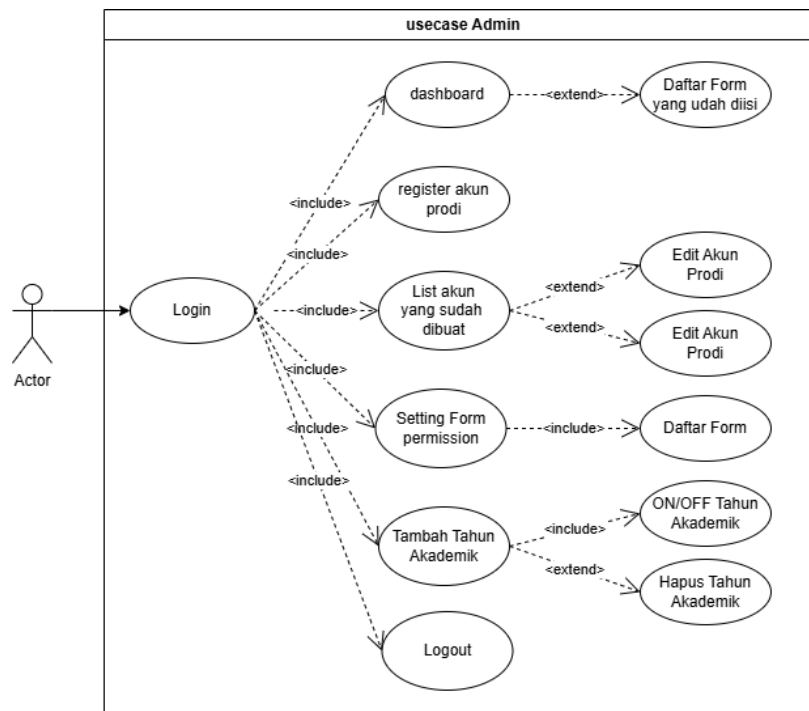
2.1.2 Perancangan Sistem

1. Perancangan UML

Unified Modeling Language (UML) adalah metode pemodelan visual yang digunakan untuk merancang dan mengembangkan perangkat lunak berbasis objek [11]. Berikut UML dari aplikasi ini:

a. *Use Case Diagram*

Use case diagram merupakan representasi visual yang menggambarkan perilaku atau fungsi-fungsi yang dijalankan oleh suatu sistem yang dirancang[12]. Diagram ini menjelaskan cara sistem merespons aksi atau interaksi dari aktor sesuai dengan situasi atau kebutuhan tertentu. Secara umum, *use case diagram* menggambarkan perilaku sistem dari perspektif pengguna, termasuk fitur-fitur utama yang dapat digunakan atau dijalankan oleh masing-masing aktor. Berikut merupakan *use case diagram* aplikasi *ILEDIN* seperti yang dapat dilihat pada Gambar berikut:

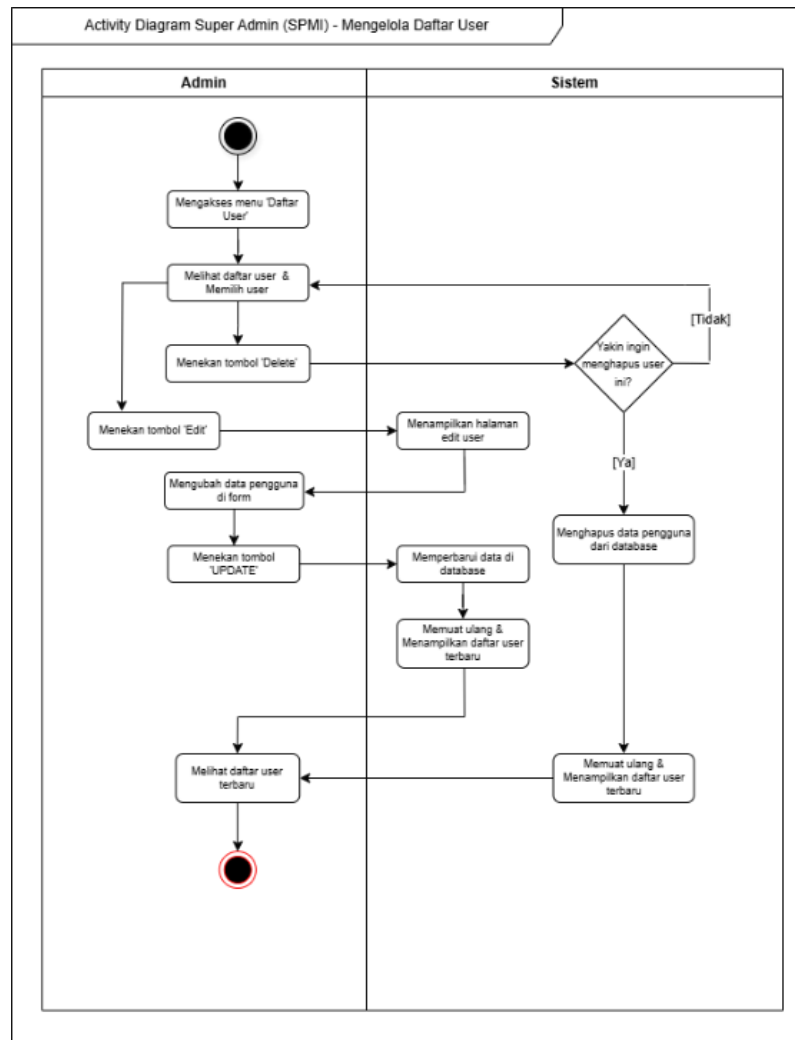


Gambar 2. 2 Use Case Diagram SPMI

b. Activity Diagram

Activity diagram merupakan representasi visual yang menunjukkan alur proses atau fungsionalitas dalam sebuah sistem informasi. Diagram ini secara rinci menggambarkan titik awal dan akhir dari suatu alur kerja (*workflow*), aktivitas-aktivitas yang berlangsung di dalamnya, serta urutan terjadinya setiap aktivitas tersebut. Selain itu, *activity* diagram juga memungkinkan pemodelan proses yang berlangsung secara paralel atau bersamaan, sehingga sangat berguna dalam menggambarkan alur sistem yang kompleks[13].

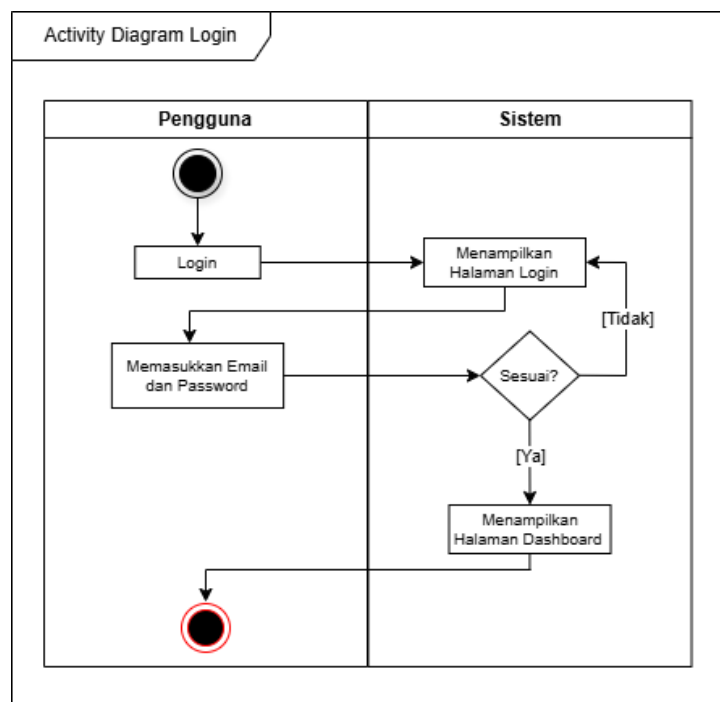
Berikut adalah *activity* diagram saat melakukan pendaftaran akun untuk prodi seperti yang dapat dilihat pada Gambar 2.3:



Gambar 2. 3 Activity Diagram Register

Activity diagram diatas memperlihatkan alur SPMI ketika menambah user baru pada sistem. Proses diawali dengan admin membuka menu *Tambah User* dan mengisi *form* pendaftaran *user*, kemudian menekan tombol *Simpan User*. Selanjutnya, sistem akan melakukan pengecekan apakah *password* dan konfirmasi *password* yang dimasukkan sudah sama, jika tidak sesuai maka sistem akan menampilkan pesan kesalahan "*Password Tidak Cocok*". Jika *password* cocok, sistem akan memeriksa apakah email yang

dimasukkan sudah terdaftar; jika email telah digunakan, akan muncul pesan *error* "Email Sudah Terdaftar". Namun, jika email belum terdaftar, data *user* akan disimpan, sistem menampilkan notifikasi bahwa proses berhasil, dan SPMI akan diarahkan ke halaman *dashboard* untuk melihat bahwa *user* baru telah berhasil ditambahkan. Berikut adalah *activity* diagram saat melakukan *login* website seperti yang dapat dilihat pada Gambar 2.4:

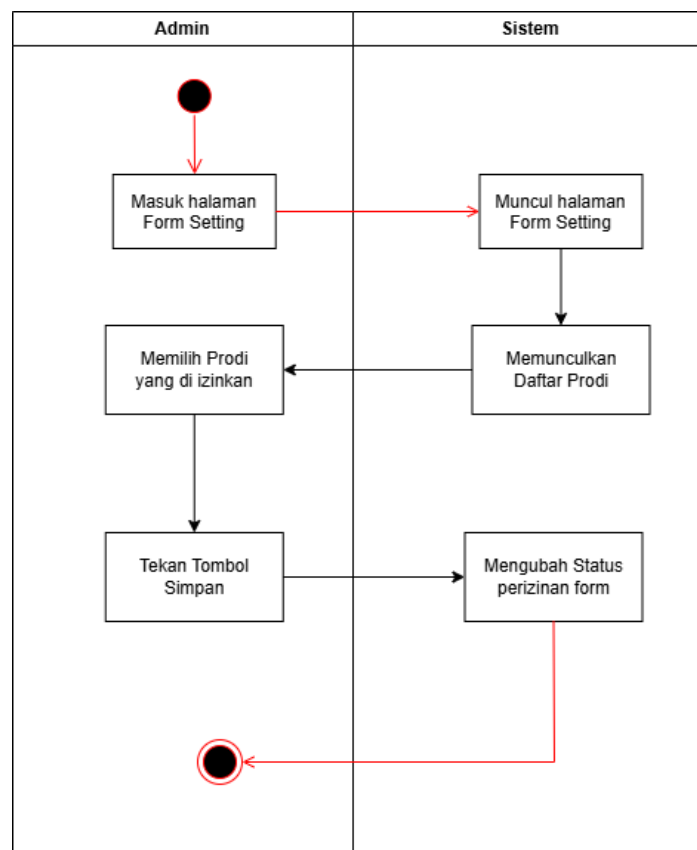


Gambar 2. 4 Activity Diagram Login

Activity diagram diatas menggambarkan alur *login* pengguna ke dalam sistem. Proses diawali ketika pengguna membuka halaman *login* dan mengisi kolom email serta *password*. Setelah itu, sistem akan melakukan pengecekan terhadap data yang diberikan. Jika data tidak valid, pengguna tetap berada di halaman *login* dan diminta untuk

memasukkan data kembali. Sebaliknya, apabila data yang dimasukkan benar, sistem akan mengarahkan pengguna ke halaman *dashboard* sebagai bukti bahwa *login* telah berhasil.

Berikut adalah *activity* diagram saat menjalankan fitur *Form Setting* seperti yang dapat dilihat pada Gambar 2.5:

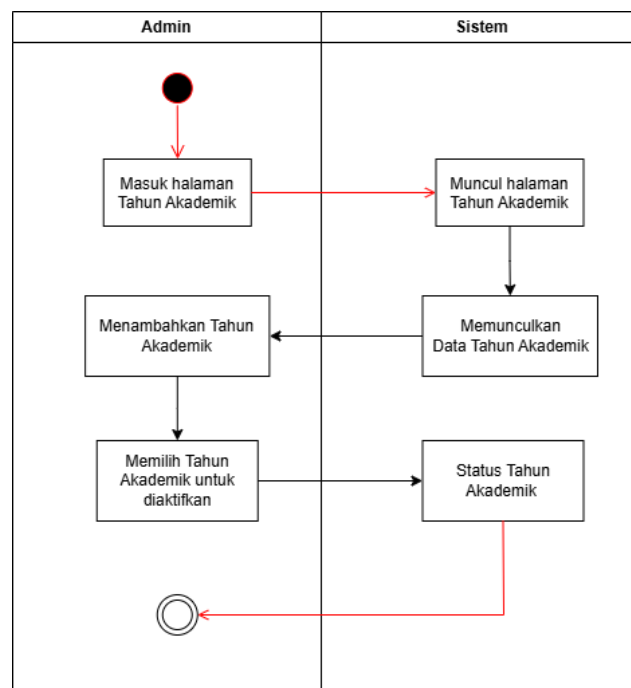


Gambar 2. 5 Activity Diagram Form Settings

Activity diagram diatas menjelaskan alur yang dilakukan SPMI saat mengelola perizinan pada menu *Form Setting*. Proses diawali ketika SPMI mengakses halaman *Form Setting*, di mana sistem langsung menampilkan daftar program studi yang tersedia. Selanjutnya, SPMI memilih program studi mana saja yang diberikan hak akses ke

form. Setelah pilihan ditentukan, SPMI menekan tombol Simpan, sehingga sistem memperbarui status perizinan *form* berdasarkan pengaturan yang telah dipilih. Proses ini selesai setelah sistem berhasil menyimpan perubahan tersebut.

Berikut adalah *activity* diagram saat menjalankan fitur Tahun Akademik seperti yang dapat dilihat pada Gambar 2.6:



Gambar 2. 6 Activity Diagram Tahun Akademik

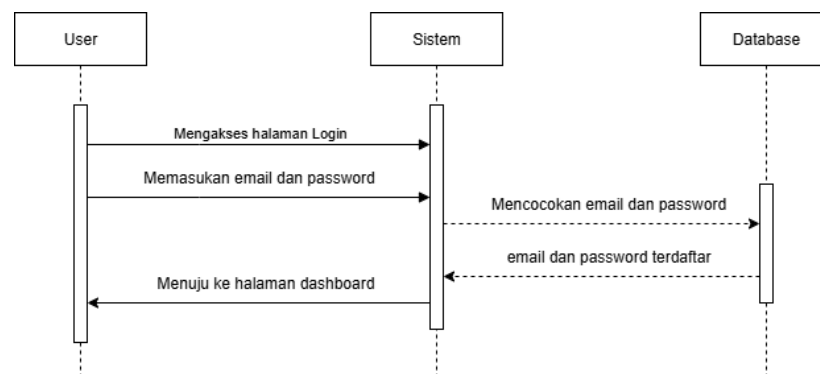
Activity diagram diatas menunjukkan proses pengelolaan tahun akademik oleh SPMI sebagai Admin. Proses dimulai dengan mengakses halaman Tahun Akademik, kemudian sistem menampilkan data Tahun Akademik yang sudah ada sebelumnya. Jika belum ada tahun akademik maka Admin bisa menambahkan tahun akademik terlebih dahulu sebelum diaktifkan. Jika sudah menentukan Tahun

Akademik mana yang akan diaktifkan, maka admin hanya perlu mengubah status menjadi aktif saja.

c. *Sequence Diagram*

Sequence Diagram merupakan alat populer dalam pengembangan sistem berbasis objek untuk menggambarkan interaksi antar objek. Diagram ini juga digunakan dalam perancangan antarmuka, baik untuk menguraikan proses bisnis menjadi aktivitas kecil guna mengidentifikasi kebutuhan pengguna, maupun untuk menganalisis perilaku sistem dalam setiap interaksi demi merancang tampilan yang sesuai [14].

Berikut adalah *Sequence* diagram saat menjalankan fitur *login*, seperti yang dapat dilihat pada Gambar 2.7:

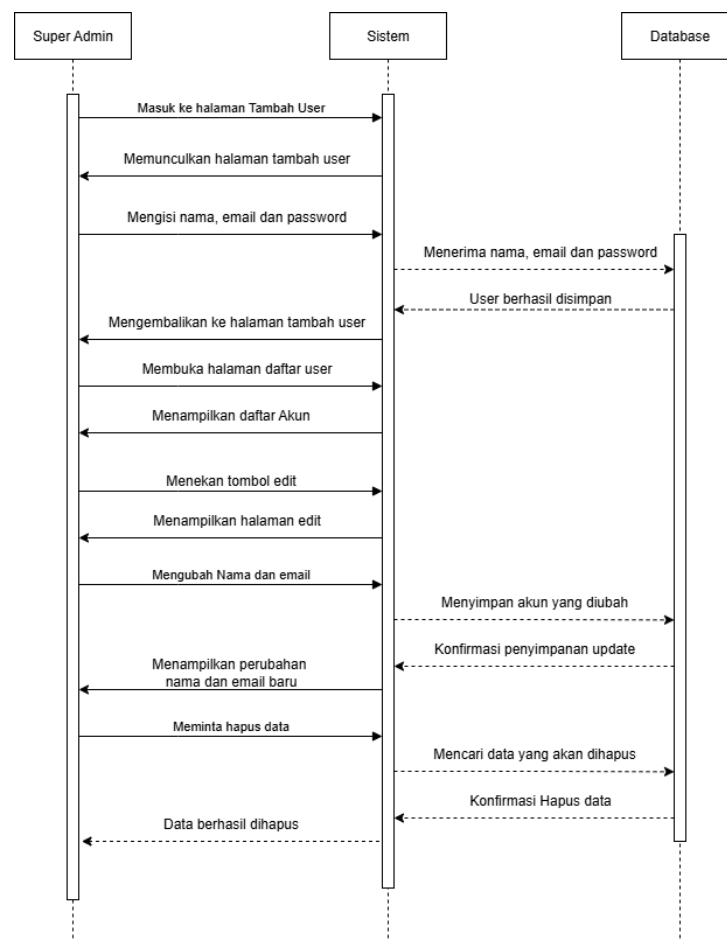


Gambar 2. 7 *Sequence Diagram Login*

Sequence Diagram diatas menjelaskan Proses dimulai ketika pengguna membuka halaman *login* dan memasukkan *email* serta *password*. Data tersebut kemudian dikirim oleh sistem ke *database* untuk dilakukan pencocokan informasi. Setelah *database*

memverifikasi bahwa *email* dan *password* sesuai dengan data yang tersimpan, sistem akan mengarahkan pengguna ke halaman *dashboard* sebagai tanda bahwa login berhasil dilakukan.

Berikut adalah *Sequence* diagram saat menjalankan fitur registrasi, seperti yang dapat dilihat pada Gambar 2.8:

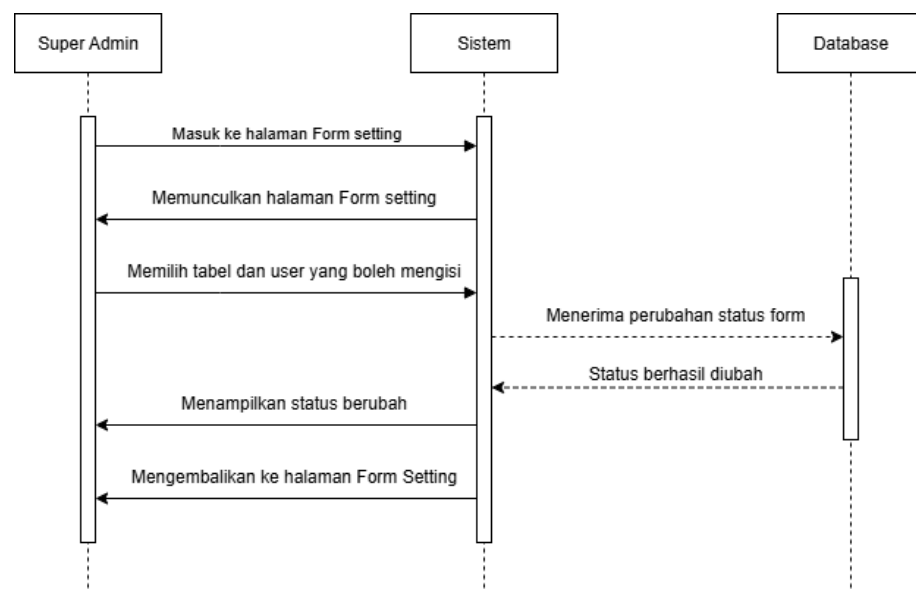


Gambar 2. 8 Sequence Diagram Registrasi

sequence Diagram tersebut menggambarkan proses pengelolaan akun oleh Super Admin, mulai dari penambahan, pengubahan, hingga penghapusan data *user*. Proses dimulai saat Super Admin membuka halaman tambah *user*, mengisi informasi seperti

nama, *email*, dan *password*, lalu sistem menyimpan data tersebut ke dalam *database*. Selanjutnya, Super Admin mengakses halaman daftar *user* dan dapat memilih akun yang ingin diedit. Setelah melakukan perubahan dan menyimpannya, sistem memperbarui data di *database* dan menampilkan hasil perubahan tersebut. Terakhir, Super Admin juga memiliki opsi untuk menghapus akun, di mana sistem akan mencari dan menghapus data *user* yang dipilih dari *database*.

Berikut adalah *Sequence* diagram saat menjalankan fitur pengaturan *form user*, seperti yang dapat dilihat pada Gambar 2.9:

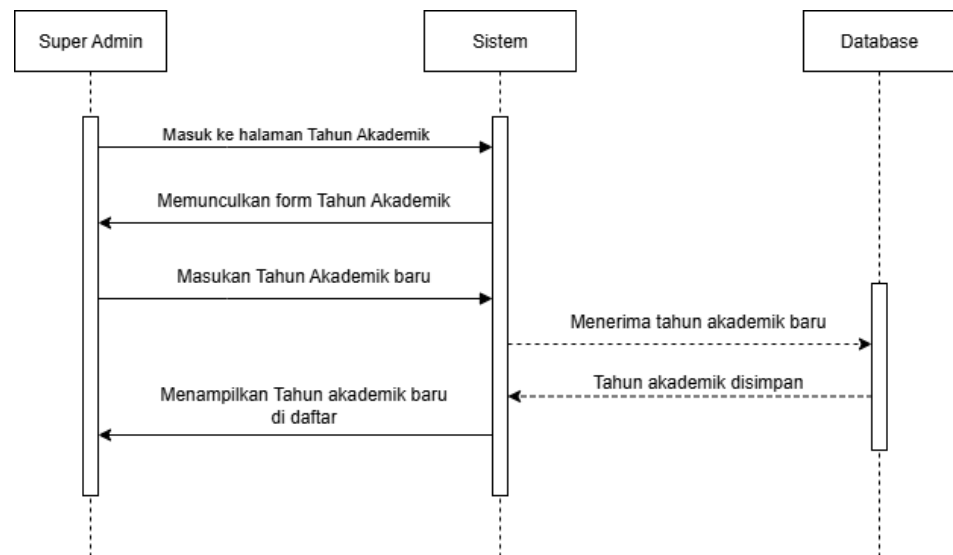


Gambar 2. 9 *Sequence* Diagram Fitur pengaturan *form user*

Sequence Diagram tersebut menjelaskan proses pengaturan form yang dilakukan oleh Super Admin. Proses dimulai saat Super Admin mengakses halaman *Form Setting*, kemudian sistem menampilkan halaman tersebut. Selanjutnya, admin memilih tabel dan pengguna yang diizinkan untuk mengisi *form*. Perubahan status *form*

kemudian dikirimkan oleh sistem ke *database*, dan setelah berhasil diperbarui, sistem menampilkan status terbaru serta mengembalikan admin ke halaman *Form Setting*.

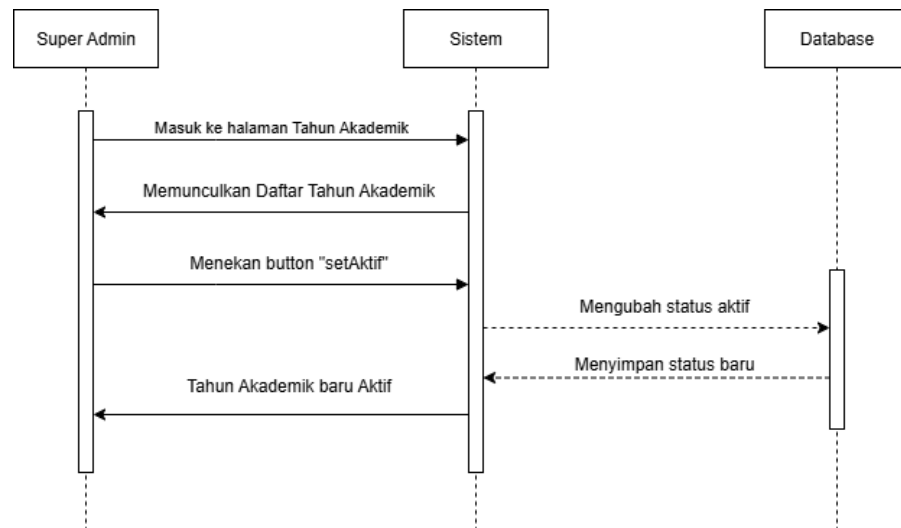
Berikut adalah *Sequence* diagram saat menjalankan fitur menambah tahun akademik, seperti yang dapat dilihat pada Gambar 2.10:



Gambar 2. 10 *Sequence Diagram* Fitur menambah tahun akademik

Sequence diagram di atas menggambarkan proses penambahan Tahun Akademik oleh Super Admin. Proses dimulai saat Super Admin masuk ke halaman Tahun Akademik, lalu sistem menampilkan *form input*. Admin kemudian mengisi Tahun Akademik baru, yang dikirim ke sistem dan diteruskan ke *database*. Setelah data disimpan, sistem menampilkan Tahun Akademik baru dalam daftar sebagai konfirmasi bahwa penambahan berhasil.

Berikut adalah *Sequence* diagram saat menjalankan fitur memilih tahun akademik, seperti yang dapat dilihat pada Gambar 2.11:



Gambar 2. 11 Sequence Diagram Fitur Memilih Tahun Akademik

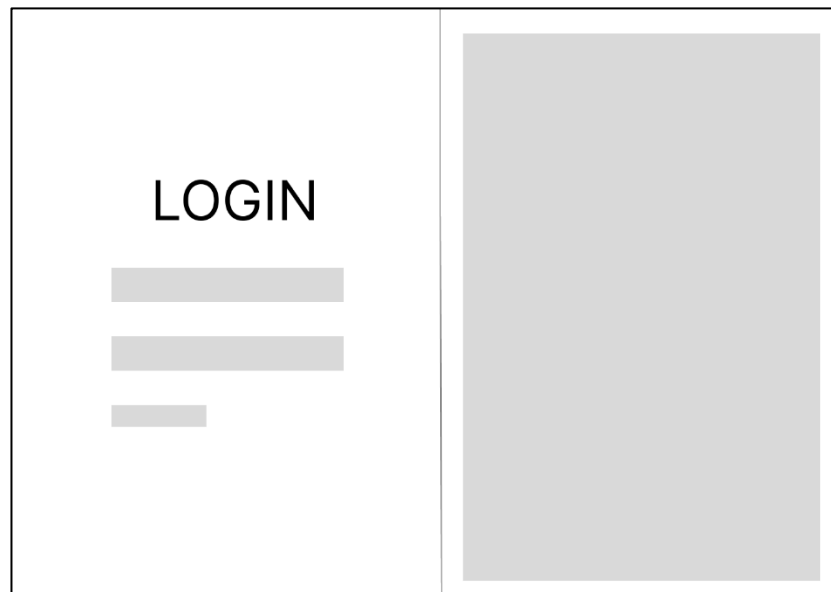
Sequence diagram di atas menjelaskan proses mengaktifkan Tahun Akademik oleh Super Admin. Proses dimulai saat Super Admin mengakses halaman Tahun Akademik, lalu sistem menampilkan daftar tahun yang tersedia. Setelah admin menekan tombol "*setAktif*", sistem mengubah status tahun akademik yang dipilih menjadi aktif dan menyimpan perubahan tersebut ke *database*. Terakhir, sistem menampilkan bahwa Tahun Akademik baru telah aktif.

2. User Interface

Pada tahap ini dilakukan pembuatan antarmuka pengguna yang akan menjadi panduan dalam pembuatan aplikasi ini. Pembuatan *UI* dipertimbangkan dari segi kenyamanan pengguna mulai dari tampilan dan tata letak.

a. Halaman *Login*

Tampilan halaman login, dapat dilihat pada Gambar 2.12.

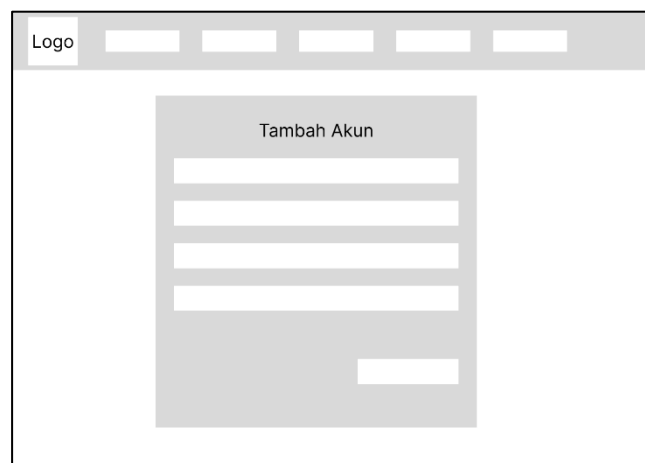


Gambar 2. 12 User Interface login

Halaman *login* adalah halaman awal yang harus diakses oleh pengguna sebelum dapat masuk ke *dashboard* aplikasi. Pada halaman ini, pengguna diwajibkan untuk memasukkan *email* dan *password* yang valid.

b. Halaman Tambah *User/Prodi*

Tampilan halaman Tambah *User/Prodi*, bisa dilihat pada Gambar 2.13.

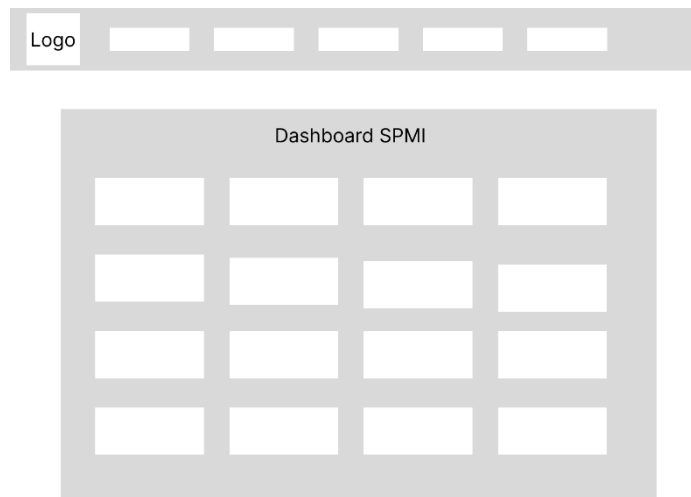


Gambar 2. 13 User Interface Tambah User/Prodi

Halaman Tambah *User/Prodi* merupakan halaman dimana SPMI dapat membuat akun dengan mengisi semua data seperti nama, *email* dan *password*.

c. Halaman *Dashboard*

Tampilan halaman *dashboard*, bisa dilihat pada Gambar 2.14.

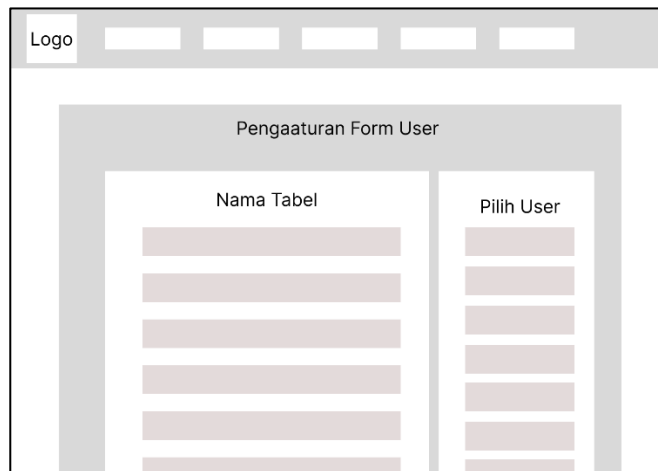


Gambar 2. 14 User Interface Dashboard

Pada halaman ini pengguna dapat melihat semua *form* yang diisi oleh prodi dan dosen yang telah diisi.

d. Halaman Pengaturan Form

Tampilan halaman Pengaturan form, dapat dilihat pada Gambar 2.15.

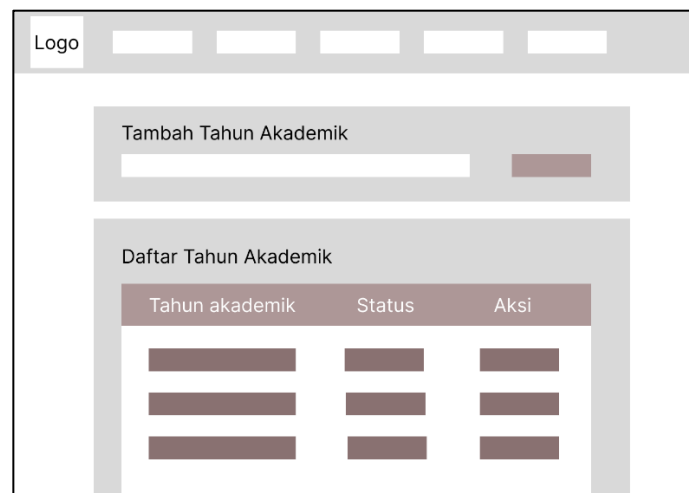


Gambar 2. 15 User Interface pengaturan form

Halaman ini merupakan salah satu fitur dimana pihak SPMI sebagai super admin memiliki akses untuk mengatur izin dari tiap *form* yang nantinya akan diisi oleh admin setiap prodi.

e. Halaman Tahun Akademik

Tampilan halaman Tahun Akademik dapat dilihat pada Gambar 2.16.



Gambar 2. 16 User Interface Tahun Akademik

Halaman Tahun Akademik ini merupakan halaman yang didalamnya menyajikan 2 fitur lagi, yaitu Tambah tahun akademik, dan Aktifkan

Tahun Akademik

1) Tambah Tahun Akademik

Pada bagian ini SPMI menambahkan tahun akademik untuk digunakan sebagai pilihan untuk set aktif tahun akademik

2) *SetAktif* Tahun Akademik

Pada bagian ini SPMI mengaktifkan tahun akademik yang tadi baru ditambahkan sebagai status Tahun Akademik Aktif.

3. Perancangan *Database*

Aplikasi *ILEDIN* memanfaatkan *MySQL* sebagai sistem manajemen basis datanya. Pilihan ini memungkinkan penyimpanan dan pengelolaan yang efisien untuk berbagai jenis informasi penting. Data yang disimpan mencakup akun pengguna, data yang diinput oleh prodi dan dosen sebagai penilaian untuk peningkatan akreditasi. Struktur dan relasi setiap tabel juga perlu dirancang supaya tidak terjadi kesalahan pengambilan dan pengisian data.

2.1.3 Implementasi

1. Pembuatan *Website* Aplikasi

Setelah tahap perancangan diselesaikan, langkah berikutnya adalah merealisasikan sistem ke dalam bentuk aplikasi yang dapat digunakan. Proses implementasi ini dilakukan dengan memanfaatkan beberapa perangkat lunak utama sebagai pendukung pengembangan aplikasi. Dapat dilihat sebagai berikut:

a. Framework Laravel

Digunakan sebagai kerangka utama dalam pengembangan aplikasi *ILEDIN* berbasis website.

b. Tailwind css

Sebuah *framework* yang biasa digunakan untuk membuat tampilan *UI* jadi lebih menarik dan dinamis

c. MySQL

Digunakan sebagai sistem manajemen basis data, yang menyimpan berbagai informasi dan data penting seperti akun pengguna, data isian *form* dan tahun akademik.

d. Laragon

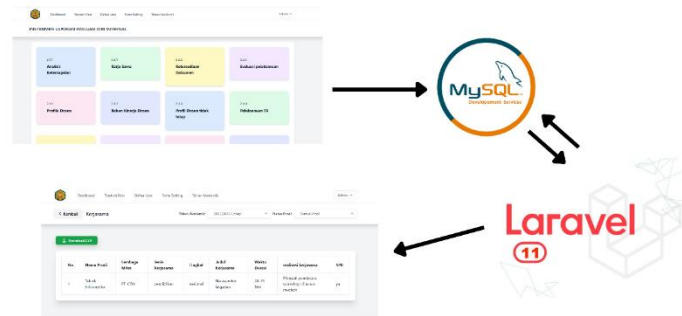
Digunakan sebagai *server local* dalam mendukung pengembangan aplikasi secara *offline* sebelum diterapkan ke *server online*.

Pengembangan *website* untuk aplikasi *ILEDIN* memiliki dua tujuan utama yang saling melengkapi:

a. *Landing Page* untuk Pemasaran: *Website* ini berfungsi sebagai gerbang digital yang menarik dan mudah digunakan oleh *user* SPMI. Halaman ini dirancang untuk memperkenalkan aplikasi *ILEDIN* secara efektif, menampilkan fitur-fitur unggulan, dan menyoroti manfaat yang ditawarkan. Tujuannya adalah saat aplikasi digunakan admin SPMI bisa dengan mudah memahami cara menggunakan aplikasi dari *ILEDIN*.

e. *Dashboard* dan Tabel data: Sisi lain dari *website* ini adalah sebuah antarmuka khusus yang memungkinkan admin SPMI untuk melihat dan

mengkoordinasi data konten aplikasi dengan mudah. *Dashboard* ini menyediakan alat-alat yang diperlukan untuk melihat dan memastikan data yang dimasukan oleh prodi dan dosen sudah benar sesuai dengan kriteria penilaian.



Gambar 2. 17 Alur Penampilan Data

Gambar di atas merupakan alur untuk menampilkan data yang dimasukan oleh prodi dan dosen yang tersimpan dalam *database*. Pada *dashboard* awal menampilkan semua *form*, lalu nanti SPMI bisa memilih tabel atau *form* mana yang akan diperiksa isinya.

2. Integrasi Sistem

Tahap integrasi sistem dalam pengembangan aplikasi *ILEDIN* merupakan langkah krusial yang menghubungkan data utama antara SPMI, Prodi dan Dosen.

Beberapa aspek penting dalam tahap integrasi ini meliputi:

- Konfigurasi *Route*: Menetapkan dan mengoptimalkan *route* sesuai dengan *controller* supaya bisa digunakan.
- Implementasi Penyimpanan data: Memastikan bahwa data yang dimasukan oleh prodi bisa tersimpan pada *database*.
- Menampilkan data: Memastikan bahwa semua data yang ada di

database bisa ditampilkan pada halaman *website*.

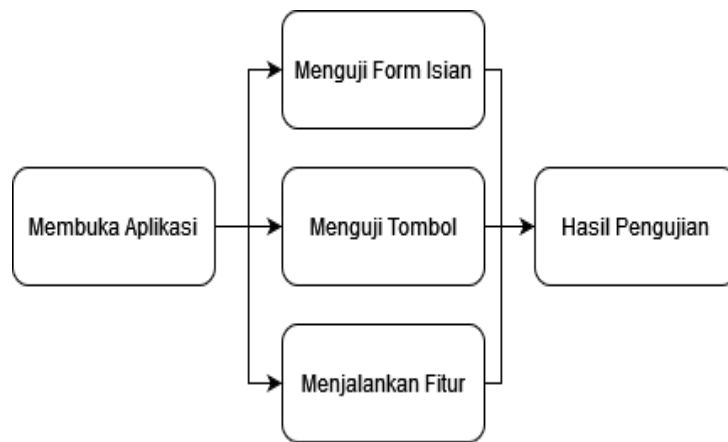
- d. Penanganan Kesalahan: Mengembangkan mekanisme untuk mengatasi berbagai skenario kesalahan koneksi atau *respons server*.
- e. Optimisasi Performa: Memastikan permintaan data dilakukan secara efisien untuk meminimalkan penggunaan sumber daya dan meningkatkan kecepatan *respons*.
- f. Keamanan Data: Menerapkan langkah-langkah keamanan seperti *enkripsi* dan *autentikasi* untuk melindungi setiap akun. Proses integrasi ini memastikan bahwa data mengalir dengan lancar, menciptakan pengalaman pengguna yang responsif dan dinamis. Keberhasilan tahap ini sangat penting untuk memastikan fungsionalitas dan keandalan secara keseluruhan.

2.1.4 Pengujian dan Evaluasi

Black box testing adalah metode pengujian kualitas perangkat lunak yang menitikberatkan pada pengujian fungsionalitas dari sistem. Tujuan dari pengujian ini adalah untuk mengidentifikasi kesalahan pada fungsi yang tidak berjalan dengan semestinya, kesalahan antarmuka, masalah dalam struktur data, performa yang tidak sesuai, serta kesalahan saat proses inisialisasi dan terminasi [15]

Pengujian Aplikasi melalui *Black Box Testing* secara langsung kepada pengguna. Tujuan pengujian fungsional aplikasi "*ILEDIN*" adalah untuk memastikan bahwa semua fitur utama dan fungsionalitas yang diharapkan berjalan sesuai spesifikasi. Ini melibatkan identifikasi pengguna target serta kasus penggunaan utama aplikasi. Metode *Black Box Testing* dipilih karena fokusnya pada pengalaman pengguna,

objektivitasnya, dan kemampuannya untuk memvalidasi fungsionalitas tanpa perlu memahami detail implementasi internal. Alur dari skenario pengujian dapat dilihat pada Gambar 2.18.

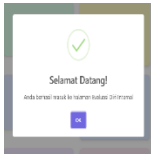
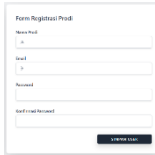
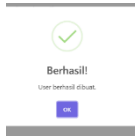
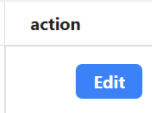
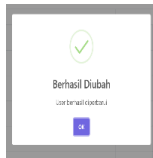


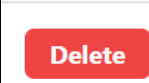
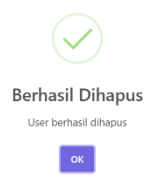
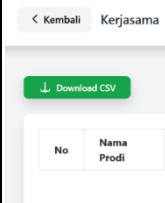
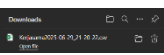
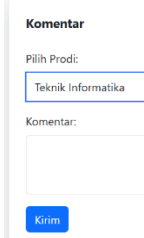
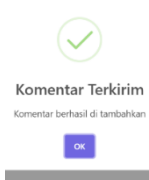
Gambar 2. 18 *Alur Skenario Pengujian*

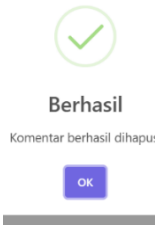
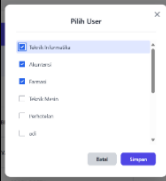
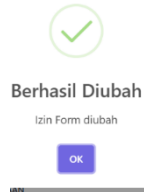
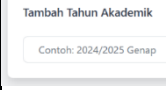
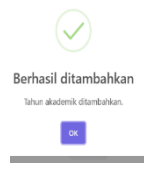
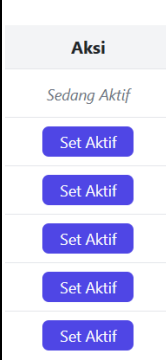
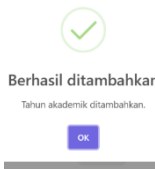
Skenario pengujian direncanakan dengan seksama untuk memastikan aplikasi diuji secara menyeluruh dan menyeluruh. Setiap fitur dalam aplikasi akan diperiksa secara detail, termasuk setiap form isian dan tombol yang ada. Berbagai variasi pengujian akan diterapkan untuk mengevaluasi kehandalan dan konsistensi aplikasi dalam berbagai situasi.

Berikut hasil pengujian, dapat dilihat pada Tabel 2.1:

Tabel 2. 1. Hasil Pengujian

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Memasukan <i>Email</i> dan <i>Password</i> .		Berhasil melakukan <i>login</i> dan masuk halaman <i>dashboard</i>		VALID
2.	Menambahka n akun untuk Prodi		Data akun berhasil disimpan		VALID
3.	Tombol edit untuk melakukan proses edit akun pada user admin prodi		Menyimpan perubahan dalam <i>database</i>		VALID

4.	Tombol hapus digunakan untuk menghapus akun yang sudah tidak digunakan lagi		Menghapus akun yang dipilih		VALID
5.	Download data yang ditampilkan pada halaman		Berhasil melakukan download file		VALID
6.	Mengirimkan komentar ke prodi		Komentar berhasil dikirim		VALID

7.	Menghapus komentar yang sudah tidak terpakai	  	Tidak bisa login dengan menampilkan pesan error		VALID
8.	Memilih prodi yang diizinkan untuk mengisi form		Menyimpan perubahan status form prodi		VALID
9.	Menambah Tahun Akademik baru		Tahun akademik baru ditambahkan		VALID
10.	Menentukan tahun akademik yang dipakai		Menyimpan perubahan tahun akademik		VALID

Hasil dari Aplikasi *ILEDIN* adalah SPMI mempunyai beberapa

fitur seperti Filter nama prodi dan tahun akademik, Tambah akun prodi, mengatur izin untuk akses *form* pada prodi, menambahkan dan menentukan tahun akademik aktif. Fitur tersebut diperlukan SPMI sebagai super admin untuk mengelola Prodi dan Dosen supaya mengisi data akreditasi dengan benar

2.1.5 Penerapan

Setelah seluruh fitur sistem selesai diimplementasikan dan melalui proses pengujian menggunakan metode *black box testing*, aplikasi *ILEDIN* mulai diterapkan secara terbatas di lingkungan internal institusi, khususnya oleh pihak SPMI dan Program Studi (Prodi). Tahapan ini bertujuan untuk menguji fungsionalitas sistem dalam kondisi nyata dan mengevaluasi sejauh mana sistem mampu memenuhi kebutuhan pengelolaan data akreditasi secara efektif.

Pada tahap ini, pengguna dari pihak SPMI mulai menggunakan sistem untuk:

- a. Mengatur akun prodi dan dosen, termasuk pemberian akses terhadap *form* akreditasi.
- b. Melakukan pengaturan *form*, seperti menentukan siapa saja yang bisa mengisi *form* tersebut.
- c. Memantau atau memeriksa proses pengisian data akreditasi dari setiap prodi oleh SPMI secara berkala.
- d. Mengelola Tahun Akademik aktif supaya data yang dikumpulkan sesuai dengan periode evaluasi akreditasi.

2.1.6 Pemeliharaan

Tahap akhir dalam metode *Waterfall* adalah pemeliharaan sistem

(*maintenance*), yang bertujuan untuk menjaga agar aplikasi tetap berjalan optimal serta mampu mengikuti perubahan kebutuhan pengguna di masa mendatang. Setelah sistem *ILEDIN* diterapkan dan digunakan secara terbatas oleh pihak SPMI dan Prodi, dilakukan evaluasi menyeluruh terhadap pengalaman pengguna, efisiensi sistem, dan kelengkapan fitur.

Berdasarkan hasil penggunaan awal dan umpan balik yang diperoleh dari pengguna sistem, beberapa saran pengembangan lanjutan yang dapat diterapkan pada tahap pemeliharaan antara lain:

a. Validasi data otomatis

Validasi diperlukan agar data yang dimasukan pengguna tidak terjadi kesalahan atau duplikasi, kolom kosong, atau data numerik yang masih bisa diisi dengan text. Hal ini diperlukan supaya menjaga integritas dan kualitas data akreditasi yang dikumpulkan.

b. Ekspor laporan ke format standar

Agar proses penyusunan laporan akreditasi jadi lebih mudah dan efisien.

2.2 Kesimpulan dan Saran

2.2.1 Kesimpulan

Penelitian ini berhasil mengembangkan aplikasi *ILEDIN* sebagai sistem informasi berbasis *web* yang mendukung proses pengelolaan data akreditasi secara digital dan terintegrasi. Adapun kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Aplikasi *ILEDIN* mampu menyediakan fitur-fitur utama seperti pengisian form akreditasi oleh prodi dan dosen, pengaturan form oleh SPMI, serta manajemen tahun akademik. Fitur-fitur ini menjawab kebutuhan proses akreditasi yang selama ini dilakukan secara manual dan memakan waktu
2. Sistem *ILEDIN* dibangun dengan pendekatan visual menggunakan UML dan menerapkan metode pengembangan perangkat lunak yang terstruktur, sehingga dapat mendukung fleksibilitas dan keberlanjutan pengelolaan data akreditasi berdasarkan standar *IAPS* 4.0 yang mencakup 23 kriteria penilaian.
3. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa seluruh fitur dalam aplikasi *ILEDIN* berjalan dengan baik dan valid. Sistem dapat menyimpan, mengubah, dan menghapus data secara efektif serta memberikan pengalaman pengguna yang stabil dan responsif.

2.2.2 Saran

Untuk penyempurnaan aplikasi *ILEDIN* ke depan, beberapa hal yang perlu diperhatikan adalah sebagai berikut:

1. Meningkatkan integrasi antar fitur agar data yang diinputkan dapat saling terhubung secara otomatis.
2. Menambahkan fitur notifikasi atau pengingat supaya Prodi dan Dosen bisa menerima peringatan dari SPMI untuk pengisian

data akreditasi.

3. Pengembangan fitur ekspor laporan akreditasi secara otomatis ke dalam format yang sesuai standar nilai akreditasi secara otomatis.
4. Peningkatan antarmuka pengguna (*UI/UX*) dengan menyesuaikan keperluan dan fungsi dari setiap fitur untuk membantu pengisi.