
Software Requirements Specification

for

Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban

Version 1.0

Prepared by

**Fiora Berliana Putri 201524045
Lamda Richo Vanjaya Sumaryadi 201524049
Nauval Ozora Mahadri 201524055**

Politeknik Negeri Bandung

28 February 2023

Table of Content

1. Introduction.....	1
1.1. Purpose	1
1.2. Document Conventions	1
1.3. Intended Audience and Reading Suggestions	2
1.4. Project Scope.....	3
1.5. References	3
2. Overall Description	4
2.1. Product Perspective	4
2.2. Product Features	4
2.2.1. Mahasiswa Umum	4
2.2.2. Mahasiswa Staf Lembaga	5
2.2.3. Admin	5
2.3. User Classes and Characteristics	6
2.4. Operating Environment	6
2.5. Design and Implementation Constraints	6
2.5.1. Tools	6
2.5.2. Bahasa Pemrograman.....	8
2.5.3. Waktu Pengembangan Aplikasi	9
2.5.4. Keamanan	9
2.5.5. Database	9
2.6. User Documentation.....	9
2.7. Business Rules.....	10
2.7.1. Mahasiswa.....	10
2.7.2. Lembaga.....	10
2.7.3. Admin	10
2.8. Assumptions and Dependencies	10
2.8.1. Assumption	10
2.8.2. Dependencies	11
3. System Design.....	11
3.1. Business Process	11
3.1.1. Core Process.....	11
3.1.2. Management Process	11
3.1.3. Support Process.....	12
1. Pendaftaran Akun	12
2. Autentikasi Akun.....	12
3. Pendaftaran Mahasiswa pada Lembaga.....	12
4. Pendaftaran Mahasiswa Polban pada Kegiatan	13
3.2. Class Diagram	13
3.3. Domain Model.....	14
3.3. Use Case Model	15

3.3.1.	Use Case Diagram.....	15
3.3.2.	Fully Dressed	16
3.3.3.	System Sequence Diagram	40
3.3.4.	Operation Contract.....	47
3.3.5.	Sequence Diagram	50
4.	External Interface Requirements	61
4.1.	User Interfaces.....	61
4.1.1.	Staf Lembaga	61
4.1.2.	Mahasiswa.....	62
4.1.3.	Admin	63
4.2.	Hardware Interfaces	63
4.2.1.	Laptop atau Komputer	63
4.2.2.	Keyboard.....	63
4.2.3.	Mouse.....	64
4.2.4.	Monitor	64
4.2.5.	Backup Storage	64
4.3.	Software Interfaces.....	64
4.3.1.	Gmail API	64
4.4.	Communications Interface	64
4.4.1.	Functional Requirements Communication	64
4.4.2.	Standar Komunikasi yang Digunakan	65
5.	Other Nonfunctional Requirements	65
5.1.	Performance Requirements	65
5.2.	Security Requirements	65
5.3.	Software Quality Attributes	65
5.3.1.	Usability.....	65
5.5.2.	Scalability	66
5.5.3.	Compatibility	66
5.5.4.	Maintainability.....	66
5.5.5.	Reliability.....	66
Appendix A: Glossary.....		68
Appendix B: User Persona		71
Appendix C: Test Case		73

Revision History

Date	Description	Version	Author
28.02.23	Pembuatan : 1. Tujuan atau <i>Purpose</i> 2. Deskripsi Umum Dokumen 3. Ruang Lingkup 4. Referensi 5. <i>Product Perspective</i>	0.1	Lamda Richo Vanjaya Sumaryadi
01.03.23	Pembuatan <i>User Classes and Characteristics</i>	0.2	Fiora Berliana Putri
02.03.23	Pembuatan : 1. <i>Product Features</i> 2. <i>Operating Environment</i> 3. <i>Design and Implementation Constraints</i>	0.3	Nauval Ozora Mahadri
03.03.23	Pembuatan <i>Performance Requirements</i>	0.4	Nauval Ozora Mahadri
07.03.23	Pembuatan <i>Security Requirements</i> dan <i>Software Quality Attributes</i>	0.5	Nauval Ozora Mahadri
08.03.23	1. Penyesuaian <i>Operating Environment</i> sesuai dengan ketentuan sebelumnya 2. Pembuatan <i>Design and Implementation Constraints</i>	0.6	Lamda Richo Vanjaya Sumaryadi
09.03.23	Pembuatan : 1. User Documentation 2. Bussiness Rules 3. Assumptions and Dependencies 4. Bussiness Process 5. Domain Model 6. Use Case Model 7. Hardware Interfaces 8. Software Interfaces 9. Communications Interfaces 10. Intended Audience and Reading Suggestions	0.9	Fiora Berliana Putri, Lamda Richo Vanjaya Sumaryadi, Nauval Ozora Mahadri
10.03.23	Pembuatan : 1. User Interfaces 2. Appendix A : Glossary 3. Appendix B : User Persona DKM, Kammi, dan Assalam	1.0	Fiora Berliana Putri, Lamda Richo Vanjaya Sumaryadi, Nauval Ozora Mahadri
23.05.23	Perubahan: 1. Class Diagram 2. Domain Model	1.1	Fiora Berliana Putri, Lamda Richo Vanjaya Sumaryadi, Nauval Ozora Mahadri

	3. Use Case Diagram 4. Detailed Design 5. Appendix C : Test Case 6. Functional dan Non-Functional Requirement		
--	--	--	--

1. Introduction

1.1. Purpose

Dokumen *Software Requirements Specification* (SRS) untuk Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban. Tujuan dari penulisan dokumen ini adalah untuk memberikan penjelasan mengenai perangkat lunak yang akan dibangun baik berupa gambaran umum maupun penjelasan detail dan menyeluruh.

Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban adalah aplikasi yang dapat mengelola mahasiswa untuk mengikuti/mendaftarkan diri sebagai pengurus Lembaga Dakwah Kampus Polban (LDK), seperti DKM, Assalam, dan Kammi. Selain itu, aplikasi ini dapat memberikan akses informasi seputar kegiatan dari lembaga di Masjid Luqmanul Hakim secara transparan kepada staf lembaga, mengelola mahasiswa atau organisasi yang ingin menggunakan sarana dan prasarana masjid, mengelola mahasiswa untuk berpartisipasi pada suatu kegiatan dari lembaga yang ada, dan mengelola pengurus lembaga dalam mengatur/mengurus kegiatannya.

Pengguna dari dokumen ini adalah para pengembang (*developer*) Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban Berbasis Website dan pengguna (*user*) dari perangkat lunak ini atau mereka yang terlibat dalam sistem. Dokumen ini akan digunakan sebagai bahan acuan dalam proses pengembangan dan sebagai bahan evaluasi pada saat proses pengembangan perangkat lunak maupun di akhir pengembangannya. Dengan adanya dokumen SRS ini diharapkan pengembangan perangkat lunak akan lebih terarah dan lebih terfokus serta tidak menimbulkan ambiguitas terutama bagi pengembang perangkat lunak ini.

1.2. Document Conventions

Dokumen ini secara garis besar terdiri dari lima bab dengan rincian sebagai berikut:

1.2.1. Bab 1 Introduction

Merupakan pengantar dokumen SRS yang berisi tujuan penulisan dokumen, deskripsi umum dokumen, saran pembacaan dokumen, ruang lingkup, dan referensi.

1.2.2. Bab 2 Overall Description

Menjelaskan terkait perspektif produk, fitur produk, pengelompokan dan karakteristik setiap pengguna, serta *environment* yang dapat digunakan dalam pengoperasian aplikasi. Lalu

terdapat pula pembahasan terkait implementasi aplikasi serta asumsi dan ketergantungan yang digunakan dalam pengembangan aplikasi ini.

1.2.3. Bab 3 System Design

Menjelaskan terkait *domain model* serta *use case model* dalam aplikasi yang dibuat untuk mempermudah *stakeholder* memahami aplikasi ini.

1.2.4. Bab 4 External Interface Requirements

Menjelaskan terkait *interface* yang ada dalam aplikasi ini seperti *user interface*, *software interface*, *hardware interface*, dan *communication interface*.

1.2.5. Bab 5 Other Nonfunctional Requirement

Menjelaskan terkait *non-functional requirement* yang akan menunjang berjalannya aplikasi.

1.3. Intended Audience and Reading Suggestions

1.3.1. Developer

Dokumen SRS akan menjadi acuan utama bagi para *developer* untuk membangun perangkat lunak sesuai dengan yang telah ditentukan dalam dokumen. Dokumen ini akan memberikan panduan tentang fitur dan fungsionalitas yang harus diimplementasikan dalam sistem, serta persyaratan teknis dan keamanan yang harus dipenuhi.

1.3.2. Manajer Proyek

Dalam dokumen ini, manajer proyek dapat memastikan bahwa proyek pengembangan perangkat lunak Masjid Lukmanul Hakim ini berjalan sesuai dengan rencana. Dokumen ini akan memberikan berbagai informasi seperti anggaran, sumber daya, batasan waktu, serta memastikan bahwa semua kebutuhan pengguna telah dipenuhi.

1.3.3. User

Pengguna aplikasi dapat memahami batasan, syarat, spesifikasi, dan juga fitur yang ada pada aplikasi. Selain itu, dokumen ini juga dapat membantu pengguna dalam memberikan *feedback* dan saran pada pengembang perangkat lunak.

1.3.4. Tester

Dokumen SRS ini akan menjadi pemandu utama bagi penguji untuk memastikan bahwa sistem aplikasi masjid berfungsi dengan benar dan memenuhi semua persyaratan yang telah

ditetapkan. Dokumen ini akan memberikan informasi tentang fitur dan fungsionalitas yang harus diuji, serta batasan dan kendala yang harus diawasi dalam pengujian.

1.3.5. Marketing

Dokumen SRS ini dapat membantu *marketing* untuk memahami kelebihan serta kekurangan dari aplikasi. Selain itu juga marketing dapat mengetahui spesifikasi, fitur, dan juga nilai lebih aplikasi yang bisa dibuat untuk mempromosikan penggunaan aplikasi. Selain itu, *marketing* juga bisa mendapat *feedback* dari calon pengguna berdasarkan dokumen ini.

1.4. Project Scope

Perangkat lunak yang akan dikembangkan adalah Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban berbasis Website, yang akan digunakan pada LDK (Lembaga Dakwah Kampus). Ruang lingkup dari aplikasi ini, meliputi fitur-fitur sebagai berikut.

- Penjadwalan Kegiatan Lembaga
- Pengelolaan Data Sarana dan Prasarana
- Pengelolaan Transaksi Peminjaman Sarana dan Prasarana
- Pengelolaan Data Akun
- Autentikasi Akun
- Pengelolaan Data Staf pada Lembaga
- Pengelolaan Data Pendaftaran pada Kegiatan
- Pengelolaan Panitia Kegiatan

1.5. References

Dokumen - dokumen yang digunakan sebagai referensi dalam pembuatan SRS ini sebagai berikut :

- IEEE Std 830-1993, IEEE Recommended Practice for Software Requirement Specifications.
- <https://repository.unikom.ac.id/35935/1/SoftwareRequirementsSpecification-Aplikasi%20Logistik%20Alat%20Tulis%20Kantor.pdf>
- <http://staff.unila.ac.id/meizano/2016/03/23/template-dan-contoh-dokumen-spesifikasi-kebutuhan-perangkat-lunak-skpl-software-requirements-specification-srs-document/>
- https://www.academia.edu/40612313/KOTA_208_Software_Requirements_Specification_version_2_5

2. Overall Description

2.1. Product Perspective

Aplikasi ini merupakan sebuah sistem yang terpisah dan mandiri, yang dirancang untuk membantu pengelolaan lembaga dan kegiatan di Masjid Lukmannul Hakim Polban.

Aplikasi ini akan berinteraksi dengan beberapa *stakeholder*, termasuk pengurus lembaga dan mahasiswa umum. Aplikasi ini tidak bergantung pada sistem lain, dan tidak ada sistem lain yang bergantung pada aplikasi ini.

2.2. Product Features

Aplikasi ini dibuat untuk pengguna yang ingin menjadi bagian dari lembaga yang ada di dalam Masjid Lukmannul Hakim Polban, bagi pengguna yang ingin mengikuti kegiatan yang diselenggarakan oleh lembaga masjid, serta bagi pengguna yang ingin meminjam inventaris yang dimiliki masjid. Berikut fitur yang utama yang dimiliki aplikasi ini:

2.2.1. Mahasiswa Umum

Untuk mahasiswa umum, aplikasi ini dapat digunakan sebagai berikut:

- Mahasiswa dapat membuat akun pada aplikasi dengan syarat memasukkan nama, nim, username, email polban, dan password [REQ-F-01]
- Aplikasi mampu melakukan autentikasi akun mahasiswa saat login berupa data email polban dan password yang sudah didaftarkan sebelumnya [REQ-F-02]
- Aplikasi mampu menampilkan informasi terkait daftar lembaga, yaitu DKM, Assalam, dan Kammi. Setiap lembaga akan menampilkan informasi umum, struktur kepengurusan, jadwal kegiatan, dan detail kegiatan. Khusus untuk lembaga DKM, terdapat informasi tambahan, yaitu informasi inventaris yang dimiliki masjid [REQ-F-03]
- Mahasiswa biasa dapat mengakses informasi mengenai daftar kegiatan pada setiap lembaga [REQ-F-04]
- Mahasiswa biasa dapat melihat informasi terkait data inventaris/sarana dan prasarana yang dimiliki masjid, yang terdiri dari data nama inventaris, deskripsi, tanggal kepemilikan, status ketersediaan, dan list peminjam [REQ-F-05]
- Mahasiswa biasa dapat melakukan peminjaman inventaris dengan mengisi data nama penanggung jawab, tujuan, tanggal, upload surat izin yang bertipe pdf [REQ-F-06]

2.2.2. Mahasiswa Staf Lembaga

Untuk mahasiswa sebagai staf lembaga, aplikasi ini dapat digunakan sebagai berikut:

- Aplikasi mampu melakukan autentikasi akun lembaga saat login berupa data email polban dan password yang sudah didaftarkan sebelumnya [REQ-F-07]
- Aplikasi mampu menyimpan data serta mengelola kegiatan di setiap lembaganya yang terdiri dari nama kegiatan, deskripsi kegiatan, jadwal pelaksanaan, dan struktur kepengurusan [REQ-F-08]
- Aplikasi mampu menambahkan dan menghapus daftar kegiatan dari setiap lembaga [REQ-F-09]
- Mahasiswa staff lembaga DKM mampu menghapus data inventaris [REQ-F-10]
- Mahasiswa staff lembaga DKM dapat menambahkan data inventaris dengan data nama inventaris, tanggal kepemilikan, deskripsi, status ketersediaan, dan list peminjam [REQ-F-15]
- Mahasiswa staff lembaga DKM dapat mengubah seluruh data inventaris seperti data nama inventaris, tanggal kepemilikan, deskripsi, status ketersediaan, dan list peminjam [REQ-F-16]
- Mahasiswa staff lembaga DKM dapat melihat semua list inventaris dengan setiap data inventarisnya berupa data nama inventaris, tanggal kepemilikan, deskripsi, status ketersediaan, dan list peminjam [REQ-F-17]

2.2.3. Admin

Untuk admin, aplikasi ini dapat digunakan sebagai berikut:

- Admin mampu membuat akun baru untuk mahasiswa staf lembaga dengan memasukkan profile picture, nama, username, email polban dan password [REQ-F-11]
- Aplikasi mampu mencatat data akun baru untuk mahasiswa umum berupa data profile picture, nama, nim, username, email polban, dan password [REQ-F-12]
- Admin dapat menghapus akun mahasiswa umum / staf lembaga [REQ-F-13]
- Admin mampu membuat lembaga berupa data profile picture dan nama lembaga [REQ-F-14]
- Admin mampu memperbarui akun mahasiswa staf lembaga berupa data profile picture, nama, nim, username, email polban, dan password [REQ-F-18]
- Admin dapat membaca lembaga berupa data profile picture dan nama lembaga [REQ-F-19]

- Admin mampu memperbarui lembaga hanya berupa data profile picture dan nama lembaga [REQ-F-20]
- Admin mampu menghapus lembaga [REQ-F-21]

2.3. User Classes and Characteristics

Karakteristik dari pengguna pada aplikasi kami tertera pada tabel berikut:

Pengguna	Definisi
Admin	Dapat melakukan create, read, update, delete akun
Staf Lembaga	Dapat melakukan create, read, update, delete kegiatan Dapat melakukan create, read, update, delete struktur kepengurusan Dapat melakukan peminjaman inventaris masjid
Mahasiswa Umum	Dapat melakukan peminjaman inventaris masjid

2.4. Operating Environment

Lingkungan Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban dapat berjalan di dua perangkat, yaitu Desktop (Website) dan Android (Website). Selain itu, aplikasi ini kompatibel dengan beberapa sistem operasi, seperti berikut ini:

- Windows 10 dan Windows 11
- Android 8.0+

Agar bisa dimuat dalam banyak perangkat, maka ukuran layar dari aplikasi ini dapat *responsive* dengan tersedia di ukuran layar small(kurang dari 640px), medium(641px hingga 1007px), dan high(lebih besar 1008px). Selain itu, resolusi layar akan tersedia pada kualitas HD, Full HD, dan 4K.

Selain harus kompatibel dengan berbagai OS dan perangkat, aplikasi juga mesti bekerja dengan baik pada koneksi internet dengan kecepatan 4G dan 5G agar bisa sesuai dengan standar kecepatan internet di Indonesia.

2.5. Design and Implementation Constraints

Untuk memaksimalkan kinerja aplikasi dan meminimalisir terjadi munculnya *bug* dan juga *error* pada aplikasi, ada beberapa batasan dan ruang lingkup yang harus dipatuhi oleh aplikasi.

Batasan - batasan tersebut diantaranya adalah :

2.5.1. Tools

Dalam pengembangan aplikasi, terdapat beberapa *tools software* yang berupa aplikasi dan *website* sebagai berikut:

1. Visual Studio Code

Tempat untuk menulis *code* baik *frontend* dan juga *backend* pada aplikasi. Alasan kami menggunakan Visual Studio Code karena aplikasinya yang sangat fleksibel bisa digunakan oleh banyak bahasa pemrograman dan juga komunitasnya yang sangat besar memudahkan kami untuk menemukan solusi apabila terjadi hambatan saat pengembangan aplikasi.

2. Google Chrome

Tentu Aplikasi ini sangat diperlukan dalam pengembangan aplikasi karena dengan adanya browser maka setiap anggota tim dapat berhubungan melalui website atau aplikasi yang bisa di akses melalui Google Chrome. Selain itu banyak sekali solusi yang bisa ditemukan saat pengembangan aplikasi melalui *browser* Google Chrome.

Alasan kami menggunakan Google Chrome sendiri karena *browser* ini mudah diakses dan umum digunakan oleh anggota kami.

3. Github

Digunakan sebagai media untuk mengakses kode program. Github memungkinkan para *programmer* untuk melakukan pekerjaan disaat yang bersamaan meskipun dalam perangkat dan di waktu yang berbeda.

4. Figma

Aplikasi ini memang ini memang dikhususkan untuk merancang sebuah UI aplikasi. Sehingga tentunya fitur - fitur yang tersedia akan membuat perancangan UI aplikasi menjadi lebih mudah dan dapat dikembangkan lebih jauh.

2.5.2. Bahasa Pemrograman

Pengembangan aplikasi akan menggunakan beberapa bahasa pemrograman. Dalam sebuah aplikasi, ada dua bagian yang harus dibangun yaitu *frontend* dan juga *backend*. Masing-masing bagian akan dikembangkan menggunakan bahasa pemrograman yang berbeda.

Untuk *frontend* sendiri diantaranya adalah :

i. HTML

HTML akan digunakan sebagai kerangka dalam membangun sebuah *website* sehingga sangat wajib untuk menyertakan HTML dalam pengembangan aplikasi ini.

ii. CSS (Bootstrap)

Dengan menggunakan *framework* Bootstrap, *website* bisa menjadi lebih responsif terhadap perubahan ukuran layar. Hal ini ditambah pula dengan terkenalnya Bootstrap dikalangan *programmer* sehingga memiliki komunitas yang cukup besar.

iii. JavaScript (React JS)

Ada beberapa alasan mengapa *react* dipilih untuk *frontend* aplikasi ini. Beberapa alasannya adalah :

- Reusability Components
- Memfasilitasi Keseluruhan Proses Penulisan Komponen
- Memastikan Rendering Lebih Cepat
- Menjamin Kestabilan Kode
- Dilengkapi dengan Perangkat Pengembangan
- Didukung oleh Komunitas yang Kuat

Sedangkan, untuk bagian *backend* akan dikembangkan dengan bahasa pemrograman yaitu Node JS. Sama seperti pemilihan React, Node JS juga memiliki banyak kelebihan yang membuatnya dipilih dalam pengembangan aplikasi ini, yaitu :

- Mampu mengeksekusi kode dengan cepat

- Mendukung proses single-thread
- Memiliki sifat non-blocking
- Mempunyai dukungan komunitas yang luas

2.5.3. Waktu Pengembangan Aplikasi

Aplikasi harus dikembangkan dalam waktu enam bulan, mulai dari tanggal awal pengembangan. Rencana pengembangan harus mencakup tahap-tahap yang jelas dan jangka waktu yang realistis.

2.5.4. Keamanan

Aplikasi harus memiliki mekanisme keamanan yang kuat untuk melindungi data pengguna, termasuk informasi pribadi seperti nomor telepon dan alamat *email*. Aplikasi juga harus mematuhi standar keamanan yang diperlukan untuk proses donasi *online*.

2.5.5. Database

Dengan mempertimbangkan jumlah pengguna aplikasi yang cukup banyak dan juga membutuhkan *database* yang fleksibel, maka *database* yang tepat untuk aplikasi ini adalah MongoDB.

2.6. User Documentation

Bantuan bagi user akan diberikan berupa dokumentasi sebagai pemandu penggunaan aplikasi. Beberapa dokumen panduannya adalah :

2.6.1. User Manual

Dokumen akan memberikan panduan yang detail tentang bagaimana cara user bisa berinteraksi dengan baik terhadap aplikasi. *User manual* akan memberitahukan secara detail bagaimana user bisa melakukan pendaftaran kegiatan dan Lembaga, serta mengajukan peminjaman barang di Mesjid Lukmanul Hakim.

2.6.2. Glossary

Berisi tentang daftar istilah teknis yang tidak umum bagi semua user aplikasi. Istilah – istilah yang akan dicantumkan meliputi penggunaan bahasa arab, istilah – istilah pada Lembaga yang biasa dipakai, dan istilah pada agama Islam.

2.6.3. Release Notes

Menjelaskan perubahan dan update versi pada perangkat lunak. Yang akan didokumentasikan adalah peningkatan fitur – fitur, perbaikan bug, dan juga pengembangan kinerja.

2.6.4. Online Help

Dokumen ini menyediakan bantuan secara daring dengan memberikan jawaban atas pertanyaan umum seputar perangkat lunak, solusi masalah teknis penggunaan perangkat lunak, dan petunjuk fitur perangkat lunak.

2.6.5. Frequently Asked Question

Menyediakan jawaban yang sering diajukan oleh user terkait penggunaan aplikasi.

2.7. Business Rules

2.7.1. Mahasiswa

1. Mahasiswa yang ingin mengakses aplikasi tersebut harus *login* dengan memasukkan *email* Polban dan *password*-nya [BR01]
2. Mahasiswa yang ingin mendaftarkan diri sebagai pengurus atau lembaga tersebut perlu mengisi formulir pendaftaran yang telah disediakan oleh setiap lembaganya [BR02]
3. Jika mahasiswa ingin meminjam sarana dan prasarana yang dimiliki suatu lembaga, mahasiswa hanya dapat meminjam ketika tidak dipinjam oleh siapapun [BR03]

2.7.2. Lembaga

1. Lembaga yang ingin mengakses aplikasi tersebut harus memasukan *username* dan *password* yang telah ditentukan sebelumnya [BR04]

2.7.3. Admin

1. Admin yang ingin mengakses aplikasi tersebut harus memasukan *username* dan *password* yang telah ditentukan sebelumnya [BR05]

2.8. Assumptions and Dependencies

Ada beberapa asumsi dan *dependency* yang harus dipenuhi dari sisi pengguna maupun server.

Hal itu diantaranya adalah :

2.8.1. Assumption

- Perangkat yang digunakan user sesuai dengan spesifikasi yang telah ditentukan oleh pihak pembuat aplikasi

- Perangkat lunak akan berjalan pada telah ditentukan dalam dokumen SRS
- User sudah memahami cara penggunaan dasar komputer sehingga bisa mengakses perangkat lunak dengan lancar
- User memiliki koneksi kecepatan internet minimal yang telah ditentukan pada aplikasi

2.8.2. Dependencies

- Harus memiliki dokumentasi penggunaan aplikasi secara terperinci
- Sistem perangkat lunak bergantung pada database untuk menyimpan semua data, sehingga ada server database yang tersedia dan terhubung dengan sistem
- Aplikasi harus mengikuti aturan dan regulasi yang berlaku terkait dengan pengumpulan data dan privasi pengguna

3. System Design

3.1. Business Process

3.1.1. Core Process

1. Penjadwalan Kegiatan Lembaga

Lembaga yang ingin melakukan penjadwalan harus sudah login terlebih dahulu pada aplikasi. Setelah itu baru dapat mengisi informasi terkait kegiatan seperti nama kegiatan, tanggal dan tempat kegiatan, ditujukan kepada siapa kegiatan tersebut, dan rundown kegiatan.

2. Peminjaman Sarana dan Prasarana

Mahasiswa yang ingin melakukan peminjaman sarana dan prasarana yang dimiliki masjid untuk kegiatannya dapat mengisi formulir terlebih dahulu serta meng-*upload* surat izin peminjaman ke aplikasi. Setelah itu, sistem akan meneruskan request peminjaman kepada staf lembaga. Lalu staf lembaga nantinya akan memeriksa surat izin peminjaman dan akan memberikan *feedback* berupa disetujui atau tidak disetujui kepada mahasiswa melalui sistem.

3.1.2. Management Process

1. Pengelolaan Staf Lembaga

Admin dapat melakukan pengelolaan terkait staf setiap lembaganya, tergantung kebijakan dari LDK. Jika LDK butuh staf lembaga baru, admin dapat melakukan create staf lembaga. Akan tetapi, jika staf lembaga yang sudah ada tidak dibutuhkan lagi, admin dapat melakukan delete staf lembaga.

2. Pengelolaan Informasi Kegiatan

Sistem akan melakukan pengelolaan terkait segala informasi kegiatan yang dimiliki oleh setiap lembaganya. Informasi kegiatan disini dapat berupa kegiatan yang akan terlaksana dan sudah terlaksana.

3. Pengelolaan Panitia Kegiatan

Sistem akan memberikan fitur pengelolaan panitia pada setiap lembaga terkait. Staf Lembaga hanya perlu memasukan daftar nama-nama dan jabatannya di kegiatan tersebut pada Sistem.

4. Pengelolaan Data Inventaris Masjid

Sistem akan melakukan pengelolaan terkait inventaris apa saja yang dimiliki oleh masjid dan lembaga. Selain itu, sistem juga akan memberi informasi kepada mahasiswa terkait sarana dan prasarana apa saja yang sedang dipinjam, akan dipinjam, dan belum dipinjam.

3.1.3. Support Process

1. Pendaftaran Akun

Bagi mahasiswa yang ingin melakukan pendaftaran akun pada aplikasi dapat memasukan data terkait email polban, password, nama, nim, prodi, dan jurusan.

2. Autentikasi Akun

Mahasiswa yang sudah berhasil melakukan *register* dapat langsung *login* dengan memasukan *email* Polban beserta *password*-nya.

3. Pendaftaran Mahasiswa pada Lembaga

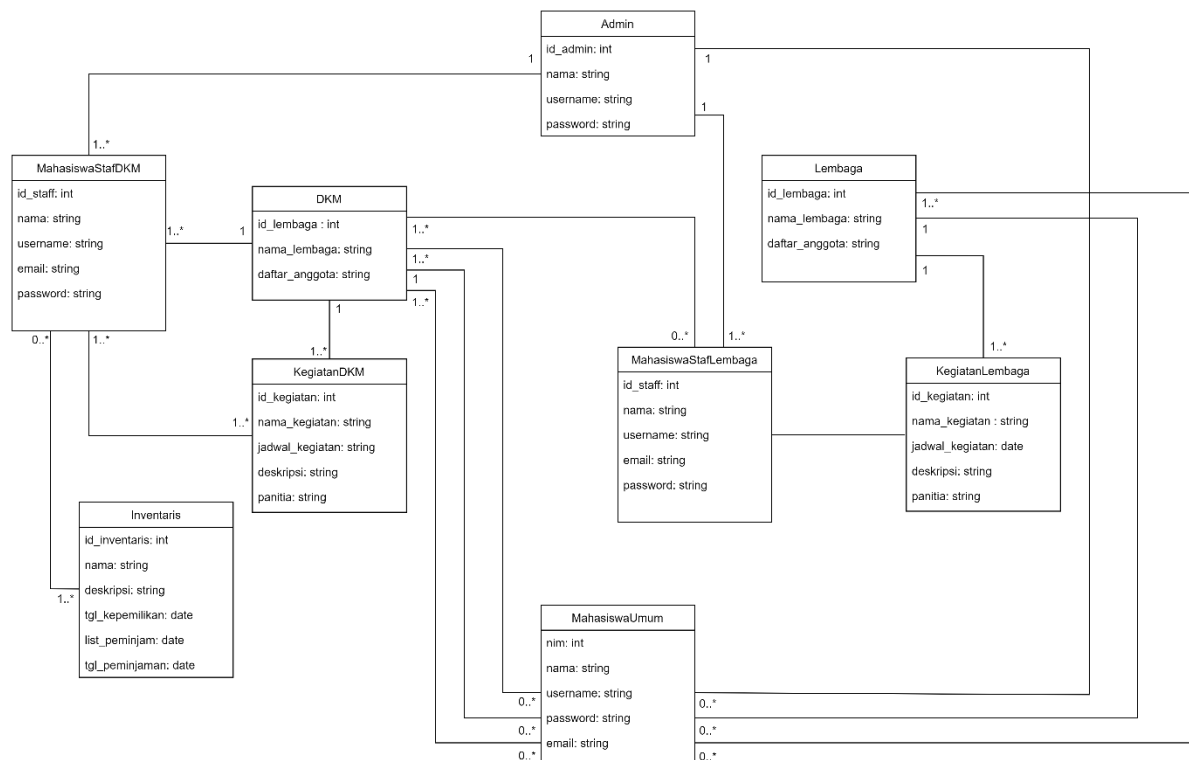
Mahasiswa yang ingin melakukan pendaftaran pada lembaga yang ada di dalam masjid, dapat dengan mengisi formulir yang mungkin untuk setiap lembaganya akan berbeda-beda. Formulir ini nantinya akan dibebaskan kepada setiap lembaga untuk diisi dengan apa saja sesuai kebutuhan mereka.

Setelah mahasiswa mengisi formulir, maka sistem akan meneruskannya ke lembaga terkait. Nantinya, staf lembaga terkait yang akan melakukan pengecekan terkait data data yang dimasukan oleh mahasiswa. Lalu, lembaga akan memberikan *feedback* kepada para mahasiswa yang sudah mendaftar berupa diterima atau tidaknya.

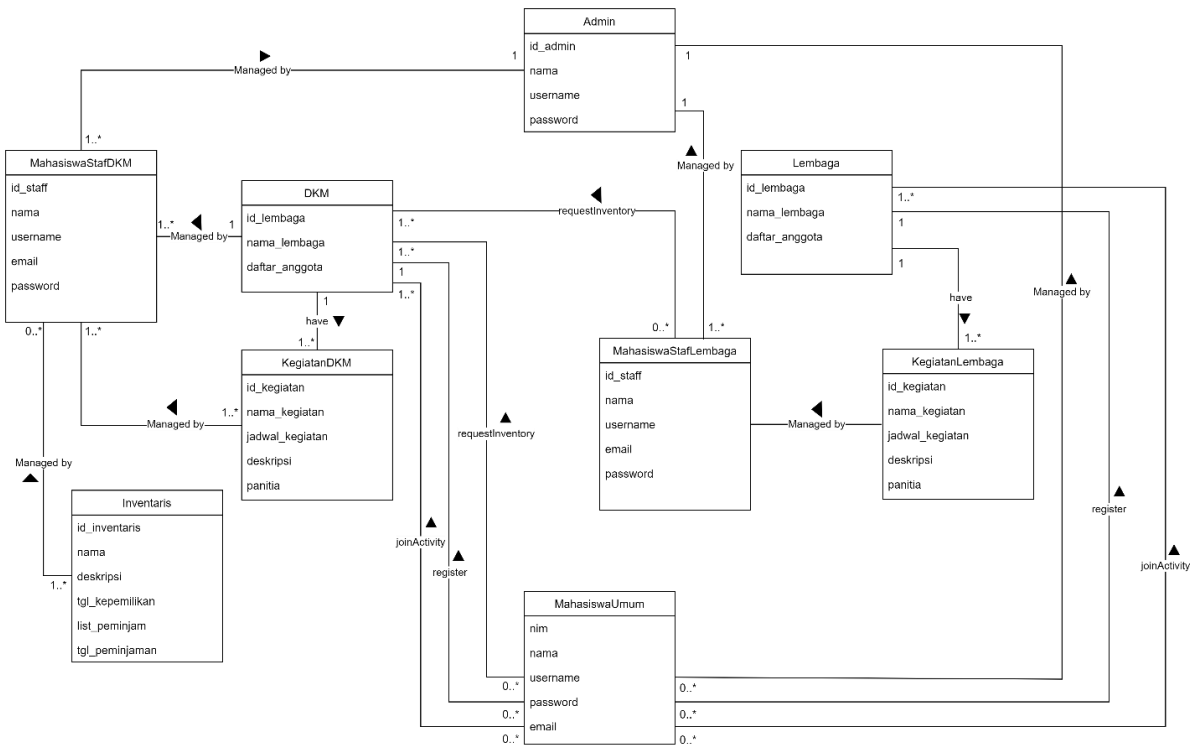
4. Pendaftaran Mahasiswa Polban pada Kegiatan

Bagi mahasiswa yang ingin mendaftar pada kegiatan-kegiatan yang dilaksanakan lembaga masjid, dapat dengan mengisikan formulir yang mungkin untuk setiap kegiatannya akan berbeda-beda. Formulir ini nantinya akan dibebaskan kepada setiap lembaga untuk diisi dengan apa saja sesuai kebutuhan mereka.

3.2. Class Diagram

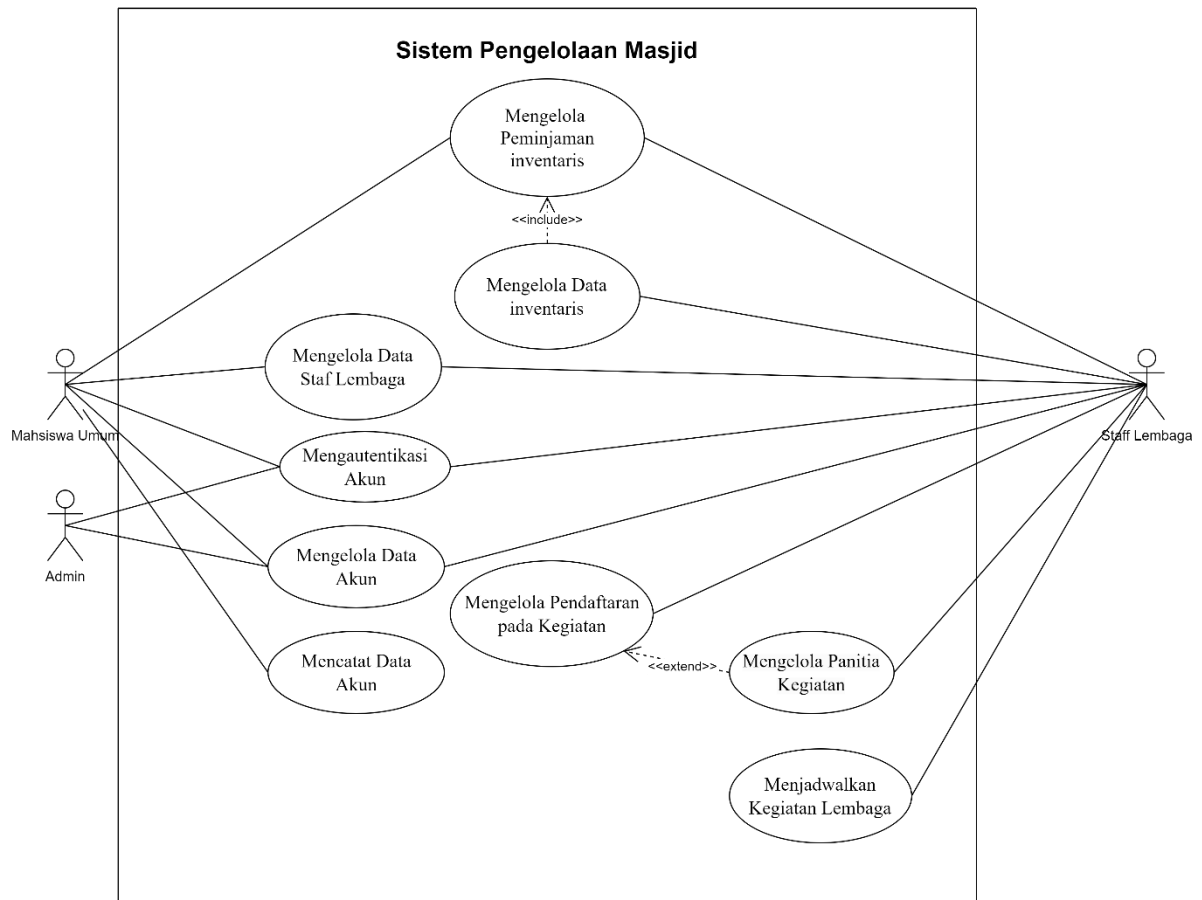


3.3. Domain Model



3.3. Use Case Model

3.3.1. Use Case Diagram



Berikut list use case pada sistem ini:

1. Menjadwalkan Kegiatan Lembaga [UC01]
2. Mengelola Data Sarana Dan Prasarana [UC02]
3. Mengelola Transaksi Peminjaman Sarana Dan Prasarana [UC03]
4. Mengelola Data Akun [UC04]
5. Mencatat Data Akun [UC05]
6. Mengautentikasi Akun [UC06]
7. Mengelola Data Staf Lembaga [UC07]
8. Melakukan Pendaftaran Pada Kegiatan [UC08]
9. Mengelola Panitia Kegiatan [UC09]

3.3.2. Fully Dressed

1. Menjadwalkan Kegiatan Lembaga

Scope : Pengelolaan Masjid Level : User Goal Primary Actor : Staf Lembaga Stakeholder and Interests : <ol style="list-style-type: none"> 1. Staf Lembaga (Kammi, Assalam, DKM) menginginkan untuk dapat menambah kegiatan baru, melihat kegiatan yang ada, menyunting kegiatan, dan menghapus kegiatan 2. Sistem menginginkan seluruh data dapat tersimpan dengan baik tanpa adanya kehilangan data Preconditions : Staf Lembaga harus sudah terdaftar pada aplikasi Success Guarantee (or Postconditions): Staf Lembaga berhasil menambah kegiatan baru, melihat kegiatan yang ada, menyunting kegiatan, dan menghapus kegiatan Main Success Scenario (or Basic Flow):		
No	Mahasiswa sebagai Staf Lembaga	Sistem
1	Login Aplikasi dengan menginput <i>username</i> dan <i>password</i>	
2		Menampilkan halaman <i>dashboard</i>
3	Menekan <i>button</i> “Kegiatan”	
4		Mengecek apakah terdapat kegiatan dalam <i>database</i> .
		Jika terdapat kegiatan, maka akan menampilkan daftar kegiatan pada Halaman Kegiatan dan menampilkan <i>button</i> : <ul style="list-style-type: none"> • Tambah Kegiatan Baru • Lihat Detail Kegiatan • Sunting Kegiatan • Hapus Kegiatan
		Jika tidak terdapat kegiatan, maka daftar kegiatan pada Halaman Kegiatan akan kosong dan menampilkan <i>button</i> “Tambah Kegiatan Baru”
5	Case 1 : Tambah Kegiatan Baru Menekan <i>button</i> “Tambah Kegiatan Baru”	
6		Menampilkan Modal Form Kegiatan Baru
7	Mengisi <i>field</i> : <ul style="list-style-type: none"> • Nama Kegiatan 	

	<ul style="list-style-type: none"> • Tempat dan Tanggal Kegiatan • Tujuan Kegiatan • Deskripsi Kegiatan 	
8	Menekan <i>button</i> “Simpan”	
9		Menyimpan informasi kegiatan baru ke dalam <i>database</i>
10		Menampilkan daftar kegiatan di Halaman Kegiatan
11	Case 2 : Melihat Detail Kegiatan Menekan <i>button</i> “Lihat Detail Kegiatan”	
12		Menampilkan detail informasi kegiatan
13	Case 3 : Sunting Kegiatan Menekan <i>button</i> “Sunting Kegiatan” pada kegiatan yang dipilih	
14		Menampilkan Modal Form Sunting Kegiatan yang setiap <i>field</i> -nya sudah terdapat <i>value</i> sebelumnya
15	Menyunting informasi kegiatan tersebut	
16	Menekan <i>button</i> “Simpan Perubahan”	
17		Menyimpan informasi kegiatan yang baru disunting ke dalam <i>database</i>
18		Menampilkan daftar kegiatan di Halaman Kegiatan
19	Case 4 : Hapus Kegiatan Menekan <i>button</i> “Hapus Kegiatan” pada kegiatan yang dipilih	
20		Menghapus kegiatan tersebut beserta detailnya dari <i>database</i>
21		Menampilkan daftar kegiatan di Halaman Kegiatan

Extensions (or Alternative Flows):

1. Jika data kegiatan tidak muncul, bisa saja disebabkan oleh tiga hal, yaitu staf lembaga memang belum menambahkan kegiatannya, koneksi internet kurang memadai, atau server yang dimiliki sistem kami sedang *down*.
 - Jika belum ada kegiatan yang ditambahkan, maka Staf Lembaga dapat melakukan Case 1 pada Main Success Scenario.
 - Jika koneksi internet kurang memadai, maka Staf Lembaga dapat melakukan *refresh* atau *restart* aplikasi.

2. Jika suatu staf lembaga mencoba menambah/menyunting/menghapus kegiatan pada lembaga yang bukan miliknya (bukan hak aksesnya), sistem akan menolak permintaan tersebut dan menampilkan pesan *error*.
3. Jika staf lembaga mencoba menambahkan jadwal kegiatan yang berulang, sistem akan meminta pengguna untuk memilih frekuensi kegiatan tersebut (misalnya harian, mingguan, atau bulanan) dan menampilkan opsi untuk mengatur jadwal berulang tersebut.

Special Requirements:

1. Tampilan kalender jadwal kegiatan harus *user-friendly*.
2. Jadwal kegiatan yang dimiliki semua lembaga harus dapat diakses oleh semua pengguna.
3. Sistem harus dapat mengirimkan notifikasi otomatis kepada semua pengguna terkait dengan jadwal kegiatan yang akan datang.

Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
3. Data Inventaris (Id Inventaris, Nama, Deskripsi, Tanggal Kepemilikan, List Peminjam, Tanggal Peminjaman)

Frequency of Occurrence: Harian

Open Issues: -

2. Mengelola Data Sarana dan Prasarana

Scope : Pengelolaan Masjid**Level :** User Goal**Primary Actor :** Staf Lembaga DKM**Stakeholder and Interests :**

1. Staf Lembaga DKM : Menginginkan untuk dapat menambah sarana dan prasarana baru, melihat sarana dan prasarana yang dimiliki, menyunting sarana dan prasarana, dan menghapus sarana dan prasarana
2. Sistem : Menginginkan seluruh data dapat tersimpan dengan baik tanpa adanya kehilangan data

Preconditions : Staf Lembaga DKM harus sudah terdaftar pada aplikasi

Success Guarantee (or Postconditions): Staf Lembaga DKM berhasil menambah sarana dan prasarana baru, melihat sarana dan prasarana yang dimiliki, menyunting sarana dan prasarana, dan menghapus sarana dan prasarana

Main Success Scenario (or Basic Flow):

No	Mahasiswa sebagai Staf Lembaga DKM	Sistem
1	Login aplikasi dengan menginput <i>username</i> dan <i>password</i>	
2		Menampilkan halaman <i>dashboard</i>
3	Menekan <i>button</i> “Inventaris”	
4		Menampilkan Halaman Sarana dan Prasarana yang berisi Inventaris Masjid
5	Case 1 : Menekan <i>button</i> “addInventory”	
6		Menampilkan form inputan data yang terdiri dari: <ul style="list-style-type: none"> • Nama • Deskripsi • Tanggal Kepemilikan • List Peminjam • Tanggal Peminjaman Serta terdapat <i>button</i> “Submit”
7	Mengisi seluruh <i>field</i> dalam form tersebut dan menekan <i>button</i> “Submit”	
8		Menyimpan data ke dalam <i>database</i> dan menampilkannya pada halaman Sarana dan Prasarana
9	Case 2 : Memilih sarana dan prasarana	

10		Menampilkan halaman informasi terkait sarana dan prasarana yang dipilih. Serta terdapat <i>button</i> “Edit” dan “Hapus”.
11	Case 2.1 : Menekan <i>button</i> “Edit”	
12		Menampilkan form inputan data yang terdiri dari: <ul style="list-style-type: none"> • Nama • Deskripsi • Tanggal Kepemilikan • List Peminjam • Tanggal Peminjaman Serta terdapat <i>button</i> “Submit”
13	Mengubah sebagian atau seluruh data dan menekan <i>button</i> “Submit”	
14		Menyimpan perubahan ke dalam database
15	Case 2.2 : Menekan <i>button</i> “Hapus”	
16		Menampilkan modal yang berisi pertanyaan “Apakah anda yakin untuk menghapus sarana dan prasarana ini?” dan menyediakan <i>button</i> “Ya” dan “Tidak”
17	Case 2.2.1 : Menekan “Tidak”	
18		Menampilkan halaman detail sarana prasarana yang sebelumnya dipilih
19	Case 2.2.2 : Menekan “Ya”	
20		Menghapus sarana dan prasarana yang ada di <i>database</i> dan menampilkan halaman detail sarana prasarana yang sebelumnya dipilih

Extensions (or Alternative Flows):

1. Jika data sarana dan prasarana tidak muncul, bisa saja disebabkan oleh tiga hal, yaitu staf lembaga DKM memang belum menambahkan kegiatannya, koneksi internet kurang memadai, atau server yang dimiliki sistem kami sedang *down*.
 - o Jika belum ada sarana dan prasarana yang ditambahkan, maka Staf Lembaga DKM dapat melakukan Case 1 pada Main Success Scenario.
 - o Jika koneksi internet kurang memadai, maka Staf Lembaga DKM dapat melakukan *refresh* atau *restart* aplikasi.
2. Jika suatu staf lembaga lain (Assalam atau Kammi) mencoba menambah/menyunting/menghapus data sarana dan prasarana, sistem akan menolak permintaan tersebut dan menampilkan pesan *error*.
3. Jika staf lembaga DKM mencoba menambahkan sarana dan prasarana yang berulang, sistem akan meminta staf lembaga untuk mengatur jumlah sarana dan prasarana yang dimiliki

Special Requirements:

1. Data sarana dan prasarana harus terintegrasi dengan sistem lain, yaitu sistem transaksi peminjaman sarana dan prasarana.

2. Sistem harus memiliki fitur *backup* dan *recovery* untuk memastikan data sarana dan prasarana tidak hilang dalam kejadian bencana atau kecelakaan lainnya.

Technology and Data Variations List:

1. Teknologi
 1. Website
 1. Sistem Operasi Windows 10 atau Windows 11
 2. Browser Chrome Version 110.0.5481.104 (Official Build) (64-bit) dan Edge Version 110.0.1587.49 (Official build) (64-bit)
 3. Resolusi Layar (Desktop, Laptop, Tablet, dan Smartphone)
 2. Aplikasi Android
 1. Sistem Operasi Android 8.0+
 2. Merk Perangkat (Sony dan Realme)
 3. Resolusi Layar (HD, Full HD, dan 4K)
2. Data
 1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
 2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
 3. Data Inventaris (Id Inventaris, Nama, Deskripsi, Tanggal Kepemilikan, List Peminjam, Tanggal Peminjaman)

Frequency of Occurrence: Harian

Open Issues: -

3. Mengelola Transaksi Peminjaman Sarana & Prasarana

Scope : Pengelolaan Masjid

Level : User Goal

Primary Actor : Mahasiswa Umum, Staf Lembaga DKM

Stakeholder and Interests :

1. Mahasiswa Umum menginginkan untuk dapat meminjam sarana dan prasarana yang dimiliki masjid
2. Staf Lembaga DKM dapat memverifikasi dan memvalidasi data peminjaman yang diajukan mahasiswa dan menginginkan *feedback* yang dikirimkan kepada Mahasiswa Umum berhasil tersampaikan.
3. Sistem menginginkan transaksi peminjaman dapat tersimpan dengan baik tanpa adanya kehilangan data

Preconditions :

1. Mahasiswa harus sudah terdaftar pada aplikasi
2. Data sarana dan prasarana telah tersedia dalam *database*
3. Sarana dan prasarana yang akan dipinjam belum dipinjam oleh pihak lain.

Success Guarantee (or Postconditions): Mahasiswa berhasil meminjam sarana dan prasarana yang dimiliki masjid

Main Success Scenario (or Basic Flow):			
No	Mahasiswa Umum	Mahasiswa sebagai Staf Lembaga DKM	Sistem
1	Login Aplikasi dengan menginput email dan password		
2			Menampilkan Halaman Dashboard
3	Menekan <i>button</i> “Sarana dan Prasarana”		
4			Menampilkan Halaman Sarana dan Prasarana yang berisi Inventaris Masjid
5	Memilih barang yang akan dipinjam		
6			Memberikan informasi barang tersebut berupa nama barang, deskripsi, dan data tanggal yang tidak bisa dipinjam
7			Memberikan <i>button</i> “Ajukan Peminjaman”
8	Menekan <i>button</i> “Ajukan Peminjaman”		
9			Menampilkan Modal Form Peminjaman Sarana dan Prasarana
10	Mengisi <i>field</i> : <ul style="list-style-type: none">Nama Penanggung JawabTujuanTanggal PeminjamanUnggah Surat Izin Peminjaman		
11	Menekan <i>button</i> “Kirim”		
12			Menyimpan data tersebut ke <i>database</i>
13			Mengirim data tersebut ke staf Lembaga DKM melalui notifikasi aplikasi
14		Login Aplikasi dengan menginput <i>username</i> dan <i>password</i>	
15		Menampilkan Halaman Dashboard	

16		Menekan <i>button</i> “Notifikasi”	
17			Menampilkan daftar semua informasi notifikasi
18		Menekan informasi terkait peminjaman sarana dan prasarana	
19			Menampilkan informasi terkait peminjaman sarana dan prasarana berupa data: <ul style="list-style-type: none"> • Nama Penanggung Jawab • Tujuan • Tanggal Peminjaman
20			Memberikan <i>button</i> “Unduh Surat Izin Peminjaman” dan <i>button</i> “Kirim Umpan Balik”
21		Mengunduh surat tersebut	
22		Menekan <i>button</i> “Kirim Umpan Balik”	
23			Menampilkan Modal Umpan Balik, berupa <i>field</i> : <ul style="list-style-type: none"> • Judul • Deskripsi
24		Mengisi <i>field</i> : <ul style="list-style-type: none"> • Judul • Deskripsi 	
25		Menekan <i>button</i> “Kirim”	
26			Menyimpan data umpan balik ke <i>database</i>
27			Mengirim ke Mahasiswa Pemohon
28	Masuk ke halaman notifikasi		
29			Menampilkan daftar semua informasi notifikasi
30	Menekan informasi terkait peminjaman sarana dan prasarana		
31			Menampilkan informasi umpan balik, berupa <i>field</i> : <ul style="list-style-type: none"> • Judul • Deskripsi

Extensions (or Alternative Flows):

1. Jika data sarana dan prasarana tidak muncul, bisa saja disebabkan oleh dua hal, yaitu koneksi internet kurang memadai, atau server yang dimiliki sistem kami sedang *down*. Maka mahasiswa dapat melakukan *refresh* atau *restart* aplikasi.
2. Jika suatu staf lembaga lain mencoba menambah/menyunting/menghapus data transaksi sarana dan prasarana pada lembaga yang bukan hak aksesnya, sistem akan menolak permintaan tersebut dan menampilkan pesan *error*.
3. Jika sarana atau prasarana yang ingin dipinjam tidak tersedia, maka:
 - Sistem memberikan pesan kesalahan kepada Mahasiswa bahwa sarana atau prasarana yang ingin dipinjam tidak tersedia.
 - Mahasiswa dapat memilih untuk mencoba untuk memesan pada tanggal yang berbeda atau membatalkan permintaan peminjaman.
4. Jika terdapat konflik jadwal saat ingin meminjam sarana dan prasarana, maka:
 - Sistem memberikan pesan kesalahan kepada Mahasiswa bahwa terdapat konflik jadwal dengan permintaan peminjaman sebelumnya.
 - Mahasiswa dapat memilih untuk mencoba untuk memesan pada tanggal yang berbeda atau membatalkan permintaan peminjaman.
5. Jika mahasiswa membatalkan permintaan peminjaman, maka:
 - Sistem membatalkan peminjaman sarana atau prasarana yang telah dipesan oleh Mahasiswa.
 - Sistem memberikan informasi kepada Staf Lembaga DKM bahwa peminjaman telah dibatalkan.
6. Jika Staf Lembaga DKM membatalkan permintaan peminjaman, maka:
 - Sistem membatalkan peminjaman sarana atau prasarana yang telah dipesan oleh Mahasiswa.
 - Sistem memberikan notifikasi kepada Mahasiswa bahwa peminjaman sarana atau prasarana telah dibatalkan oleh Staf Lembaga DKM.

Special Requirements:

1. Sistem harus dapat membatasi jumlah sarana dan prasarana yang dapat dipinjam oleh satu peminjam dalam satu waktu tertentu.
2. Sistem harus dapat membatasi waktu peminjaman sarana dan prasarana sehingga peminjam tidak dapat meminjam lebih lama dari yang diizinkan.
3. Sistem harus dapat menunjukkan ketersediaan sarana dan prasarana pada saat peminjam mencoba untuk meminjamnya
4. Sistem harus dapat memfasilitasi pembatalan peminjaman oleh peminjam atau oleh admin.
5. Sistem harus dapat memberikan notifikasi kepada peminjam yang terlambat dalam mengembalikan sarana dan prasarana.
6. Sistem harus dapat memfasilitasi perpanjangan peminjaman jika tidak ada peminjam lain yang membutuhkan sarana dan prasarana tersebut.

7. Sistem harus dapat menghitung dan memfasilitasi pembayaran denda untuk peminjam yang terlambat mengembalikan sarana dan prasarana.
8. Sistem harus dapat memverifikasi identitas peminjam dengan benar sebelum meminjamkan sarana dan prasarana.

Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
3. Data Inventaris (Id Inventaris, Nama, Deskripsi, Tanggal Kepemilikan, List Peminjam, Tanggal Peminjaman)

Frequency of Occurrence: Harian

Open Issues: -

4. Mengelola Data Akun

Scope : Pengelolaan Masjid

Level : User Goal

Primary Actor : Admin, Mahasiswa Umum, Staf Lembaga

Stakeholder and Interests :

1. Admin, Mahasiswa Umum, maupun Staf Lembaga menginginkan untuk dapat mengelola data akun
2. Sistem menginginkan seluruh data tersimpan dengan baik

Preconditions : Admin, Mahasiswa Umum, dan Staf Lembaga harus sudah terdaftar pada aplikasi

Success Guarantee (or Postconditions): Admin, Mahasiswa Umum, dan Staf Lembaga berhasil mengelola data akun masing - masing

Main Success Scenario (or Basic Flow):

No	Admin, Mahasiswa Umum, Staf Lembaga	Sistem
1	Login Aplikasi dengan menginput email dan password	
2		Menampilkan Halaman Dashboard
3	Menekan <i>button</i> “Akun”	
4		Menampilkan Halaman user profile yang berisi identitas dan data akun terdaftar pada aplikasi
8	Menekan <i>button</i> “Edit”	
9		Menampilkan Modal Form yang berisi identitas akun
10	Mengisi <i>field</i> : <ul style="list-style-type: none"> • Username • Password 	
11	Menekan <i>button</i> “Selesai”	

Extensions (or Alternative Flows):

1. Jika data Akun tidak muncul, bisa saja disebabkan oleh dua hal, yaitu koneksi internet kurang memadai, atau server yang dimiliki sistem kami sedang *down*. Maka pengguna dapat melakukan *refresh* atau *restart* aplikasi
2. Jika suatu staf lembaga dan mahasiswa umum mencoba menyunting data user profile selain miliknya, maka sistem akan menolak permintaan tersebut dan menampilkan pesan *error*

3. Jika terdapat field yang kosong, maka sistem tidak akan menyimpan perubahan. Sistem akan meminta pengguna untuk mengisi field yang kosong tersebut. Lalu setelah itu pengguna dapat melakukan *submit* lagi

Special Requirements:

1. Sistem harus dapat memberikan *alert* kepada pengguna apabila ada field yang masih belum terisi

Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
3. Data Admin (Id Admin, Nama, Username, Password)

Frequency of Occurrence: Harian

Open Issues: -

5. Mencatat Data Akun

Scope : Pengelolaan Masjid**Level :** User Goal**Primary Actor :** Mahasiswa Umum**Stakeholder and Interests :**

1. Mahasiswa Umum menginginkan untuk dapat terdaftar sebagai pengguna aplikasi
2. Sistem menginginkan seluruh permintaan pendaftaran pengguna berhasil dicatat datanya

Preconditions : Mahasiswa umum belum memiliki akun**Success Guarantee (or Postconditions):** Data mahasiswa umum berhasil dicatat atau terdaftar pada sistem**Main Success Scenario (or Basic Flow):**

No	Mahasiswa Umum	Sistem
1	Membuka Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban	
2		Menampilkan halaman login dan terdapat <i>button</i> “Daftar”
3	Menekan <i>button</i> “Daftar”	
4		Menampilkan halaman pendaftaran beserta formulir pengisian data pencatatan akun. Selain itu, terdapat <i>button</i> “Kirim”
5	Mengisi <i>field</i> , berupa: <ul style="list-style-type: none"> • Nama • NIM • <i>username</i> • <i>email</i> polban • <i>password</i> Setelah itu, menekan <i>button</i> “Kirim”	
6		Memeriksa data yang telah diinputkan. Jika <i>email</i> polban atau <i>username</i> tersebut <i>unique</i> , maka sistem akan mencatat data tersebut pada <i>database</i> dan menampilkan halaman <i>dashboard</i> . Sebaliknya, sistem akan menampilkan pesan bahwa <i>email</i> yang diinputkan salah.

Extensions (or Alternative Flows):

1. Jika pengguna tidak mengisi semua kolom yang diperlukan pada formulir pendaftaran, maka sistem akan menampilkan pesan *error* dan meminta pengguna untuk melengkapi semua *field* yang kosong.

2. Jika pengguna memasukkan *password* yang tidak memenuhi kriteria keamanan yang ditetapkan, seperti kurang dari 8 karakter atau tidak memiliki kombinasi angka dan huruf, maka sistem akan menampilkan pesan *error* dan meminta pengguna untuk memasukkan *password* yang memenuhi kriteria keamanan.
3. Jika pengguna ingin membatalkan proses registrasi, maka sistem akan mengembalikan pengguna ke halaman *login*.

Special Requirements:

1. Sistem harus memiliki persyaratan untuk *password* yang kuat dan aman, seperti minimal 8 karakter dengan kombinasi huruf besar, huruf kecil, angka, dan karakter khusus.
2. Sistem harus memiliki kebijakan privasi yang jelas dan transparan, serta memberikan pilihan kepada pengguna untuk memilih informasi apa saja yang akan dibagikan dengan pihak ketiga.
3. Sistem harus memiliki mekanisme pengamanan seperti *captcha* untuk mencegah pendaftaran otomatis oleh *bot* atau program jahat.
4. Sistem harus memiliki enkripsi untuk melindungi data penggunanya.

Technology and Data Variations List:

1. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)

Frequency of Occurrence: Harian

Open Issues: -

6. Mengautentikasi Akun

Scope : Pengelolaan Masjid**Level :** User Goal**Primary Actor :** Admin, Mahasiswa Umum, Staf Lembaga**Stakeholder and Interests :**

1. Admin, Mahasiswa Umum, Staf Lembaga : Menginginkan untuk dapat login ke dalam aplikasi
2. Sistem : Menginginkan seluruh permintaan autentikasi dari pengguna berhasil

Preconditions : Admin, Mahasiswa Umum, Staf Lembaga harus sudah terdaftar pada aplikasi**Success Guarantee (or Postconditions):** Admin, Mahasiswa Umum, Staf Lembaga berhasil login ke dalam aplikasi**Main Success Scenario (or Basic Flow):**

No	Admin, Mahasiswa Umum, Staf Lembaga	Sistem
1	Membuka aplikasi pengelolaan masjid	
2		Menampilkan halaman login dan terdapat <i>button</i> "Login"
3	Mengisi <i>field</i> : <ul style="list-style-type: none"> • Email Polban • Password Setelah mengisi <i>field</i> tersebut, mahasiswa perlu menekan <i>button</i> "Login"	
4		Mengecek data yang telah diinputkan. Jika Email Polban atau Password tersebut sesuai, maka sistem akan menampilkan halaman <i>dashboard</i> . Sebaliknya, sistem akan menampilkan pesan bahwa <i>Email/Password</i> yang diinputkan salah.

Extensions (or Alternative Flows):

1. Jika pengguna lupa password, maka dapat mengikuti langkah berikut:
 - Memilih *button* "Lupa Password"
 - Sistem akan meminta pengguna untuk melakukan verifikasi akunnya dengan memasukkan email yang didaftarkan pada sistem
 - Pengguna memasukkan email polban nya
 - Sistem akan mengirimkan kode OTP untuk memverifikasi akun dan akan menampilkan halaman untuk memasukkan kode OTP
 - Pengguna memasukkan kode OTP
 - Sistem akan memvalidasi
 - Jika kode sesuai maka sistem akan menampilkan form untuk me-reset password
 - Jika kode tidak sesuai maka sistem akan memunculkan pesan error dan meminta pengguna untuk memasukkan kode OTP yang benar

Special Requirements:

1. Proses autentikasi harus dijalankan dengan cepat dan responsif

Technology and Data Variations List:

1. Teknologi
 1. Website
 1. Sistem Operasi Windows 10 atau Windows 11
 2. Browser Chrome Version 110.0.5481.104 (Official Build) (64-bit) dan Edge Version 110.0.1587.49 (Official build) (64-bit)
 3. Resolusi Layar (Desktop, Laptop, Tablet, dan Smartphone)
 2. Aplikasi Android
 1. Sistem Operasi Android 8.0+
 2. Merk Perangkat (Sony dan Realme)
 3. Resolusi Layar (HD, Full HD, dan 4K)
2. Data
 1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
 2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
 3. Data Admin (Id Admin, Nama, Username, Password)

Frequency of Occurrence: Harian

Open Issues: -

7. Mengelola Data Staf Lembaga

Scope : Pengelolaan Masjid

Level : User Goal

Primary Actor : Mahasiswa Umum, Staf Lembaga

Stakeholder and Interests :

1. Mahasiswa Umum : Menginginkan pendaftaran keanggotaannya berhasil
2. Staf Lembaga : Menginginkan seluruh data pendaftaran tersimpan dengan baik agar dapat melakukan penyeleksian dengan cepat
3. Sistem : Menginginkan seluruh data pendaftaran tidak ada yang hilang

Preconditions : Mahasiswa Umum dan Staf Lembaga harus sudah terdaftar pada aplikasi

Success Guarantee (or Postconditions): Mahasiswa Umum berhasil melakukan pendaftaran keanggotaannya pada lembaga. Staf Lembaga dapat memberikan *feedback* apakah mahasiswa tersebut diterima atau tidak.

Main Success Scenario (or Basic Flow):

No	Mahasiswa Umum	Sistem	Mahasiswa sebagai Staff Lembaga
1	Login Aplikasi dengan menginput <i>username</i> dan <i>password</i>		

2		Menampilkan Halaman <i>Dashboard</i>	
3	Menekan <i>button</i> “Lembaga”		
4		Menampilkan 3 <i>list</i> Lembaga, yaitu DKM, Kammi, dan Assalam	
	Memilih lembaga		
5		Menampilkan lembaga yang dipilih, terdapat informasi terkait lembaga tersebut, seperti Nama Lembaga, Daftar Anggota, dan Deskripsi. Selain itu, menampilkan <i>button</i> “Daftar Lembaga”	
6	Menekan <i>button</i> “Daftar Lembaga”		
7		Menampilkan halaman registrasi, yang berisi formulir pendaftaran tergantung kebijakan lembaganya masing-masing. Selain itu, terdapat <i>button</i> “Registrasi Sekarang”	
8	Mengisi <i>field</i> yang disediakan lembaga dan menekan <i>button</i> “Registrasi Sekarang”		
9		Menampilkan bahwa proses pendaftaran berhasil. Selain itu, sistem akan mengirimkan data pendaftaran tersebut ke lembaga terkait	
10			Login Aplikasi dengan menginput <i>username</i> dan <i>password</i>
11		Menampilkan halaman <i>dashboard</i>	
12			Menekan <i>button</i> “Notifikasi”
13		Menampilkan notifikasi terkait mahasiswa yang telah melakukan pendaftaran	
14			Menekan notifikasi tersebut
15		Menampilkan data terkait pendaftaran mahasiswa tersebut. Selain itu, terdapat <i>button</i> “Terima” dan <i>button</i> “Tolak”	
16			Menerima data tersebut.

17			Case 1 : Menerima Mahasiswa Menekan <i>button</i> “Terima”
18		Menampilkan Halaman Pesan	
19			Mengirim deskripsi yang diperlukan lembaga terkait proses selanjutnya yang harus dilakukan mahasiswa pendaftar
20		Menerima deskripsi tersebut dan mengirimkan kepada mahasiswa pendaftar	
21	Menerima notifikasi tersebut		
22			Case 2 : Menolak Mahasiswa Menekan <i>button</i> “Tolak”
23		Menampilkan Halaman Pesan	
24			Mengirim deskripsi yang terkait alasan penolakan
25		Menerima deskripsi tersebut dan mengirimkan kepada mahasiswa pendaftar	
26	Menerima notifikasi tersebut		

Extensions (or Alternative Flows):

1. Jika Mahasiswa Umum tidak mengisi semua field yang diminta dan menekan *button* “Registrasi Sekarang”, maka data yang sudah terisi tidak dapat di *submit*. Lalu, sistem akan meminta pengguna untuk mengisi seluruh field data.
2. Jika Mahasiswa Umum melakukan pendaftaran lebih dari satu kali, maka sistem akan memberi pesan error dan meminta Mahasiswa Umum untuk mencoba mendaftar di lembaga lain.

Special Requirements:

1. Jika terdapat rincian lain (seperti tes lanjutan) dalam alur pendaftaran lembaga tertentu, maka sistem harus memberi informasi kepada Mahasiswa Umum melalui notifikasi pada aplikasi.

Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)

3. Data Lembaga

Frequency of Occurrence: Harian

Open Issues: -

8. Mengelola Data Pendaftaran pada Kegiatan

Scope : Pengelolaan Masjid

Level : User Goal

Primary Actor : Mahasiswa Umum

Stakeholder and Interests :

1. Mahasiswa Umum : Menginginkan untuk dapat terdaftar dalam sebuah kegiatan
2. Sistem : Menginginkan seluruh pendaftaran tersimpan dengan baik

Preconditions : Mahasiswa Umum harus sudah terdaftar pada aplikasi

Success Guarantee (or Postconditions): Mahasiswa Umum berhasil mendaftarkan dirinya dalam sebuah kegiatan

Main Success Scenario (or Basic Flow):

No	Mahasiswa Umum	Sistem
1	Login Aplikasi dengan menginput <i>username</i> dan <i>password</i>	
2		Menampilkan Halaman <i>dashboard</i>
3	Menekan <i>button</i> “Kegiatan”	
4		Menampilkan seluruh kegiatan yang ada
5	Memilih kegiatan yang akan diikuti	
6		Menampilkan informasi lengkap terkait kegiatan tersebut serta terdapat <i>button</i> “Ikuti Kegiatan”
7	Menekan <i>button</i> “Ikuti Kegiatan”	
8		Menampilkan form yang harus diisi (form berisi sesuai kebutuhan lembaga) serta <i>button</i> “Submit”
9	Mengisi seluruh field dalam form dan menekan <i>button</i> “Submit”	
10		Menyimpan hasil isi form ke <i>database</i> dan menampilkan informasi bahwa pendaftaran berhasil

Extensions (or Alternative Flows): Jika Mahasiswa Umum tidak mengisi semua field yang diminta dan menekan *button* “Registrasi “Submit”, maka data yang sudah terisi tidak dapat di submit. Lalu, sistem akan meminta pengguna untuk melengkapi field data yang masih kosong.

Special Requirements:

1. Sistem harus dapat melakukan validasi data pendaftaran agar data yang masuk ke dalam sistem benar dan tidak ada duplikasi data.

2. Sistem harus mampu mengatur batasan kuota pendaftaran agar peserta tidak melebihi kapasitas kegiatan yang diadakan.

Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
3. Data Admin (Id Admin, Nama, Username, Password)

Frequency of Occurrence: Harian

Open Issues: -

9. Mengelola Panitia Kegiatan

Scope : Pengelolaan Masjid**Level :** User Goal**Primary Actor :** Staff Lembaga**Stakeholder and Interests :**

1. Mahasiswa Umum menginginkan untuk dapat terdaftar dalam sebuah kegiatan
2. Sistem menginginkan seluruh pendaftaran tersimpan dengan baik

Preconditions : Mahasiswa Umum harus sudah terdaftar pada aplikasi**Success Guarantee (or Postconditions):** Mahasiswa Umum berhasil mendaftarkan dirinya dalam sebuah kegiatan**Main Success Scenario (or Basic Flow):**

No	Mahasiswa sebagai Staf Lembaga	Sistem
1	Login aplikasi dengan menginput <i>username</i> dan <i>password</i>	
2		Menampilkan halaman <i>dashboard</i>
3	Menekan <i>button</i> “Kegiatan”	
4		Menampilkan halaman kegiatan yang berisi daftar seluruh kegiatan lembaga tersebut
5	Memilih kegiatan yang akan diatur	
6		Menampilkan detail kegiatan yang dipilih. Jika sudah terdapat struktur panitia, sistem akan menampilkan struktur panitia beserta <i>button</i> “Sunting Struktur Panitia” dan “Hapus Struktur Panitia”. Namun, jika belum ada, sistem akan menyediakan <i>button</i> “Tambah Struktur Panitia”.
7	Case 1 : Menekan “Tambah Struktur Panitia”	
8		Menyediakan form inputan data yang berisi nama panitia beserta perannya. Selain itu, terdapat <i>button</i> “Simpan” dan “Cancel”
9	Case 1.1 : Menekan “Cancel”	
10		Menampilkan halaman detail kegiatan.

11	Case 1.2 : Menekan “Simpan” Staf lembaga mengisi seluruh nama panitia beserta perannya dan menekan <i>button</i> “Simpan”.	
12		Menyimpan ke <i>database</i> dan menampilkan halaman detail kegiatan yang terdapat struktur panitianya
13	Case 2 : Menekan “Sunting Struktur Panitia”	
14		Menampilkan struktur panitia yang terdapat nama dan perannya. Selain itu, terdapat <i>button</i> “Simpan” dan “Cancel”
15	Case 2.1 : Menekan “Cancel”	
16		Menampilkan halaman detail kegiatan.
17	Case 2.1 : Menekan “Simpan” Mengganti sebagian atau seluruh isinya dan menekan <i>button</i> “Simpan”	
18		Menyimpan ke <i>database</i> dan menampilkan halaman detail kegiatan yang terdapat struktur panitianya
19	Case 3 : Menekan “Hapus Struktur Panitia”	
20		Menampilkan modal yang berisi pertanyaan “Apakah anda yakin untuk menghapus struktur ini?” dan menyediakan <i>button</i> “Ya” dan “Tidak”.
21	Case 3.1 : Menekan “Tidak”	
22		Menampilkan halaman detail kegiatan.
23	Case 3.2 : Menekan “Ya”	
24		Menghapus struktur panitia yang ada di <i>database</i> dan menampilkan halaman detail kegiatan

Extensions (or Alternative Flows): Jika data sarana dan prasarana tidak muncul, bisa saja disebabkan oleh tiga hal, yaitu staf lembaga memang belum menambahkan daftar kepanitiaannya, koneksi internet kurang memadai, atau server yang dimiliki sistem kami sedang *down*.

Special Requirements:

1. Data struktur panitia kegiatan harus terintegrasi dengan sistem lain, yaitu sistem Data pendaftaran pada kegiatan
2. Sistem harus memiliki fitur *backup* dan *recovery* untuk memastikan data struktur panitia tidak hilang dalam kejadian bencana atau kecelakaan lainnya.

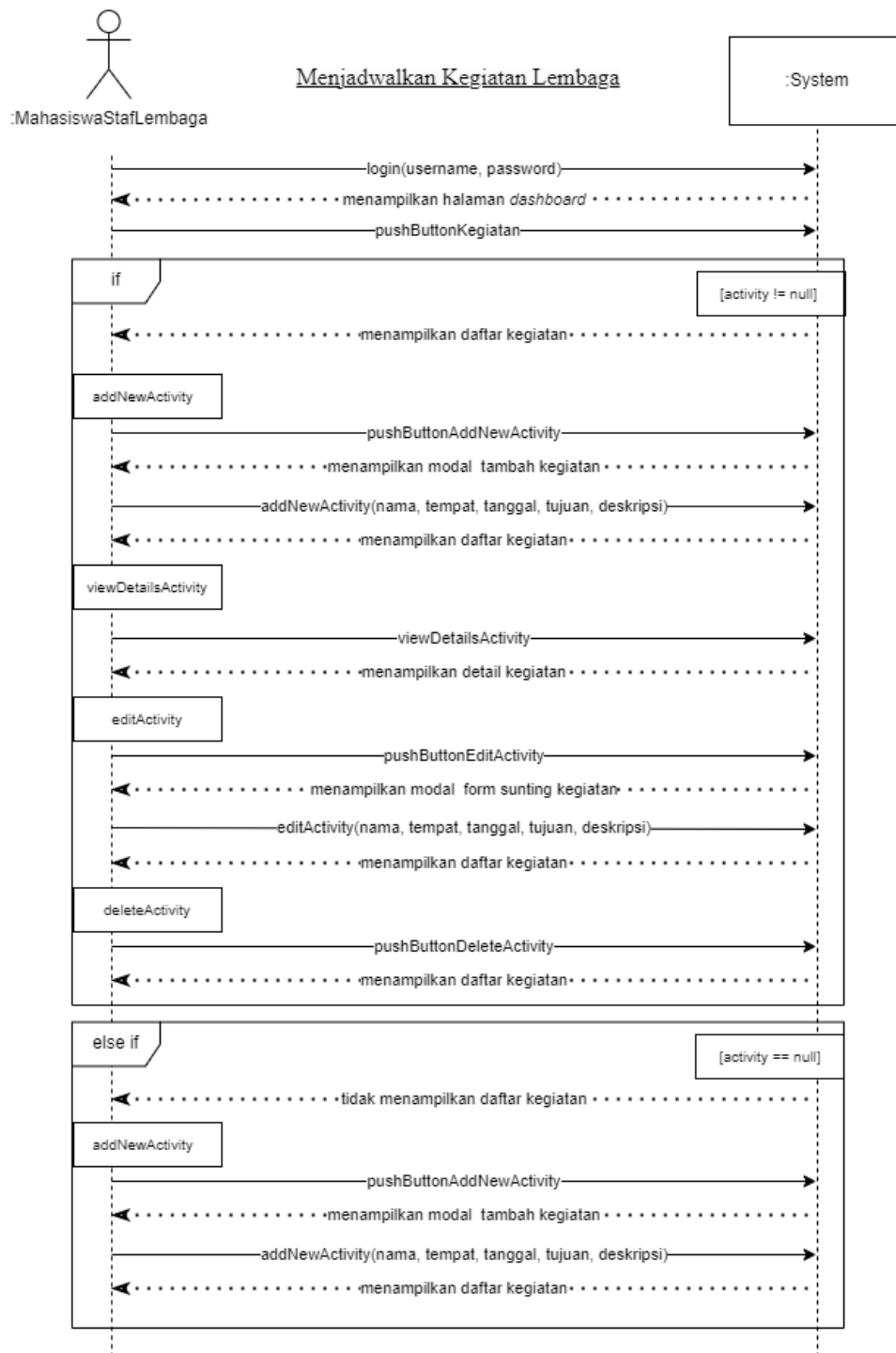
Technology and Data Variations List:

1. Data Staf Lembaga (Id Staff, Nama, Username, Password, Email)
2. Data Mahasiswa Umum (Nama, NIM, Username, Password, Email)
3. Data Admin (Id Admin, Nama, Username, Password)

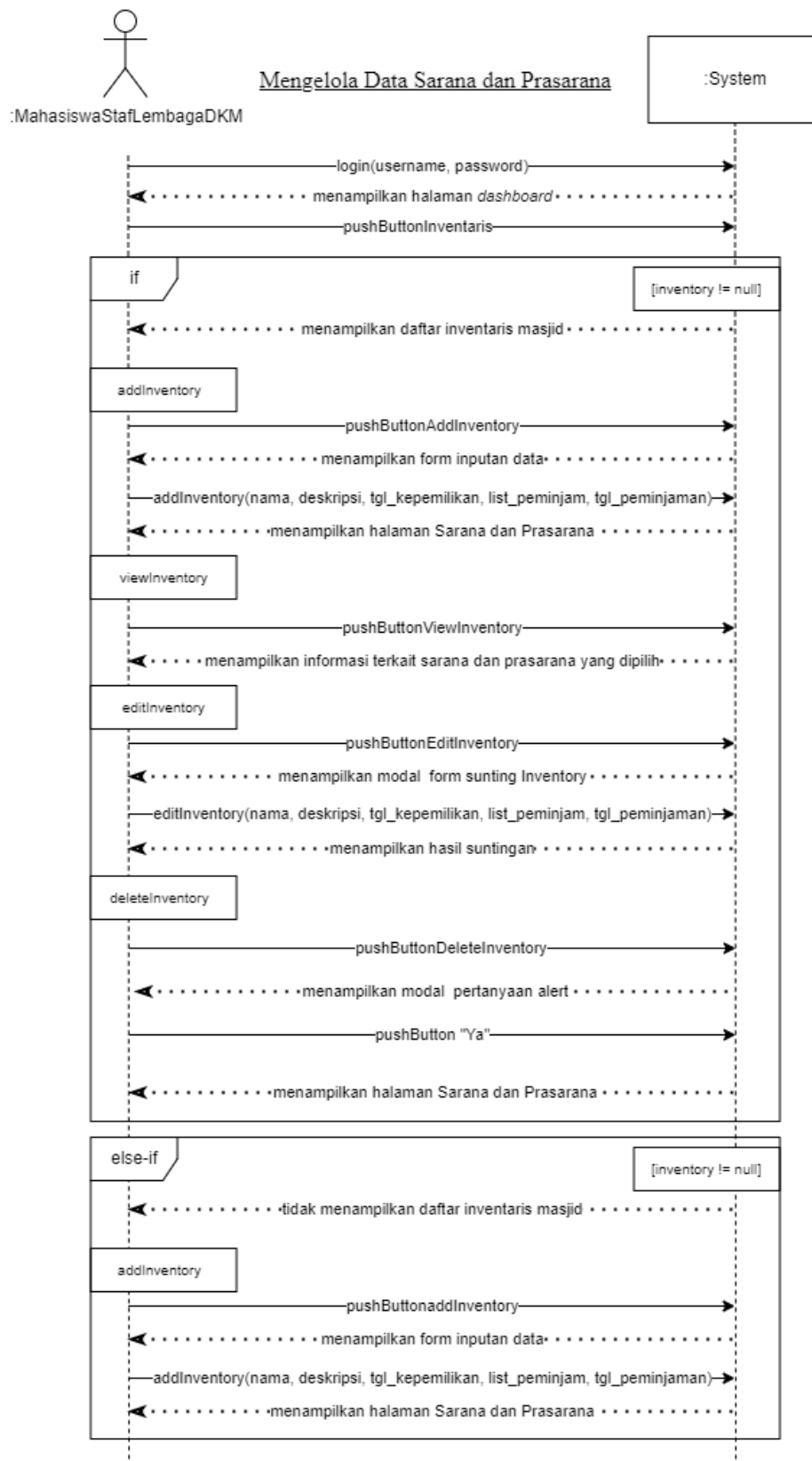
Frequency of Occurrence: Harian**Open Issues:** -

3.3.3. System Sequence Diagram

1. Menjadwalkan Kegiatan Lembaga

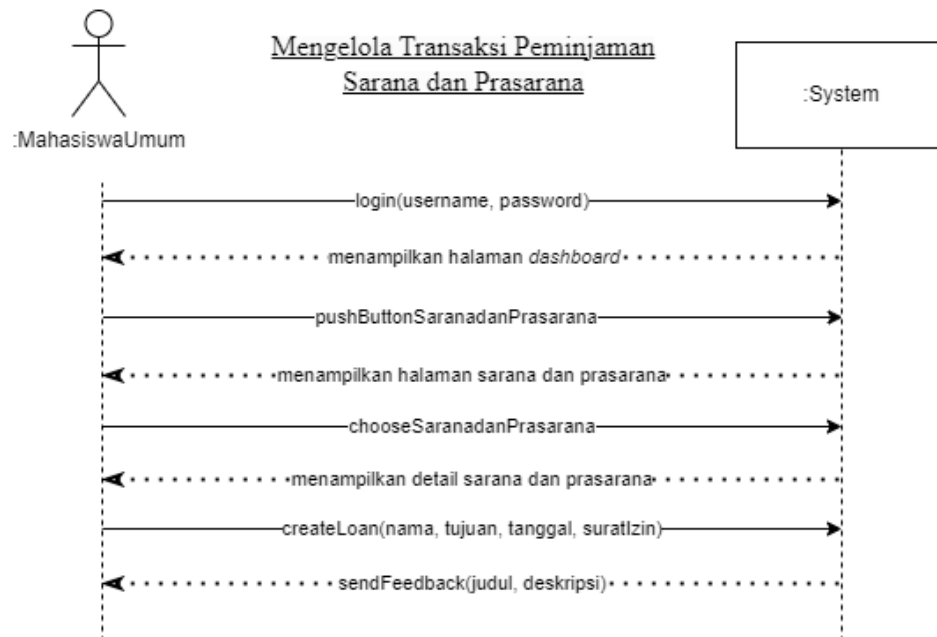


2. Mengelola Data Sarana dan Prasarana

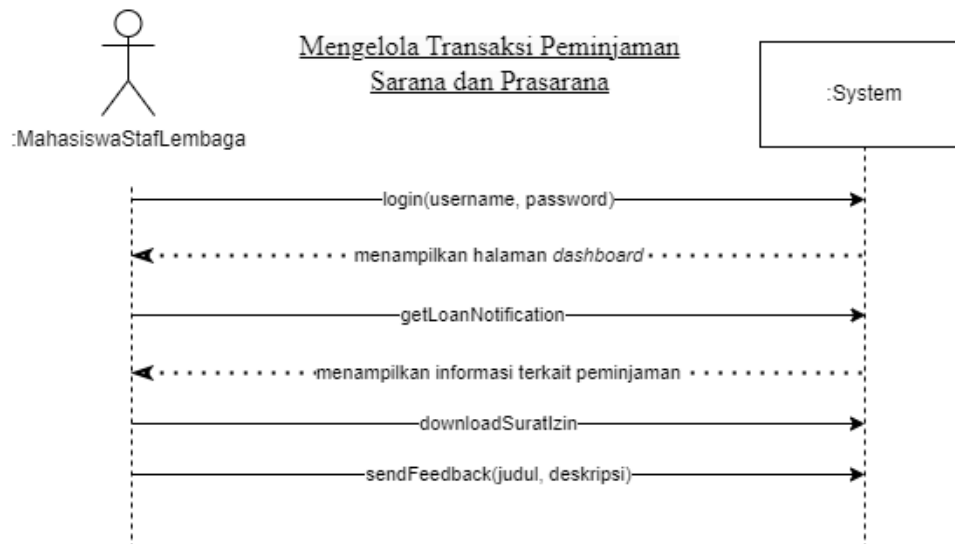


3. Mengelola Transaksi Peminjaman Sarana dan Prasarana

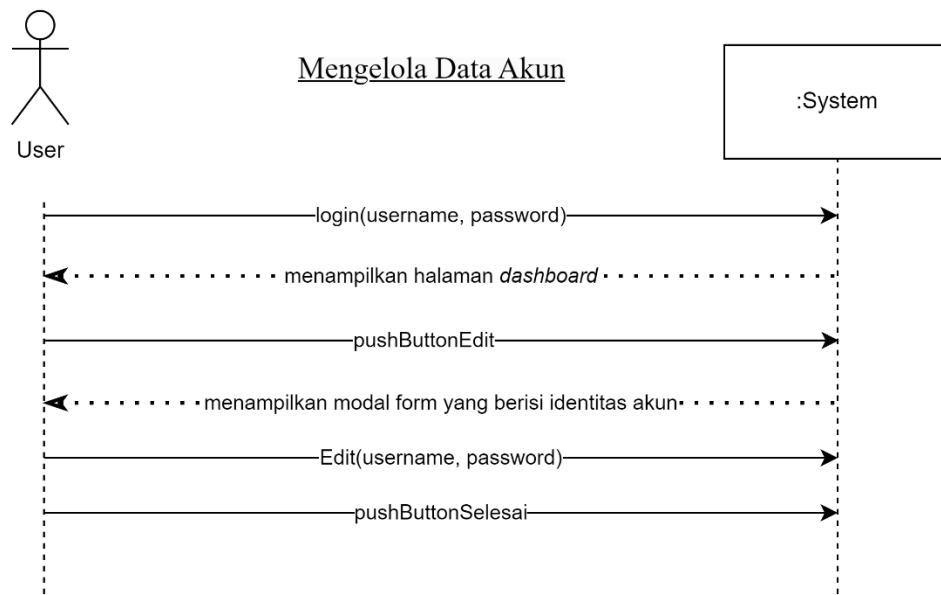
- Interaksi Mahasiswa Umum dengan Sistem



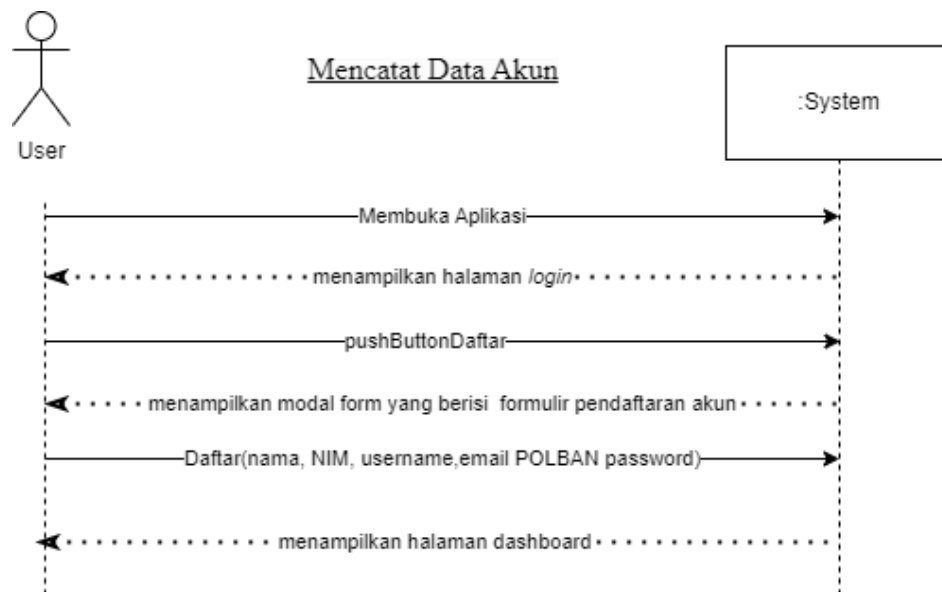
- Interaksi Staf Lembaga dengan Sistem



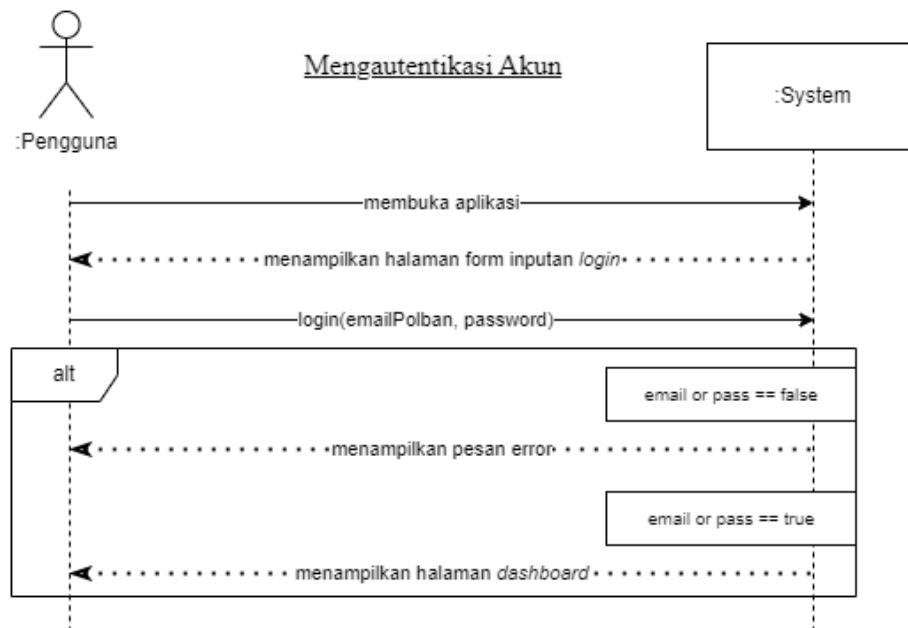
4. Mengelola Data Akun



5. Mencatat Data Akun

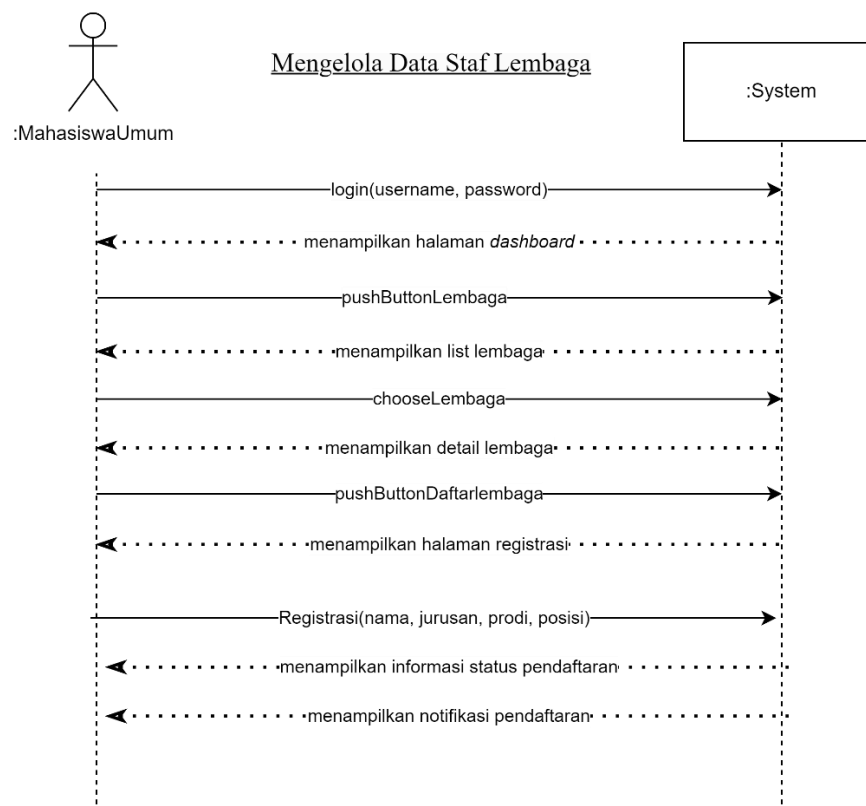


6. Mengautentikasi Akun

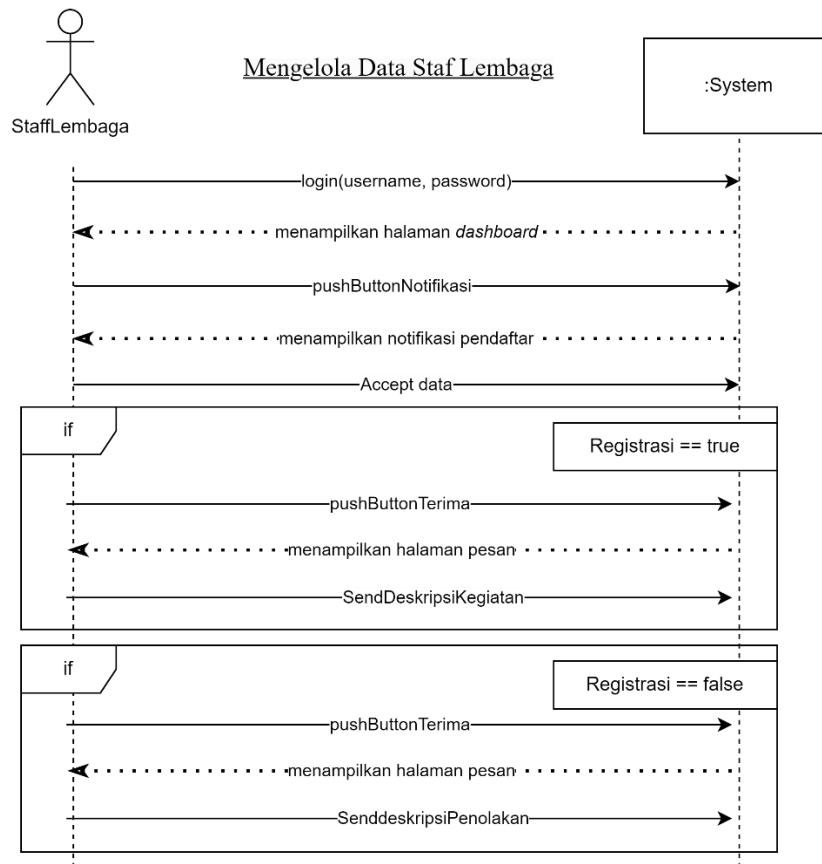


7. Mengelola Data Staf Lembaga

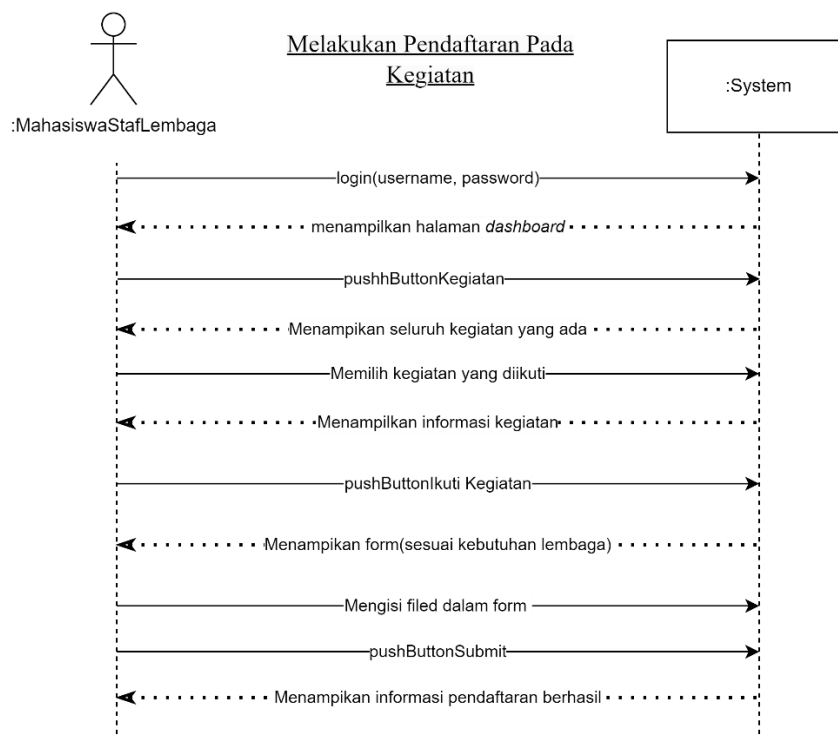
- Interaksi Mahasiswa Umum dengan Sistem



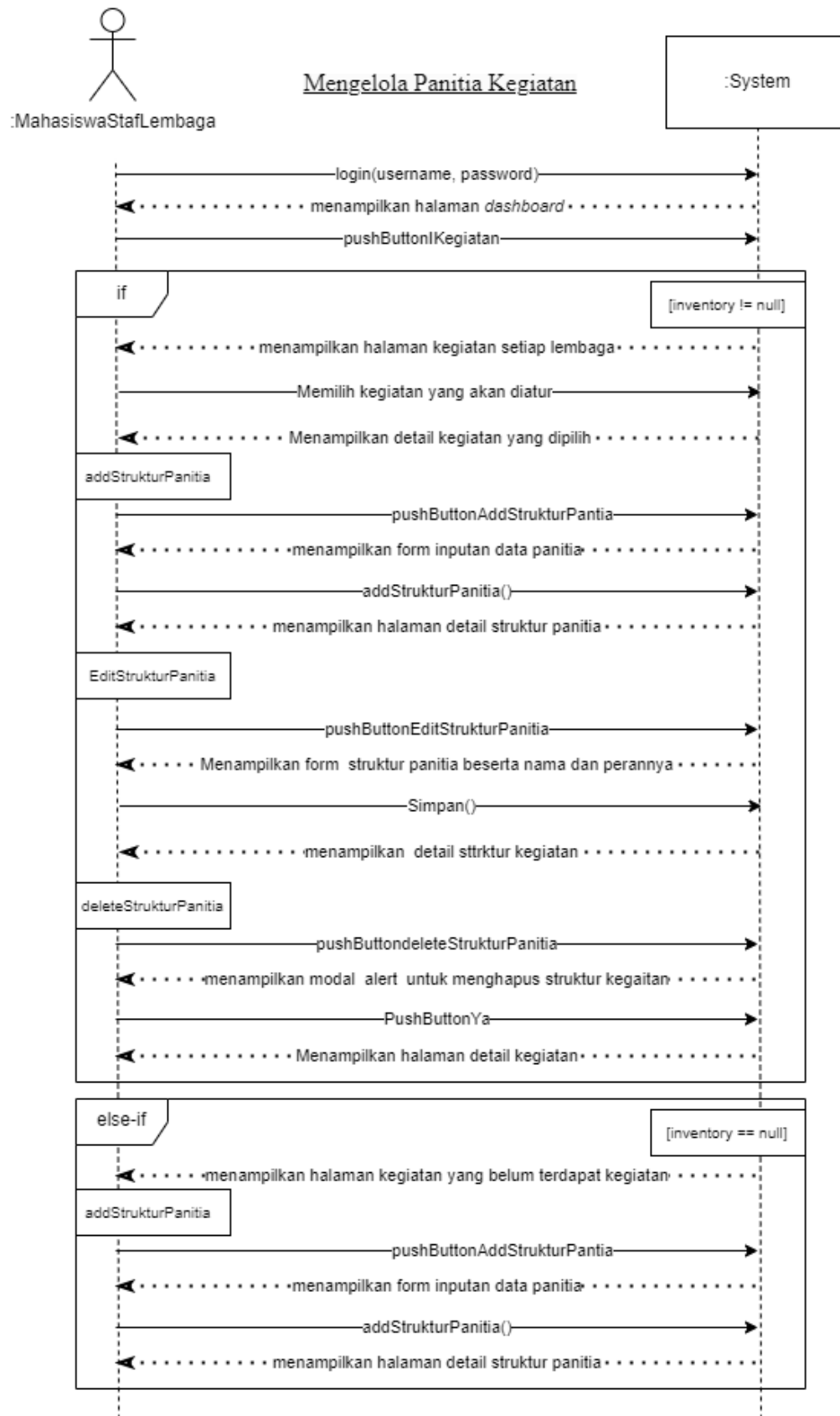
- Interaksi Staff Lembaga dengan Sistem



8. Melakukan Pendaftaran pada Kegiatan



9. Mengelola Panitia Kegiatan



3.3.4. Operation Contract

1. Menjadwalkan Kegiatan Lembaga

Contract CO1: addNewActivity

Operation	: addNewActivity(nama:string, tempat: string, tanggal: date, tujuan: string, deskripsi: string)
Cross References	: Use Cases: Menjadwalkan Kegiatan Lembaga
Preconditions	: Staf Lembaga harus sudah terdaftar pada aplikasi
Postconditions	: <ul style="list-style-type: none"> • Kegiatan baru berhasil dibuat • Pengguna dapat melihat kegiatan baru yang telah ditambahkan pada daftar kegiatan

Contract CO2: editActivity

Operation	: editActivity(nama:string, tempat: string, tanggal: date, tujuan: string, deskripsi: string)
Cross References	: Use Cases: Menjadwalkan Kegiatan Lembaga
Preconditions	: Staf Lembaga harus sudah terdaftar pada aplikasi
Postconditions	: <ul style="list-style-type: none"> • Kegiatan berhasil diubah • Pengguna dapat melihat kegiatan yang telah diubah pada daftar kegiatan

2. Mengelola Data Sarana dan Prasarana

Contract CO3: addInventory

Operation	: addInventory(nama: string, deskripsi: string, tgl_kepemilikan: date, list_peminjam: list, tgl_peminjaman: date)
Cross References	: Use Cases: Mengelola Data Sarana dan Prasarana
Preconditions	: Staf Lembaga harus sudah terdaftar pada aplikasi
Postconditions	: <ul style="list-style-type: none"> • Sarana dan prasarana baru berhasil ditambahkan ke inventaris masjid • Pengguna dapat melihat daftar sarana dan prasarana masjid yang telah ditambahkan

Contract CO4: editInventory

Operation	: editInventory(nama: string, deskripsi: string, tgl_kepemilikan: date, list_peminjam: list, tgl_peminjaman: date)
Cross References	: Use Cases: Mengelola Data Sarana dan Prasarana
Preconditions	: Staf Lembaga harus sudah terdaftar pada aplikasi
Postconditions	: <ul style="list-style-type: none"> • Sarana dan prasarana berhasil disunting • Pengguna dapat melihat daftar sarana dan prasarana masjid yang telah disunting

3. Mengelola Transaksi Peminjaman Sarana dan Prasarana

Contract CO5: createLoan

Operation	: createLoan(nama: string, tujuan: string, tanggal: date, suratIzin: file)
Cross References	: Use Cases: Mengelola Transaksi Peminjaman Sarana dan Prasarana
Preconditions	: <ul style="list-style-type: none"> • Mahasiswa Umum harus sudah terdaftar pada aplikasi • Data sarana dan prasarana telah tersedia dalam database • Sarana dan prasarana yang akan dipinjam belum dipinjam oleh pihak lain
Postconditions	: <ul style="list-style-type: none"> • Permintaan peminjaman berhasil dibuat

Contract CO6: sendFeedback

Operation	: sendFeedback(judul: string, deskripsi: string)
Cross References	: Use Cases: Menjadwalkan Kegiatan Lembaga
Preconditions	: <ul style="list-style-type: none"> • Mahasiswa Umum harus sudah terdaftar pada aplikasi • Mahasiswa Umum harus sudah mengajukan peminjaman sarana dan prasarana pada sistem
Postconditions	: <ul style="list-style-type: none"> • <i>Feedback</i> berhasil tersampaikan kepada Mahasiswa Umum • Mahasiswa Umum dapat melihat <i>feedback</i> yang diterima berupa disetujui atau tidak

4. Mengelola Data Akun

Contract CO7: Edit

Operation	: Edit(username : string, password : string)
Cross References	: Use Cases: Mengelola Data Akun
Preconditions	: Akun sudah terdaftar pada aplikasi
Postconditions	: Sebagian informasi akun berhasil dibuat

5. Mencatat Data Akun

Contract CO8: Daftar

Operation	: Daftar(nama, username, NIM, Email POLBAN, password)
Cross References	: Use Cases: Mencatat Data Akun
Preconditions	: -
Postconditions	: <ul style="list-style-type: none"> • Akun berhasil didaftarkan pada aplikasi • user bisa mengakses aplikasi

6. Mengautentikasi Akun

Contract CO9: login

Operation	: login(emailPolban: string, password: string)
Cross References	: Use Cases: Mengautentikasi Akun
Preconditions	: Akun pengguna sudah terdaftar lebih dulu di aplikasi
Postconditions	: <ul style="list-style-type: none"> • Autentikasi akun berhasil • Pengguna berhasil masuk ke halaman <i>dashboard</i>

7. Mengelola Data Staf Lembaga

Contract C10: Registrasi

Operation	: registrasi(nama, jurusan, prodi, posisi)
Cross References	: Use Cases: Mengelola pendaftaran pada Lembaga
Preconditions	: Akun pengguna sudah terdaftar lebih dulu di aplikasi
Postconditions	: <ul style="list-style-type: none"> • user berhasil terdaftar pada lembaga • user ditolak masuk pada lembaga

8. Melakukan Pendaftaran pada Kegiatan

Contract C11: registerKegiatan

Operation	: registerKegiatan()
Cross References	: Use Cases: MengelolaData Pendaftaran pada kegiatan
Preconditions	: Sudah terdaftar pada aplikasi
Postconditions	: User diterima menjadi partisipan kegiatan

9. Mengelola Panitia Kegiatan

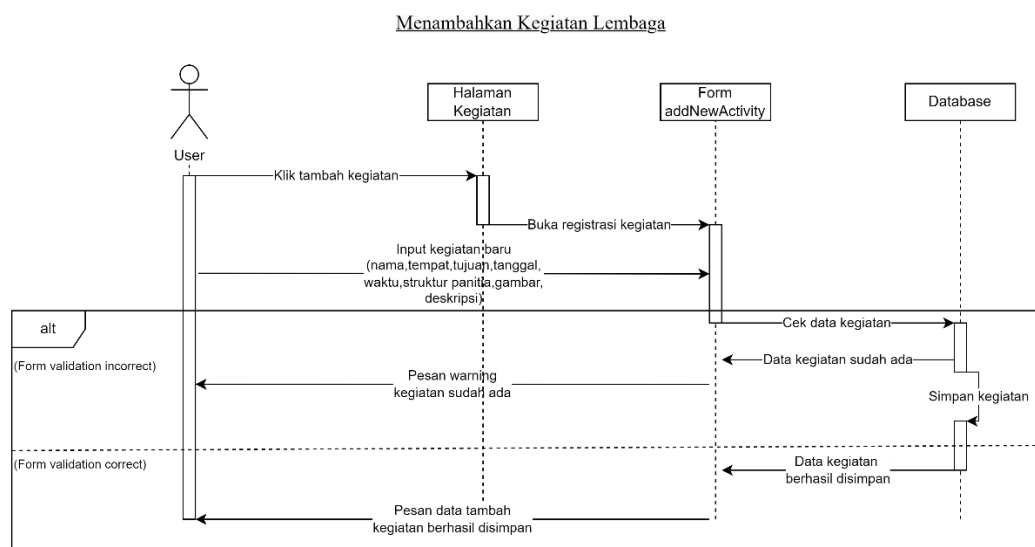
Contract C12: addStrukturPanitia

Operation	: addStrukturPanitia
Cross References	: Use Cases: Mengelola Panitia Kegiatan
Preconditions	: Kegiatan sudah ditambahkan pada aplikasi
Postconditions	: <ul style="list-style-type: none"> Struktur panitia berhasil dibuat Mahasiswa umum dan staff dapat melihat strktur panitia dalam detial deskripsi kegiatan

3.3.5. Sequence Diagram

1. Menjadwalkan Kegiatan Lembaga

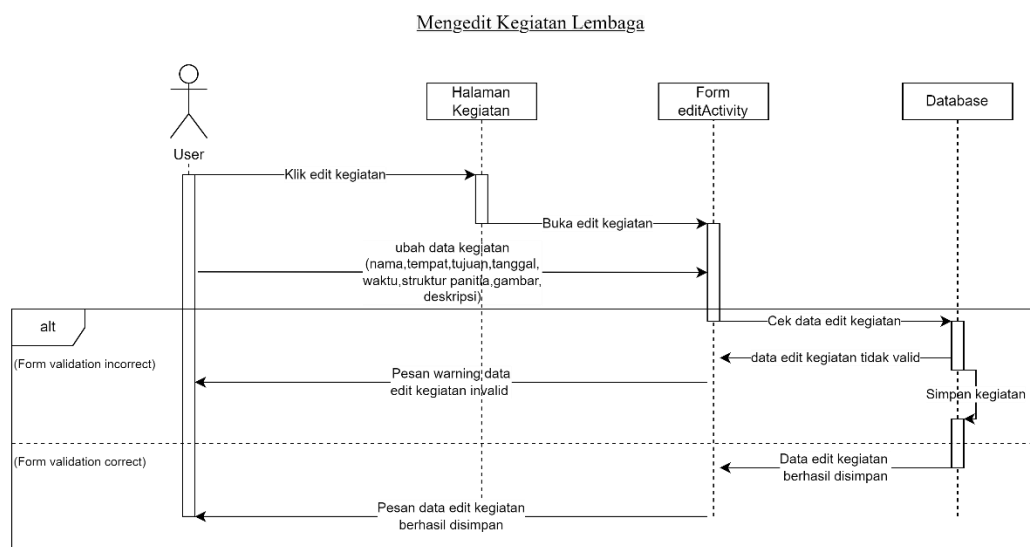
a. Menambahkan Kegiatan Lembaga



No SD	SD_01
Nama SD	Menambahkan Kegiatan Lembaga
Deskripsi	Mahasiswa Staf Lembaga menambahkan kegiatan baru

ID Use Case	UC01
ID Functional Requirement	REQ-F-04, REQ-F-09
ID Non-Functional Requirement	REQ-NF-04
ID Business Rules	-
Nama Class Terkait	Kegiatan_Lembaga, Kegiatan_DKM
Method Terkait	addNewActivity()

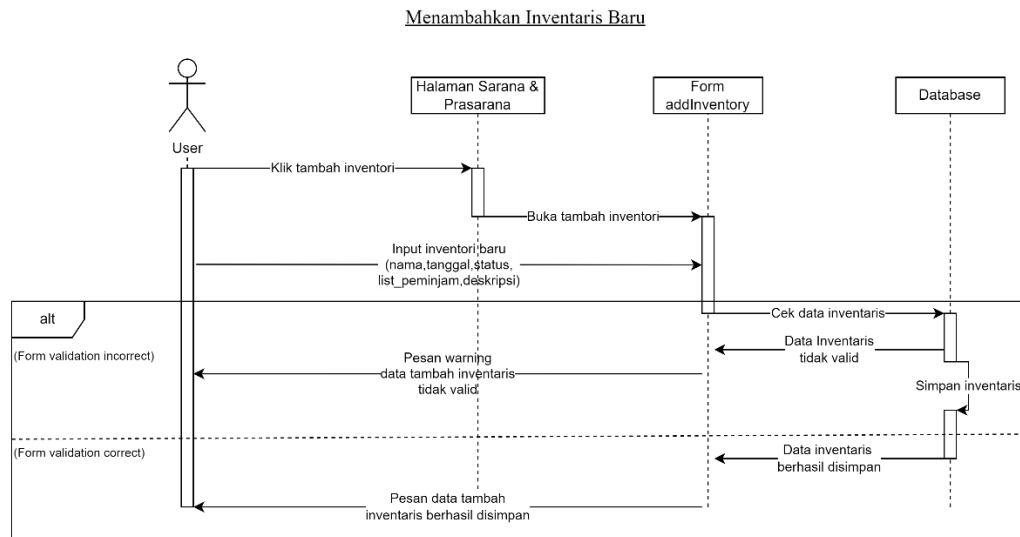
b. Mengedit Kegiatan Lembaga



No SD	SD_02
Nama SD	Mengedit Kegiatan Lembaga
Deskripsi	Mahasiswa Staf Lembaga mengedit kegiatan yang sudah ada
ID Use Case	UC01
ID Functional Requirement	REQ-F-04, REQ-F-09
ID Non-Functional Requirement	REQ-NF-04
ID Business Rules	-
Nama Class Terkait	Kegiatan_Lembaga, Kegiatan_DKM
Method Terkait	editActivity()

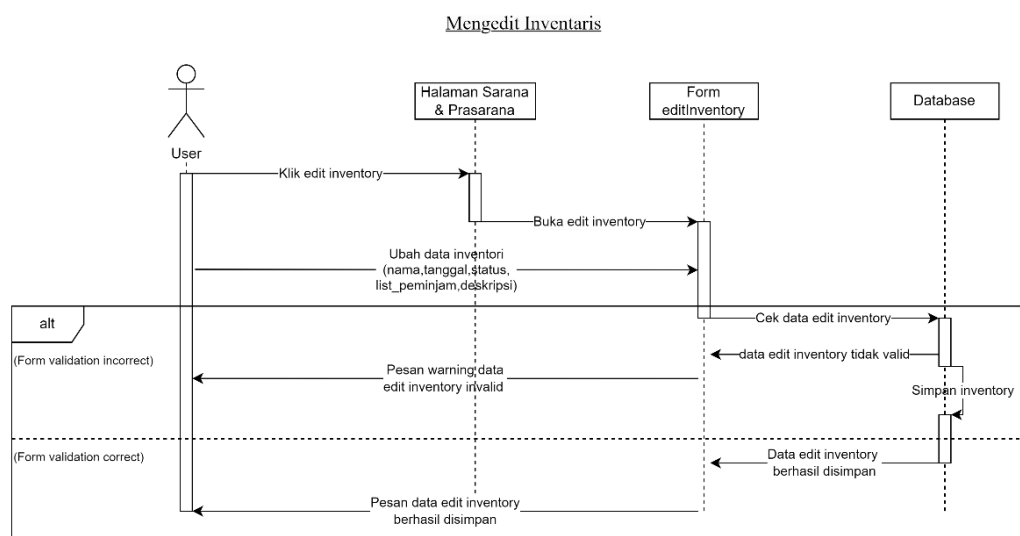
2. Mengelola Data Sarana dan Prasarana

a. Menambahkan Inventaris Baru



No SD	SD_03
Nama SD	Menambahkan Inventaris Baru
Deskripsi	Mahasiswa Staf Lembaga DKM menambahkan inventaris baru yang dimiliki mesjid
ID Use Case	UC02
ID Functional Requirement	REQ-F-15, REQ-F-17, REQ-F-05
ID Non-Functional Requirement	REQ-NF-16, REQ-NF-17, REQ-NF-18, REQ-NF-19, REQ-NF-21, REQ-NF-25
ID Business Rules	-
Nama Class Terkait	Inventaris
Method Terkait	CreateInventory()

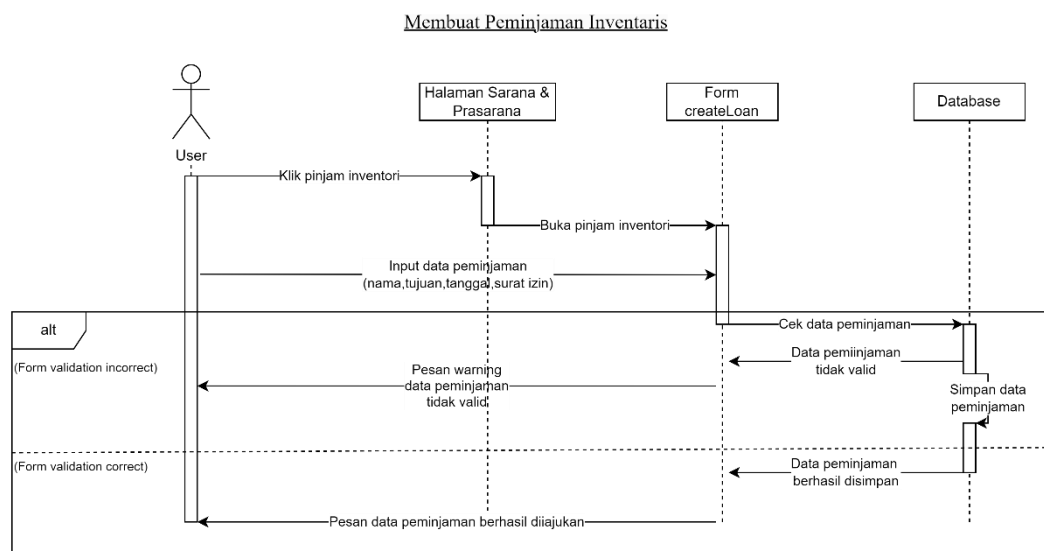
b. Mengedit Inventaris



No SD	SD_04
Nama SD	Mengedit Inventaris
Deskripsi	Mahasiswa Staf Lembaga DKM mengedit inventaris yang dimiliki mesjid
ID Use Case	UC02
ID Functional Requirement	REQ-F-16
ID Non-Functional Requirement	REQ-NF-16, REQ-NF-17, REQ-NF-18, REQ-NF-19, REQ-NF-21, REQ-NF-25
ID Business Rules	-
Nama Class Terkait	Inventaris
Method Terkait	updateInventory()

3. Mengelola Transaksi Peminjaman Sarana dan Prasarana

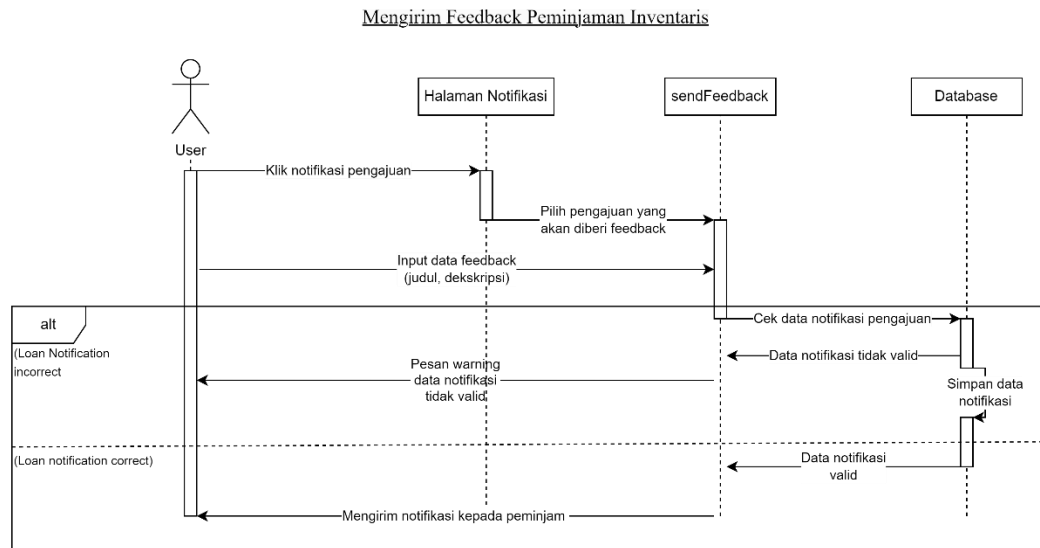
a. Membuat Peminjaman Inventaris



No SD	SD_05
Nama SD	Membuat Peminjaman Inventaris
Deskripsi	Mahasiswa Umum meminjam inventaris yang dimiliki mesjid melewati perantara Staf Lembaga DKM
ID Use Case	UC03
ID Functional Requirement	REQ-F-06,
ID Non-Functional Requirement	REQ-NF-11, REQ-NF-16, REQ-NF-17, REQ-NF-18, REQ-NF-19, REQ-NF-21, REQ-NF-25
ID Business Rules	
Nama Class Terkait	Inventaris

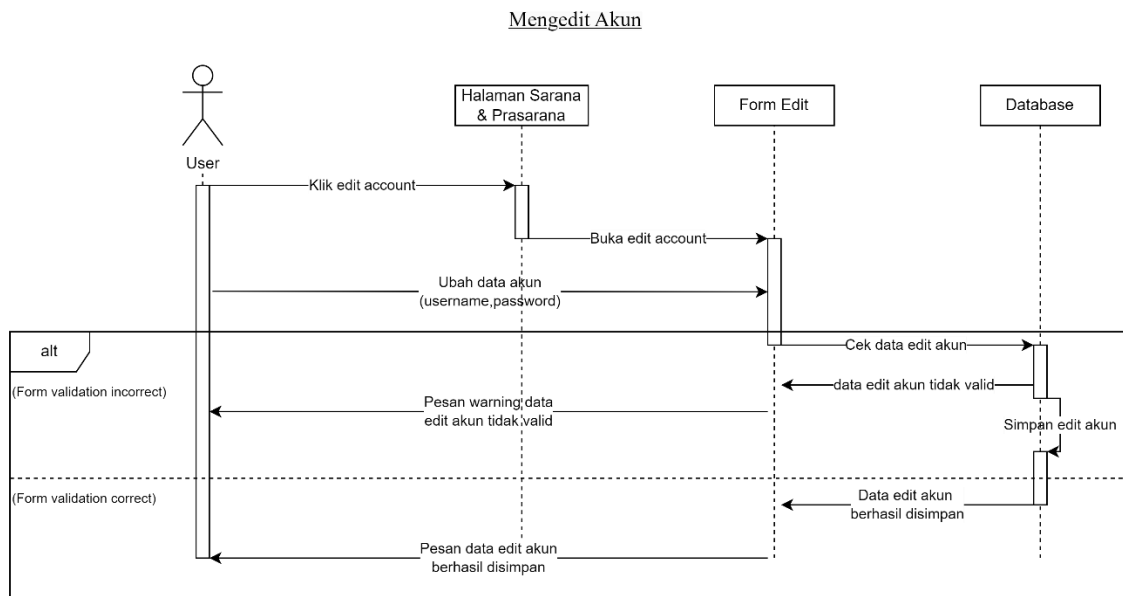
Method Terkait	createLoan()
-----------------------	--------------

b. Mengirim Feedback Peminjaman Inventaris



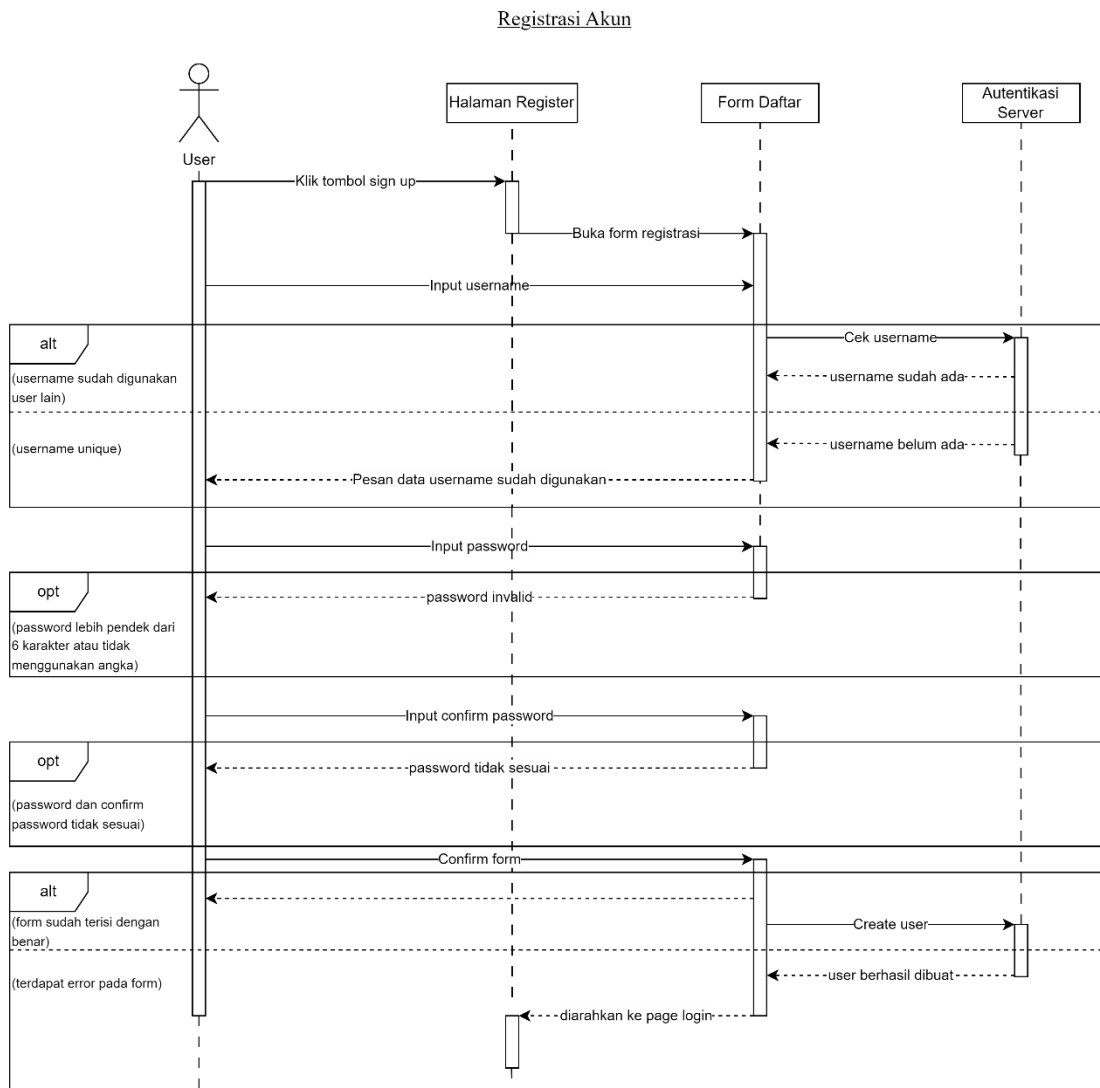
No SD	SD_06
Nama SD	Mengirim Feedback Peminjaman Inventaris
Deskripsi	Mahasiswa Staf Lembaga DKM mengirim feedback kepada Mahasiswa Umum terkait status peminjaman inventarisnya
ID Use Case	UC03
ID Functional Requirement	REQ-F-17
ID Non-Functional Requirement	REQ-NF-16, REQ-NF-17, REQ-NF-18, REQ-NF-19, REQ-NF-21, REQ-NF-25
ID Business Rules	BR-03
Nama Class Terkait	Inventaris
Method Terkait	sendFeedback()

4. Mengelola Data Akun



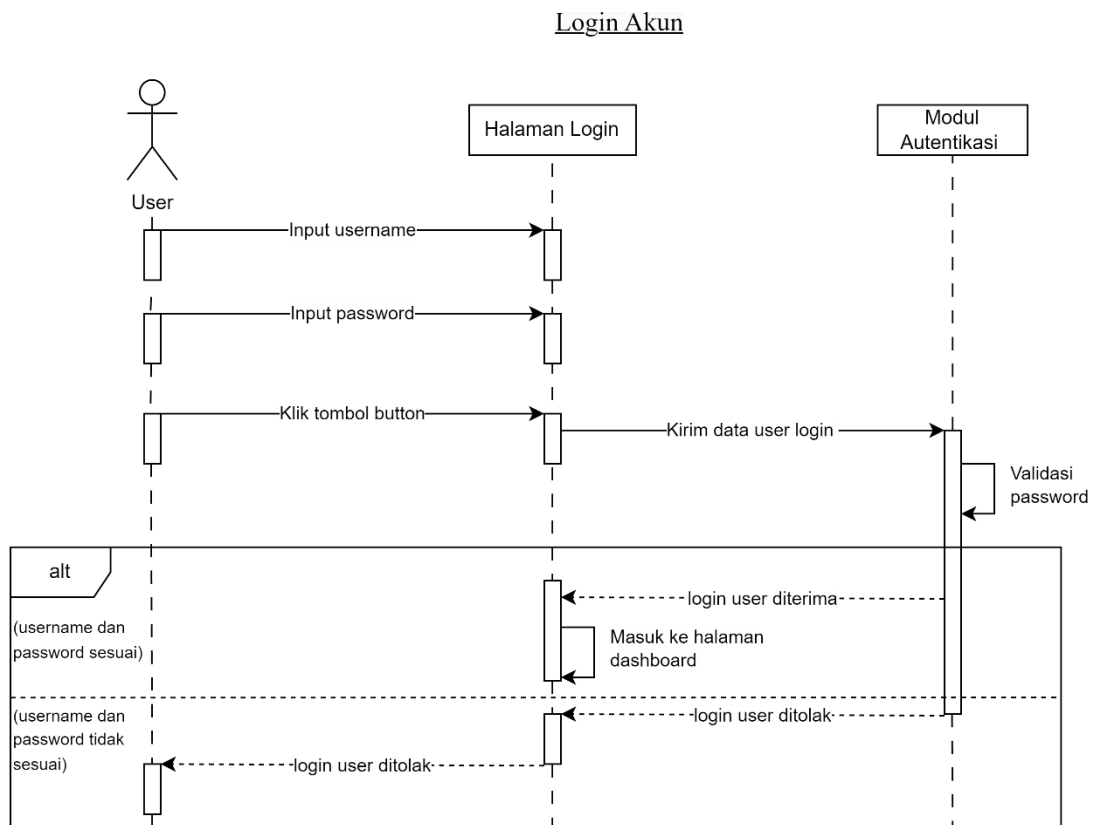
No SD	SD_07
Nama SD	Mengedit Akun
Deskripsi	Pengguna mengedit akun yang dimilikinya
ID Use Case	UC04
ID Functional Requirement	REQ-F-01, REQ-F-02
ID Non-Functional Requirement	-
ID Business Rules	BR-01, BR-05
Nama Class Terkait	Admin, Mahasiswa_Staf_DKM, Mahasiswa_Staf_Lembaga, Mahasiswa_Umum
Method Terkait	Edit()

5. Mencatat Data Akun



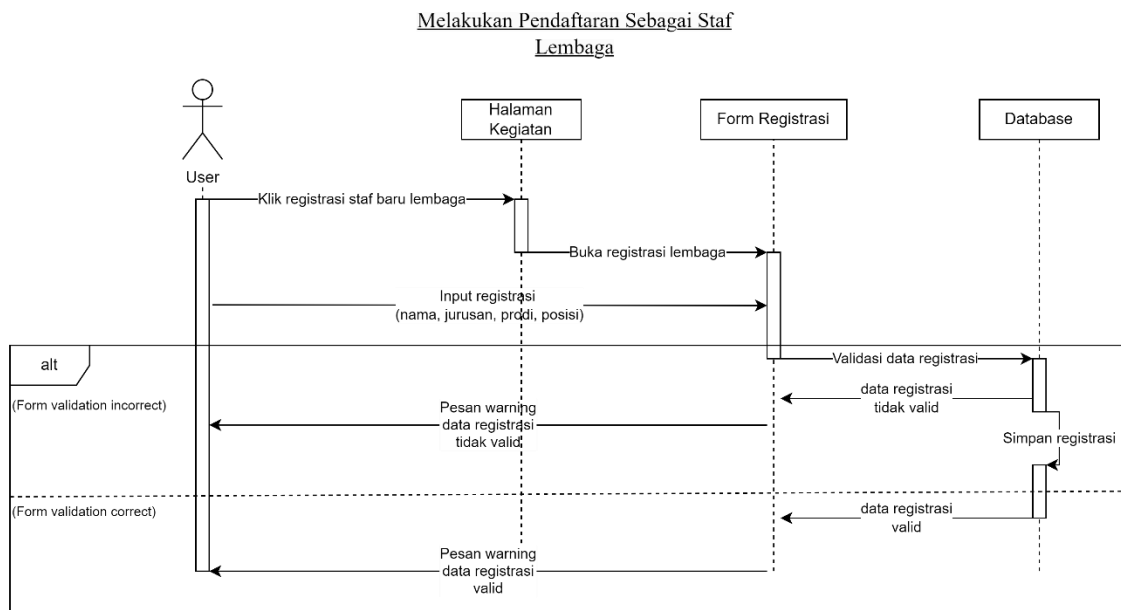
No SD	SD_08
Nama SD	Registrasi Akun
Deskripsi	Mahasiswa Umum membuat akun dalam aplikasi
ID Use Case	UC05
ID Functional Requirement	REQ-F-01
ID Non-Functional Requirement	REQ-NF-01
ID Business Rules	BR-01
Nama Class Terkait	Mahasiswa_Umum
Method Terkait	Daftar()

6. Mengautentikasi Akun



No SD	SD_09
Nama SD	Login Akun
Deskripsi	Pengguna login ke dalam aplikasi dengan akun pribadinya masing-masing
ID Use Case	UC06
ID Functional Requirement	REQ-F-02, REQ-F-07
ID Non-Functional Requirement	REQ-NF-02
ID Business Rules	BR-01
Nama Class Terkait	Mahasiswa_Staf_Lembaga, Mahasiswa_Staf_DKM, Mahasiswa_Umum
Method Terkait	login()

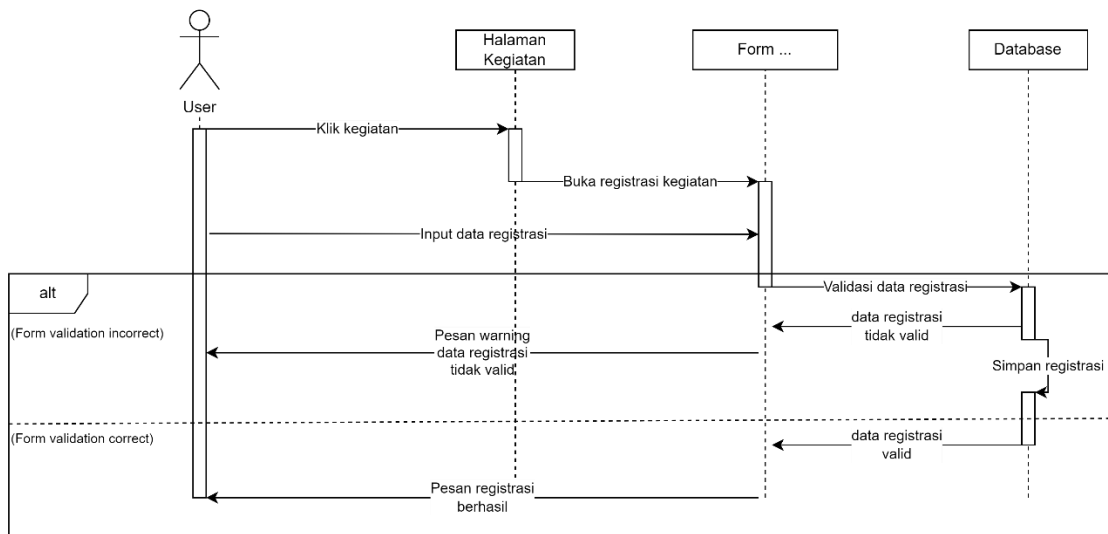
7. Mengelola Data Staf Lembaga



No SD	SD_10
Nama SD	Melakukan Pendaftaran Sebagai Staf Lembaga
Deskripsi	Mahasiswa Umum melakukan pendaftaran sebagai staf lembaga baru
ID Use Case	UC07
ID Functional Requirement	REQ-F-11
ID Non-Functional Requirement	REQ-NF-08
ID Business Rules	-
Nama Class Terkait	Mahasiswa_Umum
Method Terkait	Registrasi()

8. Melakukan Pendaftaran pada Kegiatan

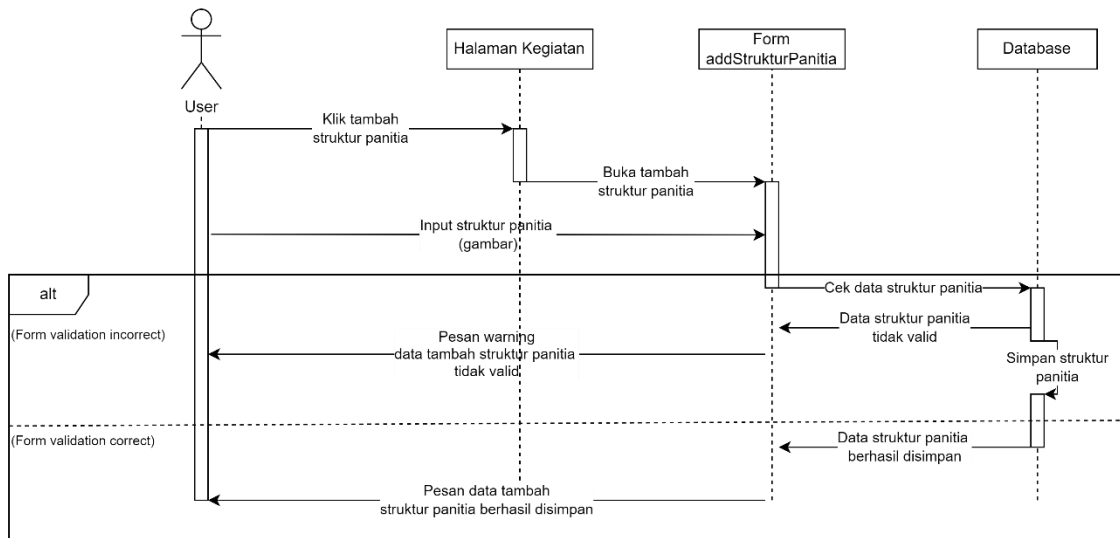
Melakukan Pendaftaran Pada Kegiatan Lembaga



No SD	SD_11
Nama SD	Melakukan Pendaftaran Pada Kegiatan Lembaga
Deskripsi	Mahasiswa Umum melakukan pendaftaran untuk mengikuti kegiatan yang diselenggarakan oleh lembaga
ID Use Case	UC08
ID Functional Requirement	REQ-F-09
ID Non-Functional Requirement	REQ-NF-08, REQ-NF-09,
ID Business Rules	-
Nama Class Terkait	Kegiatan_DKM, Kegiatan_Lembaga, Mahasiswa_Umum
Method Terkait	registerKegiatan()

9. Mengelola Panitia Kegiatan

Menambahkan Struktur Panitia



No SD	SD_12
Nama SD	Menambahkan Struktur Panitia
Deskripsi	Mahasiswa Staf Lembaga menambahkan struktur panitia dari kegiatan yang akan diselenggarakan
ID Use Case	UC09
ID Functional Requirement	REQ-F-08
ID Non-Functional Requirement	REQ-NF-04
ID Business Rules	-
Nama Class Terkait	Mahasiswa_Staf_lembaga
Method Terkait	addStrukturPanitia()

4. External Interface Requirements

4.1. User Interfaces

Aplikasi Pengelolaan Lembaga Dakwah Kampus Polban ini digunakan oleh Staf Lembaga yang ada di Lembaga Dakwah Kampus Polban, seluruh mahasiswa umum Polban, dan pengelola aplikasi (Admin).

Untuk dapat memudahkan pelaksanaan proses masukan dan keluaran baik dari data maupun informasi antara aplikasi dan pengguna, maka perlu rancangan *user interface*. *User interface* yang dirancang pada aplikasi ini menggunakan website yang bisa *responsive* di perangkat *desktop*, *laptop*, maupun *android*.

Pada aplikasi ini, terdapat tiga rancangan *interface* yang terdiri untuk Staf Lembaga, Mahasiswa Umum, dan Admin. Rancangan *user interface* yang diperlukan dalam aplikasi ini adalah:

4.1.1. Staf Lembaga

1. Halaman Login

Halaman autentikasi staf lembaga yang terdapat form berupa *field email* dan *password*. Selain itu, terdapat tombol untuk memulihkan kata sandi dan terdapat tombol untuk masuk ke halaman register (pendaftaran akun).

2. Halaman Profil Lembaga

Halaman yang berisi semua data terkait lembaga tersebut, seperti foto profil lembaga, nama lembaga, struktur kepengurusan, dan jadwal berupa kegiatan yang sudah pernah dilaksanakan dan kegiatan yang akan dilaksanakan. Selain itu, terdapat tombol untuk mengubah data tersebut.

3. Halaman Kegiatan

Halaman untuk mengelola data kegiatan lembaga tersebut, seperti pengelolaan informasi detail kegiatan dan pengelolaan panitia pada kegiatan terkait.

4. Halaman Sarana dan Prasarana (DKM)

Halaman khusus lembaga DKM untuk mengelola sarana dan prasarana. Halaman ini akan berisi *list* inventaris beserta detail informasi inventaris tersebut yang dimiliki Masjid Lukmannul Hakim Polban. Terdapat *action* untuk melakukan CRUD pada *list* inventaris tersebut.

5. Halaman Penerimaan Mahasiswa

Halaman untuk mengelola data mahasiswa yang sudah melakukan pendaftaran pada suatu lembaga. Selain itu, terdapat tombol *action* yang jika diklik akan menampilkan modal “*pop up*” sebagai perantara untuk memberikan *feedback*-nya kepada mahasiswa bersangkutan.

6. Halaman Notifikasi

Halaman untuk mendapatkan data informasi terbaru dari mahasiswa yang sudah melakukan pendaftaran pada lembaga atau dari mahasiswa yang sudah melakukan pengajuan peminjaman sarana dan prasarana.

4.1.2. Mahasiswa

1. Halaman Register

Halaman yang berisi *form* berupa *field* Input nama, NIM, *username*, *email* polban, dan *password*. Selain itu, terdapat tombol “Login” jika mahasiswa tersebut sudah pernah melakukan pendaftaran akun.

2. Halaman Login

Sama seperti halaman login yang ada pada staf lembaga, halaman autentikasi mahasiswa umum akan terdapat *form* berupa *field email* dan *password*. Selain itu, terdapat tombol untuk memulihkan kata sandi dan terdapat tombol untuk masuk ke halaman register (pendaftaran akun).

3. Halaman Dashboard

Halaman yang berisi *list* tiga lembaga dakwah kampus, data kegiatan yang akan datang, kegiatan yang sudah berlalu, serta navigasi bar yang terdiri dari tombol halaman profil pengguna, halaman sarana dan prasarana, halaman lembaga, dan sign out.

4. Halaman Lembaga

Halaman yang berisi data-data lengkap terkait tiap lembaga yang ada dan dapat melakukan pengelolaan informasi tersebut.

5. Halaman Sarana dan Prasarana

Halaman yang digunakan untuk meminjam sarana dan prasarana masjid.

4.1.3. Admin

1. Halaman Login

Sama seperti halaman login yang ada pada staf lembaga atau mahasiswa, halaman autentikasi admin akan terdapat *form* berupa *field username* dan *password*. Selain itu, terdapat tombol untuk memulihkan kata sandi dan terdapat tombol untuk masuk ke halaman register (pendaftaran akun).

2. Halaman Staf lembaga

Halaman ini berisi list seluruh akun staf lembaga yang ada dalam Lembaga Dakwah Kampus, baik DKM, Assalam, maupun Kammi. Pada halaman ini, Admin dapat membuat akun staf lembaga yang baru dan dapat pula melakukan edit pada akun staf lembaga yang sudah ada.

3. Halaman Mahasiswa Umum

Halaman ini berisi list seluruh akun mahasiswa umum yang sudah melakukan registrasi pada aplikasi kami. Pada halaman ini, Admin hanya dapat melihat rincian akun setiap mahasiswa dan menghapus akun tersebut, tanpa dapat mengedit ataupun membuat akun baru untuk mahasiswa.

4. Halaman Lembaga

Halaman ini berisi list dari lembaga yang ada dalam Lembaga Dakwah Kampus. Untuk saat ini terdapat 3 lembaga yaitu DKM, Assalam, dan Kammi. Namun, jika suatu saat terdapat kebutuhan untuk menambah atau bahkan menghapus lembaga, itu dapat dilakukan pada halaman ini.

4.2. Hardware Interfaces

Hardware interface yang dibutuhkan untuk membantu proses pembangunan dari aplikasi ini yaitu:

4.2.1. Laptop atau Komputer

Perangkat utama yang dibutuhkan untuk membantu proses pembangunan aplikasi dengan spesifikasi minimum sebagai berikut:

- Processor Intel(R) Core(TM) I7-3630QM CPU @2.40GHz (8CPUs), ~2.4GHz
- Memory 12288MB RAM
- GPU NVIDIA GeForce GT 640M
- VRAM 2 GB

4.2.2. Keyboard

Salah satu alat untuk proses menginputkan informasi yang dibutuhkan oleh sistem. Baik berupa karakter, angka maupun simbol lainnya.

4.2.3. Mouse

Dapat membantu sistem untuk dapat mengenali masukan dari *developer* dengan melakukan klik.

4.2.4. Monitor

Dapat membantu *developer* untuk mengetahui dan menampilkan apa yang menjadi *output* dari sistem. Serta menjadi perantara komunikasi antara sistem dengan *developer*.

4.2.5. Backup Storage

Aplikasi ini akan menghasilkan data penting. Oleh karena itu, *backup storage* seperti *cloud storage* diperlukan untuk memastikan data tetap aman dan terhindar dari kehilangan data.

4.3. Software Interfaces

Dalam *software interface* hanya ada satu, yaitu:

4.3.1. Gmail API

Gmail API merupakan salah satu fitur milik *google* yang membantu *application server* untuk berinteraksi dengan *email* dari *google*, yaitu gmail dengan menggunakan API yang di sediakan oleh gmail. Pada aplikasi ini, Gmail API digunakan untuk membantu bagi mahasiswa, staf lembaga, maupun admin dalam memberikan fitur forgot *password*.

4.4. Communications Interface

4.4.1. Functional Requirements Communication

Agar perangkat lunak bisa berkomunikasi dengan *user*-nya, maka setiap *user* wajib menambahkan alamat *email* unik yang mereka miliki. Dengan adanya *email* terdaftar pada aplikasi, maka perangkat lunak bisa mengirimkan berbagai pesan dan notifikasi yang harus disampaikan pada user-nya. Seperti pesan penerimaan anggota, pesan hasil pengajuan pinjaman barang, dll.

Selain itu, aplikasi juga harus kompatibel dengan banyak web *browser* seperti Google Chrome, Opera, Mozilla, dan Microsoft Edge.

4.4.2. Standar Komunikasi yang Digunakan

Untuk protokol komunikasi yang akan diterapkan pada perangkat lunak ini adalah *Hypertext Transfer Protocol Secure*(HTTPS). Protokol ini sangat umum digunakan untuk mengenkripsi data yang ditransfer antara *server* web dan *browser* web. Protokol ini juga sangat baik untuk melindungi privasi dan keamanan pengguna.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- Halaman harus selesai dimuat dalam waktu kurang dari 3 detik pada kecepatan koneksi internet minimal 3 mbps. [REQ-NF-05]
- Sistem harus memiliki tingkat disponibilitas minimal 99,9%. [REQ-NF-06]

5.2. Security Requirements

- Aplikasi harus terlindungi dari serangan seperti serangan SQL injection, cross-site scripting (XSS), dan serangan DDoS. [REQ-NF-12]
- Aplikasi hanya akan memberikan izin masuk bagi pengguna yang memiliki *email* dan *password* yang sudah terdaftar sebelumnya. [REQ-NF-13]
- Aplikasi harus memastikan bahwa data tidak dapat dicuri melalui serangan seperti *phishing* atau serangan *man-in-the-middle*. [REQ-NF-14]
- Aplikasi harus memiliki perencanaan pemulihan darurat untuk memastikan bahwa data dan sistem dapat dipulihkan setelah terjadi kegagalan atau serangan keamanan. [REQ-NF-15]

5.3. Software Quality Attributes

Atribut aplikasi yang dijelaskan pada bagian ini antara lain kemudahan (*usability*), *scalability*, kesesuaian (*compatibility*), *maintainability*, dan *reliability* dari aplikasi yang akan dibangun.

5.3.1. Usability

- User dapat melakukan pendaftaran akun dengan mudah. [REQ-NF-01]
- User dapat melakukan login pada aplikasi dengan mudah. [REQ-NF-02]
- Simbol dan logo yang digunakan dalam aplikasi harus mudah dimengerti oleh user (masyarakat POLBAN). [REQ-NF-03]
- User bisa mengetahui informasi kegiatan setiap lembaga dengan mudah. [REQ-NF-04]

5.5.2. Scalability

- Aplikasi dapat berjalan normal dengan maksimal pendaftar sebanyak 800 pendaftar dalam 20 menit. [REQ-NF-08]
- Aplikasi dapat berjalan normal dengan Peningkatan jumlah mahasiswa yang ingin mendaftarkan diri di suatu kegiatan pada lembaga yang ada secara membludak sebanyak 400 mahasiswa dalam rentang waktu 30 menit. [REQ-NF-09]
- Sistem harus bisa menangani pengunjung pada aplikasi dengan jumlah 1000 pengunjung dalam rentang waktu 5 menit. [REQ-NF-10]
- Aplikasi dapat menangani kegiatan peminjaman inventari barang masjid dengan jumlah 50 peminjaman dalam 5 menit. [REQ-NF-11]

5.5.3. Compatibility

- Aplikasi harus kompatibel di berbagai perangkat seluler seperti Android dan Windows. [REQ-NF-16]
- Aplikasi harus tersedia dengan resolusi layar HD, Full HD, dan 4K. [REQ-NF-17]
- Ukuran layar *small*(kurang dari 640px), *medium*(641px hingga 1007px), dan *high*(lebih besar 1008px). [REQ-NF-18]
- Aplikasi bekerja dengan baik di kecepatan koneksi internet 4G dan 5G. [REQ-NF-19]

5.5.4. Maintainability

- Aplikasi harus memiliki sistem monitoring performa agar admin/pengurus aplikasi mengetahui bila ada masalah pada aplikasi lebih cepat. [REQ-NF-20]
- Struktur kode harus teratur, bersih, dan mudah untuk dibaca oleh programmer. [REQ-NF-21]
- Aplikasi harus dilakukan perawatan rutin setiap harinya dengan jangka waktu pengerjaan 3 - 4 jam sehari tanpa mengganggu kegiatan aplikasi yang sedang berjalan. [REQ-NF-22]
- Aplikasi dapat melakukan pembaharuan setiap 3 - 6 bulan sekali. [REQ-NF-23]

5.5.5. Reliability

- Sistem harus memiliki waktu tanggap rata-rata kurang dari 200 milidetik pada beban penuh. [REQ-NF-06]

- Aplikasi harus responsif dan memberikan waktu tanggapan yang cepat kepada pengguna. Kinerja aplikasi harus terjamin meskipun terdapat beban tinggi atau saat melakukan operasi yang kompleks. **[REQ-NF-25]**

Appendix A: Glossary

Istilah, singkatan, dan akronim yang digunakan dalam dokumen ini dapat dilihat pada tabel berikut:

Istilah/Singkatan/Akronim	Definisi
Developer	Seseorang yang terlibat dalam pembuatan, pengembangan, dan pemeliharaan aplikasi komputer, situs web, atau perangkat lunak. Developer dapat bekerja sebagai bagian dari tim pengembangan perangkat lunak atau sebagai <i>freelancer</i> .
User	Seseorang yang menggunakan atau memanfaatkan suatu produk atau layanan. Dalam konteks teknologi informasi, user adalah orang yang menggunakan perangkat lunak atau aplikasi komputer, situs web, atau perangkat teknologi lainnya.
Environment	Environment merujuk pada berbagai faktor lingkungan yang mempengaruhi pengembangan perangkat lunak, seperti hardware, software, dan infrastruktur teknologi lainnya. Environment juga mencakup faktor non-teknis seperti kebijakan organisasi, budaya perusahaan, dan lingkungan kerja.
Domain Model	Representasi abstrak dari suatu domain atau bisnis yang diwakili oleh perangkat lunak yang sedang dikembangkan. Domain Model mewakili konsep-konsep, entitas, dan hubungan antara entitas dalam domain bisnis yang ingin diotomatisasi dengan perangkat lunak.
Use Case Model	Representasi dari fitur dan fungsi yang diinginkan dalam perangkat lunak yang sedang dikembangkan, yang digambarkan dalam bentuk skenario atau urutan interaksi antara pengguna dan sistem.
Stakeholder	Individu atau kelompok yang memiliki kepentingan atau terpengaruh oleh suatu proyek atau keputusan yang diambil.
Interface	Titik kontak antara dua atau lebih sistem atau komponen yang memungkinkan pertukaran informasi atau data.
User Interface (UI)	Bagian dari aplikasi atau sistem yang digunakan oleh pengguna untuk berinteraksi dengan perangkat lunak. UI bisa berupa antarmuka grafis, antarmuka teks, atau kombinasi keduanya.
Software Interface	Cara untuk berkomunikasi antara dua perangkat lunak atau program melalui protokol atau format data yang telah ditentukan.

Hardware Interface	Cara untuk berkomunikasi dengan perangkat keras melalui protokol atau format data yang telah ditentukan.
Communication Interface	Sebuah sistem yang mengatur interaksi dan pertukaran informasi antara dua atau lebih perangkat atau sistem. Antarmuka komunikasi dapat terdiri dari perangkat keras, perangkat lunak, atau keduanya.
Non Functional Requirement	Kebutuhan yang tidak berkaitan dengan fitur atau fungsionalitas suatu sistem, melainkan lebih berkaitan dengan aspek-aspek lain seperti kinerja, keamanan, keandalan, dan sebagainya.
Tester	Seseorang yang bertanggung jawab untuk mengevaluasi kualitas dan kinerja suatu sistem atau aplikasi.
Marketing	Aktivitas yang dilakukan untuk mempromosikan, menjual, dan mendistribusikan produk atau jasa suatu perusahaan.
Desktop	Sebuah komputer pribadi.
Android	Sistem operasi <i>open-source</i> yang dirancang khusus untuk perangkat <i>mobile</i> .
Responsive	Suatu desain atau teknologi yang memungkinkan sebuah website atau aplikasi untuk menyesuaikan tampilan dan fungsinya dengan ukuran dan jenis perangkat yang digunakan oleh pengguna.
Bug	Kesalahan atau kegagalan pada sebuah program komputer atau sistem yang menyebabkan tidak berfungsinya sistem tersebut sesuai dengan yang diharapkan.
Error	Pesan kesalahan atau indikasi bahwa terjadi kegagalan dalam sebuah program atau sistem.
Tools	Alat atau perangkat yang digunakan untuk membantu menyelesaikan tugas atau pekerjaan tertentu.
Database	Kumpulan data yang terorganisir secara sistematis dalam suatu sistem komputer.
Frontend	Bagian yang terlihat oleh pengguna atau <i>user interface</i> . Umumnya terdiri dari elemen-elemen visual seperti <i>layout</i> , warna, <i>font</i> , gambar, dan animasi yang membentuk tampilan antarmuka pengguna.
Backend	Bagian yang tidak terlihat oleh pengguna dan terletak di belakang layar. Biasanya berfungsi sebagai penghubung antara <i>frontend</i> dengan <i>database</i> dan menyediakan data atau informasi yang diperlukan oleh <i>frontend</i> .
Programmer	Seseorang yang memiliki keterampilan dan kemampuan dalam membuat, mengembangkan, dan memelihara perangkat lunak atau program komputer.

Core Process	Serangkaian tindakan atau kegiatan utama yang dilakukan oleh suatu organisasi atau perusahaan untuk mencapai tujuan bisnis atau misi perusahaan.
Management Process	Serangkaian tindakan atau kegiatan yang dilakukan oleh manajemen suatu organisasi untuk mengelola sumber daya dan operasi bisnis agar dapat mencapai tujuan organisasi secara efektif dan efisien
Support Process	Serangkaian tindakan atau kegiatan yang dilakukan untuk mendukung operasi inti atau core process suatu organisasi.
HTML	HyperText Markup Language adalah bahasa pemrograman yang digunakan untuk membuat dan mengatur tampilan halaman web.
CSS	Cascading Style Sheets adalah bahasa stylesheet yang digunakan untuk mengatur tampilan halaman web.
Javascript	Bahasa pemrograman tingkat tinggi yang digunakan terutama untuk membuat halaman web yang interaktif dan dinamis.
React Js	Sebuah kerangka kerja (<i>framework</i>) JavaScript yang digunakan untuk membuat antarmuka pengguna (<i>user interface/UI</i>) pada aplikasi web.
Node Js	Sebuah runtime environment yang digunakan untuk menjalankan aplikasi JavaScript di sisi <i>server</i> atau <i>backend</i> .

Appendix B: User Persona

1. User Persona DKM



Ketua DKM'20 Lukmannul Hakim

About

Reza is the head of DKM batch 2020. He was elected and trusted to lead various activities at the institution so that it can get even better.

Skills

- Public Speaking
- Social Network
- Decision Making
- Read Situation
- Leading

Goals & Needs

- Can arrange all activities in a systematic and planned manner
- Can provide facilities for students who want to register for dkm institutions
- Can provide facilities for students or lecturers who want to use the mosque hall

Frustrations


- Lack of requests for applicants who want to enter the DKM institution
- Details of the activities are still unclear
- There are activities that fail due to lack of initiative

REZA RAHADIAN

🏠 20 📍 Bandung

🎒 College Student 💕 Single

2. User Persona Kammi



Riri Putri - Treasurer in the Community Development Department

Demography

Age: 20	Education: High School
Gender: Female	Profession: Student
Status: Single	Location: Bandung

Bio

Riri is Treasurer In The Department Of Community Development At The Kammi Institute Which Is Active In The Social Sector (Helping The Community), The Political Field (Studying Hot Issues), And Also Da'wah.

Goals

- Provide Benefits To The Wider Community


Quote

"Earn Your Leadership Everyday".

Frustration

- The Schedule Of Activities Is Too Short
- Can Get Donations From Various Parties To Carry Out An Activity

3. User Persona Assalam



"Eat well, Live well"

More Information

Age: 21 Education: Unknown
Status: Single Location: Bandung

About

Ozoran San is the Head of the Syiar Department at the Assalam Institution of the Luqmanul Hakim Mosque. He is active and leads many activities in the Assalam institution

Goals

- Organize activities that can increase the spiritual capacity of members
- Spreading Islamic da'wah in the Polban environment in order to achieve its vision, Polban Madani.

Painpoints

- Lack of quantity of human resources as teachers/mentors/murabbi
- Some activities did not reach the target
- Need a strategy to hook the Polban's Student into participating in study activities

Ozora San
Head of Broadcasting Department

Appendix C: Test Case

Berikut *link* dari *test case* pengujian fitur Pengelolaan Inventaris,

<https://docs.google.com/spreadsheets/d/1wqC1U7T0HHiq598ctPqs5RIvzruYNT7F/edit#gid=276861608>