



## Aplikasi Mobile untuk Otomatisasi Penerbitan Surat Mahasiswa

Salim Athari <sup>1\*</sup>, Daru Prasetyawan <sup>2</sup>, Rahmadhan Gatra <sup>1</sup>

<sup>1</sup> Pusat Teknologi Informasi dan Pangkalan Data, UIN Sunan Kalijaga, Indonesia

<sup>2</sup> Program Studi Informatika, UIN Sunan Kalijaga, Indonesia

\* Korespondensi: [salim.athari@uin-suka.ac.id](mailto:salim.athari@uin-suka.ac.id)

**Sitasi:** Athari, S.; Prasetyawan, D.; Gatra, R. (2025). Aplikasi Mobile untuk Otomatisasi Penerbitan Surat Mahasiswa. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 7(2), 252-263.

<https://doi.org/10.35746/jtim.v7i2.630>

Diterima: 11-11-2024

Direvisi: 03-01-2025

Disetujui: 21-01-2025



**Copyright:** © 2025 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

**Abstract:** Effective management of correspondence is a crucial requirement in various organizations to enhance operational efficiency, particularly in supporting lectures in higher education. The issuance of letters for student needs must adhere to specific requirements that must be fulfilled before the letter is issued. This process can sometimes be lengthy as the examination is conducted by officers. The application of information technology enables the acceleration of letter issuance by automating the verification of the necessary requirements. The use of mobile technology also facilitates flexible access for users at any time and from any location. Mobile applications are designed to address user needs in terms of functionality, ease of use, and optimal performance. This study aims to develop a mobile-based automatic letter system that assists students in obtaining letters required during lectures. The application operates automatically, meaning that the fulfillment of the requirements necessary for issuing letters can occur without the intervention of officers to verify them. The development of this system follows the Software Development Life Cycle (SDLC), which encompasses needs analysis, design, implementation, and system testing. Application development incorporates data obtained from other existing systems such as academic information systems, personnel information systems, and electronic official manuscript systems. This application utilizes the Application Programming Interface (API) to communicate with other applications. The development of this system employs the Dart programming language and the Flutter framework. From the results of User Acceptance Testing (UAT), a score of 93.19% was achieved, indicating that the application was well received by users.

**Keywords:** Mobile apps, Automation, Mail, Students, Flutter

**Abstrak:** Pengelolaan surat-menyurat yang efektif menjadi kebutuhan utama dalam berbagai organisasi untuk meningkatkan efisiensi operasional, termasuk di dalamnya dalam mendukung perkuliahan di perguruan tinggi. Penerbitan surat untuk kebutuhan mahasiswa harus melalui persyaratan-persyaratan yang harus dipenuhi sebelum surat tersebut diterbitkan. Proses tersebut terkadang membutuhkan waktu yang lama karena pemeriksaannya dilakukan oleh petugas. Penerapan teknologi informasi memungkinkan untuk mempercepat penerbitan surat dengan mengotomatisasi pemeriksaan persyaratan yang diperlukan. Penggunaan teknologi *mobile* juga memungkinkan akses yang fleksibel bagi pengguna kapan saja dan di mana saja. Aplikasi mobile dirancang untuk memenuhi kebutuhan pengguna dalam hal fungsionalitas, kemudahan penggunaan, dan kinerja yang optimal. Penelitian ini bertujuan untuk mengembangkan sistem surat otomatis berbasis mobile yang dapat membantu mahasiswa dalam menerbitkan surat yang dibutuhkan selama perkuliahan. Aplikasi dapat bekerja secara otomatis, artinya pemenuhan persyaratan yang dibutuhkan dalam penerbitan surat dapat dilakukan tanpa campur tangan petugas untuk memeriksanya. Pengembangan sistem ini mengacu pada *Software Development Life*

*Cycle (SDLC)*, yang meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian sistem. Pengembangan aplikasi melibatkan data yang diperoleh dari sistem lain yang telah ada sebelumnya seperti sistem informasi akademik, sistem informasi kepegawaian, dan sistem naskah dinas elektronik. Aplikasi ini menggunakan *Application Programming Interface (API)* untuk berkomunikasi dengan aplikasi lain. Pengembangan sistem ini menggunakan bahasa pemrograman *Dart* dan *framework Flutter*. Dari hasil *User Acceptance Testing (UAT)* diperoleh nilai sebesar 93,19%, artinya aplikasi dapat diterima dengan baik oleh pengguna.

**Kata kunci:** Aplikasi mobile, Otomatisasi, Surat, Mahasiswa, Flutter

## 1. Pendahuluan

Surat merupakan salah satu alat komunikasi antara dua pihak, baik dalam bentuk kertas maupun elektronik. Surat merupakan sarana komunikasi yang berisi tulisan, pernyataan, kabar, harapan, perintah, laporan atau tanggapan dari pihak satu ke pihak lain [1]. Di bidang pendidikan, khususnya Perguruan Tinggi, surat dijadikan sebuah dokumen administrasi yang sangat penting. Berbagai jenis surat akan selalu dibutuhkan untuk mendukung kegiatan perkuliahan. Terdapat berbagai jenis surat yang diperlukan oleh mahasiswa untuk menunjang kegiatan perkuliahan seperti surat keterangan masih kuliah, surat keterangan berkelakuan baik, surat keterangan habis teori, surat keterangan lulus, surat izin penelitian, surat izin observasi, surat permohonan rekomendasi penelitian, surat izin penelitian, dan lainnya. Misalnya, surat keterangan biasanya digunakan oleh mahasiswa sebagai syarat untuk mengikuti kegiatan tertentu seperti perlombaan, pendaftaran beasiswa, atau kegiatan lainnya.

Penerbitan surat untuk kebutuhan mahasiswa harus memenuhi persyaratan-persyaratan yang ditetapkan untuk setiap jenis surat. Petugas Tata Usaha harus melakukan proses verifikasi serta memeriksa persyaratan-persyaratan yang harus dipenuhi sebelum menerbitkan surat yang diminta oleh mahasiswa. Sebagai contoh, untuk menerbitkan surat keterangan masih kuliah petugas harus memastikan bahwa mahasiswa tersebut benar-benar masih dalam status aktif/tidak sedang cuti atau belum lulus. Contoh lainnya adalah surat izin penelitian tugas akhir, petugas harus memastikan bahwa mahasiswa tersebut sedang mengambil matakuliah tugas akhir. Disamping itu, mahasiswa juga harus menyiapkan persyaratan-persyaratan yang diperlukan. Hal ini mengakibatkan proses penerbitan surat memakan waktu yang cukup lama.

Penggunaan piranti *mobile* tidak dapat lagi dipisahkan dalam kehidupan masyarakat. Selain itu, Teknologi Informasi khususnya teknologi *mobile* telah berkembang dengan pesat. Hal ini ditandai dengan banyaknya aplikasi *mobile* yang banyak digunakan untuk mendukung kebutuhan-sehari, seperti aplikasi *marketplace*, aplikasi transportasi, dan sebagainya. Dalam mendukung kelancaran administrasi akademik, perkembangan teknologi *mobile* juga dapat dimanfaatkan untuk mempermudah pekerjaan tersebut. Aplikasi *mobile* merupakan program perangkat lunak yang berjalan pada piranti *mobile* [2]. Perangkat *mobile* biasanya sudah tertanam aplikasi yang melekat pada perangkat tersebut ketika proses manufaktur atau biasanya disebut sebagai aplikasi bawaan. Selain itu, aplikasi *mobile* selain bawaan juga dapat dipasang (*install*) pada *smartphone*. Aplikasi ini dapat di-*download* dari toko aplikasi seperti *Play store* atau *iTunes*, atau dari distribusi perangkat lunak *platform* lainnya. Aplikasi *mobile* mempunyai banyak kelebihan jika dibandingkan dengan aplikasi web dan desktop. Aplikasi *mobile* memiliki antar muka pengguna dan *user experience* yang lebih mudah digunakan digunakan. Selain itu, beberapa aplikasi *mobile* dapat digunakan tanpa harus terhubung dengan *internet* [3].

Pengembangan aplikasi *mobile* berkembang pesat dengan kepentingan ekonomi dan ilmiah yang substansial [4]. Pengembangan aplikasi *mobile* adalah serangkaian proses dan prosedur untuk membuat perangkat lunak yang dapat berjalan pada perangkat nirkabel, seperti *smartphone* dan perangkat genggam lainnya. Proses pengembangan aplikasi

melibatkan pembuatan perangkat lunak yang dapat di-*install* dan mengimplementasikan layanan *backend* menggunakan *Application Programming Interface* (API). Aplikasi surat telah banyak digunakan dalam sistem administrasi modern [5]. Pengembangan aplikasi *mobile* sebelumnya telah digunakan untuk administrasi/pencatatan surat keluar dan surat masuk sehingga mempermudah proses administrasi data surat masuk dan surat keluar [6]. Pengembangan aplikasi *mobile* untuk manajemen surat masuk dan surat keluar juga telah dilakukan di Perum Bulog Subdivre Dumai dengan tujuan untuk mempermudah penginformasian surat masuk kepada pimpinan, pengalokasian tindak lanjut dan disposisi terhadap surat masuk, dan untuk mempercepat pencarian data [7].

Dalam penelitian ini, peneliti akan mengembangkan sebuah aplikasi *mobile* untuk otomatisasi surat mahasiswa. Persyaratan-persyaratan penerbitan surat yang biasanya diperiksa secara manual, oleh sistem akan dilakukan secara otomatis *web service*. Sebagai contoh dalam penerbitan surat keterangan aktif, sistem akan memastikan bahwa mahasiswa tersebut benar-benar aktif dengan meminta informasi dari Sistem Informasi Akademik melalui *web service*. Dalam penerbitan surat tidak sedang menerima beasiswa, sistem akan mengambil informasi dari Sistem Beasiswa. Selain itu untuk memastikan bahwa pejabat yang menerbitkan surat tersebut benar, sistem akan berkomunikasi dengan Sistem Tata Naskah Dinas Elektronik dan Sistem Informasi Kepegawaian.

## 2. Metodologi Penelitian

Seperti pengembangan aplikasi web, pengembangan aplikasi *mobile* juga mengacu pada pengembangan perangkat lunak tradisional. Tetapi pengembangan aplikasi *mobile* juga mempertimbangkan fitur-fitur khusus pada perangkat *mobile*. Aplikasi *mobile* memerlukan berbagai informasi untuk memenuhi kebutuhan pengguna [8]. Metode penelitian yang digunakan dalam penelitian ini mengacu pada *Software Development Life Cycle* (SDLC). SDLC merupakan sebuah proses yang meliputi perancangan, pengembangan, dan pengujian perangkat lunak [9]. Setiap jenis SDLC digunakan dirancang untuk kebutuhan khusus guna mengakomodasi sumber daya yang tersedia [10]. Pengembangan perangkat lunak mencakup kegiatan identifikasi/analisis persyaratan, perancangan, pengkodean, pengujian, serta produksi dan perawatan aplikasi [11]. Alur pengembangan aplikasi *mobile* untuk Otomatisasi Penerbitan Surat Mahasiswa dapat dilihat pada Gambar 1.



**Gambar 1.** Alur pengembangan aplikasi

Penelitian ini dimulai dari studi literatur dan pengumpulan persyaratan yang diperlukan dalam pengembangan aplikasi. Pada tahap ini, semua informasi dikumpulkan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional. Dalam pengumpulan informasi, metode yang digunakan adalah observasi dan dokumentasi. Dalam pengembangan aplikasi *mobile* untuk otomatisasi surat, observasi dilakukan untuk mengamati proses kerja dalam penerbitan surat mahasiswa. Sedangkan teknik dokumentasi digunakan untuk menganalisis dokumen-dokumen yang tersedia terkait dengan proses penerbitan surat mahasiswa.

Pada tahap ini, semua informasi terkait penerbitan surat mahasiswa yang telah dikumpulkan sebelumnya dianalisis untuk mendefinisikan solusi teknis dan spesifikasi yang dibutuhkan. Tujuan utama dari tahap ini adalah untuk memastikan semua orang mengetahui setiap elemen persyaratan yang diperlukan [12]. Tahap analisis bertujuan untuk mengumpulkan kebutuhan secara lengkap [13].

Setelah spesifikasi sistem telah didapatkan, selanjutnya spesifikasi tersebut digambarkan dalam sebuah rancangan aplikasi. Perancangan sistem otomatisasi surat mahasiswa meliputi perancangan arsitektur dan perancangan alur sistem. Perancangan arsitektur yang menggambarkan susunan keseluruhan komponen-komponen sistem yang saling berkomunikasi. Perancangan proses menggambarkan proses yang berjalan di dalam sistem.

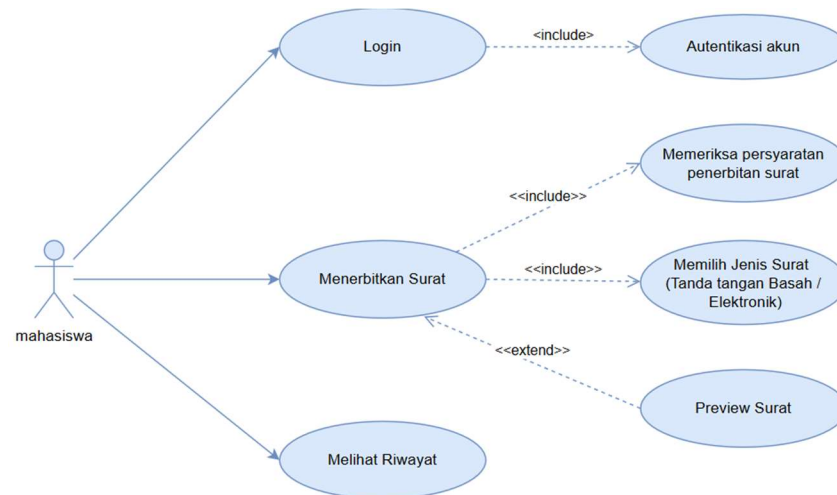
Pada tahap implementasi, rancangan-rancangan yang telah dibuat sebelumnya diterjemahkan menjadi kode-kode program secara keseluruhan menggunakan bahasa pemrograman tertentu. Untuk membangun aplikasi, penelitian ini menggunakan bahasa pemrograman *Dart* dan *framework Flutter*. *Dart* merupakan bahasa pemrograman berorientasi objek, berbasis kelas yang menggunakan sintaks *C-style* dan dapat dikompilasi menjadi kode asli atau JavaScript [14]. *Flutter* merupakan *framework* opensource untuk mengembangkan aplikasi multi-platform yang dapat dikompilasi secara native dari satu basis kode. *Flutter* adalah *framework* lintas platform yang menargetkan pengembangan aplikasi seluler berperforma tinggi [15]. Dengan *Flutter*, memungkinkan pengembangan membangun aplikasi untuk sistem operasi Android dan IOS dalam satu waktu [16].

Selanjutnya aplikasi dievaluasi menggunakan teknik black box testing untuk memastikan kesesuaian fungsional aplikasi yang dikembangkan dengan spesifikasi [2]. Sedangkan teknik User Acceptance Testing (UAT) digunakan untuk mengukur kepuasan pengguna dalam menggunakan aplikasi.

### 3. Hasil dan Pembahasan

#### 3.1. Analisis Persyaratan

Sistem Otomatisasi Surat Mahasiswa adalah sistem yang memungkinkan penerbitan surat dilakukan secara otomatis, artinya permohonan penerbitan surat secara otomatis disetujui apabila syarat-syarat yang diperlukan terpenuhi. Gambar 1 menunjukkan *use case* Sistem Otomatisasi Surat Mahasiswa. Mahasiswa berperan sebagai aktor yang akan menggunakan sistem tersebut. Untuk menerbitkan surat di dalam sistem ini, mahasiswa diharuskan login terlebih dahulu. Proses login diperlukan untuk memperoleh informasi mahasiswa tersebut, seperti nama, NIM, Program Studi, dan Fakultas. Proses ini akan berhubungan dengan *API Lightweight Directory Access Protocol (LDAP)* dan *API Akademik*. Mahasiswa login ke dalam Sistem Otomatisasi Surat dengan mengisikan NIM dan password. Sistem Otomatisasi Surat akan meminta otentikasi kepada *API LDAP*. Jika *request* yang diminta tidak ada di dalam *API LDAP*, sistem akan menolak proses login. Jika permintaan otentikasi akun terpenuhi, selanjutnya sistem akan meminta informasi data mahasiswa dari *API Akademik*. Informasi inilah yang akan digunakan dalam penerbitan surat. *Use Case* Sistem Otomatisasi Surat Mahasiswa dapat dilihat pada Gambar 2.



**Gambar 2.** Use Case Sistem Otomatisasi Surat Mahasiswa

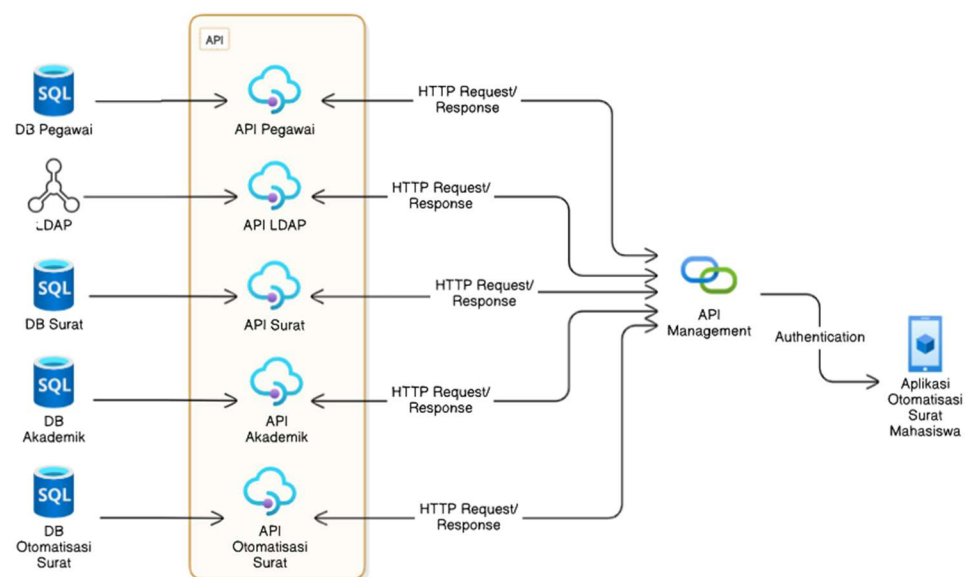
Setelah proses login selesai, mahasiswa diminta untuk memilih jenis surat yang akan diterbitkan. Hal ini dimaksudkan untuk membedakan jenis surat yang akan diterbitkan karena setiap jenis surat memiliki karakteristik dan persyaratan masing-masing. Persyaratan untuk masing-masing jenis surat disajikan pada Tabel 1.

**Tabel 1.** Syarat penerbitan surat mahasiswa

Jenis Surat	Persyaratan
Surat Keterangan Masih Kuliah	Status mahasiswa harus aktif Kelengkapan Data Pribadi Mahasiswa (Nomor Handphone dan Alamat Tinggal)
Surat Keterangan Berkelakuan Baik	Status mahasiswa harus aktif Tidak memiliki catatan kurang terpuji
Surat Keterangan Habis Teori	Status mahasiswa harus aktif Bebas Teori
Surat Keterangan Tidak Sedang Menerima Beasiswa	Status mahasiswa harus aktif Tidak sedang menerima beasiswa Tidak sedang menerima pembayaran beasiswa
Surat Keterangan Pindah Studi	Status mahasiswa harus aktif
Surat Keterangan Lulus	Status mahasiswa harus sudah lulus Tanggal yudisium tidak lebih dari 3 bulan
Surat Permohonan Izin Observasi	Status mahasiswa harus aktif
Surat Permohonan Izin Kuliah Lapangan (PPL/PLP/PKL/KP/PPS)	Status mahasiswa harus aktif Sedang mengambil matakuliah lapangan (PPL/PLP/PKL/KP/PPS)
Surat Permohonan Izin Penelitian Matakuliah	Status mahasiswa harus aktif
Surat Permohonan Studi Pendahuluan	Status mahasiswa harus aktif Sedang mengambil matakuliah Tugas Akhir Kelengkapan Data Pribadi Mahasiswa (Nomor Handphone dan Alamat Tinggal)
Surat Permohonan Izin Penelitian Tugas Akhir	Status mahasiswa harus aktif Sedang mengambil matakuliah Tugas Akhir
Surat Permohonan Rekomendasi Penelitian	Status mahasiswa harus aktif

### 3.2. Perancangan Arsitektur

Aplikasi otomatisasi surat mahasiswa berbasis *mobile* melibatkan beberapa API diantaranya API LDAP, API Akademik, API Pegawai, API Surat, dan API Otomatisasi Surat. Aplikasi otomatisasi surat mahasiswa meminta data kepada API *gateway* melalui *HTTP Request*. Selanjutnya API *gateway* meneruskan permintaan ke *service* terkait. API LDAP digunakan untuk otentikasi login dengan menerima parameter *username* dan *password*, kemudian akan merespon dengan mengirimkan data pengguna yang tersimpan di dalam LDAP apabila *username* dan *password* sesuai, atau mengirimkan pesan otentikasi gagal jika *username* dan *password* tidak sesuai. API Akademik menyediakan informasi tentang mahasiswa dan perkuliahan, termasuk didalamnya status mahasiswa seperti apakah mahasiswa tersebut sedang aktif atau tidak, apakah mahasiswa tersebut sudah mengambil matakuliah yang disyaratkan dalam penerbitan surat, dan lain sebagainya. Perancangan Arsitektur sistem otomatisasi surat mahasiswa disajikan pada Gambar 3.



**Gambar 3.** Arsitektur sistem otomatisasi surat mahasiswa

API Pegawai digunakan untuk menyediakan data yang terkait dengan kepegawaian. Data terkait dengan kepegawaian yang diperlukan dalam penerbitan surat mahasiswa adalah pejabat penanda tangan surat dan unit yang mengeluarkan surat. Surat yang diterbitkan harus memiliki nomor surat yang terdaftar di dalam Tata Naskah Dinas Elektronik (TNDE). Layanan yang menyediakan nomor surat tersebut adalah API Surat. Aplikasi otomatisasi surat mahasiswa mengirimkan permintaan kepada API Surat untuk mengirimkan nomor surat sesuai dengan jenis surat dan unit yang menerbitkan surat tersebut. Kemudian surat yang sudah bernomor dan sudah memiliki tanda tangan dikirim kembali kepada API Surat dan disimpan kedalam *database* surat. API Otomatisasi Surat Mahasiswa merupakan layanan yang khusus melayani aplikasi otomatisasi surat mahasiswa, seperti untuk menghubungkan dengan *database* otomatisasi surat mahasiswa.

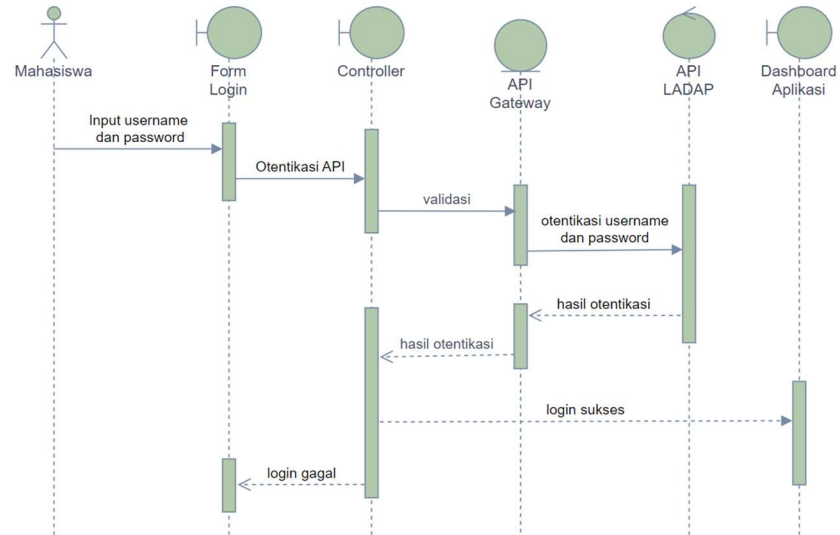
### 3.3. Perancangan Proses

Perancangan proses aplikasi otomatisasi surat mahasiswa digambarkan melalui *sequence diagram*.

#### 3.3.1. Sequence Diagram Proses Login

Proses login dimulai dengan mengisikan *username* (NIM) dan *password* pada halaman login. Aplikasi akan meminta otentikasi dengan mengirimkan *username* dan *password* ke API *gateway*. Selanjutnya API *gateway* akan meneruskan permintaan otentikasi *username* dan *password* ke API LDAP. API LDAP akan mengirimkan respon hasil otentikasi ke API

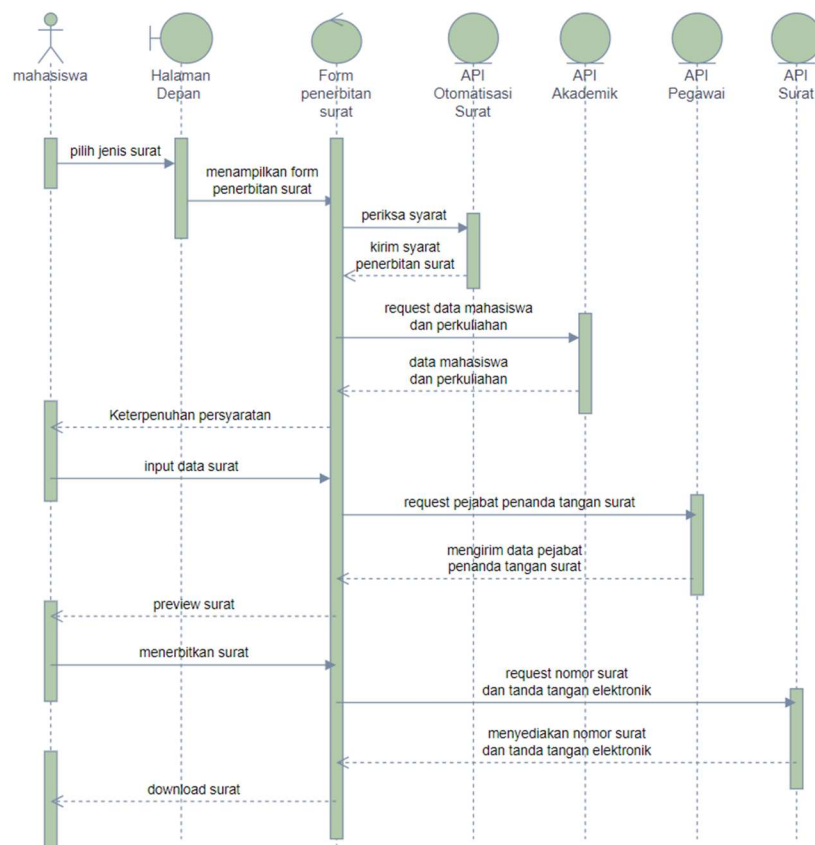
gateway kemudian diteruskan ke aplikasi. Apabila otentikasi sesuai aplikasi akan menampilkan halaman utama aplikasi yang menampilkan daftar jenis surat. Sedangkan apabila otentikasi tidak sesuai maka aplikasi akan dikembalikan lagi ke halaman login. *Sequence diagram* proses login dapat dilihat pada Gambar 4.



Gambar 4. *Sequence diagram* proses login

### 3.3.2. Sequence Diagram Penerbitan surat

Sequence diagram proses penerbitan surat dapat dilihat pada Gambar 5.



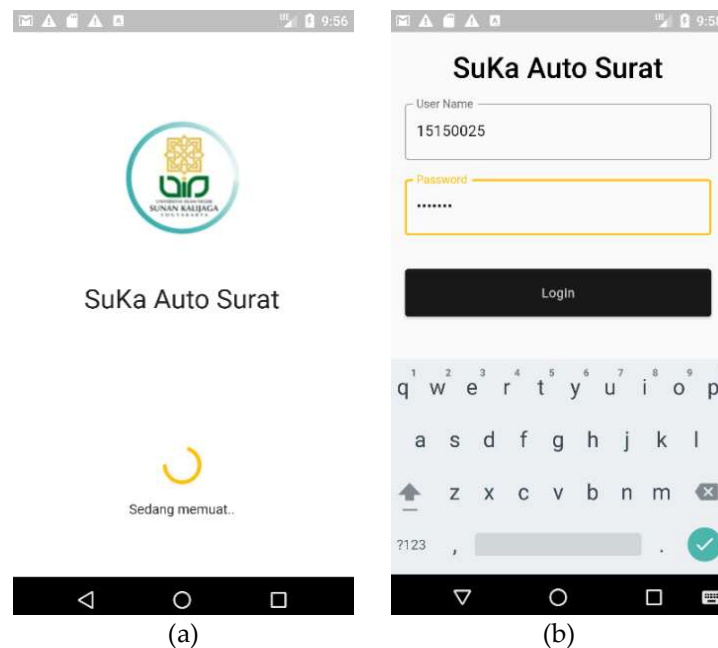
Gambar 5. *Sequence diagram* proses penerbitan surat

Proses penerbitan surat dimulai dengan memilih jenis surat yang akan diterbitkan. Aplikasi akan menyajikan form yang berisi syarat untuk jenis surat yang dipilih. Syarat penerbitan surat ini diperoleh dari API Otomatisasi Surat, sehingga aplikasi otomatisasi surat meminta data syarat dari API Otomatisasi Surat dengan mengirimkan parameter jenis surat. Selanjutnya API Otomatisasi Surat mengirimkan syarat penerbitan surat berdasarkan jenis surat yang dimaksud. Kemudian aplikasi meminta data mahasiswa dan perkuliahan kepada API Akademik sesuai kondisi syarat yang diperlukan. API Akademik menyediakan data mahasiswa dan perkuliahan sesuai dengan yang diminta oleh aplikasi. Setelah itu, aplikasi akan membandingkan antara syarat yang diperlukan untuk penerbitan jenis surat tersebut dengan kondisi mahasiswa dan perkuliahan. Apabila syarat yang diperlukan terpenuhi, maka mahasiswa diminta untuk mengisi data surat, seperti tujuan penerima surat, dan keperluan surat. Setelah data tersebut diisikan, aplikasi akan meminta data pejabat penanda tangan surat sesuai dengan fakultas dari mahasiswa tersebut. Mahasiswa diminta untuk memeriksa surat yang akan diterbitkan. Jika surat tersebut sudah sesuai, mahasiswa dapat menerbitkan surat tersebut. Dalam proses penerbitan surat tersebut, aplikasi meminta nomor surat dan tanda tangan elektronik kepada API Surat sesuai dengan unit penerbit surat dan pejabat penanda tangan surat.

### 3.4. Implementasi/Pengkodean

#### 3.4.1. Halaman Depan dan Login

Ketika aplikasi pertama kali dijalankan, aplikasi akan menyajikan halaman depan. Halaman login digunakan untuk memberikan akses aplikasi kepada mahasiswa. Pada halaman ini terdapat *field username* dan *password* yang harus diisi oleh mahasiswa dan mengklik tombol login untuk memperoleh akses ke dalam aplikasi. Proses login dilakukan dengan melakukan otentikasi atau validasi *username* dan *password* melalui API LDAP. *Username* dan *password* dikirimkan ke *server* LDAP melalui API LDAP dan mengembalikan status validasi pengguna. Jika *username* valid, maka halaman aplikasi akan diarahkan ke halaman *dashboard*. Halaman depan dan halaman login aplikasi *mobile* otomatisasi surat mahasiswa dapat dilihat pada Gambar 6.



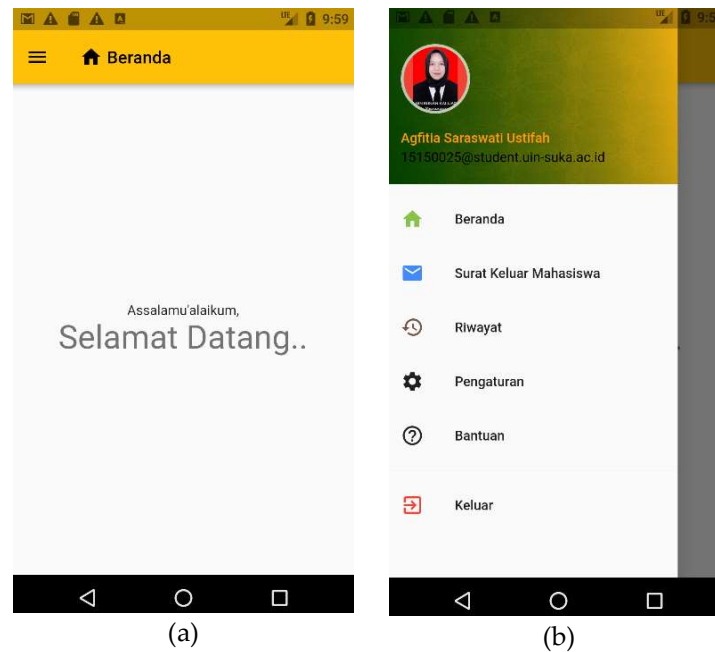
Gambar 6. (a) Tampilan halaman depan; (b) Halaman login

#### 3.4.2. Halaman Dashboard dan Menu

Halaman *dashboard* akan ditampilkan setelah berhasil login. Halaman *dashboard* hanya berisi ucapan selamat datang. Pada bagian kiri atas terdapat icon untuk



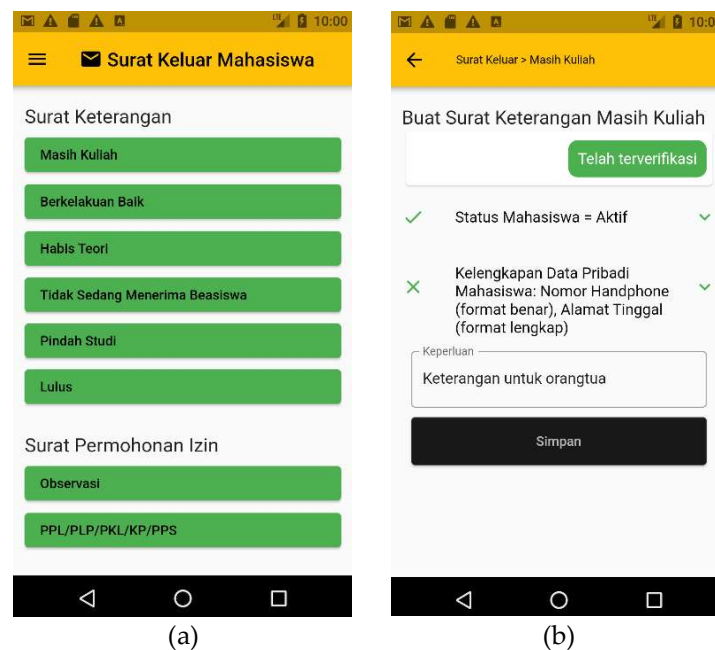
menampilkan menu yang disediakan. Halaman *dashboard* dan menu aplikasi *mobile* otomatisasi surat mahasiswa dapat dilihat pada Gambar 7.



Gambar 7. (a) Tampilan *dashboard*; (b) Daftar menu

### 3.4.3. Halaman Surat Keluar dan Pembuatan Surat

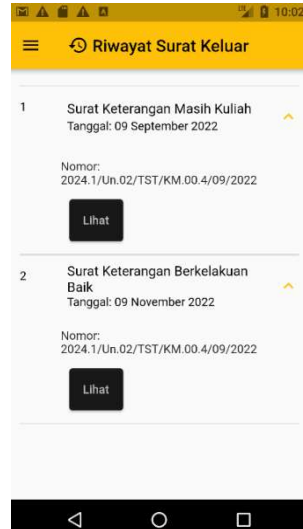
Halaman surat keluar mahasiswa berisi daftar jenis surat yang akan diterbitkan. Form pembuatan surat berisikan syarat-syarat yang diperlukan dalam penerbitan surat dan form isian yang dibutuhkan. Halaman surat keluar dan form pembuatan surat pada aplikasi *mobile* otomatisasi surat mahasiswa dapat dilihat pada Gambar 8.



Gambar 8. (a) Halaman surat keluar; (b) Form pembuatan surat

### 3.4.4. Halaman Riwayat Surat Keluar

Halaman riwayat surat keluar digunakan untuk menyajikan arsip surat yang pernah dibuat sebelumnya. Halaman riwayat surat keluar aplikasi *mobile* otomatisasi surat mahasiswa dapat dilihat pada Gambar 9.



Gambar 9. Halaman riwayat surat keluar

### 3.5. Pengujian

Pengujian fungsional pada aplikasi otomatisasi surat mahasiswa dilakukan menggunakan metode *black box testing*. Hasil pengujian black box disajikan pada Tabel 2.

Tabel 2. Pengujian *black box*

Input	Output yang diharapkan	Output	Hasil
Login dengan NIM dan <i>password</i> yang benar	Login berhasil dan menampilkan <i>dashboard</i>	Login berhasil dan menampilkan <i>dashboard</i>	Sesuai
Login dengan NIM dan <i>password</i> yang benar	Login gagal dan menampilkan pesan kesalahan	Login gagal dan menampilkan pesan kesalahan	Sesuai
Menekan icon menu	Menampilkan daftar menu	Menampilkan daftar menu	Sesuai
Memilih menu surat keluar	Manampilkan daftar jenis surat	Manampilkan daftar jenis surat	Sesuai
Memilih jenis surat	Menampilkan syarat yang harus dipenuhi untuk setiap jenis surat beserta status keterpenuhannya	Menampilkan syarat yang harus dipenuhi untuk setiap jenis surat beserta status keterpenuhannya	Sesuai
Membuat surat pada kondisi masih ada syarat yang tidak terpenuhi	Tidak dapat membuat surat dan menampilkan pesan kesalahan	Tidak dapat membuat surat dan menampilkan pesan kesalahan	Sesuai
Membuat surat dengan kondisi semua syarat terpenuhi	Menyimpan draft surat dan menampilkan pilihan jenis tanda tangan	Menyimpan draft surat dan menampilkan pilihan jenis tanda tangan	Sesuai
Memilih jenis tanda tangan basah	Menampilkan surat dengan kolom tanda tangan kosong	Menampilkan surat dengan kolom tanda tangan kosong	Sesuai
Memilih jenis tanda tangan digital	Menampilkan surat dengan kolom tanda tangan barcode dengan nomor surat sudah terisi	Menampilkan surat dengan kolom tanda tangan barcode dengan nomor surat sudah terisi	Sesuai
Memilih menu riwayat surat keluar	Menampilkan daftar riwayat penerbitan surat	Menampilkan daftar riwayat penerbitan surat	Sesuai
Memilih salah satu surat dari daftar riwayat surat keluar	Mengunduh file surat yang telah dibuat sebelumnya	Mengunduh file surat yang telah dibuat sebelumnya	Sesuai

Hasil pengujian black box menunjukkan bahwa aplikasi otomatisasi surat memiliki tingkat keberhasilan 100%.

Untuk memastikan aplikasi dapat memenuhi ekspektasi dari pengguna, aplikasi perlu diuji menggunakan teknik *User Acceptance Testing (UAT)*. Pengujian UAT dilakukan dengan memberikan pertanyaan kepada pengguna terkait dengan penggunaan aplikasi. Selanjutnya jawaban dari pengguna dibobot sesuai dengan jawaban yang dipilih dengan ketentuan Sangat Setuju (SS) = 5, Setuju (S) = 4, Netral (N) = 3, Tidak Setuju (TS) = 2, dan Sangat Tidak Setuju (STS) = 1. Hasil jawaban kuesioner UAT disajikan pada Tabel 3.

**Tabel 3.** Hasil jawaban kuesioner UAT

	Pernyataan	SS	S	N	TS	STS	Skor
<b>Fungsionalitas</b>							
1.	Aplikasi otomatisasi surat mempermudah saya dalam menerbitkan surat	12	-	-	-	-	60
2.	Keluaran yang dihasilkan aplikasi sesuai dengan yang saya harapkan	10	2	-	-	-	58
3.	Aplikasi memberikan informasi yang benar sesuai dengan persyaratan yang dibutuhkan dalam menerbitkan surat	11	1	-	-	-	59
<b>Usability</b>							
4.	Aplikasi otomatisasi surat memiliki antarmuka yang mudah dipahami	7	4	1	-	-	54
5.	Aplikasi memiliki navigasi aplikasi yang jelas	7	3	2	-	-	53
6.	Aplikasi memiliki langkah-langkah yang sederhana dalam menerbitkan surat	8	1	3	-	-	53
<b>Performance</b>							
7.	Aplikasi memberikan respon yang cepat saat digunakan	8	2	2	-	-	54
8.	Selama menggunakan aplikasi, saya tidak mengalami keterlambatan dalam melakukan operasi tertentu	9	2	1	-	-	56
<b>Reliability</b>							
9.	Saya dapat menerbitkan surat dari awal hingga akhir tanpa kesalahan	10	1	1	-	-	57
10.	Aplikasi bekerja secara stabil saat digunakan	8	2	2	-	-	54
<b>Kepuasan Keseluruhan</b>							
11.	Secara keseluruhan, saya puas menggunakan aplikasi ini untuk kebutuhan penerbitan surat	10	2	-	-	-	58
12.	Saya akan merekomendasikan aplikasi ini kepada mahasiswa lain untuk menerbitkan surat mahasiswa	9	1	2	-	-	55

Dari Tabel 3 dapat diperoleh total skor yang diperoleh dari kuesioner UAT adalah 671 dan total skor ideal adalah 720, sehingga nilai pengujian UAT aplikasi otomatisasi surat sebesar 93,19%.

#### 4. Kesimpulan

Aplikasi *mobile* otomatisasi penerbitan surat mahasiswa dikembangkan dengan mengintegrasikan beberapa data dan sistem yang sudah ada sebelumnya seperti LDAP, Sistem Informasi Akademik, Sistem Kepegawaian, dan Sistem Tata Naskah Dinas Elektronik (TNDE). LDAP digunakan untuk autentikasi pengguna. Sistem Informasi Akademik digunakan untuk mendapatkan informasi terkait dengan mahasiswa dan perkuliahan. Sistem TNDE digunakan untuk menangani permintaan terkait administrasi persuratan, seperti penomoran dan pengesahan surat. Melalui aplikasi yang dikembangkan ini, mahasiswa dapat membuat surat sendiri untuk keperluan

perkuliahan. Pemeriksaan persyaratan penerbitan surat dilakukan oleh sistem dengan membandingkan informasi dari Sistem Informasi Akademik dengan persyaratan yang ditentukan untuk setiap jenis surat. Surat yang dihasilkan dari aplikasi ini memiliki dua jenis pengesahan, yaitu dengan tanda tangan basah dan tanda tangan elektronik. Dari hasil *User Acceptance Testing (UAT)* diperoleh nilai sebesar 93,19%, artinya aplikasi dapat diterima dengan baik oleh pengguna. Aplikasi ini hanya menyediakan fitur untuk keperluan surat mahasiswa saja, sehingga aplikasi ini dapat dikembangkan lebih lanjut untuk keperluan lainnya seperti untuk kebutuhan persuratan pegawai

## Referensi

- [1] V. O. Sari, "Keterampilan Menulis Surat yang Baik dan Benar," *Edukasi Lingua Sastra*, vol. 16, no. 2, pp. 92–100, 2018, doi: <https://doi.org/10.47637/elsa.v16i2.97>.
- [2] A. Alhaddad, A. Andrews, and Z. Abdalla, "FSMApp: Testing mobile apps," in *Advances in Computers*, vol. 130, A. R. Hurson, Ed., Elsevier, 2023, pp. 1–65. doi: <https://doi.org/10.1016/bs.adcom.2022.09.001>.
- [3] R. Annisa and M. R. Baihaqi, "PENERAPAN METODE WATERFALL DALAM MERANCANG APLIKASI MOBILE GURU DAN ORANG TUA," *Jurnal Teknik Informatika Kaputama (JTik)*, vol. 5, no. 2, 2021, Accessed: Nov. 25, 2024. [Online]. Available: <https://jurnal.kaputama.ac.id/index.php/JTIK/article/view/543>
- [4] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges," *IEEE Access*, vol. 6, pp. 17711–17728, 2018, doi: <https://doi.org/10.1109/ACCESS.2018.2818724>.
- [5] Z. Zainal, A. Amna, and H. Syahputra, "SISTEM SURAT-MENYURAT BERBASIS WEB DI KANTOR DESA BALE MUSARA", vol. 1, no. 3, 2022. Accessed: Nov. 25, 2024. [Online]. Available: <https://journal.unimar-amni.ac.id/index.php/ocean/article/download/373/297>
- [6] W. Riezkiadi, N. Wahyu Rahadi, and R. Purwanto, "Pengembangan Sistem Administrasi Surat Masuk dan Keluar Berbasis Desktop dan Mobile (Studi Kasus Politeknik Negeri Cilacap)," *JTT (Jurnal Teknologi Terapan)*, vol. 2, no. 1, pp. 29–36, 2016, doi: <https://doi.org/10.31884/jtt.v2i1.26>.
- [7] R. Hidayatullah, V. Asmawati, and D. Wahyuni, "Aplikasi Manajemen Surat Masuk dan Surat Keluar Berbasis Mobile di Perum Bulog Subdivre Dumai," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 2, pp. 66–80, 2020, doi: <https://doi.org/10.14421/jiska.2020.52-02>.
- [8] M. Melinda, S. R. Ramadhan Na, Y. Nurdin, and Y. Yunidar, "Implementation of System Development Life Cycle (SDLC) on IoT-Based Lending Locker Application," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 982–987, Aug. 2023, doi: <https://doi.org/10.29207/resti.v7i4.5047>.
- [9] M. Hossain, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *International Journal For Multidisciplinary Research*, Sep. 2023, doi: <https://doi.org/10.36948/ijfmr.2023.v05i05.6223>.
- [10] O. J. Okesola, A. A. Adebiyi, A. A. Owoade, O. Adeaga, O. Adeyemi, and I. Odun-Ayo, "Software Requirement in Iterative SDLC Model," in *Advances in Intelligent Systems and Computing*, Springer, 2020, pp. 26–34. doi: [https://doi.org/10.1007/978-3-030-51965-0\\_2](https://doi.org/10.1007/978-3-030-51965-0_2).
- [11] J. de Vicente Mohino, J. Bermejo Higuera, J. R. Bermejo Higuera, and J. A. Sicilia Montalvo, "The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies," 2019. doi: <https://doi.org/10.3390/electronics8111218>.
- [12] N. Dwivedi, D. Katiyar, and G. Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 3, Mar. 2022, [Online]. Available: <https://journal.ijresm.com/index.php/ijresm/article/view/1881>
- [13] D. Prasetyawan and P. D. Rahmanto, "Pengembangan Sistem Seleksi Proposal Penelitian Berbasis Web Service Menggunakan REST API," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 6, no. 3, pp. 283–295, Sep. 2024, doi: <https://doi.org/10.35746/jtim.v6i3.585>.
- [14] M. Syafiih, "Digitizing Arudh and Qowafi Classics as Android-Based Student Learning Media Using Flutter," *International Journal of Engineering, Science and Information Technology*, vol. 2, no. 2, pp. 129–135, 2022, doi: <https://doi.org/10.52088/ijesty.v2i2.292>.
- [15] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, "Application Development Using Flutter," *International Research Journal of Modernization in Engineering Technology and Science @International Research Journal of Modernization in Engineering*, vol. 2, no. 08, pp. 2582–5208, 2020, [Online]. Available: [https://www.irjmets.com/uploadedfiles/paper/volume2/issue\\_8\\_august\\_2020/3180/1628083124.pdf](https://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf)
- [16] N. Kuzmin, K. Ignatiev, and D. Grafov, "Experience of Developing a Mobile Application Using Flutter," in *Information Science and Applications*, K. J. Kim and H.-Y. Kim, Eds., Singapore: Springer Singapore, 2020, pp. 571–575. doi: [https://doi.org/10.1007/978-981-15-1465-4\\_56](https://doi.org/10.1007/978-981-15-1465-4_56).