

Pengembangan Sistem Notifikasi Real-Time untuk Aplikasi Manajemen Persuratan Multiplatform menggunakan Firebase Cloud Messaging dan Application Programming Interface

Asep Wahyudin¹, Ani Anisyah^{*2}, Deffin Ahmaddifa³

^{1,2,3}Program Studi Ilmu Komputer/Universitas Pendidikan Indonesia; Jl. Dr. Setiabudi No.229, Isola, Kec. Sukasari, Kota Bandung, Jawa Barat 40154

e-mail: ¹away@upi.edu, ^{*2}anianisyah@upi.edu, ³deffin@upi.edu

Abstrak

Dalam era digital yang semakin maju, layanan persuratan mengalami transformasi dari sistem konvensional ke elektronik yang dapat diakses melalui aplikasi berbasis website maupun mobile. Layanan ini memegang peranan penting dalam mendukung aktivitas komunikasi formal di berbagai sektor. Namun, salah satu tantangan utama yang dihadapi adalah memastikan informasi dapat diterima secara tepat waktu. Pengembangan sistem notifikasi menjadi solusi yang dapat ditawarkan untuk mengatasi tantangan ini. Penelitian ini dilakukan untuk mengembangkan Sistem Notifikasi Otomatis dalam Konteks Manajemen Surat Menyurat dengan menggunakan metode pengembangan perangkat lunak Waterfall. Pengembangan sistem notifikasi ini dirancang dengan mengintegrasikan Application Programming Interface (API) dan Firebase Cloud Messaging (FCM) sebagai mekanisme pengiriman pesan notifikasi untuk lintas platform secara real time. Hasil penelitian menunjukkan bahwa sistem notifikasi otomatis ini dapat digunakan untuk memudahkan stakeholder dan penerima surat untuk mengetahui informasi surat masuk atau surat keluar secara real time.

Kata kunci : Manajemen Persuratan, Notifikasi, Application Programming Interface, Firebase Cloud Messaging, Waterfall

Abstract

In the rapidly modern digital world, correspondence services have evolved from traditional to electronic systems reachable via website-based or mobile applications. This service is essential for supporting formal communication processes across a variety of areas. However, one of the most significant issues is ensuring that information is obtained in a timely manner. A potential solution to this issue is to develop a notification system. The purpose of this research was to create an Automatic Notification System in the context of correspondence management utilizing the Waterfall software development methodology. This notification system was developed by integrating the Application Programming Interface (API) with Firebase Cloud Messaging (FCM) as an interface for sending cross-platform notifications in real time. The results of the research show that this automatic notification system can be utilized to help stakeholders and letter receivers find out about incoming or outgoing letters in real time.

Keywords: Correspondence management, Notifications, Application Programming Interface, Firebase Cloud Messaging, Waterfall

1. PENDAHULUAN

Peran manajemen kantor dalam suatu instansi sangat penting. Peran ini mencakup pencatatan administrasi, manajemen surat, dan manajemen alur kerja atau proses bisnis. Menurut [1], segala dokumentasi kegiatan dalam suatu instansi atau organisasi dipastikan sebagai alat bukti untuk tujuan legalitas dan data arsip termasuk di dalamnya surat yang dikirim dan diterima menjadi informasi yang berguna bagi instansi atau organisasi.

Dalam era digitalisasi ini, layanan persuratan telah mengalami transformasi. Pada umumnya, sudah banyak instansi telah memiliki aplikasi untuk menunjang layanan ini. Layanan ini memegang peranan penting dalam mendukung aktivitas komunikasi formal, baik di sektor pemerintahan, perusahaan, maupun institusi pendidikan. Namun, salah satu tantangan utama yang dihadapi adalah bagaimana memastikan pesan atau informasi yang dikirim melalui sistem persuratan diterima,

dipahami, dan ditindaklanjuti oleh penerima secara tepat waktu. Permasalahan seperti keterlambatan dalam merespon surat, hilangnya notifikasi terkait dokumen penting, atau kesalahan dalam pengelolaan dokumen sering kali terjadi karena kurangnya mekanisme pengingat atau pemberitahuan yang efektif. Hal ini dapat mengakibatkan dampak negatif, seperti penundaan dalam pengambilan keputusan, menurunnya produktivitas, dan bahkan risiko kehilangan peluang strategis. Selain itu, tren digitalisasi yang meningkat mendorong integrasi sistem notifikasi ini dapat dirasakan di berbagai platform namun sistem ini masih belum banyak dikembangkan. Sehingga dibutuhkan pengembangan sistem notifikasi yang generik. Hal tersebut menjadi permasalahan, terutama pada sistem informasi manajemen perkantoran yang memerlukan penanganan yang cepat.

Perkembangan teknologi yang sangat pesat telah mengubah banyak aspek dalam proses komunikasi data. Perkembangan komunikasi data semakin memudahkan masyarakat dalam memperoleh informasi. Komunikasi merupakan hal yang sangat penting dalam kehidupan sehari-hari, termasuk dalam kegiatan organisasi suatu instansi. Notifikasi merupakan salah satu metode komunikasi melalui jaringan internet yang memberikan layanan penting [2]. Artinya, layanan notifikasi ini membantu pengguna mengelola tugasnya, mengubah informasi pada aplikasi yang digunakannya, serta dapat mengingatkan pengguna akan tugas yang sedang dikerjakannya. Fitur notifikasi manajemen persuratan ini memungkinkan pemangku kepentingan diberitahu tentang penerimaan surat sehingga bisa segera menindaklanjuti surat yang diterimanya. Selain itu, penerima surat juga akan mengetahui jika terdapat surat yang dikirimkan.

Selama beberapa tahun terakhir, permintaan pengguna untuk aplikasi mobile telah mengalami peningkatan yang signifikan. Pergeseran ini dapat dikaitkan dengan beberapa faktor yang menyoroti keunggulan dan kemudahan yang ditawarkan aplikasi mobile. Misalnya, aplikasi mobile memanfaatkan konektivitas internet yang konstan, sehingga menjadikannya ideal untuk pembaruan dan pemberitahuan secara *real-time*. Aplikasi mobile dapat mengirimkan notifikasi langsung ke pengguna, menyediakan pembaruan yang tepat waktu dan relevan. Hal ini menyebabkan peningkatan keterlibatan dan retensi pengguna melalui peringatan dan pengingat, peningkatan keterlibatan pengguna, dan kemampuan untuk tetap mendapat informasi secara instan. Selain itu, membuat pemberitahuan bersifat generik dapat memudahkan pengembang untuk mengembangkan aplikasi di berbagai platform, misalnya situs web dan mobile.

Beberapa penelitian terkait perancangan aplikasi sistem notifikasi otomatis dalam konteks pengelolaan surat menyurat. Mekanisme notifikasi yang telah diimplementasikan antara lain dengan menggunakan *SMS Gateway* [3-4] pada perangkat *smartphone*. Mekanisme kedua adalah *Firebase Cloud Messaging* (FCM) yang digunakan untuk sistem berbasis website [5] dan sistem berbasis mobile [6]. Selanjutnya mekanisme ketiga menggunakan *Application Programming Interface* (API) yang dirancang untuk aplikasi surat menyurat berbasis website dengan memanfaatkan Telegram API [7] [8]. Berdasarkan hasil penelitian ini, belum ada pengembangan model notifikasi yang dibuat untuk aplikasi multiplatform seperti aplikasi mobile dan website yang terintegrasi. Selain itu, belum ada fitur manajemen notifikasi untuk manajemen surat menyurat. Permasalahan ini berkaitan dengan permintaan pengguna terhadap aplikasi terintegrasi yang dapat diakses pada website dan aplikasi mobile dimana saja dan kapan saja. Oleh karena itu, model notifikasi akan lebih efisien dan menghemat waktu dalam proses pengembangan suatu aplikasi jika modelnya dibuat generik, dengan kata lain, model notifikasi dapat digunakan oleh berbagai platform seperti situs web dan mobile.

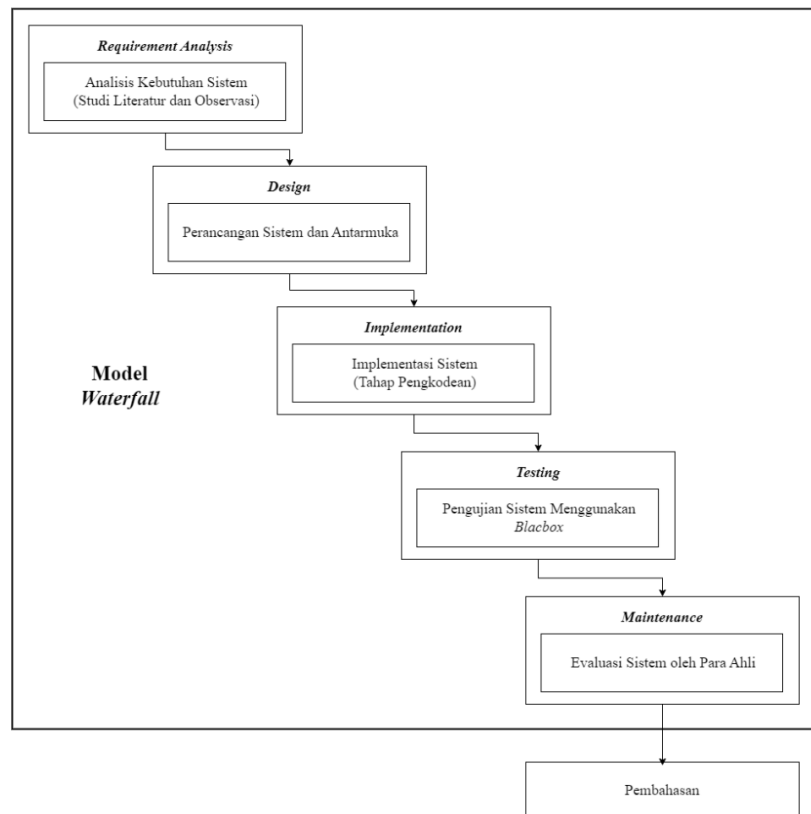
Model notifikasi generik akan lebih efisien dan menghemat waktu dalam proses pengembangan sistem multiplatform. Untuk membuat suatu model menjadi generik dibutuhkan sebuah teknologi yang disebut *Application Programming Interface* (API). API berfungsi sebagai perantara beberapa aplikasi baik yang berada pada platform yang sama maupun lintas platform sehingga dapat saling berkomunikasi[9].

Arsitektur yang digunakan untuk membuat model pada API ini adalah RESTful API. Cara kerja RESTful API ini adalah klien REST akan mengakses data atau resource pada server REST yang mana setiap resource akan dibedakan berdasarkan sebuah ID global atau URI (Universal Resource Identifiers). Mekanisme yang digunakan dalam pengiriman notifikasi atau pesan pada API tersebut adalah *Firebase Cloud Messaging* (FCM). FCM dapat mengirimkan pesan secara real time kepada klien yang disediakan oleh Google secara gratis[10]. Model notifikasi ini memiliki beberapa request atau fitur seperti melihat manajemen notifikasi berdasarkan hak akses pengguna, menambahkan dan mengirimkan pesan atau notifikasi kepada klien [5].

Oleh karena itu, untuk mengatasi permasalahan di atas, penelitian ini bermaksud untuk merancang sebuah aplikasi notifikasi otomatis multiplatform yang dapat diakses pada website dan

mobile dengan fitur pengelolaan surat seperti membuat surat masuk dan surat keluar. Pengembangan ini menggunakan metode *waterfall* dengan menggunakan mekanisme pengiriman notifikasi berbasis cloud yang disediakan oleh Google, yaitu Firebase Cloud Messaging (FCM). Dengan menggunakan FCM dan API ini diharapkan pengiriman pesan notifikasi dapat dilakukan lintas platform secara real time.

2. METODE PENELITIAN



Gambar 1 Desain Penelitian

Gambar 1 menggambarkan desain penelitian yang dilakukan dengan menerapkan metode *waterfall*. Berikut ini adalah penjelasan dari setiap tahapan yang telah disusun:

1. Tahap pertama adalah menganalisis kebutuhan sistem. Langkah ini dilakukan dengan melakukan studi literatur untuk mendapatkan data dari penelitian yang relevan tentang sistem notifikasi otomatis dalam sistem manajemen surat. Selain itu, dilakukan observasi pada sistem persuratan yang sedang berjalan, yaitu dengan melakukan observasi dan analisis Sistem Manajemen Persuratan di FPMIPA UPI. Setelah itu, data-data dari kegiatan tersebut digunakan untuk mengidentifikasi, mengevaluasi, dan menginterpretasikan kebutuhan sistem yang akan dibangun.
2. Tahap desain sistem dilakukan dengan melakukan perancangan sistem menggunakan diagram UML seperti *use case diagram*, *activity diagram*, *sequence diagram*, diagram basis data, dan desain antarmuka pengguna sistem.
3. Tahap implementasi dilakukan dengan mengubah sistem desain menjadi kode program menggunakan bahasa pemrograman Jawa Script.
4. Tahap pengujian sistem dilakukan dengan melakukan pengujian metode Blackbox dan pengujian ahli untuk memastikan sistem yang dibangun berfungsi dengan baik dan sesuai dengan kebutuhan sistem.
5. Tahap pemeliharaan dilakukan dengan melakukan perbaikan kode sistem sebagai hasil perbaikan sistem pengujian oleh penguji atau validasi ahli.

3. HASIL DAN PEMBAHASAN

- A. Analisis Kebutuhan Sistem
 - 1) Hasil Studi Pustaka

Hasil dari studi pustaka terkait notifikasi otomatis dalam konteks pengelolaan surat menyurat menunjukkan bahwa terdapat beberapa aplikasi manajemen kantor yang sudah menggunakan fitur notifikasi real-time sebagai sarana komunikasi antar pemangku kepentingan dengan beragam cara. Berdasarkan studi pustaka yang didapatkan, pada Tabel 1 disajikan rangkuman teknologi dan fitur pengelolaan surat menyurat yang sudah pernah diterapkan di beberapa penelitian.

Tabel 1 Ringkasan Teknologi Sistem Notifikasi

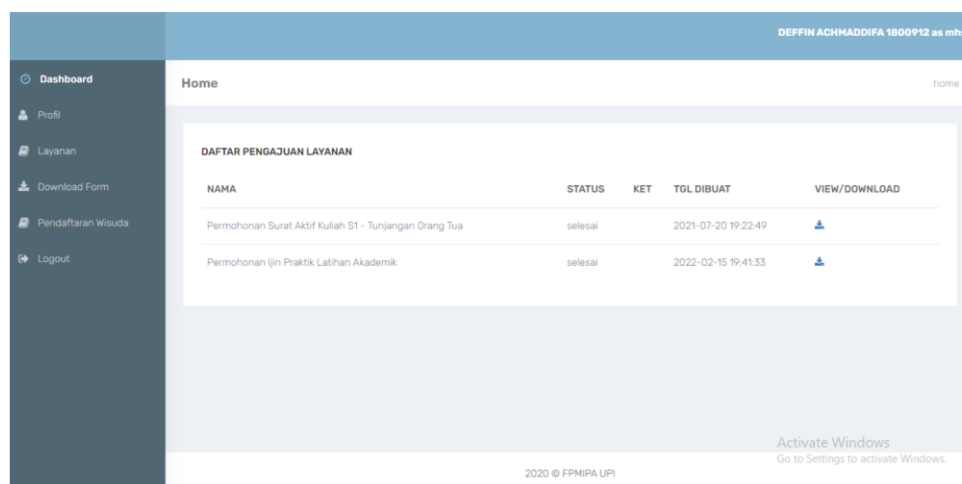
No.	Teknologi Sistem Notifikasi	Keuntungan	Kekurangan	Peneliti
1.	SMS Gateway	Pengiriman informasi atau pesan berlangsung cepat dan biaya yang dikeluarkan relatif murah (hanya untuk jangka waktu pendek)	Notifikasi hanya dikirim ke smartphone dan jika sistem sedang error karena jaringan komunikasi, sistem tidak dapat mengirim ulang pesan Mekanisme ini membutuhkan biaya yang besar untuk membayar pulsa setiap kali mengirim SMS notifikasi dalam jangka waktu yang lama.	[3], [4]
2.	Firebase Cloud Messaging (FCM)	Mekanisme teknologi ini membantu pengiriman dan penerimaan pesan notifikasi di latar depan (onMessage/foreground) dan pesan di latar belakang (onBackgroundMessage) di iOS, Android, dan berbasis web tanpa biaya. FCM juga dapat mengirim pesan menggunakan segmen target yang telah ditentukan sebelumnya menggunakan demografi dan perilaku. Selain itu, pesan dapat dikirim hanya ke perangkat yang telah berlangganan topik tertentu, dan dapat mengirim pesan hanya ke satu perangkat untuk mendapatkan informasi data terperinci.	Layanan FCM dapat berjalan lebih lancar dan optimal jika menggunakan protokol HTTPS, bukan HTTP.	[5], [6]
3.	Application Programming Interface (API) with Telegram Messaging Application	Memfasilitasi transfer pesan antara dua atau lebih perangkat melalui media sosial telegram.	Pengiriman pesan notifikasi hanya dapat diakses melalui smartphone dan memerlukan aplikasi pihak ketiga	[7], [8]
4.	PushJS	Sama seperti FCM, PushJS dapat menerima pesan secara real-time dan dapat dikembangkan menggunakan <i>service worker</i> yang menerima pesan di latar depan (onMessage) dan latar belakang	Pesan hanya dapat diakses dengan menggunakan aplikasi berbasis web dan memerlukan pihak	[11]

		(onBackgroundMessage). Artinya, pengguna dapat menerima pesan tanpa membuka atau mengakses aplikasi.	ketiga untuk menambahkan pesan.	
5.	Progressive Web Apps (PWA)	Aplikasi web dapat berjalan seperti aplikasi web mobile menggunakan browser. Pengguna dapat menerima pemberitahuan secara real-time	Sistem tidak dapat menerima pesan untuk aplikasi android dan hanya menerima pemberitahuan pesan	[12]

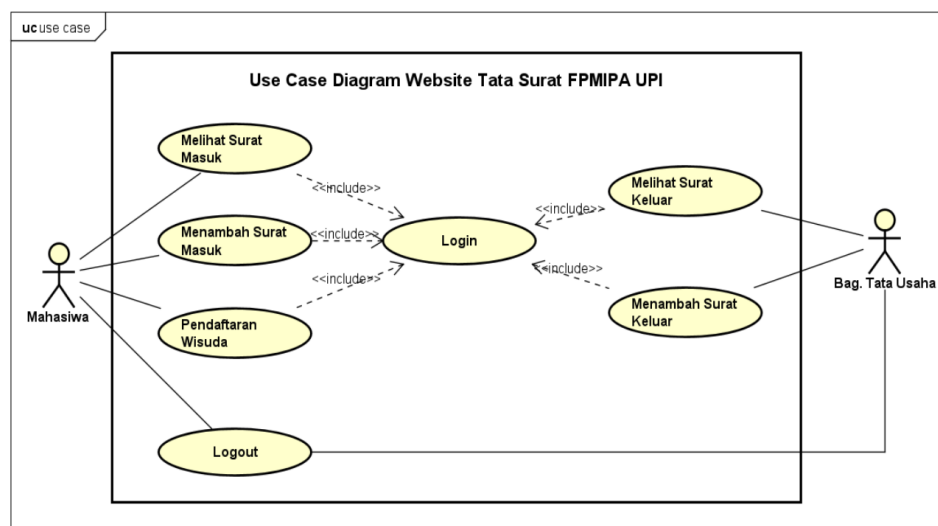
Berdasarkan kajian penelitian-penelitian tersebut, belum ada penelitian yang mengembangkan atau menggunakan model notifikasi untuk sistem generik yang diimplementasikan pada platform website dan mobile application. Untuk mengatasi masalah ini, *Application Programming Interface* (API) dapat digunakan sebagai media komunikasi bagi beberapa aplikasi atau aplikasi multiplatform untuk saling berkomunikasi dan berintegrasi satu sama lain.

2) Hasil Observasi dan Analisis Sistem Berjalan

Analisis sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas cara kerja sistem, dan permasalahan yang dihadapi oleh sistem yang akan dijadikan dasar perancangan usulan. Analisis sistem yang sedang berjalan ini dilakukan untuk mendukung data hasil studi pustaka. Pada tahapan ini, dilakukan observasi dan analisis pada Website Sistem Manajemen Persuratan di Fakultas Pendidikan Matematika dan IPA (FPMIPA) Universitas Pendidikan Indonesia (Gambar 2).



Gambar 2 Website Manajemen Persuratan FPMIPA UPI



Gambar 3 Use Case Diagram pada Sistem Manajemen Persuratan FPMIPA UPI

Pada Gambar 3 disajikan gambar use case pada manajemen persuratan FPMIPA UPI. Terdapat dua aktor yang berinteraksi dengan sistem, yaitu mahasiswa dan bagian Tata Usaha. Kedua aktor ini mempunyai use case login dan logout ke sistem. Mahasiswa memiliki peran antara lain melihat surat masuk, menambah surat masuk, dan registrasi wisuda. Sedangkan bagian Tata Usaha memiliki peran antara lain melihat surat keluar dan menambah surat keluar. Mahasiswa dan bagian Tata Usaha harus login terlebih dahulu untuk mengakses perannya masing-masing. Definisi aktor dapat dilihat pada Tabel 2.

Tabel 2 Daftar Use Case Pada Sistem Manajemen Persuratan FPMIPA UPI

No.	Use Case	Deskripsi
1.	Login	Proses atau tindakan yang digunakan oleh aktor untuk mendapatkan akses ke sistem.
2.	Logout	Proses atau tindakan yang digunakan oleh aktor untuk menutup akses ke situs web
3.	Melihat Surat Masuk dari petugas	Mahasiswa dapat menerima surat masuk yang telah diminta dan dibuat oleh admin.
4.	Memambah Surat Masuk ke petugas	Mahasiswa meminta pembuatan surat dan mengirimkannya ke admin
5.	Mendaftar Wisuda	Mahasiswa dapat mengirimkan formulir pendaftaran wisuda ke admin untuk melengkapi administrasi wisuda
6.	Melihat Surat Keluar dari mahasiswa	Admin dapat melihat surat keluar yang telah dibuat untuk mahasiswa di menu keluar
7.	Membuat Surat Keluar ke mahasiswa	Admin dapat membuat dan mengirimkan surat permintaan ke mahasiswa.

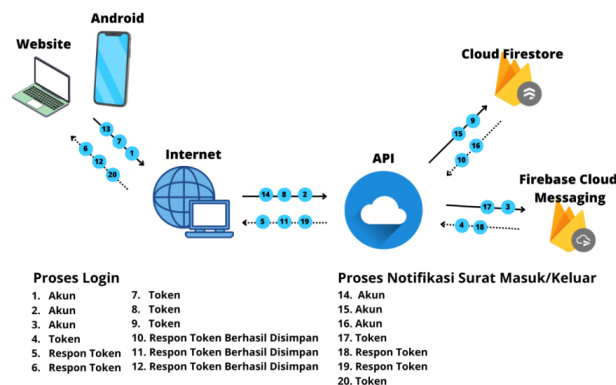
Berdasarkan analisis sistem yang ada, terdapat beberapa keterbatasan pada sistem, misalnya sistem belum memiliki sistem notifikasi dan manajemen notifikasi. Tujuan dari sistem yang diusulkan adalah memudahkan pengguna untuk mendapatkan notifikasi surat. Selain itu, pengembangan manajemen notifikasi membantu pengguna untuk mencari dan membaca notifikasi surat. Tabel 3 di bawah ini menjelaskan deskripsi evaluasi sistem yang ada berdasarkan analisis sistem yang ada.

Tabel 3 Daftar Evaluasi Sistem Manajemen Persuratan FPMIPA UPI

No.	Keterbatasan Sistem	Solusi yang diusulkan
1.	Belum ada fitur untuk pengiriman pesan/notifikasi surat masuk dan keluar	Mengembangkan sistem notifikasi otomatis untuk surat masuk dan keluar menggunakan sistem firebase.
2.	Belum ada fitur untuk manajemen notifikasi pada multi-platform (website dan mobile application)	Mengembangkan manajemen notifikasi menggunakan Application Programming Interface.

B. Hasil Desain Sistem

1) Desain Sistem Notifikasi



Gambar 4 Desain Sistem Notifikasi Real-time

Sebelum melakukan pengkodean, langkah yang perlu dilakukan adalah membuat desain model sistem notifikasi yang akan dibangun. Gambar 4 menunjukkan desain sistem notifikasi yang menggambarkan alur notifikasi pada aplikasi. Berikut ini adalah alur model notifikasi yang dibangun.

- a) Pengguna harus memastikan bahwa aplikasi terhubung dengan internet. Kemudian, pengguna harus melakukan login ke aplikasi untuk mendapatkan akses melalui API.
- b) API mengirimkan data ke akun pengguna berupa token unik untuk didaftarkan oleh FCM
- c) Token yang sudah didaftarkan oleh FCM akan dikirimkan oleh API dengan mengirimkan respons token ke aplikasi, dimana token ini digunakan untuk mengirim dan menerima notifikasi.
- d) Aplikasi akan melakukan *callback* API untuk menyimpan token ke dalam basis data Firebase
- e) Terdapat beberapa kondisi, jika pengguna melakukan permintaan surat oleh aktor mahasiswa, token akan disimpan ke dalam data surat keluar. Sedangkan, jika pengguna sebagai admin, token akan disimpan ke dalam data surat masuk. Selain itu, jika pengguna mengakses aplikasi melalui *smartphone*, token akan disimpan dengan nama *mobile key*. Selanjutnya, jika pengguna mengakses melalui *website*, token akan disimpan dengan nama *web key*.
- f) API akan memberikan respon berupa pesan yang menunjukkan token sudah berhasil disimpan

Untuk menjalankan sistem notifikasi, aktor mahasiswa harus membuat surat masuk untuk admin sedangkan admin harus membuat surat keluar untuk mahasiswa. Ketika surat dibuat, aplikasi akan memanggil API untuk menyimpan data notifikasi ke dalam database dan akan mengirimkan notifikasi ke pengguna. Data permintaan yang dikirim oleh aplikasi ditunjukkan pada Gambar 5.



Gambar 5 Data (Request Body) API untuk pengiriman notifikasi

Pada request body API terdapat kondisi dimana apabila request body dengan key "to" kosong atau tidak terisi, maka API akan mengambil semua data token Admin dari database. Namun apabila request body dengan key "to" terisi, maka API akan mengambil data token berdasarkan request body dengan key "to". Kemudian, API akan mengirimkan notifikasi tersebut ke FCM dan FCM akan mengirimkan pesan ke aplikasi yang terdaftar di database agar pengguna dapat melihat notifikasi tersebut di *smartphone* atau *website* pengguna. Apabila pengguna tidak sedang mengakses aplikasi, maka notifikasi tersebut akan diterima di *background message*.

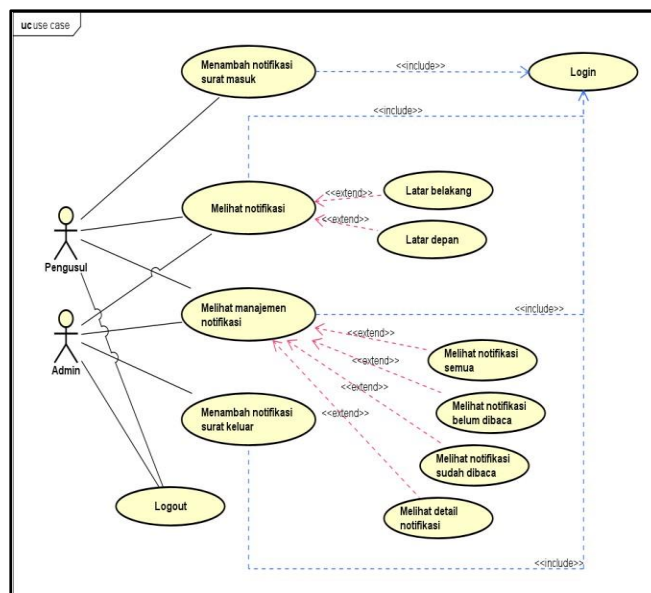
2) Use Case Diagram

Diagram use case untuk sistem yang diusulkan digambarkan pada Gambar 6. Pada diagram ini terdapat dua aktor dan rincian peran aktor dijelaskan pada Tabel 4. Aktor pertama adalah mahasiswa/pengusul, dan aktor kedua adalah admin. Aktor memiliki akses yang sama yaitu dapat melihat notifikasi saat aplikasi dibuka (*foreground notification*) dan aplikasi ditutup (*background notification*). Selain itu, aktor juga dapat melihat manajemen notifikasi seperti melihat semua notifikasi, melihat notifikasi yang belum dibaca, melihat notifikasi yang telah dibaca, dan melihat detail notifikasi. Namun, perbedaan akses untuk kedua aktor tersebut adalah aktor mahasiswa dapat menambahkan notifikasi surat masuk. Sementara Admin dapat menambahkan notifikasi surat keluar. Rincian use case dijelaskan pada Tabel 5

Tabel 4 Daftar Aktor pada use case Sistem notifikasi yang diusulkan

No.	Aktor	Deskripsi
1.	Pengusul	Aktor memiliki akses untuk membuat surat masuk, menambahkan notifikasi surat masuk, melihat notifikasi, dan melihat manajemen notifikasi.

2.	Admin	Aktor memiliki akses untuk membuat surat keluar, menambahkan notifikasi surat keluar, melihat notifikasi, dan melihat manajemen notifikasi.
----	-------	---



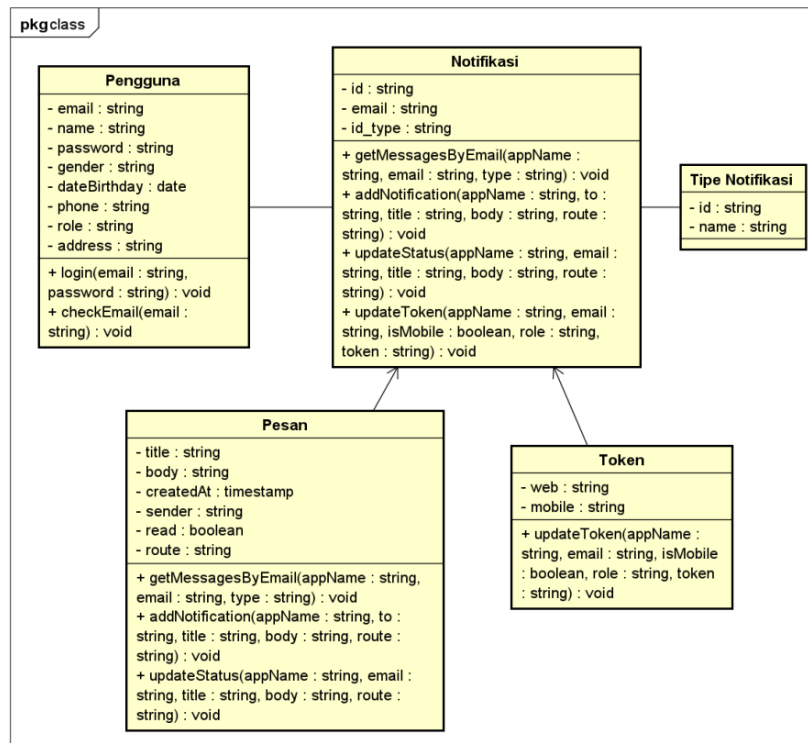
Gambar 6 Diagram use case sistem notifikasi yang diusulkan

Tabel 5 Deskripsi use case sistem notifikasi yang diusulkan

No.	Use Case	Deskripsi
1.	Login	Pengguna yang mengusulkan pembuatan surat (pengusul/mahasiswa) dan Admin harus masuk untuk mendapatkan akses masing-masing dan memperoleh token yang digunakan untuk menerima notifikasi.
2.	Menambah Notifikasi Surat Masuk	Aktor dapat membuat surat masuk untuk menambahkan notifikasi surat masuk bagi admin
3.	Melihat Notifikasi	Dua aktor dapat melihat notifikasi satu sama lain
4.	Melihat manajemen notifikasi	Dua aktor dapat melihat daftar notifikasi, memfilter notifikasi, dan melihat detail notifikasi
5.	Menambahkan notifikasi surat keluar	Admin dapat membuat data surat keluar untuk menambahkan notifikasi surat keluar bagi siswa.

3) Class Diagram

Gambar 7 menggambarkan diagram kelas dari sistem aplikasi yang diusulkan. Diagram kelas mencakup beberapa kelas seperti kelas pengguna, kelas notifikasi, kelas jenis notifikasi, kelas pesan, dan kelas token. Kelas notifikasi dan kelas pengguna memiliki hubungan untuk menggambarkan bahwa pengguna sebagai aktor yang mengirim dan menerima notifikasi. Selain itu, kelas notifikasi memiliki hubungan dengan kelas tipe notifikasi. Kemudian, tipe notifikasi akan mengatur siapa yang dapat menerima notifikasi sehingga tipe kelas notifikasi memiliki hubungan dengan kelas pengguna.



Gambar 7 Diagram class sistem notifikasi yang diusulkan

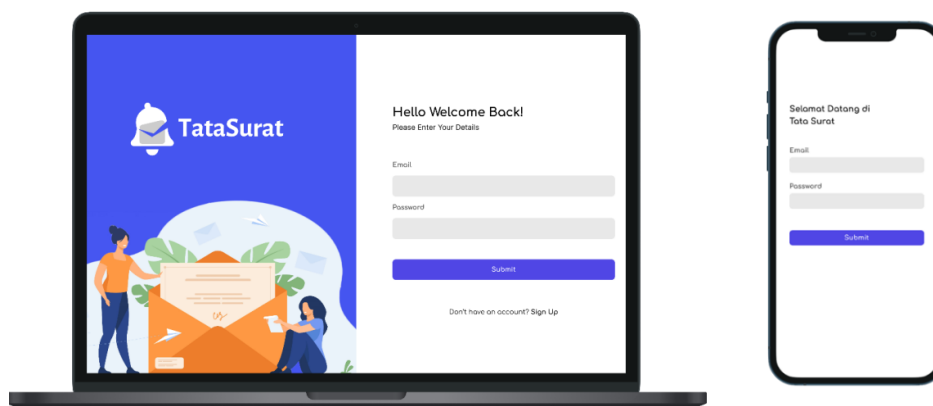
Terdapat dua jenis notifikasi, notifikasi surat masuk dikirim oleh siswa sedangkan notifikasi surat keluar dikirim oleh admin. Kelas pesan dan kelas token memiliki hubungan pewarisan dari notifikasi karena jenis notifikasi tersebut berisi pesan atau token.

4) Desain User Interface Sistem

Sistem antarmuka pengguna menggunakan perangkat lunak Figma. Desain user interface yang dibangun merupakan transformasi daftar analisis kebutuhan sistem menjadi desain yang akan diimplementasikan pada tahap berikutnya. Pada tahap desain ini dibangun antarmuka pengguna untuk aplikasi mobile dan aplikasi web seperti pada penjelasan di bawah ini.

a) User Interface Fungsi Login

Gambar 8 menunjukkan antarmuka pengguna untuk halaman login yang berisi formulir untuk mengisi email dan kata sandi.

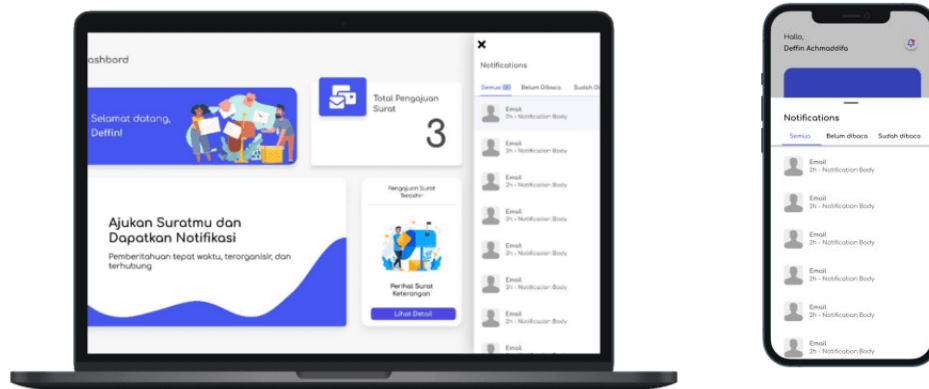


Gambar 8 Desain UI untuk form login di halaman website (kiri) dan mobile (kanan)

b) User Interface untuk fungsi manajemen notifikasi

Gambar 9 di sebelah kiri menunjukkan tampilan manajemen notifikasi saat pengguna mengakses aplikasi web dan Gambar 9 di sebelah kanan menunjukkan tampilan notifikasi pada aplikasi halaman mobile. Tampilan pada Gambar 9 di sebelah kanan akan muncul saat tombol notifikasi ditekan. Pada tampilan ini, terdapat daftar notifikasi yang masuk. Tampilan ini juga berisi filter untuk notifikasi

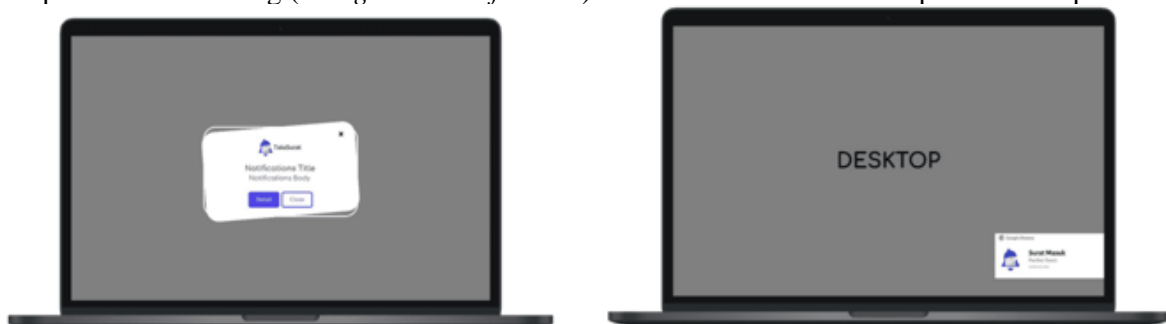
seperti semua notifikasi, notifikasi yang belum dibaca, dan notifikasi yang sudah dibaca. Setiap item notifikasi akan menampilkan detail notifikasi berupa judul notifikasi, pengirim, detail notifikasi, dan waktu pengiriman surat.



Gambar 9 Desain UI untuk manajemen notifikasi di halaman website (kiri) dan mobile (kanan)

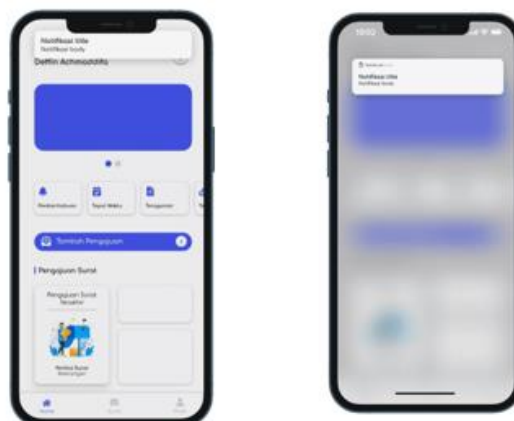
c) User Interface Notifikasi

Gambar 10 adalah desain halaman notifikasi yang ditampilkan kepada pengguna saat aplikasi dibuka di situs web. Antarmuka berisi logo, judul notifikasi, notifikasi pesan, tombol detail, dan tombol tutup. Gambar 10 di sisi kiri adalah antarmuka pengguna untuk pesan latar depan (*foreground notification*) di halaman situs web, sedangkan Gambar 10 di sisi kanan adalah antarmuka pengguna untuk pesan latar belakang (*background notification*) di halaman situs web saat aplikasi ditutup.



Gambar 10 Desain UI pop up notifikasi pada halaman website (kiri) dan desktop (kanan)

Pada aplikasi mobile, Gambar 10 pada sisi kiri merupakan tampilan notifikasi saat pengguna membuka aplikasi, sedangkan Gambar 10 pada sisi kanan merupakan tampilan notifikasi saat aplikasi mobile ketika ditutup.



Gambar 11 Desain pop up notifikasi pada aplikasi mobile

d. Pengembangan Sistem Notifikasi

Prototipe sistem dibuat dengan menggunakan pemrograman Jawa Script. Aplikasi berbasis web menggunakan framework ReactJS sedangkan aplikasi berbasis mobile menggunakan framework React native. Sementara itu, model notifikasi dirancang menggunakan API. Framework yang digunakan

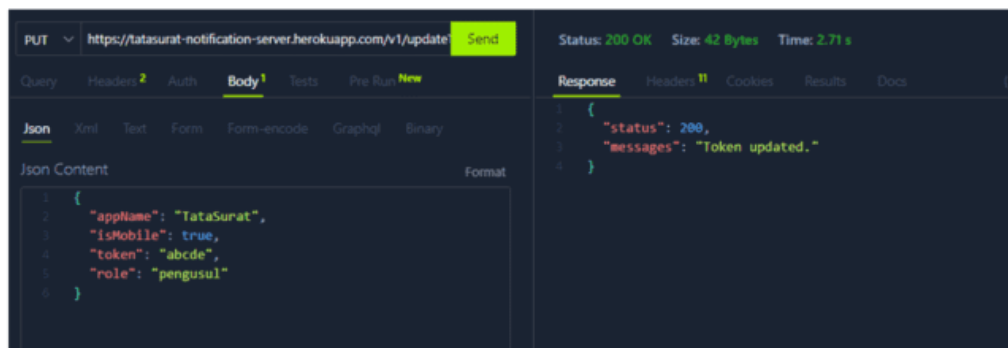
adalah Express JS sehingga model notifikasi dapat diakses oleh berbagai sistem.

e. Pengembangan API untuk kebutuhan sistem notifikasi

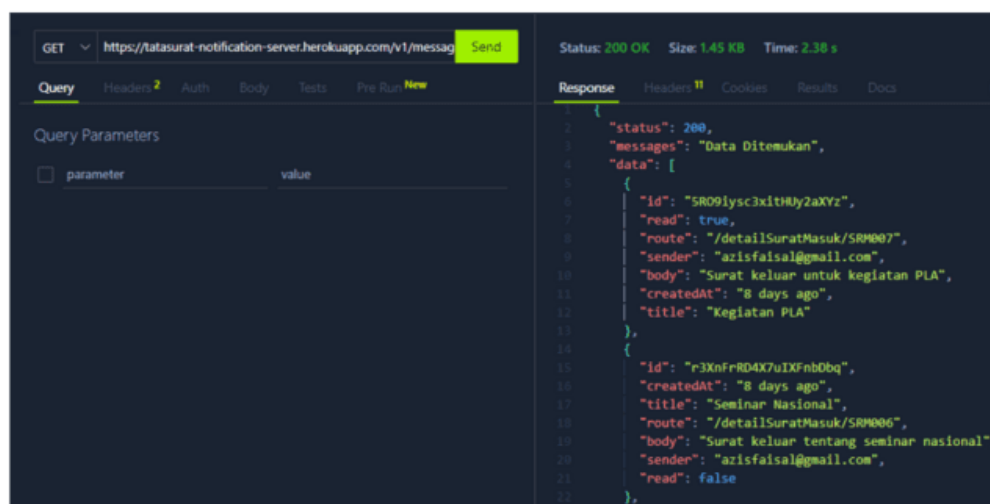
Model notifikasi yang dibangun dapat diakses menggunakan aplikasi Postman. Tabel 6 menjelaskan daftar API yang dibuat untuk sistem notifikasi model.

Tabel 6 Daftar API untuk kebutuhan sistem notifikasi

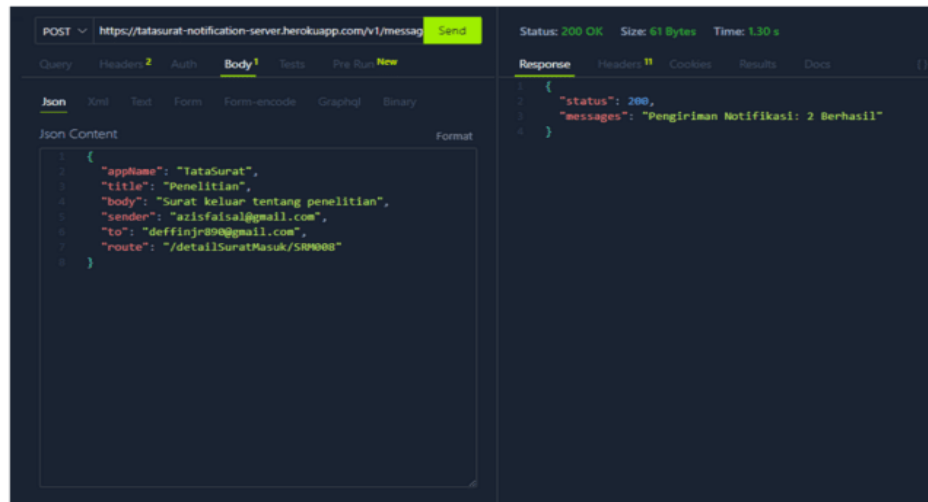
No.	API Path	Method	Body	Parameter	Deskripsi	No Gambar
1.	/updateToken	PUT	Key appName, isMobile, token, role	Email	API untuk mengupdate token	12
2.	/messages	GET	-	Application name, email, type of letter	API untuk melihat dan membaca daftar notifikasi	13
3.	/messages	POST	Key appName, title, body, sender, to, and route	-	API untuk mengirim pesan notifikasi	14
4.	/updateStatus	POST	Key appName, title, body, route	email	API untuk mengubah status notifikasi	15



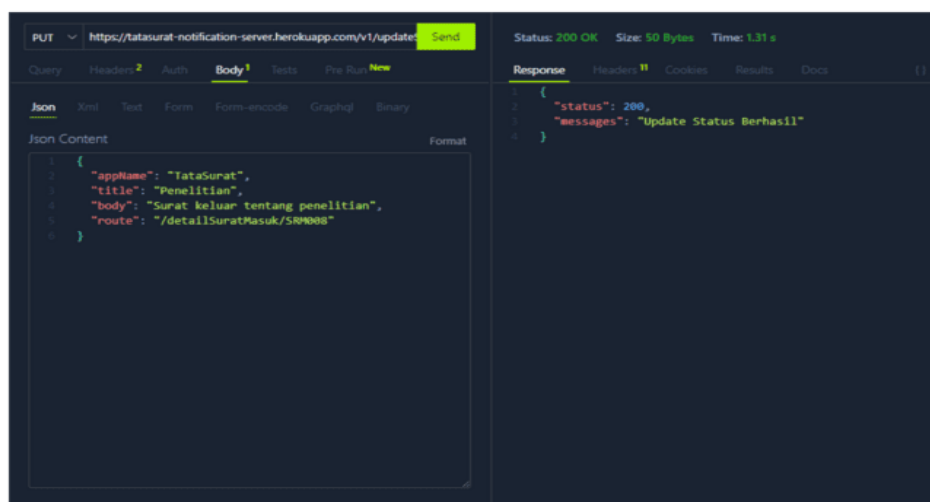
Gambar 12 API untuk update token



Gambar 13 API untuk melihat dan membaca daftar notifikasi



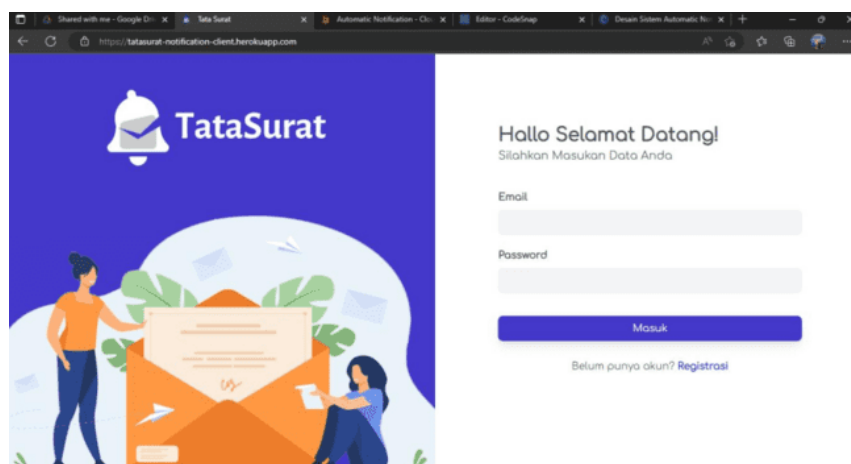
Gambar 14 API untuk mengirim notifikasi



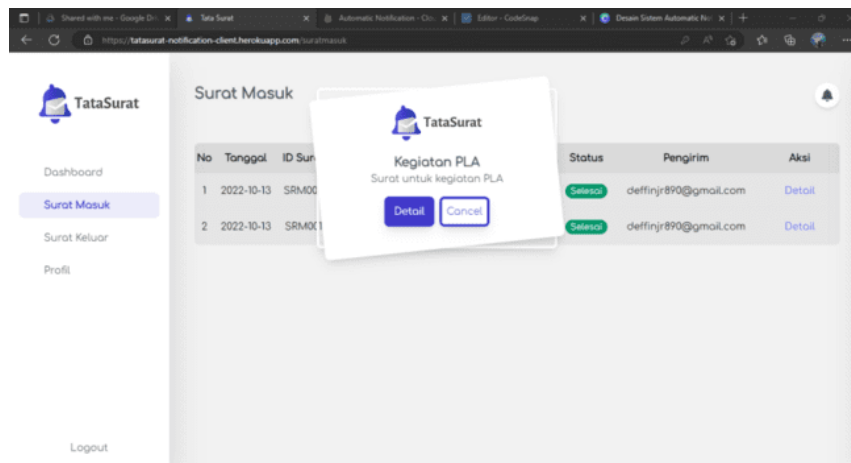
Gambar 15 API untuk mengupdate status notifikasi

f. Hasil Pengkodean

Gambar 16 merupakan hasil implementasi koding untuk halaman login pada website. Pada halaman ini terdapat form login berupa email, password, dan tombol login. Gambar 17 merupakan tampilan notifikasi yang diterima oleh pengguna ketika membuka aplikasi berbasis website. Pada tampilan ini terdapat logo aplikasi, judul notifikasi, detail notifikasi, tombol detail, dan tombol tutup. Kemudian, gambar 18 merupakan tampilan notifikasi ketika aplikasi berbasis web ditutup.



Gambar 16 Hasil implementasi kode untuk halaman login di website

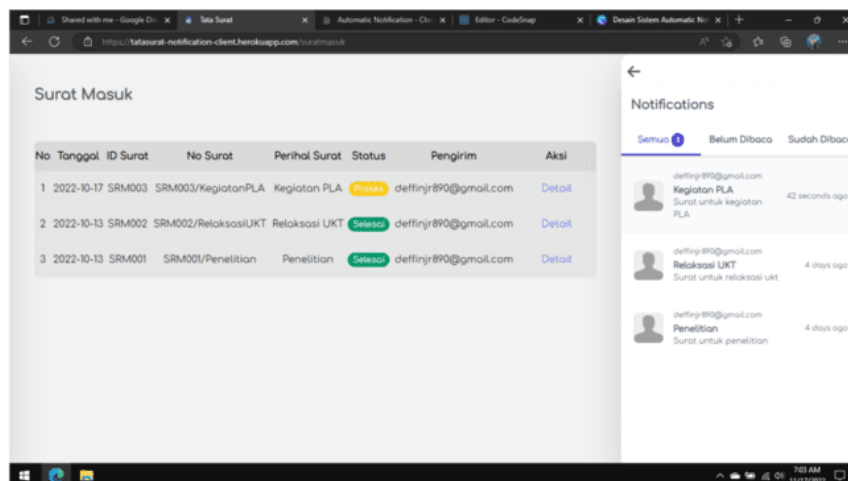


Gambar 17 Hasil implementasi kode untuk foreground notification di halaman website

Gambar 19 merupakan implementasi sistem manajemen notifikasi ketika pengguna mengakses aplikasi berbasis website. Tampilan ini akan muncul di sisi kanan desktop/website dan muncul ketika tombol notifikasi ditekan. Pada tampilan ini terdapat daftar notifikasi yang masuk untuk pengguna. Tampilan ini juga berisi filter notifikasi meliputi daftar semua notifikasi, daftar notifikasi yang belum dibaca, dan daftar notifikasi yang sudah dibaca. Setiap item notifikasi akan menampilkan judul notifikasi, pengirim, detail notifikasi, dan waktu pengiriman surat.



Gambar 18 Hasil implementasi kode untuk background notification di halaman website



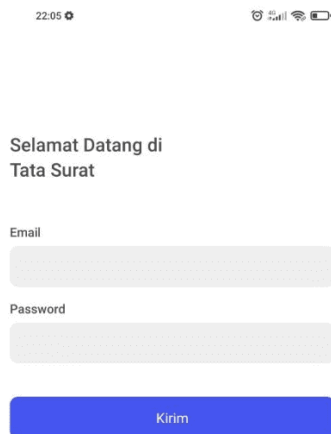
Gambar 19 Hasil implementasi kode untuk halaman manajemen notifikasi di halaman website ketika aplikasi dibuka

Gambar 20 merupakan implementasi sistem manajemen notifikasi saat pengguna tidak mengakses aplikasi berbasis situs web. Tampilan ini berada di sisi kanan desktop dan akan muncul saat tombol notifikasi ditekan. Pada tampilan ini terdapat daftar notifikasi yang masuk. Setiap item notifikasi berisi logo aplikasi, judul notifikasi, detail notifikasi, dan waktu pengiriman surat.

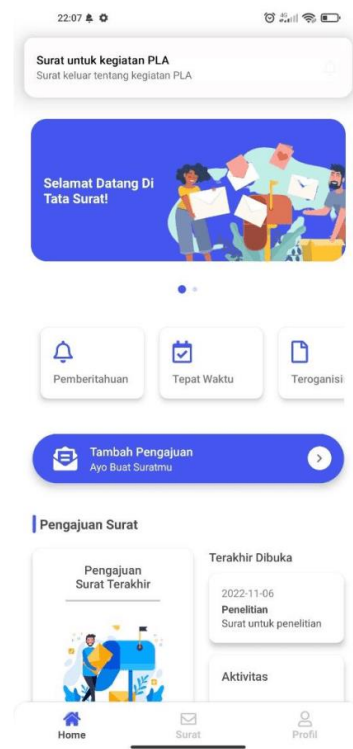


Gambar 20 Hasil implementasi kode untuk halaman manajemen notifikasi di halaman website ketika aplikasi ditutup

Sementara itu, hasil implementasi kode untuk aplikasi mobile ditunjukkan pada Gambar 21 dan Gambar 22. Gambar 21 merupakan halaman *login* pada aplikasi mobile dan Gambar 22 merupakan pop up notifikasi menggunakan mekanisme *foreground* yang diterima oleh pengguna saat aplikasi dibuka. Pop up notifikasi berisi notifikasi judul dan notifikasi pesan detail.

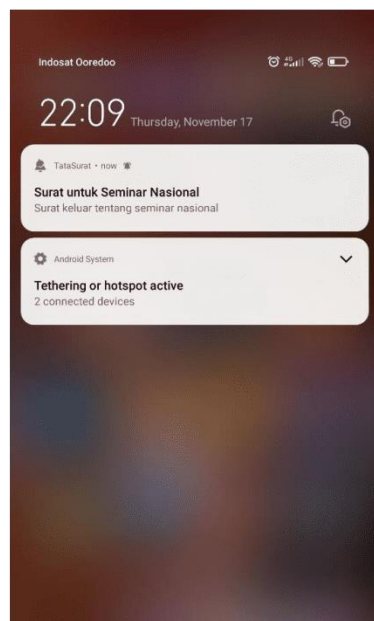


Gambar 21 Hasil implementasi kode untuk halaman login di halaman mobile

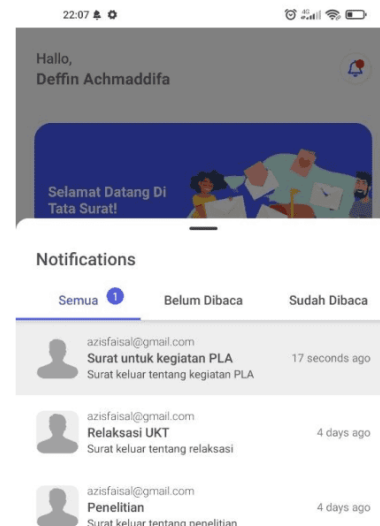


Gambar 21 Hasil implementasi kode untuk pop up notifikasi di halaman mobile

Hasil implementasi kode *background notification* pada halaman mobile ditunjukkan pada Gambar 23. Kemudian, Gambar 24 menampilkan hasil pengelolaan notifikasi pada aplikasi mobile yang berisi data detail notifikasi seperti judul, pengirim, detail notifikasi, dan waktu notifikasi. Selain itu, terdapat menu filter untuk melihat notifikasi meliputi semua notifikasi, notifikasi yang belum terbaca, dan notifikasi yang sudah terbaca.



Gambar 23 Hasil implementasi kode untuk notifikasi pada perangkat mobile



Gambar 24 Hasil implementasi kode untuk halaman daftar notifikasi pada perangkat mobile

C. Pengujian Sistem

Pengujian aplikasi menggunakan metode Blackbox untuk memastikan bahwa aplikasi yang dikembangkan sesuai dengan yang diharapkan dan memenuhi kebutuhan pengguna aplikasi, serta memastikan aplikasi bebas bug. Skenario pengujian untuk prototipe ini ditunjukkan pada Tabel 7. Dari skenario pengujian yang dijalankan, semua skenario pengujian lulus uji yang menandakan semua fungsional sistem berjalan dengan baik.

Tabel 7 Daftar skenario pengujian sistem dengan menggunakan metode blackbox

Item Pengujian	Skenario	Hasil Aktual	Status Pengujian
API Notifikasi	Mengirim token untuk aplikasi klien	Berhasil mengirim token sebagai respons dari FCM untuk aplikasi klien	Valid
	Menyimpan/memperbarui token untuk aplikasi klien	Berhasil mengirim respons, menyimpan/memperbarui token berdasarkan pengguna autentikasi dan platform yang digunakan pada basis data Firebase	Valid
	Mengirim pemberitahuan surat masuk untuk aplikasi klien	Berhasil mengirim respons lalu mengirim notifikasi untuk semua akun admin	Valid
	Mengirim pemberitahuan surat keluar untuk aplikasi klien	Berhasil mengirim respons lalu mengirim notifikasi untuk akun surat permintaan	Valid
	Menyimpan data pemberitahuan untuk aplikasi klien	Berhasil mengirim respons dan menyimpan data notifikasi ke dalam basis data Firestore berdasarkan pengguna autentikasi	Valid
	Mengirim data pemberitahuan berdasarkan pengguna	Berhasil mendapatkan respons sebagai data notifikasi berdasarkan pengguna autentikasi	Valid

	otentikasi untuk aplikasi klien		
	Memperbarui status pemberitahuan untuk aplikasi klien	Berhasil mendapatkan respons dan memperbarui notifikasi pada basis data Firebase	Valid
Fitur Sistem Notifikasi Persuratan	Mengisi data yang benar ke dalam formulir login	Proses login berhasil dan sistem menampilkan dasbor sesuai dengan autentikasi pengguna	Valid
	Mengisi data yang salah ke dalam formulir login	Sistem menampilkan pesan peringatan untuk menginformasikan bahwa data salah	Valid
	Mendapatkan token dari FCM melalui API	Sistem berhasil mendaftarkan token pada FCM untuk pengiriman dan penerimaan notifikasi	Valid
	Menyimpan token FCM ke dalam database Firebase melalui API	Sistem berhasil menyimpan token FCM pada basis data firebase sesuai dengan autentikasi pengguna dan platform yang diakses oleh pengguna	Valid
	Melihat notifikasi saat aplikasi terbuka	Sistem menampilkan notifikasi saat aplikasi terbuka	Valid
	Melihat notifikasi saat aplikasi ditutup	Sistem menampilkan notifikasi saat aplikasi ditutup	Valid
	Melihat halaman notifikasi manajemen berdasarkan pengguna yang diautentikasi	Sistem menampilkan halaman notifikasi manajemen berdasarkan autentikasi pengguna	Valid
	Melihat daftar notifikasi Filter notifikasi	Sistem menampilkan daftar notifikasi sesuai dengan filter yang aktif	Valid
	Memperbarui notifikasi status	Notifikasi status pembaruan sistem menjadi status “merah” setelah pengguna mengklik notifikasi	Valid
	Mengklik notifikasi	Sistem menampilkan detail notifikasi surat masuk/keluar	Valid

D. Hasil Tahapan Pemeliharaan

Tahap pemeliharaan untuk penelitian ini adalah melakukan pembaruan terhadap prototipe berdasarkan evaluasi ahli. Evaluasi ahli dilakukan dengan metode wawancara kepada dua orang pakar di bidang pengembangan perangkat lunak dari perusahaan yang berbeda. Hasil wawancara menyatakan model notifikasi memiliki fungsi sistem dengan baik. Artinya dapat digunakan oleh pemangku kepentingan untuk menerima surat notifikasi secara real-time. Selain itu, notifikasi model prototipe bersifat generik dan dapat digunakan dalam aplikasi multiplatform. Namun, ada beberapa catatan revisi sebagai berikut:

- Notifikasi tidak muncul saat aplikasi ditutup
- Notifikasi manajemen di halaman admin menampilkan daftar notifikasi masuk dan keluar selain itu halaman admin hanya menampilkan notifikasi surat masuk.

Setelah evaluasi pakar, tahap selanjutnya adalah melakukan kode perbaikan bug berdasarkan daftar revisi. Kemudian, setelah melakukan pembaharuan dan penambahan kode dilakukan pengujian kembali untuk memastikan revisi telah diselesaikan. Hasil kode revisi adalah:

- Notifikasi muncul saat aplikasi ditutup
- Notifikasi manajemen telah menampilkan daftar notifikasi berdasarkan penggunaan akses

4. KESIMPULAN

Model notifikasi yang dibuat pada penelitian ini menggunakan teknologi *Application Programming Interface* (API) dan *Firebase Cloud Messaging* (FCM) yang bertujuan agar model notifikasi ini dapat digunakan pada aplikasi berbasis website dan mobile. Hasil penelitian menunjukkan bahwa sistem notifikasi otomatis ini dapat digunakan dan bermanfaat dalam pengelolaan surat menyurat untuk memudahkan para stakeholder dan penerima surat mengetahui surat masuk/keluar

secara real time dengan notifikasi. Saran yang dapat dikembangkan untuk penelitian ini adalah mengembangkan fitur *Progressive Web Apps* (PWA) sehingga pengguna dapat mengakses aplikasi berbasis website melalui browser di smartphone pengguna tanpa harus menginstall aplikasi mobile selain itu pengguna masih dapat melihat notifikasi ketika membuka atau menutup aplikasi.

DAFTAR PUSTAKA

- [1] J. M. I. Budidarma, Jurnal MIB Volume 3 No 3 Juli 2019. Green Press, 2019.
- [2] M. Z. Faried, A. Mulwinda, and Y. Primadiyono, "Pengembangan Aplikasi Android Bimbingan Skripsi dengan Fitur Notifikasi," Jurnal Teknik Elektro, vol. 9, no. 2, pp. 74-79, 2017.
- [3] A. Firdaus, N. Santoso, and F. Amalia, "Pembangunan Sistem Aplikasi Pelayanan Administrasi Desa Berbasis Web dengan Fitur Notifikasi Sms Gateway (Studi Kasus Desa Pelangwot)," Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, vol. 3, no. 2, pp. 1429-1434, 2019.
- [4] A. H. Sulasmoro, I. Y. Sari, and Y. F. Sabanise, "Integrasi SMS Gateway Untuk Pengembangan Sistem Informasi Surat Pada Kantor Kepala Desa (Studi kasus: Desa Balapulung Kulon Kab. Tegal)," Jurnal Informatika: Jurnal Pengembangan IT, vol. 3, no. 1, pp. 56-60, Jan. 2018, doi: 10.30591/jpit.v3i1.678.
- [5] A. Faisol and F. Rahmadian, "REALTIME NOTIFICATION PADA APLIKASI BERBASIS WEB MENGGUNAKAN FIREBASE CLOUD MESSAGING (FCM)," Jurnal Mnemonic, vol. 1, no. 2, pp. 14-17, Dec. 2019, doi: 10.36040/mnemonic.v1i2.32.
- [6] S. Aulia Ulfa, H. Ajie, and M. F. Duskarnaen, "Pengembangan Aplikasi Layanan Notifikasi Surat Resmi Digital Di Universitas Negeri Jakarta," PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer, vol. 2, no. 2, pp. 143-147, Dec. 2018, doi: 10.21009/pinter.2.2.8.
- [7] B. G. Lainsamputty and S. Supriyadi, "Perancangan Notifikasi Pesan Disposisi Pada Sistem Pengelolaan Surat Dinas Menggunakan Layanan API Media Social," AITI, vol. 16, no. 2, pp. 165-186, Jun. 2020, doi: 10.24246/aiti.v16i2.165-186.
- [8] I. Puspitasari, A. A. Suryanto, A. Nurlifa, and S. L. Jannah, "Sistem Pengelolaan Surat Dengan Memanfaatkan Application Programming Interface (Api) Untuk Notifikasi Pesan Disposisi.," in SEMINAR NASIONAL TEKNOLOGI & SAINS, 2022.
- [9] I. Kurniawan and F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android ," JITSI : Jurnal Ilmiah Teknologi Sistem Informasi, vol. 1, no. 4, 2020.
- [10] M. Ilhami, "Pengenalan Google Firebase Untuk Hybrid Mobile Apps Berbasis Cordova," Jurnal Ilmiah IT CIDA, vol. 3, no. 1, Apr. 2018, doi: 10.55635/jic.v3i1.47.
- [11] A. Rahmatulloh, A. N. Rachman, and F. Anwar, "Implementasi Web Push Notification pada Sistem Informasi Manajemen Arsip Menggunakan PUSHJS," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 6, no. 3, p. 327, May 2019, doi: 10.25126/jtiik.201963936.
- [12] F. M. Hamzah, S. Olii, and M. S. Tuloli, "Implementasi Progressive Web Apps Pada Sistem Informasi Disposisi Surat Dengan Teknologi Service Worker," Journal of Systems and Information Technology, pp. 70-81, 2021