

**IMPLEMENTASI *CLOUD MESSAGING* PADA APLIKASI
PENGEMBALIAN BUKU PERPUSTAKAAN BERBASIS
MOBILE
(STUDI KASUS UKRIDA)**

***CLOUD MESSAGING IMPLEMENTATION ON MOBILE-
BASED LIBRARY BOOK APPLICATION
(A CASE STUDY AT UKRIDA)***

Ferry Chandra¹, Benisius²

Program Studi Teknik Informatika
Universitas Kristen Krida Wacana
Jl. Tanjung Duren Raya no. 4, Jakarta Barat
¹ferry.2012tin011@civitas.ukrida.ac.id, ²ehba@ukrida.ac.id

Abstrak

Seiring dengan berkembangnya teknologi, kebutuhan akan perangkat komunikasi yang praktis dan *mobile* juga semakin meningkat. Dengan berbagai kelebihanannya *smartphone* saat ini telah menjadi perangkat komunikasi yang populer. Salah satu fitur yang ditawarkan oleh *smartphone* adalah *cloud messaging*. *Cloud messaging* adalah layanan gratis dari Google yang memungkinkan dilakukannya pengiriman data dari *server* aplikasi pihak ketiga ke aplikasinya sendiri. Untuknya *cloud messaging* dapat berfungsi sebagai pengingat. Penelitian ini bertujuan untuk mengimplementasikan *cloud messaging* pada aplikasi android perpustakaan sehingga dapat mengingatkan batas waktu pengembalian buku kepada *user*. Aplikasi ini dibuat dengan menggunakan teknologi *web* untuk pengambilan data yang ada di *server* dan Android Studio dalam pengembangan aplikasi Androidnya. Hasil pengujian terhadap aplikasi ini dilakukan melalui uji coba. Hasil yang ditemukan adalah *cloud messaging* dapat diimplementasikan pada aplikasi pengingat pengembalian buku perpustakaan.

Kata kunci: Android, *cloud messaging*, perpustakaan

Abstract

Along with the development of technology, the need for practical and mobile communication devices is also increasing. With the advantages provided, today's smartphone has become a popular communication device. One feature offered by smartphones is cloud messaging. Cloud messaging is a free service from Google that enables the data transmission from third party app servers to the application. For it, cloud messaging can serve as a reminder. This research aims to implement cloud messaging in android-based library application so it can remind the user when the library book needs to be returned. This app was created using web technology for retrieving existing data on server and Android Studio to develop it's the Android app. The application testing was done through self-test. The results showed cloud messaging could be implemented in the library book reminder application.

Keywords: Android, *cloud messaging*, library

Tanggal Terima Naskah : 30 Oktober 2017
Tanggal Persetujuan Naskah : 30 November 2017

1. PENDAHULUAN

Penggunaan perangkat telekomunikasi saat ini hampir tidak dapat dilepaskan dari keseharian manusia. *Google* pada survei tahun 2015 mencatat 43% dari koneksi Internet dilakukan melalui perangkat *smartphone*. Angka ini bertambah 15% dari tahun sebelumnya [1]. *Smartphone* dengan berbagai kelebihanannya secara perlahan mulai menggantikan keberadaan *handphone*. Salah satu fitur menarik pada *smartphone* adalah *cloud messaging*. *Cloud messaging* adalah sebuah layanan gratis yang memungkinkan kita mengirimkan pesan dari *server* ke aplikasi *client* [2]. *Cloud messaging* dapat berfungsi sebagai pengingat, *chat messaging*, atau latar belakang proses dari suatu aplikasi. *Cloud messaging* membuat pengguna tidak perlu selalu mengecek aplikasi mereka karena aplikasi dapat menerima notifikasi dari *server*. Contoh penerapan *Cloud messaging* dapat dijumpai pada aplikasi facebook, LINE, instagram, dan lain-lain. Pada penelitian ini *cloud messaging* diterapkan pada aplikasi perpustakaan Ukrida sebagai pengingat bagi *user* setiap kali mendekati batas waktu peminjaman.

2. KONSEP DASAR

2.1 Aplikasi Mobile Android

Aplikasi *mobile* adalah program komputer yang didesain untuk dapat berjalan di atas perangkat *mobile*, seperti *smartphone*, *tablet*, atau jam [3]. Saat ini Android merupakan *mobile platform* paling populer di pasaran dengan persentase pengguna mencapai 70% [4]. Android adalah sistem operasi berbasis Linux (kernel 2.6) untuk perangkat bergerak, bersifat *open source* yang awalnya dikembangkan oleh Android Inc. sebelum selanjutnya dibeli Google pada 2005 [5].

2.2 Google Cloud Messaging (GCM)

GCM adalah sebuah layanan dari Google yang digunakan untuk mengirimkan data dari *server* ke Android *device*. GCM dapat mengirimkan pesan singkat untuk memberitahu aplikasi bahwa terdapat data baru atau notifikasi. Layanan GCM menangani seluruh antrian pesan dan pengiriman ke aplikasi pengguna Android. Kelebihan dari GCM adalah layanan ini tidak memiliki kuota maksimal pengiriman pesan dan bersifat gratis. GCM memiliki karakteristik utama sebagai berikut [2]:

- a. Mengizinkan *server* aplikasi pihak ketiga untuk mengirimkan pesan ke aplikasi Android miliknya.
- b. Jika menggunakan *GCM Cloud Connection Server*, dapat menerima pesan *upstream* dari perangkat Android pengguna.
- c. Aplikasi pada perangkat Android tidak harus berjalan untuk dapat menerima pesan. Sistem akan membangunkan aplikasi melalui *broadcast intent* ketika pesan sampai ke perangkat.
- d. GCM tidak menyediakan *user interface* atau penanganan lainnya untuk pesan. *GCM* hanya mengirimkan pesan mentah ke aplikasi Android.
- e. GCM membutuhkan perangkat Android minimal versi 2.2 atau lebih baru yang memasang aplikasi *Google Play Store*.
- f. GCM menggunakan koneksi yang sudah ada pada layanan Google. Untuk perangkat dengan versi 3.0 atau sebelumnya, perangkat tersebut membutuhkan *login* ke akun Google. Untuk versi 4.0.4 (*Ice Cream Sandwich*) ke atas, hal ini tidak dibutuhkan.



Gambar 1. Arsitektur Google *Cloud Messaging*

Arsitektur Google *cloud messaging* terdiri dari *server*, *GCM server*, dan *client device* (Gambar 1). Berikut ini adalah cara bagaimana antarkomponen saling berinteraksi [2]:

- *GCM Connection Servers* menerima pesan dari *server* aplikasi pihak ketiga dan mengirimkan pesan tersebut ke aplikasi Android (*client app*) yang mendukung GCM. Saat ini, Google menyediakan koneksi ke *server* melalui HTTP dan XMPP.
- *Server* aplikasi pihak ketiga (*3rd-Party App Server*) adalah komponen yang dibuat/diimplementasikan oleh *developer* untuk bekerja dengan koneksi *server* GCM yang diinginkan. *Server* aplikasi mengirimkan pesan kepada *server* GCM; *server* GCM mendaftarkan dan menyimpan pesan tersebut, kemudian mengirimkan ke perangkat jika perangkat dalam keadaan *online*.
- *Client App* adalah aplikasi Android yang berjalan pada perangkat yang mendukung GCM. Untuk menerima pesan dari GCM, perangkat Android tersebut harus mendaftar ke GCM dan mendapatkan ID dari registrasi tersebut. Jika menggunakan koneksi *server* melalui XMPP, aplikasi pengguna (*client*) dapat mengirimkan pesan balik ke koneksi *server*.

3. METODE PENELITIAN

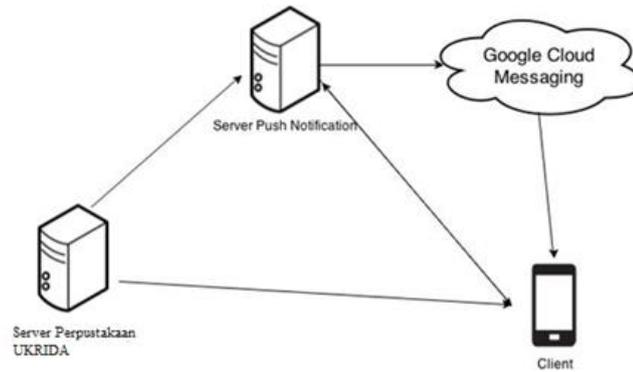
Penelitian dilakukan melalui beberapa tahapan sebagai berikut:

- a. Analisis
Pada tahap Analisis dilakukan pemodelan sistem berdasarkan analisis dari *business process* di Perpustakaan Ukrida. Pendataan *business process* dilakukan melalui wawancara dengan pimpinan unit Perpustakaan. Dari *Business process* yang diperoleh selanjutnya dilakukan pemodelan yang meliputi *use case diagram*, *use case narrative*, dan *activity diagram*.
- b. Perancangan
Pada tahap ini dilakukan perancangan sistem yang meliputi perancangan *database*, alur sistem, dan antarmuka sistem.
- c. Implementasi
Hasil perancangan sistem dari tahap sebelumnya akan diimplementasikan di tahap ini. Implementasi yang dilakukan adalah meliputi pengodean, instalasi di *server* sampai dengan pendistribusian aplikasi.
- d. Evaluasi
Sistem yang telah dibuat perlu dievaluasi untuk mendapatkan hasil yang *valid*. Pada tahapan ini evaluasi dilakukan dengan menggunakan metode *blackbox*. *Blackbox testing* adalah salah satu jenis pengujian perangkat lunak yang fokus pada kebutuhan fungsional dari perangkat lunak tersebut. Penguji atau *tester* hanya cukup mengerti tentang masukan (*input*) dan keluaran (*output*) yang dihasilkan, tanpa mengetahui sistem internal yang bekerja [6].

4. ANALISIS DAN PERANCANGAN

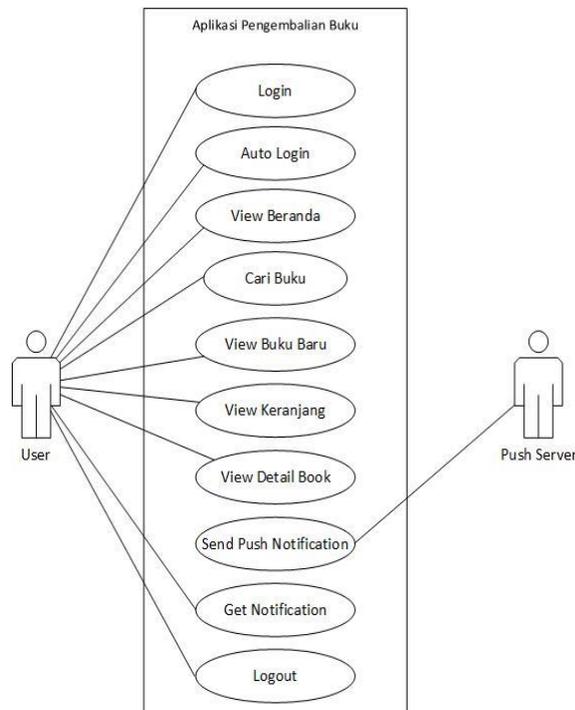
Pendataan awal untuk pemodelan sistem dilakukan melalui hasil pengamatan di lapangan dan wawancara dengan Kepala Perpustakaan Ukrida. Berdasarkan proses tersebut, disepakati bahwa *business process* untuk pengiriman notifikasi adalah sebagai berikut:

- Notifikasi akan dikirimkan kepada mahasiswa yang sudah berstatus terlambat atau potensial terlambat mengembalikan buku.
- Mahasiswa dikategorikan potensial terlambat jika tanggal kembali kurang dari satu hari sebelum batas waktu.



Gambar 2. Arsitektur sistem

Arsitektur sistem yang dirancang terdiri dari *server* perpustakaan, *server push notification*, *server google cloud messaging*, dan *smartphone client*. *Server* perpustakaan berperan menyediakan data perpustakaan seperti transaksi peminjaman dan akun anggota sedangkan *server push notification* berfungsi untuk menyimpan data gcmid yang akan digunakan dalam mengirimkan notifikasi ke *client*.



Gambar 3. Use Case Diagram

Gambar 3 merupakan *use case diagram* dari sistem yang dikembangkan. Terdapat dua aktor dengan *use case* terkait penggunaan *cloud messaging* sebagai berikut:

1. *User*

User merupakan mahasiswa Ukrida yang memiliki akses untuk melakukan beberapa hal berikut:

a. *Login*

User diwajibkan untuk *login* sebelum menggunakan aplikasi ini karena hanya mahasiswa Ukrida yang bisa melakukan *login*.

b. *Auto Login*

Auto Login berfungsi apabila *user* sudah melakukan *login* dan belum melakukan *logout*.

c. *Get Notification*

User dapat menerima notifikasi yang dikirimkan oleh *push server*. Notifikasi yang diterima berupa pemberitahuan mengenai keterlambatan pengguna dalam mengembalikan buku dan pemberitahuan saat buku yang dipinjam hampir jatuh tempo.

d. *Logout*

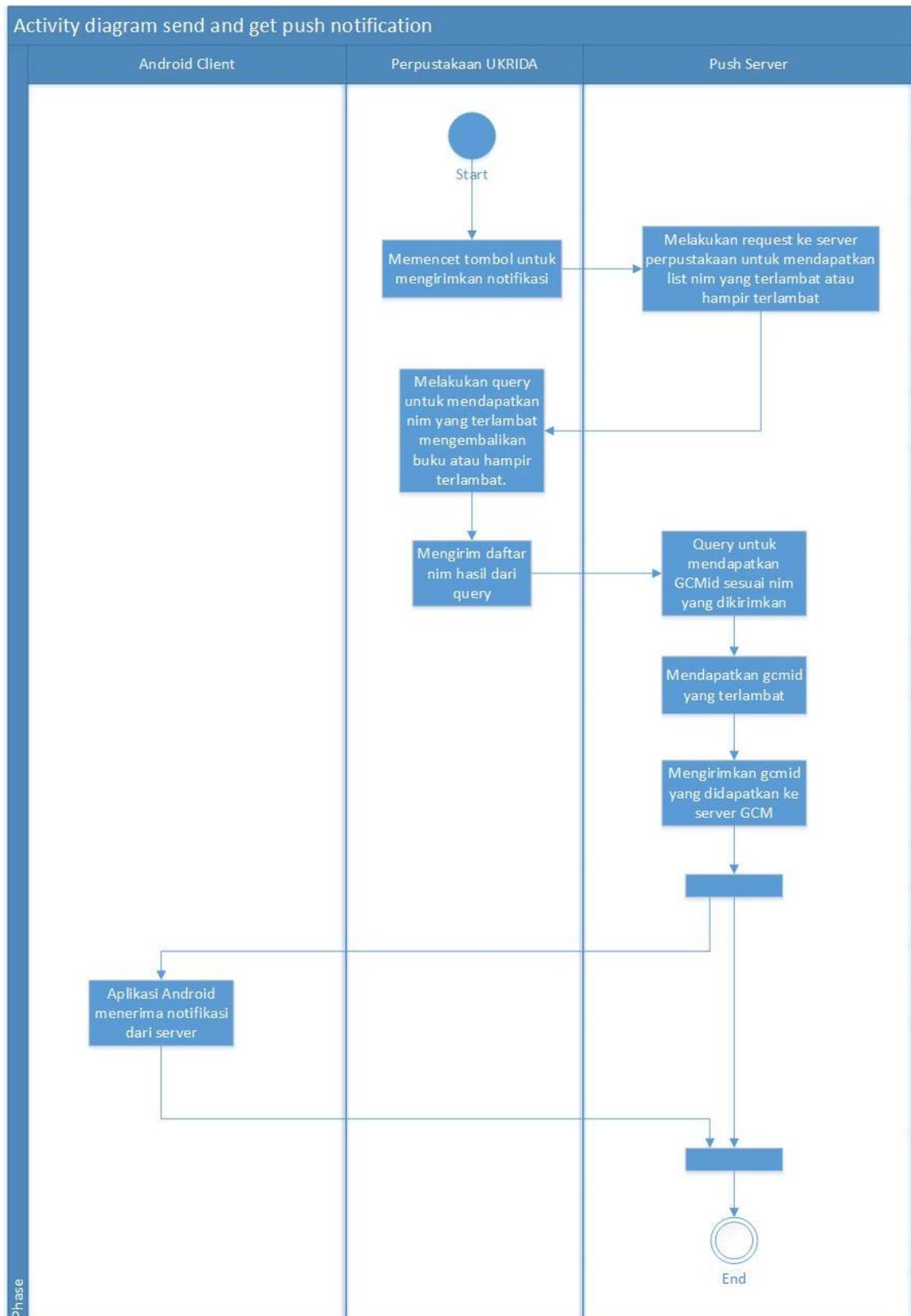
User dapat melakukan *logout* setelah *login*.

2. *Push Server*

Push Server berperan mengirimkan notifikasi ke aplikasi *mobile user*. Dalam hal ini, notifikasi yang dikirimkan berupa *push up notification* bagi peminjaman yang terlambat maupun masa peminjaman yang hampir jatuh tempo. Pengiriman notifikasi dilakukan dengan memanggil suatu fungsi yang akan mengirimkan *request* ke *server* perpustakaan untuk mendapatkan daftar NIM dengan parameter tanggal kemarin kemudian mencari transaksi peminjaman yang kurang sehari dari batas waktu peminjaman. Dari daftar NIM yang diterima, *push server* akan menyortir *gcmid* berdasarkan daftar NIM di *database push server*. *Gcmid ter-filter* selanjutnya akan dikirimkan ke *server* Google yang beralamat di <https://android.googleapis.com/gcm/send> dengan parameter *gcmid* dan pesan.

Get and send push notification adalah aktivitas utama pada sistem ini. Alur aktifitas dari proses tersebut adalah seperti pada gambar 4 dan dapat dijabarkan sebagai berikut:

- Langkah 1: Administrator membuka halaman atau url untuk mengirimkan notifikasi.
- Langkah 2: Sistem melakukan *query* untuk mendapatkan daftar NIM/*user* yang terlambat atau hampir terlambat mengembalikan buku dari *server* perpustakaan.
- Langkah 3: Daftar NIM/*user* dikirimkan ke *Push Server*.
- Langkah 4: Sistem di *Push Server* menerima dan melakukan *query* untuk mendapatkan *gcmid* dari daftar NIM/*user* yang diterima.
- Langkah 5: *Push Server* mengirimkan data *gcmid* ke *server* GCM.
- Langkah 6: *Server* GCM mengirimkan *push notification* ke *Android Client* dan mengirimkan laporan ke *Push Server* dalam bentuk JSON.
- Langkah 7: *Android Client* menerima dan menampilkan pesan.



Gambar 4. Activity diagram Get and Send Push Notification

Tabel 1 merupakan rancangan struktur tabel yang dibutuhkan untuk implementasi *database* di *push server*.

Tabel 1 Struktur tabel gcm

Kolom	Tipe Data	Deskripsi
id	int(11)	Id merupakan <i>primary key</i>
gcmid	text	Gcmid berisikan tentang gcmid dari suatu <i>device</i>
nim	varchar(50)	Nim mahasiswa yang melakukan <i>login</i>
active	tinyint(4)	Status aktif <i>device</i>

Struktur basis data pada *push server* hanya memiliki tabel gcm yang berisikan NIM mahasiswa anggota perpustakaan dan GCM (*Google Cloud Messaging*) ID. Gcmid merupakan id dari suatu *device* android yang digunakan sebagai tujuan untuk mengirimkan *push notification*. Gcmid akan dihasilkan menggunakan fungsi `gcm.register` ketika *user* melakukan proses *login*. *Client* memiliki hubungan dengan *server push* untuk menyimpan daftar gcmid dan nim pada *server push*. *Client* juga memiliki hubungan dengan *server* perpustakaan karena seluruh data tentang buku dan *login* disimpan di sana. *Push server* memiliki hubungan dengan *server* perpustakaan Ukrida untuk memperoleh daftar NIM yang terlambat, kemudian mengirimkan ke *server* google, dan diterima oleh *client*. Kolom *active* menandakan apakah GCM ID sedang dalam keadaan aktif atau nonaktif (*logout*).

5. IMPLEMENTASI DAN EVALUASI

Implementasi yang dilakukan meliputi implementasi di sisi *push server*, *server* perpustakaan, dan *app client*.

5.1 Implementasi *Push Server*

Server dengan basis data dipasang secara *online* di alamat <http://27.50.19.51:1206/gcm>.

1) Implementasi basis data MySQL

Basis data bernama *push server* yang dibutuhkan untuk menyimpan data NIM, *active*, dan gcmid.

2) Implementasi koneksi PHP

Dalam mengolah data yang tersimpan pada basis data, dibuat *file* untuk menghubungkan *server* dengan tabel yang akan diolah datanya. *File* ini menggunakan fungsi PHP `mysqli()` untuk membuat koneksi dengan *database*.

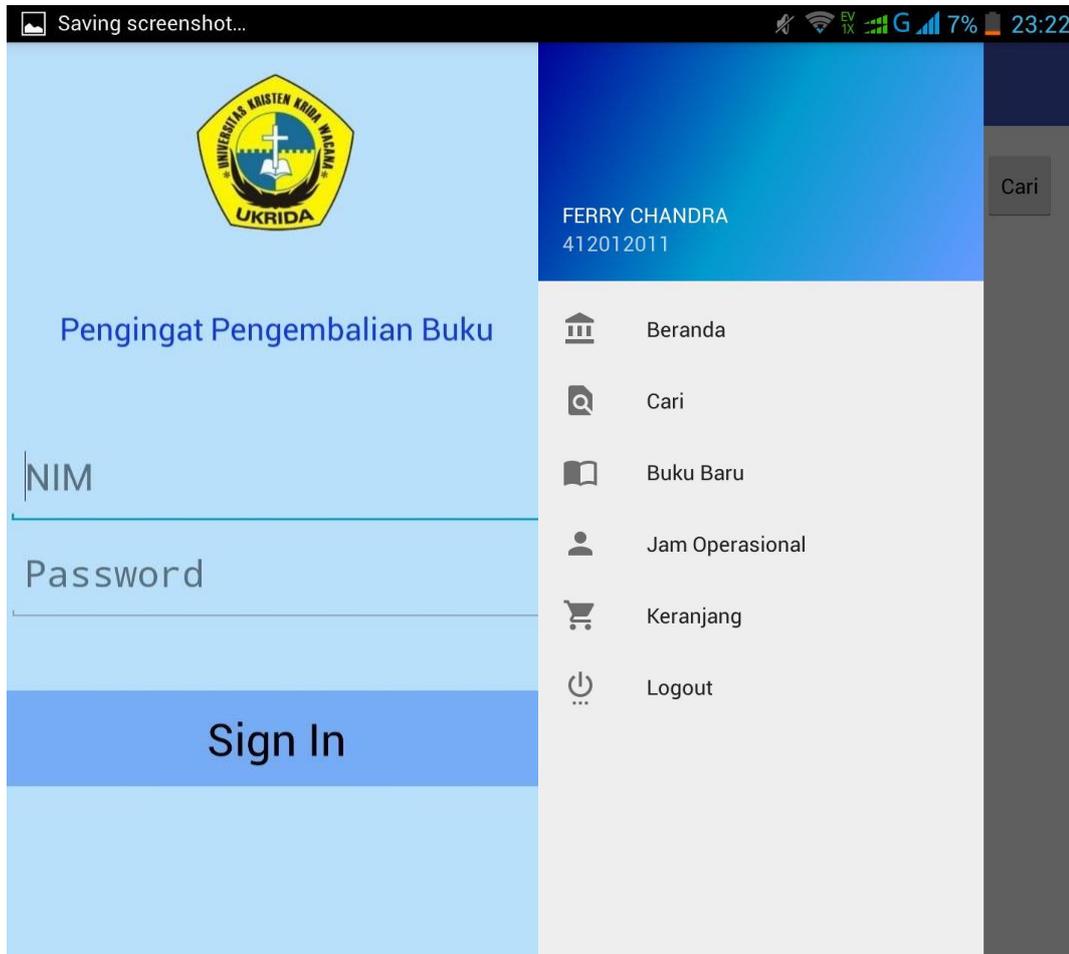
3) Implementasi *Send Push Notification*.

Halaman *send push* merupakan halaman dimana perpustakaan dapat mengirimkan notifikasi. Notifikasi yang dikirimkan akan langsung masuk ke android *user* yang sudah terpasang aplikasi pengingat pengembalian buku Ukrida dan memiliki daftar buku yang hampir atau sudah terlambat. Notifikasi dikirimkan dalam bentuk tulisan, getaran, suara, dan lampu LED pada android.

5.2 Implementasi *Server Perpustakaan*

Implementasi di *server* perpustakaan adalah dengan mengintegrasikan *Application Programming Interface* (API) di *server* perpustakaan yang beralamat di <http://ejournal.ukrida.ac.id/api>. API berfungsi untuk mengambil daftar buku secara acak (*random*), terbaru, sesuai pencarian, melakukan pengecekan saat *login*, mendapatkan daftar

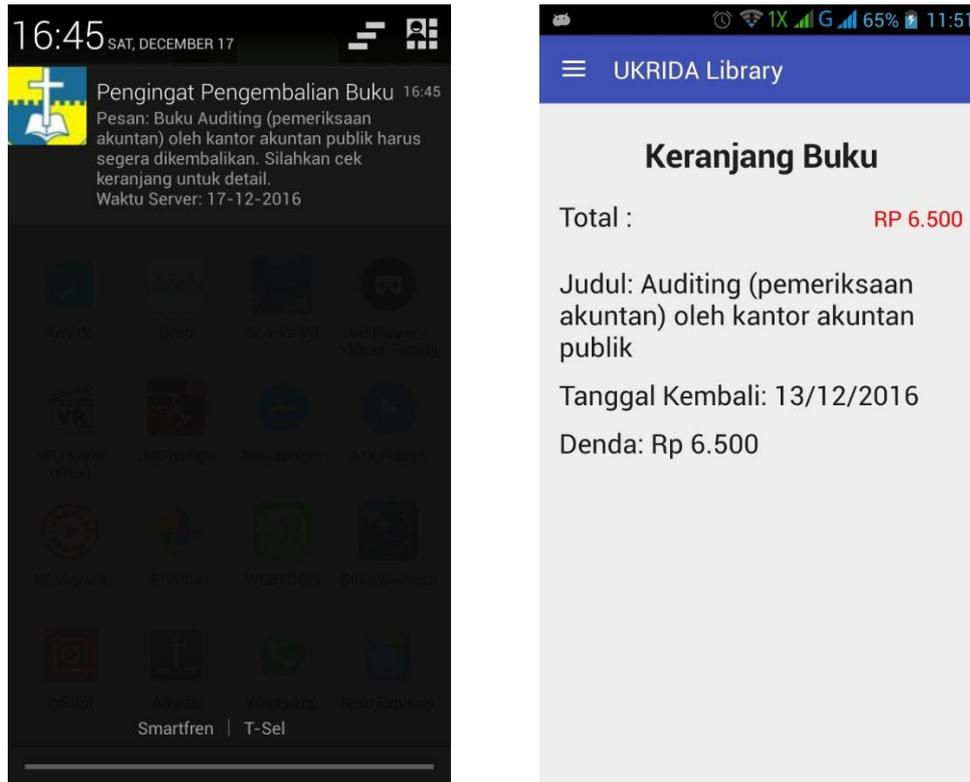
buku yang sedang dipinjam, dan mengambil data detail peminjaman berdasarkan id. API selanjutnya akan mengembalikan data dalam format JSON.



Gambar 5. Tampilan halaman *login* dan menu navigasi

5.3 Implementasi *Client App*

Setelah aplikasi terpasang di Android *client* maka halaman *login* seperti pada gambar 5 akan tampil pada saat aplikasi pertama kali dijalankan. Apabila *user* berhasil *login* maka menu navigasi dapat diakses dengan menekan tombol menu utama pada bagian kiri atas layar. *User* selanjutnya akan menerima *push notification* berupa peringatan dan pengingat tentang buku yang sudah habis atau akan habis masa pinjamnya (Gambar 6).

Gambar 6. Tampilan *push notification*

Sistem yang telah diimplementasikan kemudian diuji dengan metode *black box* yang merupakan metode pengujian dengan memfokuskan pada fungsional sistem yang telah dibangun. Pengujian dilakukan dengan menggunakan *smartphone* Smartfren Andromax V, Asus Zenfone 4, dan Xiaomi Mi 4i yang masing-masing memiliki sistem operasi Android OS v4.2.2 *Jelly Bean*, Android v4.4.2 *Kitkat*, dan Android v5.0.2 *Lollipop*.

Hasil pengujian seperti pada tabel 2 memperlihatkan bahwa *output* dari sistem telah berjalan sesuai dengan yang diharapkan.

Tabel 2. Pengujian *Black Box*

No	Skenario Pengujian	Hasil diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Login</i> menggunakan NIM dan <i>password</i>	Masuk ke dalam tampilan beranda	Sesuai harapan	<i>Valid</i>
2	Proses menampilkan menu navigasi	Muncul menu navigasi	Sesuai harapan	<i>Valid</i>
3	Menampilkan <i>detail</i> buku	Muncul <i>detail</i> buku sesuai buku yang di pilih	Sesuai harapan	<i>Valid</i>
4	Proses <i>logout</i>	Aplikasi keluar ke halaman <i>login</i>	Sesuai harapan	<i>Valid</i>
5	Proses pengiriman <i>push notification</i>	User dengan kategori terlambat atau hamper terlambat menerima notifikasi	Sesuai harapan	<i>Valid</i>

6. KESIMPULAN

Berdasarkan implementasi dan evaluasi yang telah dilakukan maka dapat diperoleh kesimpulan bahwa Google *cloud messaging* dapat diimplementasikan pada aplikasi Pengingat Pengembalian Buku Perpustakaan sebagai pengingat bagi mahasiswa ketika masa peminjaman buku hampir atau sudah habis.

REFERENSI

- [1]. Google Inc. Trended Data. Internet: <https://www.consumerbarometer.com/en/trending/?countryCode=ID&category=TRN-NOFILTER-ALL>; Accessed on January 12th, 2016.
- [2]. Developer Android. Google Cloud Messaging For Android. Internet: <https://developers.google.com/cloud-messaging/gcm>; accessed on January 8th, 2016.
- [3]. Statcounter. Internet: <http://gs.statcounter.com/os-market-share/mobile/worldwide>; accessed on on January 6th, 2016.
- [4]. Lee, Wei-Meng. 2012. *Beginning Android™ 4 Application Development*. Indiana: John Wiley & Sons, Inc.
- [5]. Khan, M. E. 2011. *Different Approaches To Black Box Testing Technique for Finding Errors. International Journal of Software Engineering & Applications (IJSEA)* Vol. 2, No. 4